

Oracle® Clusterware

Clusterware 管理およびデプロイメント・ガイド

19c

F16192-07(原本部品番号:E95727-10)

2023年8月

タイトルおよび著作権情報

Oracle Clusterware Clusterware管理およびデプロイメント・ガイド19c

F16192-07

[Copyright ©](#) 2007, 2023, Oracle and/or its affiliates.

原著者: Subhash Chandra, Eric Belden

原協力者: Richard Strohm

原協力者: David Jimenez Alvarez, Troy Anthony, Ram Avudaiappan, Mark Bauer, Devang Bagaria, Suman Bezawada, Gajanan Bhat, Burt Clouse, Ian Cookson, Jonathan Creighton, Mark Fuller, Apostolos Giannakidis, Angad Gokakkar, Donald Graves, John Grout, Vikash Gunreddy, Andrey Gusev, Winston Huang, Shankar Iyer, Sameer Joshi, Ashwinee Khaladkar, Roland Knapp, Erich Kreisler, Karen Li, Barb Lundhild, Manuel Garcia Maciel, Saar Maoz, John McHugh, Markus Michalewicz, Lars Mortimer, Anil Nair, Siva Nandan, Philip Newlan, Gia-Khanh Nguyen, Balaji Pagadala, Srinivas Poovala, Kamalesh Ramaswamy, Sampath Ravindhran, Kevin Reardon, Dipak Saggi, Duane Smith, Janet Stern, Su Tang, Manasvi Nishith Vohra, James Warnes, Douglas Williams, Soo Huey Wong

目次

- [表一覧](#)
- [タイトルおよび著作権情報](#)
- [はじめに](#)
 - [対象読者](#)
 - [ドキュメントのアクセシビリティ](#)
 - [関連ドキュメント](#)
 - [表記規則](#)
- [このリリースでの『Oracle Clusterware管理およびデプロイメント・ガイド』の変更点](#)
 - [Oracle Clusterwareリリース19cの変更点](#)
 - [Oracle Clusterware 19cで非推奨となった機能](#)
 - [Oracle Clusterware 19cでサポートが終了した機能](#)
- [1 Oracle Clusterwareの概要](#)
 - [Oracle Clusterwareの概要](#)
 - [Oracle Clusterwareのシステム要件の理解](#)
 - [Oracle Clusterwareのハードウェアの概念および要件](#)
 - [Oracle Clusterwareのオペレーティング・システムの概念および要件](#)
 - [Oracle Clusterwareのソフトウェアの概念および要件](#)
 - [Oracle Clusterwareのネットワーク構成の概念](#)
 - [単一クライアント・アクセス名\(SCAN\)](#)
 - [アドレスの手動構成](#)
 - [Oracle Clusterwareのプラットフォーム固有のソフトウェア・コンポーネントの概要](#)
 - [Oracle Clusterware技術スタック](#)
 - [クラスタ・レディ・サービス技術スタック](#)
 - [Oracle高可用性サービス技術スタック](#)
 - [トランスポート層セキュリティ暗号スイート管理](#)
 - [WindowsシステムでのOracle Clusterwareプロセス](#)
 - [Oracle Databaseの高可用性オプション](#)
 - [Oracle Clusterwareのインストールの概要](#)
 - [Oracle Clusterwareのリリースの互換性](#)
 - [Oracle Clusterwareのアップグレードおよびパッチ適用の概要](#)
 - [グリッド・インフラストラクチャ管理リポジトリの概要](#)
 - [ドメイン・サービス・クラスタの概要](#)
 - [Oracle Clusterware環境の管理の概要](#)
 - [コマンドの評価の概要](#)
 - [グリッド環境でのOracle Clusterwareのクローニングおよび拡張の概要](#)
 - [Oracle Clusterwareの高可用性フレームワークおよびAPIの概要](#)
 - [クラスタの時間管理の概要](#)
 - [クラスタの時間管理のアクティブ化および非アクティブ化](#)
- [2 Oracle Clusterwareの構成と管理](#)
 - [ロール別管理](#)
 - [クラスタ管理者の管理](#)

- [ロール区分の構成](#)
- [グリッド設定ウィザードを使用したOracle Grid Infrastructureの構成](#)
 - [単一ノードの構成](#)
 - [複数ノードの構成](#)
 - [Oracle Grid Infrastructureのアップグレード](#)
 - [サイレント・モードでの構成ウィザードの実行](#)
 - [Oracle Grid Infrastructureホームの移動およびパッチ適用](#)
- [サーバーの重みベースのノード削除](#)
- [Oracle Database Quality of Service Managementの概要](#)
- [グリッド・ネーミング・サービスの概要](#)
 - [GNSおよびGNS仮想IPアドレスのネットワーク管理タスク](#)
 - [グリッド・ネーミング・サービス構成オプションの理解](#)
 - [高可用性グリッド・ネーミング・サービス](#)
 - [アドレスの自動構成オプション](#)
 - [アドレスの静的な自動構成オプション](#)
 - [アドレスの共有GNSオプション](#)
- [グリッド・ネーミング・サービスの管理](#)
 - [高可用性GNSの構成](#)
 - [プライマリGNSインスタンスおよびセカンダリGNSインスタンスの削除](#)
 - [SRVCTLを使用したGNSの起動と停止](#)
 - [GNSサーバーまたはGNSクライアント・クラスタへのクラスタの変換](#)
 - [非GNSクラスタのGNSサーバー・クラスタへの変換](#)
 - [非GNSクラスタのクライアント・クラスタへの変換](#)
 - [GNSを実行しているシングル・クラスタのサーバー・クラスタへの変換](#)
 - [GNSを実行しているシングル・クラスタのGNSクライアント・クラスタへの変換](#)
 - [別のクラスタへのGNSの移動](#)
 - [IPv4からIPv6ネットワークへの移行時のGNSサブドメインの変更](#)
 - [DNSによるクラスタの名前解決をGNSにローリング変換](#)
- [ノード障害分離](#)
 - [IPMI用のサーバー・ハードウェア構成](#)
 - [CRSCTLを使用したIPMIベース障害分離のインストール後の構成](#)
 - [Oracle ClusterwareでのIPMIインストール後の構成](#)
 - [CRSCTLを使用したIPMI構成の変更](#)
 - [CRSCTLを使用したIPMI構成の削除](#)
- [手動構成ネットワーク上のネットワーク・アドレスの理解](#)
 - [ネットワーク・アドレスの構成要件の理解](#)
 - [IPv6アドレスのフォーマットについて](#)
 - [名前解決およびネットワーク・リソースのアドレス・タイプ](#)
 - [SCANアドレスおよびクライアント・サービス接続の理解](#)
 - [有効なノード・チェックにおけるSCANリスナーおよびサービス登録制限](#)
- [共有単一クライアント・アクセス名の構成](#)
 - [共有単一クライアント・アクセス名の構成について](#)
 - [共有SCANの使用の構成](#)

- [手動構成システム上のネットワーク・アドレスの変更](#)
 - [SRVCTLを使用した仮想IPアドレスの変更](#)
 - [Oracle Clusterwareのプライベート・ネットワーク構成の変更](#)
 - [プライベート・ネットワークおよびネットワーク・インタフェースについて](#)
 - [冗長なインターコネクトの使用](#)
 - [OIFCFGを使用したインタフェース名の変更結果](#)
 - [ネットワーク・インタフェースの変更](#)
 - [SRVCTLを使用したネットワークの作成](#)
 - [クラスタ内のネットワーク・アドレス構成](#)
 - [SRVCTLを使用した、静的IPv4アドレスの静的IPv6アドレスへの変更](#)
 - [SRVCTLを使用した、動的IPv4アドレスの動的IPv6アドレスへの変更](#)
 - [IPv4ネットワークのIPv4およびIPv6ネットワークへの変更](#)
 - [SRVCTLを使用したVIPアドレスのネットワークのIPv4からIPv6への移行](#)
- [クラスタ間依存性プロキシ](#)
- [3 ポリシー・ベースのクラスタおよび容量の管理](#)
 - [サーバー・プールおよびポリシー・ベース管理の概要](#)
 - [サーバー・プールおよびサーバーのカテゴリ化](#)
 - [サーバー・プールおよびポリシーベース管理](#)
 - [サーバー・プールの動作](#)
 - [デフォルト・サーバー・プール](#)
 - [空きサーバー・プール](#)
 - [汎用サーバー・プール](#)
 - [サーバー・プール属性](#)
 - [サーバー・プールを使用した、Oracle Clusterwareでの新しいサーバーの割当て方法](#)
 - [サーバー・プールからサーバー・プールへのサーバーの移動](#)
 - [デフォルト属性を使用したサーバー・プールの管理](#)
 - [サーバーのカテゴリ化の概要](#)
 - [クラスタ構成ポリシーおよびポリシー・セットの概要](#)
 - [ロード対応のリソース配置](#)
 - [サーバー構成およびサーバー状態の属性](#)
 - [データベース・サーバーのメモリー不足の管理](#)
 - [サーバー・カテゴリ属性](#)
 - [ポリシー・セット構成の例](#)
- [4 Oracle Flex Clusters](#)
 - [Oracle Flex Clusterの概要](#)
 - [Oracle Flex Clusterの管理](#)
 - [クラスタ・モードの変更](#)
 - [Oracle Clusterwareの標準クラスタからのOracle Flex Clusterへの変更](#)
 - [Oracle拡張クラスタ](#)
 - [Oracle拡張クラスタの構成](#)
- [5 Oracle Cluster Registryおよび投票ファイルの管理](#)
 - [Oracle Cluster RegistryおよびOracle Local Registryの管理](#)
 - [Oracle Automatic Storage ManagementへのOracle Cluster Registryの移行](#)

- [Oracle ASMから他のタイプの記憶域へのOracle Cluster Registryの移行](#)
- [Oracle Cluster Registryの場所の追加、置換、修復および削除](#)
 - [Oracle Cluster Registryの場所の追加](#)
 - [Oracle Cluster Registryの場所の削除](#)
 - [Oracle Cluster Registryの場所の置換](#)
 - [ローカル・ノードのOracle Cluster Registry構成の修復](#)
 - [Oracle Cluster Registryのデータ損失保護メカニズムの無効化](#)
- [Oracle Cluster Registryのバックアップ](#)
 - [バックアップ・ファイルのリスト](#)
 - [バックアップ場所の変更](#)
- [Oracle Cluster Registryのリストア](#)
 - [Oracle Cluster Registryの問題の確認方法](#)
 - [LinuxまたはUnixシステムでのOracle Cluster Registryのリストア](#)
 - [WindowsシステムでのOracle Cluster Registryのリストア](#)
 - [Oracle Restart環境でのOracle Cluster Registryのリストア](#)
- [Oracle Cluster Registryの問題の診断](#)
- [エクスポート・コマンドおよびインポート・コマンドを使用したOracle Cluster Registryの管理](#)
 - [LinuxまたはUNIXシステムでのOracle Cluster Registryの内容のインポート](#)
 - [WindowsシステムでのOracle Cluster Registryの内容のインポート](#)
- [Oracle Local Registry](#)
- [Oracle Cluster Registry構成のアップグレードおよびダウングレード](#)
- [投票ファイルの管理](#)
 - [Oracle ASMでの投票ファイルの格納](#)
 - [投票ファイルのバックアップ](#)
 - [投票ファイルのリストア](#)
 - [投票ファイルの追加、削除または移行](#)
 - [Oracle ASMに格納されている投票ファイルの変更](#)
 - [Oracle ASMに格納されていない投票ファイルの変更](#)
 - [Oracle ASMへの投票ファイルの移行](#)
 - [投票ファイルの場所の確認](#)
- [6 クラスタ・ノードの追加および削除](#)
 - [クラスタ・ノードを追加するための必須ステップ](#)
 - [LinuxおよびUNIXシステムでのクラスタ・ノードの追加および削除](#)
 - [LinuxおよびUNIXシステムでのクラスタ・ノードの追加](#)
 - [LinuxおよびUNIXシステムでのクラスタ・ノードの削除](#)
 - [Windowsシステムでのクラスタ・ノードの追加および削除](#)
 - [Windowsシステムでのクラスタへのノードの追加](#)
 - [Windowsシステムでのクラスタ・ノードの削除](#)
- [7 Oracle Clusterwareのクローニング](#)
 - [Oracle Clusterwareのクローニングの概要](#)
 - [Oracle Grid Infrastructureホームをクローニングするための準備](#)
 - [ステップ1: Oracle Clusterwareのインストール](#)
 - [ステップ2: 実行中のソフトウェアの停止](#)

- [ステップ3: Oracle Grid Infrastructureホームのコピーの作成](#)
 - [方法1: 設定ウィザードを使用してOracle Grid Infrastructureホームのコピーを作成する](#)
 - [方法2a: そのコピーから不要なファイルを削除することによりOracle Grid Infrastructureホームのコピーを作成する](#)
 - [方法2b: -Xオプションを使用してOracle Grid Infrastructureホームのコピーを作成する](#)
- [Oracle Clusterwareのクローニングによるクラスタの作成](#)
 - [ステップ1: 新規クラスタ・ノードの準備](#)
 - [ステップ2: Oracle Grid Infrastructureホームのデプロイ](#)
 - [ステップ3: gridSetup.shユーティリティの実行](#)
- [クローニングを使用したクラスタへのノードの追加](#)
- [クローニング時に生成されたログ・ファイルの検索および表示](#)
- [8 Oracle Clusterwareを使用したアプリケーションの高可用性の実現](#)
 - [Oracle Clusterwareリソースおよびエージェント](#)
 - [Oracle Clusterwareのリソース](#)
 - [仮想マシンのリソース](#)
 - [リソース・グループ](#)
 - [リソース・グループ・タイプ](#)
 - [リソース・グループの使用](#)
 - [Oracle Clusterwareのリソース・タイプ](#)
 - [Oracle Clusterwareのエージェント](#)
 - [Oracle Clusterwareの組み込みエージェント](#)
 - [アクション・スクリプト](#)
 - [エージェントの作成](#)
 - [CおよびC++エージェントの作成およびデプロイ](#)
 - [Oracle Clusterwareへのリソースの登録](#)
 - [Oracle Clusterwareを使用した高可用性の実現の概要](#)
 - [リソース属性](#)
 - [リソースの状態](#)
 - [リソース依存性](#)
 - [起動依存性](#)
 - [attraction](#)
 - [dispersion](#)
 - [exclusion](#)
 - [hard](#)
 - [pullup](#)
 - [weak](#)
 - [停止依存性](#)
 - [hard](#)
 - [リソース状態のリカバリに対するリソース依存性の影響](#)
 - [リソースの配置](#)
 - [リソースとしてのアプリケーションの登録](#)

- [Oracle Clusterwareで管理されるアプリケーションVIPの作成](#)
 - [Oracle Enterprise ManagerによるアプリケーションVIPの追加](#)
- [ユーザー定義のリソースの追加](#)
 - [デプロイメント・スキームの決定](#)
 - [特定のサーバー・プールへのリソースの追加](#)
 - [サーバー固有のデプロイメントを使用したリソースの追加](#)
 - [generic_applicationリソース・タイプを使用するリソースの作成](#)
- [Oracle Enterprise Managerを使用したリソースの追加](#)
- [リソース権限の変更](#)
- [アプリケーションの配置ポリシー](#)
- [アプリケーションおよびアプリケーション・リソースの登録解除](#)
- [リソースの管理](#)
 - [アプリケーション・リソースの登録](#)
 - [アプリケーション・リソースの起動](#)
 - [アプリケーションおよびアプリケーション・リソースの再配置](#)
 - [アプリケーションおよびアプリケーション・リソースの停止](#)
 - [クラスタウェア・アプリケーションおよびアプリケーション・リソースのステータス情報の表示](#)
- [Oracle Clusterwareによるリソースの自動再起動の管理](#)
 - [Oracle Clusterwareリソースの自動再起動の回避](#)
 - [Oracle Clusterwareリソースの再起動試行カウンタの自動管理](#)
- [A クラスタ検証ユーティリティのリファレンス](#)
 - [クラスタ検証ユーティリティについて](#)
 - [CVUの概要](#)
 - [CVUの操作上のノート](#)
 - [CVUのインストール要件](#)
 - [CVUの使用方法](#)
 - [CVUの構成ファイル](#)
 - [権限およびセキュリティ](#)
 - [CVUヘルプの使用](#)
 - [非推奨およびサポート対象外のCLUVFYコマンド](#)
 - [特別なCVU項目](#)
 - [修正スクリプトの生成](#)
 - [CVUを使用した、インストールの前提条件を満たしているかどうかの確認](#)
 - [Oracle Database 10gリリース1または2でのCVUの使用](#)
 - [開始基準および終了基準](#)
 - [冗長モードおよびUNKNOWN出力](#)
 - [CVUのノード・リストのショートカット](#)
- [クラスタ検証ユーティリティのコマンド・リファレンス](#)
 - [cluvfy comp acfs](#)
 - [cluvfy comp admprv](#)
 - [cluvfy comp asm](#)
 - [cluvfy comp baseline](#)
 - [cluvfy comp clocksync](#)

- [cluvfy comp clu](#)
- [cluvfy comp clumgr](#)
- [cluvfy comp containerhost](#)
- [cluvfy comp crs](#)
- [cluvfy comp dhcp](#)
- [cluvfy comp dns](#)
- [cluvfy comp freespace](#)
- [cluvfy comp gns](#)
- [cluvfy comp gnpn](#)
- [cluvfy comp ha](#)
- [cluvfy comp healthcheck](#)
- [cluvfy comp nodeapp](#)
- [cluvfy comp nodecon](#)
- [cluvfy comp nodereach](#)
- [cluvfy comp ocr](#)
- [cluvfy comp ohasd](#)
- [cluvfy comp olr](#)
- [cluvfy comp peer](#)
- [cluvfy comp scan](#)
- [cluvfy comp software](#)
- [cluvfy comp space](#)
- [cluvfy comp ssa](#)
- [cluvfy comp sys](#)
- [cluvfy comp vdisk](#)
- [cluvfy comp vital](#)
- [cluvfy stage acfscfg](#)
- [cluvfy stage appcluster](#)
- [cluvfy stage cfs](#)
- [cluvfy stage crsinst](#)
- [cluvfy stage dbcfg](#)
- [cluvfy stage -pre dbinst](#)
- [cluvfy stage hacfg](#)
- [cluvfy stage hwos](#)
- [cluvfy stage nodeadd](#)
- [cluvfy stage nodedel](#)
- [cluvfy stage osupg](#)
- [cluvfy stage patch](#)
- [CVUのトラブルシューティングおよび診断出力](#)
 - [トレースの有効化](#)
 - [クラスタ検証ユーティリティの既知の問題](#)
 - [クラスタ検証ユーティリティによってサポートされるデータベースのバージョン](#)
 - [Linuxの共有記憶域アクセス性\(ssa\)チェックのレポートの制限](#)
 - [Red Hat Linuxでの共有ディスクの検出](#)

- [B Oracle Clusterwareのリソース・リファレンス](#)
 - [リソース属性](#)
 - [構成可能なリソース属性](#)
 - [ACL](#)
 - [ACTION_SCRIPT](#)
 - [ACTION_TIMEOUT](#)
 - [ACTIONS](#)
 - [ACTIVE_PLACEMENT](#)
 - [AGENT_FILENAME](#)
 - [ALERT_TEMPLATE](#)
 - [AUTO_START](#)
 - [CARDINALITY](#)
 - [CARDINALITY_ID](#)
 - [CHECK_INTERVAL](#)
 - [CHECK_TIMEOUT](#)
 - [CLEAN_TIMEOUT](#)
 - [CRITICAL_RESOURCES](#)
 - [DELETE_TIMEOUT](#)
 - [DESCRIPTION](#)
 - [ENABLED](#)
 - [FAILURE_INTERVAL](#)
 - [FAILURE_THRESHOLD](#)
 - [HOSTING_MEMBERS](#)
 - [INSTANCE_FAILOVER](#)
 - [INTERMEDIATE_TIMEOUT](#)
 - [LOAD](#)
 - [MODIFY_TIMEOUT](#)
 - [NAME](#)
 - [OFFLINE_CHECK_INTERVAL](#)
 - [ONLINE_RELOCATION_TIMEOUT](#)
 - [PLACEMENT](#)
 - [RELOCATE_KIND](#)
 - [RELOCATE_BY_DEPENDENCY](#)
 - [RESTART_ATTEMPTS](#)
 - [SCRIPT_TIMEOUT](#)
 - [SERVER_CATEGORY](#)
 - [SERVER_POOLS](#)
 - [START_CONCURRENCY](#)
 - [START_DEPENDENCIES](#)
 - [START_TIMEOUT](#)
 - [STOP_CONCURRENCY](#)
 - [STOP_DEPENDENCIES](#)
 - [STOP_TIMEOUT](#)

- [UPTIME_THRESHOLD](#)
 - [USER_WORKLOAD](#)
 - [USE_STICKINESS](#)
- [読取り専用のリソース属性](#)
 - [ACTION_FAILURE_EVENT_TEMPLATE](#)
 - [INSTANCE_COUNT](#)
 - [INTERNAL_STATE](#)
 - [LAST_SERVER](#)
 - [LAST_STATE_CHANGE](#)
 - [PROFILE_CHANGE_EVENT_TEMPLATE](#)
 - [RESOURCE_LIST](#)
 - [RESTART_COUNT](#)
 - [STATE](#)
 - [STATE_CHANGE_EVENT_TEMPLATE](#)
 - [STATE_DETAILS](#)
 - [TARGET](#)
 - [TARGET_SERVER](#)
 - [TYPE](#)
- [非推奨のリソース属性](#)
 - [DEGREE](#)
- [サード・パーティ製アプリケーションのアクション・スクリプトの例](#)
- [C OLSNODESのコマンド・リファレンス](#)
 - [OLSNODESの使用](#)
 - [概要](#)
 - [操作上のノート](#)
 - [OLSNODESコマンドのサマリー](#)
 - [構文](#)
 - [例](#)
- [D Oracle Interface Configurationツール\(OIFCFG\)のコマンド・リファレンス](#)
 - [OIFCFGコマンドライン・インタフェースの起動](#)
 - [OIFCFGの使用方法的概要](#)
 - [OIFCFGコマンドの書式](#)
 - [OIFCFGコマンド](#)
 - [OIFCFGコマンドのパラメータ](#)
 - [OIFCFGの使用上のノート](#)
 - [OIFCFGの例](#)
- [E Oracle Clusterware制御\(CRSCTL\)ユーティリティ・リファレンス](#)
 - [CRSCTLの概要](#)
 - [クラスタ用に変更された\(クラスタ対応\)コマンド](#)
 - [CRSCTLの操作上のノート](#)
 - [非推奨のサブプログラムまたはコマンド](#)
 - [デュアル環境のCRSCTLコマンド](#)
 - [crsctl check css](#)

- [crsctl check evm](#)
- [crsctl get hostname](#)
- [crsctl add resource](#)
- [crsctl delete resource](#)
- [crsctl eval add resource](#)
- [crsctl eval fail resource](#)
- [crsctl eval relocate resource](#)
- [crsctl eval modify resource](#)
- [crsctl eval start resource](#)
- [crsctl eval stop resource](#)
- [crsctl getperm resource](#)
- [crsctl modify resource](#)
- [crsctl relocate resource](#)
- [crsctl restart resource](#)
- [crsctl setperm resource](#)
- [crsctl start resource](#)
- [crsctl status resource](#)
- [crsctl stop resource](#)
- [crsctl add resourcegroup](#)
- [crsctl check resourcegroup](#)
- [crsctl delete resourcegroup](#)
- [crsctl eval add resourcegroup](#)
- [crsctl eval fail resourcegroup](#)
- [crsctl eval relocate resourcegroup](#)
- [crsctl eval start resourcegroup](#)
- [crsctl eval stop resourcegroup](#)
- [crsctl export resourcegroup](#)
- [crsctl modify resourcegroup](#)
- [crsctl relocate resourcegroup](#)
- [crsctl restart resourcegroup](#)
- [crsctl start resourcegroup](#)
- [crsctl status resourcegroup](#)
- [crsctl stop resourcegroup](#)
- [crsctl add resourcegrouptype](#)
- [crsctl delete resourcegrouptype](#)
- [crsctl modify resourcegrouptype](#)
- [crsctl get tracefileopts](#)
- [crsctl set tracefileopts](#)
- [crsctl add type](#)
- [crsctl delete type](#)
- [crsctl getperm type](#)
- [crsctl modify type](#)
- [crsctl setperm type](#)

- [crsctl status type](#)
- [Oracle RAC環境のCRSCTLコマンド](#)
 - [crsctl request action](#)
 - [crsctl add category](#)
 - [crsctl delete category](#)
 - [crsctl modify category](#)
 - [crsctl status category](#)
 - [crsctl check cluster](#)
 - [crsctl start cluster](#)
 - [crsctl stop cluster](#)
 - [crsctl get cluster class](#)
 - [crsctl get cluster configuration](#)
 - [crsctl set cluster disabledtlsciphersuite](#)
 - [crsctl get cluster extended](#)
 - [crsctl get cluster hubsize](#)
 - [crsctl set cluster hubsize](#)
 - [crsctl get cluster mode](#)
 - [crsctl set cluster mode](#)
 - [crsctl get cluster name](#)
 - [crsctl add cluster site](#)
 - [crsctl delete cluster site](#)
 - [crsctl modify cluster site](#)
 - [crsctl query cluster site](#)
 - [crsctl get cluster tlsciphersuite](#)
 - [crsctl get cluster type](#)
 - [crsctl set cluster type](#)
 - [crsctl get cpu equivalency](#)
 - [crsctl set cpu equivalency](#)
 - [crsctl check crs](#)
 - [crsctl config crs](#)
 - [crsctl disable crs](#)
 - [crsctl enable crs](#)
 - [crsctl start crs](#)
 - [crsctl stop crs](#)
 - [crsctl query crs activeversion](#)
 - [crsctl add crs administrator](#)
 - [crsctl delete crs administrator](#)
 - [crsctl query crs administrator](#)
 - [crsctl query crs autostart](#)
 - [crsctl set crs autostart](#)
 - [crsctl query crs releasepatch](#)
 - [crsctl query crs releaseversion](#)
 - [crsctl query crs site](#)

- [crsctl query crs softwarepatch](#)
- [crsctl query crs softwareversion](#)
- [crsctl get css](#)
- [crsctl pin css](#)
- [crsctl set css](#)
- [crsctl unpin css](#)
- [crsctl unset css](#)
- [crsctl get css ipmiaddr](#)
- [crsctl set css ipmiaddr](#)
- [crsctl set css ipmiadmin](#)
- [crsctl query css ipmiconfig](#)
- [crsctl unset css ipmiconfig](#)
- [crsctl query css ipmidevice](#)
- [crsctl get css noautorestart](#)
- [crsctl set css noautorestart](#)
- [crsctl add css votedisk](#)
- [crsctl delete css votedisk](#)
- [crsctl query css votedisk](#)
- [crsctl check ctss](#)
- [crsctl discover dhcp](#)
- [crsctl get clientid dhcp](#)
- [crsctl release dhcp](#)
- [crsctl request dhcp](#)
- [crsctl replace discoverystring](#)
- [crsctl query dns](#)
- [crsctl start ip](#)
- [crsctl status ip](#)
- [crsctl stop ip](#)
- [crsctl lsmodules](#)
- [crsctl create member_cluster_configuration](#)
- [crsctl delete member_cluster_configuration](#)
- [crsctl query member_cluster_configuration](#)
- [crsctl delete node](#)
- [crsctl get nodename](#)
- [crsctl get node role](#)
- [crsctl add policy](#)
- [crsctl delete policy](#)
- [crsctl eval activate policy](#)
- [crsctl modify policy](#)
- [crsctl status policy](#)
- [crsctl create policyset](#)
- [crsctl modify policyset](#)
- [crsctl status policyset](#)

- [crsctl check resource](#)
- [crsctl relocate resource](#)
- [crsctl get resource use](#)
- [crsctl set resource use](#)
- [crsctl start rollingpatch](#)
- [crsctl stop rollingpatch](#)
- [crsctl start rollingupgrade](#)
- [crsctl eval add server](#)
- [crsctl eval delete server](#)
- [crsctl eval relocate server](#)
- [crsctl modify server](#)
- [crsctl relocate server](#)
- [crsctl status server](#)
- [crsctl get server css_critical](#)
- [crsctl set server css_critical](#)
- [crsctl get server label](#)
- [crsctl set server label](#)
- [crsctl add serverpool](#)
- [crsctl delete serverpool](#)
- [crsctl eval add serverpool](#)
- [crsctl eval delete serverpool](#)
- [crsctl eval modify serverpool](#)
- [crsctl getperm serverpool](#)
- [crsctl modify serverpool](#)
- [crsctl setperm serverpool](#)
- [crsctl status serverpool](#)
- [crsctl query socket udp](#)
- [crsctl start testdns](#)
- [crsctl status testdns](#)
- [crsctl stop testdns](#)
- [crsctl replace votedisk](#)
- [crsctl add wallet](#)
- [crsctl delete wallet](#)
- [crsctl modify wallet](#)
- [crsctl query wallet](#)
- [Oracle Restart環境のCRSCTLコマンド](#)
 - [crsctl check has](#)
 - [crsctl config has](#)
 - [crsctl disable has](#)
 - [crsctl enable has](#)
 - [crsctl query has releaseversion](#)
 - [crsctl query has softwareversion](#)
 - [crsctl start has](#)

- [crsctl stop has](#)
- [トラブルシューティングおよび診断出力](#)
 - [crsctl set logを使用した動的デバッグ](#)
 - [コンポーネント・レベルのデバッグ](#)
 - [Oracle Clusterwareモジュールのデバッグの有効化](#)
 - [Oracle Clusterwareリソースのデバッグの有効化](#)
- [F サーバー制御\(SRVCTL\)コマンド・リファレンス](#)
 - [SRVCTLの使用方法](#)
 - [単一文字ではなくキーワードとしてのコマンド・パラメータの指定](#)
 - [SRVCTLオブジェクトの値の文字セットおよび大文字小文字の区別](#)
 - [SRVCTLヘルプの使用](#)
 - [SRVCTLの権限とセキュリティ](#)
 - [追加のSRVCTLトピック](#)
 - [非推奨のSRVCTLサブプログラムまたはコマンド](#)
 - [すべてのSRVCTLコマンドの単一文字パラメータ](#)
 - [その他のSRVCTLコマンドおよびパラメータ](#)
 - [SRVCTLのコマンド・リファレンス](#)
 - [srvctl config all](#)
 - [srvctl add asm](#)
 - [srvctl config asm](#)
 - [srvctl disable asm](#)
 - [srvctl enable asm](#)
 - [srvctl getenv asm](#)
 - [srvctl modify asm](#)
 - [srvctl predict asm](#)
 - [srvctl relocate asm](#)
 - [srvctl remove asm](#)
 - [srvctl setenv asm](#)
 - [srvctl start asm](#)
 - [srvctl status asm](#)
 - [srvctl stop asm](#)
 - [srvctl unsetenv asm](#)
 - [srvctl add asmnetwork](#)
 - [srvctl config asmnetwork](#)
 - [srvctl modify asmnetwork](#)
 - [srvctl remove asmnetwork](#)
 - [srvctl add cdp](#)
 - [srvctl disable cdp](#)
 - [srvctl enable cdp](#)
 - [srvctl modify cdp](#)
 - [srvctl relocate cdp](#)
 - [srvctl remove cdp](#)
 - [srvctl start cdp](#)

- [srvctl status cdp](#)
- [srvctl stop cdp](#)
- [srvctl add cdpproxy](#)
- [srvctl config cdpproxy](#)
- [srvctl disable cdpproxy](#)
- [srvctl enable cdpproxy](#)
- [srvctl modify cdpproxy](#)
- [srvctl relocate cdpproxy](#)
- [srvctl remove cdpproxy](#)
- [srvctl start cdpproxy](#)
- [srvctl status cdpproxy](#)
- [srvctl stop cdpproxy](#)
- [srvctl add cvu](#)
- [srvctl config cvu](#)
- [srvctl disable cvu](#)
- [srvctl enable cvu](#)
- [srvctl modify cvu](#)
- [srvctl relocate cvu](#)
- [srvctl remove cvu](#)
- [srvctl start cvu](#)
- [srvctl status cvu](#)
- [srvctl stop cvu](#)
- [srvctl add exportfs](#)
- [srvctl config exportfs](#)
- [srvctl disable exportfs](#)
- [srvctl enable exportfs](#)
- [srvctl modify exportfs](#)
- [srvctl remove exportfs](#)
- [srvctl start exportfs](#)
- [srvctl status exportfs](#)
- [srvctl stop exportfs](#)
- [srvctl add filesystem](#)
- [srvctl config filesystem](#)
- [srvctl disable filesystem](#)
- [srvctl enable filesystem](#)
- [srvctl modify filesystem](#)
- [srvctl predict filesystem](#)
- [srvctl remove filesystem](#)
- [srvctl start filesystem](#)
- [srvctl status filesystem](#)
- [srvctl stop filesystem](#)
- [srvctl add gns](#)
- [srvctl config gns](#)

- [srvctl disable gns](#)
- [srvctl enable gns](#)
- [srvctl export gns](#)
- [srvctl import gns](#)
- [srvctl modify gns](#)
- [srvctl relocate gns](#)
- [srvctl remove gns](#)
- [srvctl start gns](#)
- [srvctl status gns](#)
- [srvctl stop gns](#)
- [srvctl update gns](#)
- [srvctl add havip](#)
- [srvctl config havip](#)
- [srvctl disable havip](#)
- [srvctl enable havip](#)
- [srvctl modify havip](#)
- [srvctl relocate havip](#)
- [srvctl remove havip](#)
- [srvctl start havip](#)
- [srvctl status havip](#)
- [srvctl stop havip](#)
- [srvctl add ioserver](#)
- [srvctl config ioserver](#)
- [srvctl disable ioserver](#)
- [srvctl enable ioserver](#)
- [srvctl getenv ioserver](#)
- [srvctl modify ioserver](#)
- [srvctl relocate ioserver](#)
- [srvctl remove ioserver](#)
- [srvctl setenv ioserver](#)
- [srvctl start ioserver](#)
- [srvctl status ioserver](#)
- [srvctl stop ioserver](#)
- [srvctl unsetenv ioserver](#)
- [srvctl add mgmtdb](#)
- [srvctl config mgmtdb](#)
- [srvctl disable mgmtdb](#)
- [srvctl enable mgmtdb](#)
- [srvctl getenv mgmtdb](#)
- [srvctl modify mgmtdb](#)
- [srvctl relocate mgmtdb](#)
- [srvctl remove mgmtdb](#)
- [srvctl setenv mgmtdb](#)

- [srvctl start mgmtldb](#)
- [srvctl status mgmtldb](#)
- [srvctl stop mgmtldb](#)
- [srvctl unsetenv mgmtldb](#)
- [srvctl add mgmtlsnr](#)
- [srvctl config mgmtlsnr](#)
- [srvctl disable mgmtlsnr](#)
- [srvctl enable mgmtlsnr](#)
- [srvctl getenv mgmtlsnr](#)
- [srvctl modify mgmtlsnr](#)
- [srvctl remove mgmtlsnr](#)
- [srvctl setenv mgmtlsnr](#)
- [srvctl start mgmtlsnr](#)
- [srvctl status mgmtlsnr](#)
- [srvctl stop mgmtlsnr](#)
- [srvctl unsetenv mgmtlsnr](#)
- [srvctl add mountfs](#)
- [srvctl config mountfs](#)
- [srvctl disable mountfs](#)
- [srvctl enable mountfs](#)
- [srvctl modify mountfs](#)
- [srvctl remove mountfs](#)
- [srvctl start mountfs](#)
- [srvctl status mountfs](#)
- [srvctl stop mountfs](#)
- [srvctl add ovmm](#)
- [srvctl config ovmm](#)
- [srvctl modify ovmm](#)
- [srvctl remove ovmm](#)
- [srvctl add qosmserver](#)
- [srvctl config qosmserver](#)
- [srvctl disable qosmserver](#)
- [srvctl enable qosmserver](#)
- [srvctl modify qosmserver](#)
- [srvctl predict qosmserver](#)
- [srvctl relocate qosmserver](#)
- [srvctl remove qosmserver](#)
- [srvctl start qosmserver](#)
- [srvctl status qosmserver](#)
- [srvctl stop qosmserver](#)
- [srvctl add rhpclient](#)
- [srvctl config rhpclient](#)
- [srvctl disable rhpclient](#)

- [srvctl enable rhpcclient](#)
- [srvctl modify rhpcclient](#)
- [srvctl relocate rhpcclient](#)
- [srvctl remove rhpcclient](#)
- [srvctl start rhpcclient](#)
- [srvctl status rhpcclient](#)
- [srvctl stop rhpcclient](#)
- [srvctl add rhpserver](#)
- [srvctl config rhpserver](#)
- [srvctl disable rhpserver](#)
- [srvctl enable rhpserver](#)
- [srvctl modify rhpserver](#)
- [srvctl relocate rhpserver](#)
- [srvctl remove rhpserver](#)
- [srvctl start rhpserver](#)
- [srvctl status rhpserver](#)
- [srvctl stop rhpserver](#)
- [srvctl add vm](#)
- [srvctl check vm](#)
- [srvctl config vm](#)
- [srvctl disable vm](#)
- [srvctl enable vm](#)
- [srvctl modify vm](#)
- [srvctl relocate vm](#)
- [srvctl remove vm](#)
- [srvctl start vm](#)
- [srvctl status vm](#)
- [srvctl stop vm](#)
- [G Oracle Clusterware C Application Program Interfaceエージェント・フレームワーク](#)
 - [エージェント・フレームワークのデータ型](#)
 - [エージェント・フレームワークのコンテキストの初期化および永続性](#)
 - [CおよびC++エントリ・ポイント機能のプロトタイプ](#)
 - [CおよびC++エントリ・ポイント・タイプおよびコード](#)
 - [CおよびC++エントリ・ポイント・ファンクションの戻り値](#)
 - [マルチスレッドに関する注意事項](#)
 - [非推奨のAPI](#)
 - [APIリファレンス](#)
 - [clsagfw_add_type\(\)](#)
 - [clsagfw_check_resource\(\)](#)
 - [clsagfw_create_attr_iterator\(\)](#)
 - [clsagfw_delete_cookie\(\)](#)
 - [clsagfw_exit2\(\)](#)
 - [clsagfw_get_attr_from_iterator\(\)](#)

- [clsagfw_get_attrvalue\(\)](#)
- [clsagfw_get_check_type\(\)](#)
- [clsagfw_get_cmdid\(\)](#)
- [clsagfw_get_cookie\(\)](#)
- [clsagfw_get_request_action_name\(\)](#)
- [clsagfw_get_resource_id\(\)](#)
- [clsagfw_get_resource_name\(\)](#)
- [clsagfw_get_retry_count\(\)](#)
- [clsagfw_get_type_name\(\)](#)
- [clsagfw_init\(\)](#)
- [clsagfw_is_cmd_timedout\(\)](#)
- [clsagfw_log\(\)](#)
- [clsagfw_modify_attribute\(\)](#)
- [clsagfw_reset_attr_iterator\(\)](#)
- [clsagfw_send_status2\(\)](#)
- [clsagfw_set_cookie\(\)](#)
- [clsagfw_set_entrpoint\(\)](#)
- [clsagfw_set_exitcb\(\)](#)
- [clsagfw_set_resource_state_label\(\)](#)
- [clsagfw_startup\(\)](#)
- [エージェントの例](#)
- [H Oracle Clusterware C Application Program Interface](#)
 - [Oracle Clusterwareのプログラミング・インタフェース\(C API\)について](#)
 - [概要](#)
 - [操作上のノート](#)
 - [非推奨のCLSCRS API](#)
 - [既存のCLSCRS APIの変更点](#)
 - [対話形式CLSCRS API](#)
 - [非対話形式CLSCRS API](#)
 - [コマンド評価API](#)
 - [clscrs_whatif_set_activepolicy](#)
 - [clscrs_whatif_fail_resource](#)
 - [clscrs_whatif_register_resource](#)
 - [clscrs_whatif_relocate_resource](#)
 - [clscrs_whatif_start_resource](#)
 - [clscrs_whatif_stop_resource](#)
 - [clscrs_whatif_register_serverpool](#)
 - [clscrs_whatif_unregister_serverpool](#)
 - [clscrs_whatif_add_server](#)
 - [clscrs_whatif_delete_server](#)
 - [clscrs_whatif_relocate_server](#)
 - [サーバーのカテゴリ化API](#)
 - [clscrs_servercategory_create](#)

- [clscrs_servercategory_destroy](#)
 - [clscrs_register_servercategory](#)
 - [clscrs_unregister_servercategory](#)
 - [clscrs_get_server_by_category](#)
 - [clscrs_register_server](#)
- [STAT3 API](#)
 - [clscrs_stat3](#)
- [その他のAPI](#)
 - [clscrs_get_error_details](#)
 - [clscrs_request_action](#)
 - [clscrs_restart_resource](#)
 - [clscrs_start_resource_in_pools](#)
 - [clscrs_stop_resource_in_pools](#)
- [I Oracle Cluster Registryユーティリティ・リファレンス](#)
 - [OCRCONFIGについて](#)
 - [OCRCONFIGコマンド・リファレンス](#)
 - [ocrconfig -add](#)
 - [ocrconfig -backuploc](#)
 - [ocrconfig -copy](#)
 - [ocrconfig -delete](#)
 - [ocrconfig -downgrade](#)
 - [ocrconfig -export](#)
 - [ocrconfig -import](#)
 - [ocrconfig -manualbackup](#)
 - [ocrconfig -overwrite](#)
 - [ocrconfig -repair](#)
 - [ocrconfig -replace](#)
 - [ocrconfig -restore](#)
 - [ocrconfig -showbackup](#)
 - [ocrconfig -showbackuploc](#)
 - [ocrconfig -upgrade](#)
 - [Oracle Cluster Registryのトラブルシューティングおよび診断出力](#)
 - [Oracle Cluster Registryのトラブルシューティング](#)
 - [OCRCHECKユーティリティの使用](#)
 - [構文](#)
 - [例](#)
 - [OCRDUMPユーティリティを使用したOracle Cluster Registryの内容の表示](#)
 - [OCRDUMPユーティリティの構文およびオプション](#)
 - [OCRDUMPユーティリティの例](#)
 - [OCRDUMPユーティリティの出力例](#)
- [J Oracle Clusterwareのトラブルシューティング](#)
 - [クラスタ・リソース・アクティビティ・ログを使用したクラスタ・リソースの障害の監視](#)
 - [crsctl query calog](#)

- [crsctl get calog maxsize](#)
 - [crsctl get calog retentiontime](#)
 - [crsctl set calog maxsize](#)
 - [crsctl set calog retentiontime](#)
- [Oracle Clusterwareの診断およびアラート・ログ・データ](#)
- [診断収集スクリプト](#)
- [Oracle拡張クラスタの記憶域の分割](#)
- [ローリング・アップグレードおよびドライバのインストールに関する問題](#)
- [ゾーン委任のテスト](#)
- [Oracle Clusterwareアラート](#)
 - [Diagnostic Record Unique IDを使用したアラート・メッセージ](#)
- [用語集](#)
- [索引](#)

表一覧

- [1-1 Oracle Clusterwareコンポーネントに関連付けられたプロセスおよびサービスのリスト](#)
- [2-1 ネットワーク・インタフェース、プライベートIPアドレスおよびプライベート・ホスト名の記憶域](#)
- [3-1 サーバー・プール属性](#)
- [3-2 サーバー・プール属性の構成例](#)
- [3-3 処理後のサーバー・プール構成](#)
- [3-4 サーバー構成属性](#)
- [3-5 サーバー状態の属性](#)
- [3-6 サーバー・カテゴリ属性](#)
- [7-1 クローニングのログ・ファイルおよび説明](#)
- [7-2 Oracleインベントリ・ディレクトリの場所の検索](#)
- [8-1 リソース・グループの依存性のタイプおよび修飾子](#)
- [8-2 エージェント・フレームワークの監視特性](#)
- [8-3 予想されるリソースの状態](#)
- [A-1 CVUの様々な検証の実行](#)
- [A-2 非推奨およびサポート対象外のCluvfyコマンド](#)
- [A-3 cluvfy comp acfsコマンドのパラメータ](#)
- [A-4 cluvfy comp admpvコマンドのパラメータ](#)
- [A-5 cluvfy comp asmコマンドのパラメータ](#)
- [A-6 cluvfy comp baselineコマンドのパラメータ](#)
- [A-7 cluvfy comp clocksyncコマンドのパラメータ](#)
- [A-8 cluvfy comp cluコマンドのパラメータ](#)
- [A-9 cluvfy comp clumgrコマンドのパラメータ](#)
- [A-10 cluvfy comp containerhostコマンドのパラメータ](#)
- [A-11 cluvfy comp clumgrコマンドのパラメータ](#)
- [A-12 cluvfy comp dhcpコマンドのパラメータ](#)
- [A-13 cluvfy comp dnsコマンドのパラメータ](#)
- [A-14 cluvfy comp gnsコマンドのパラメータ](#)
- [A-15 cluvfy comp gnpnコマンドのパラメータ](#)
- [A-16 cluvfy comp healthcheckコマンドのパラメータ](#)
- [A-17 cluvfy comp nodeappコマンドのパラメータ](#)
- [A-18 cluvfy comp nodeconコマンドのパラメータ](#)
- [A-19 cluvfy comp nodereachコマンドのパラメータ](#)
- [A-20 cluvfy comp ocrコマンドのパラメータ](#)
- [A-21 cluvfy comp ocrコマンドのパラメータ](#)
- [A-22 cluvfy comp ocrコマンドのパラメータ](#)
- [A-23 cluvfy comp peerコマンドのパラメータ](#)
- [A-24 cluvfy comp softwareコマンドのパラメータ](#)
- [A-25 cluvfy comp spaceコマンドのパラメータ](#)
- [A-26 cluvfy comp ssaコマンドのパラメータ](#)
- [A-27 cluvfy comp sysコマンドのパラメータ](#)
- [A-28 cluvfy comp vdiskコマンドのパラメータ](#)

- [A-29 cluvfy comp vitalコマンドのパラメータ](#)
- [A-30 cluvfy stage acfscfgコマンドのパラメータ](#)
- [A-31 cluvfy stage appclusterコマンドのパラメータ](#)
- [A-32 cluvfy stage cfsコマンドのパラメータ](#)
- [A-33 cluvfy stage \[-pre | -post\] crsinstコマンドのパラメータ](#)
- [A-34 cluvfy stage -pre dbcfgコマンドのパラメータ](#)
- [A-35 cluvfy stage -pre dbinstコマンドのパラメータ](#)
- [A-36 cluvfy stage hacfgコマンドのパラメータ](#)
- [A-37 cluvfy stage hwosコマンドのパラメータ](#)
- [A-38 cluvfy stage nodeaddコマンドのパラメータ](#)
- [A-39 cluvfy stage nodedelコマンドのパラメータ](#)
- [A-40 cluvfy stage osupgコマンドのパラメータ](#)
- [A-41 cluvfy stage patchコマンドのパラメータ](#)
- [C-1 OLSNODESのコマンド・オプション](#)
- [D-1 OIFCFGコマンド](#)
- [E-1 非推奨のCRSCTLコマンドおよびその代替](#)
- [E-2 crsctl add resourceコマンドのパラメータ](#)
- [E-3 crsctl delete resourceコマンドのパラメータ](#)
- [E-4 crsctl eval add resourceコマンドのパラメータ](#)
- [E-5 crsctl eval fail resourceコマンドのパラメータ](#)
- [E-6 crsctl eval relocate resourceコマンドのパラメータ](#)
- [E-7 crsctl eval modify resourceコマンドのパラメータ](#)
- [E-8 crsctl eval start resourceコマンドのパラメータ](#)
- [E-9 crsctl eval stop resourceコマンドのパラメータ](#)
- [E-10 crsctl getperm resourceコマンドのパラメータ](#)
- [E-11 crsctl modify resourceコマンドのパラメータ](#)
- [E-12 crsctl relocate resourceコマンドのパラメータ](#)
- [E-13 crsctl restart resourceコマンドのパラメータ](#)
- [E-14 crsctl setperm resourceコマンドのパラメータ](#)
- [E-15 crsctl start resourceコマンドのパラメータ](#)
- [E-16 crsctl status resourceコマンドのパラメータ](#)
- [E-17 crsctl stop resourceコマンドのパラメータ](#)
- [E-18 crsctl add resourcegroupコマンドのパラメータ](#)
- [E-19 crsctl check resourcegroupコマンドのパラメータ](#)
- [E-20 crsctl delete resourcegroupコマンドのパラメータ](#)
- [E-21 crsctl eval add resourcegroupコマンドのパラメータ](#)
- [E-22 crsctl eval fail resourcegroupコマンドのパラメータ](#)
- [E-23 crsctl eval relocate resourcegroupコマンドのパラメータ](#)
- [E-24 crsctl eval start resourcegroupコマンドのパラメータ](#)
- [E-25 crsctl eval stop resourcegroupコマンドのパラメータ](#)
- [E-26 crsctl export resourceコマンドのパラメータ](#)
- [E-27 crsctl modify resourcegroupコマンドのパラメータ](#)
- [E-28 crsctl relocate resourcegroupコマンドのパラメータ](#)

- [E-29 crsctl restart resourcegroupコマンドのパラメータ](#)
- [E-30 crsctl start resourcegroupコマンドのパラメータ](#)
- [E-31 crsctl status resourcegroupコマンドのパラメータ](#)
- [E-32 crsctl stop resourcegroupコマンドのパラメータ](#)
- [E-33 crsctl add resourcegrouptypeコマンドのパラメータ](#)
- [E-34 crsctl delete resourcegrouptypeコマンドのパラメータ](#)
- [E-35 crsctl set tracefileoptsコマンドのパラメータ](#)
- [E-36 crsctl add typeコマンドのパラメータ](#)
- [E-37 crsctl getperm typeコマンドのパラメータ](#)
- [E-38 crsctl modify typeコマンドのパラメータ](#)
- [E-39 crsctl setperm typeコマンドのパラメータ](#)
- [E-40 crsctl status typeコマンドのパラメータ](#)
- [E-41 crsctl request actionコマンドのパラメータ](#)
- [E-42 crsctl add categoryコマンドのパラメータ](#)
- [E-43 crsctl modify categoryコマンドのパラメータ](#)
- [E-44 crsctl status categoryコマンドのパラメータ](#)
- [E-45 crsctl start clusterコマンドのパラメータ](#)
- [E-46 crsctl stop clusterコマンドのパラメータ](#)
- [E-47 crsctl modify cluster siteコマンドのパラメータ](#)
- [E-48 crsctl start crsコマンドのパラメータ](#)
- [E-49 crsctl add crs administratorコマンドのパラメータ](#)
- [E-50 crsctl delete crs administratorのパラメータ](#)
- [E-51 crsctl set crs autostartコマンドのパラメータ](#)
- [E-52 crsctl add css votediskコマンドのパラメータ](#)
- [E-53 crsctl delete css votediskコマンドのパラメータ](#)
- [E-54 crsctl discover dhcpコマンドのパラメータ](#)
- [E-55 crsctl get clientid dhcpコマンドのパラメータ](#)
- [E-56 crsctl release dhcpコマンドのパラメータ](#)
- [E-57 crsctl request dhcpコマンドのパラメータ](#)
- [E-58 crsctl query dnsコマンドのパラメータ](#)
- [E-59 crsctl start ipコマンドのパラメータ](#)
- [E-60 crsctl stop ipコマンドのパラメータ](#)
- [E-61 crsctl create member_cluster_configurationコマンドのパラメータ](#)
- [E-62 crsctl add policyコマンドのパラメータ](#)
- [E-63 crsctl eval activate policyコマンドのパラメータ](#)
- [E-64 crsctl modify policyコマンドのパラメータ](#)
- [E-65 crsctl status policyコマンドのパラメータ](#)
- [E-66 crsctl modify policysetコマンドのパラメータ](#)
- [E-67 crsctl check resourceコマンドのパラメータ](#)
- [E-68 crsctl relocate resourceコマンドのパラメータ](#)
- [E-69 crsctl eval add serverコマンドのパラメータ](#)
- [E-70 crsctl eval delete serverコマンドのパラメータ](#)
- [E-71 crsctl eval relocate serverのパラメータ](#)

- [E-72 crsctl modify serverコマンドのパラメータ](#)
- [E-73 crsctl relocate serverコマンドのパラメータ](#)
- [E-74 crsctl status serverコマンドのパラメータ](#)
- [E-75 crsctl add serverpoolコマンドのパラメータ](#)
- [E-76 crsctl eval add serverpoolコマンドのパラメータ](#)
- [E-77 crsctl eval delete serverpoolコマンドのパラメータ](#)
- [E-78 crsctl eval modify serverpoolコマンドのパラメータ](#)
- [E-79 crsctl getperm serverpoolコマンドのパラメータ](#)
- [E-80 crsctl modify serverpoolコマンドのパラメータ](#)
- [E-81 crsctl setperm serverpoolコマンドのパラメータ](#)
- [E-82 crsctl status serverpoolコマンドのパラメータ](#)
- [E-83 crsctl query socket udpコマンドのパラメータ](#)
- [E-84 crsctl start testdnsコマンドのパラメータ](#)
- [E-85 crsctl status testdnsコマンドのパラメータ](#)
- [E-86 crsctl stop testdnsコマンドのパラメータ](#)
- [E-87 crsctl replace votediskコマンドのパラメータ](#)
- [E-88 crsctl add walletコマンドのパラメータ](#)
- [E-89 crsctl delete walletコマンドのパラメータ](#)
- [E-90 crsctl modify walletコマンドのパラメータ](#)
- [E-91 crsctl query walletコマンドのパラメータ](#)
- [E-92 CRS、CSSおよびEVMモジュールのコンポーネント](#)
- [E-93 クラスタ同期サービス\(CSS\)モジュールおよびその機能](#)
- [E-94 通信\(COMM\)モジュールおよびその機能](#)
- [E-95 Oracle Clusterware \(CRS\)モジュールおよびその機能](#)
- [E-96 Oracle Cluster Registry \(OCR\)コンポーネント名](#)
- [E-97 OCRCONFIGモジュールおよびその機能](#)
- [F-1 SRVCTLオブジェクト名の文字列制限](#)
- [F-2 SRVCTLコマンドで非推奨になった単一文字パラメータ](#)
- [F-3 SRVCTLで非推奨になったコマンドおよびパラメータ](#)
- [F-4 オブジェクト・キーワードおよび短縮形](#)
- [F-5 srvctl add asmコマンドのパラメータ](#)
- [F-6 srvctl config asmコマンドのパラメータ](#)
- [F-7 srvctl disable asmコマンドのパラメータ](#)
- [F-8 srvctl enable asmコマンドのパラメータ](#)
- [F-9 srvctl relocate asmコマンドのパラメータ](#)
- [F-10 srvctl remove asmコマンドのパラメータ](#)
- [F-11 srvctl setenv asmコマンドのパラメータ](#)
- [F-12 srvctl start asmコマンドのパラメータ](#)
- [F-13 srvctl status asmコマンドのパラメータ](#)
- [F-14 srvctl stop asmコマンドのパラメータ](#)
- [F-15 srvctl add asmnetworkコマンドのパラメータ](#)
- [F-16 srvctl modify asmnetworkコマンドのパラメータ](#)
- [F-17 srvctl remove asmnetworkコマンドのパラメータ](#)

- [F-18 srvctl add cdpコマンドのパラメータ](#)
- [F-19 srvctl modify cdpコマンドのパラメータ](#)
- [F-20 srvctl relocate cdpコマンドのパラメータ](#)
- [F-21 srvctl start cdpコマンドのパラメータ](#)
- [F-22 srvctl add cdpproxyコマンドのパラメータ](#)
- [F-23 srvctl config cdpproxyコマンドのパラメータ](#)
- [F-24 srvctl disable cdpproxyコマンドのパラメータ](#)
- [F-25 srvctl enable cdpproxyコマンドのパラメータ](#)
- [F-26 srvctl modify cdpproxyコマンドのパラメータ](#)
- [F-27 srvctl relocate cdpproxyコマンドのパラメータ](#)
- [F-28 srvctl remove cdpproxyコマンドのパラメータ](#)
- [F-29 srvctl start cdpproxyコマンドのパラメータ](#)
- [F-30 srvctl status cdpproxyコマンドのパラメータ](#)
- [F-31 srvctl stop cdpproxyコマンドのパラメータ](#)
- [F-32 srvctl add exportfsコマンドのパラメータ](#)
- [F-33 srvctl config exportfsコマンドのパラメータ](#)
- [F-34 srvctl modify exportfsコマンドのパラメータ](#)
- [F-35 srvctl remove exportfsコマンドのパラメータ](#)
- [F-36 srvctl start exportfsコマンドのパラメータ](#)
- [F-37 srvctl status exportfsコマンドのパラメータ](#)
- [F-38 srvctl stop exportfsコマンドのパラメータ](#)
- [F-39 srvctl add filesystemコマンドのパラメータ](#)
- [F-40 srvctl config filesystemコマンドのパラメータ](#)
- [F-41 srvctl disable filesystemコマンドのパラメータ](#)
- [F-42 srvctl enable filesystemコマンドのパラメータ](#)
- [F-43 srvctl modify filesystemコマンドのパラメータ](#)
- [F-44 srvctl predict filesystemコマンドのパラメータ](#)
- [F-45 srvctl remove filesystemコマンドのパラメータ](#)
- [F-46 srvctl start filesystemコマンドのパラメータ](#)
- [F-47 srvctl status filesystemコマンドのパラメータ](#)
- [F-48 srvctl stop filesystemコマンドのパラメータ](#)
- [F-49 srvctl add gnsコマンドのパラメータ](#)
- [F-50 srvctl config gnsコマンドのパラメータ](#)
- [F-51 srvctl export gnsコマンドのパラメータ](#)
- [F-52 srvctl modify gnsコマンドのパラメータ](#)
- [F-53 srvctl start gnsコマンドのパラメータ](#)
- [F-54 srvctl update gnsコマンドのパラメータ](#)
- [F-55 srvctl add havipコマンドのパラメータ](#)
- [F-56 srvctl config havipコマンドのパラメータ](#)
- [F-57 srvctl disable havipコマンドのパラメータ](#)
- [F-58 srvctl enable havipコマンドのパラメータ](#)
- [F-59 srvctl modify havipコマンドのパラメータ](#)
- [F-60 srvctl relocate havipコマンドのパラメータ](#)

- [F-61 `srvctl remove havip`コマンドのパラメータ](#)
- [F-62 `srvctl start havip`コマンドのパラメータ](#)
- [F-63 `srvctl stop havip`コマンドのパラメータ](#)
- [F-64 `srvctl add ioserver`コマンドのパラメータ](#)
- [F-65 `srvctl modify ioserver`コマンドのパラメータ](#)
- [F-66 `srvctl relocate ioserver`コマンドのパラメータ](#)
- [F-67 `srvctl setenv ioserver`コマンドのパラメータ](#)
- [F-68 `srvctl status ioserver`のパラメータ](#)
- [F-69 `srvctl modify mgmtldb`コマンドのパラメータ](#)
- [F-70 `srvctl remove mgmtldb`コマンドのパラメータ](#)
- [F-71 `srvctl setenv mgmtldb`コマンドのパラメータ](#)
- [F-72 `srvctl start mgmtldb`コマンドのパラメータ](#)
- [F-73 `srvctl stop mgmtldb`コマンドのパラメータ](#)
- [F-74 `srvctl add mgmtlsnr`コマンドのパラメータ](#)
- [F-75 `srvctl setenv mgmtlsnr`コマンドのパラメータ](#)
- [F-76 `srvctl add mountfs`コマンドのパラメータ](#)
- [F-77 `srvctl disable mountfs`コマンドのパラメータ](#)
- [F-78 `srvctl enable mountfs`コマンドのパラメータ](#)
- [F-79 `srvctl modify mountfs`コマンドのパラメータ](#)
- [F-80 `srvctl start mountfs`コマンドのパラメータ](#)
- [F-81 `srvctl stop mountfs`コマンドのパラメータ](#)
- [F-82 `srvctl add ovmm`コマンドのパラメータ](#)
- [F-83 `srvctl modify ovmm`コマンドのパラメータ](#)
- [F-84 `srvctl modify qosmserver`コマンドのパラメータ](#)
- [F-85 `srvctl add rhpcclient`コマンドのパラメータ](#)
- [F-86 `srvctl modify rhpcclient`コマンドのパラメータ](#)
- [F-87 `srvctl add rhpserver`コマンドのパラメータ](#)
- [F-88 `srvctl modify rhpserver`コマンドのパラメータ](#)
- [F-89 `srvctl remove rhpserver`コマンドのパラメータ](#)
- [F-90 `srvctl add vm`コマンドのパラメータ](#)
- [F-91 `srvctl check vm`コマンドのパラメータ](#)
- [F-92 `srvctl disable vm`コマンドのパラメータ](#)
- [F-93 `srvctl enable vm`コマンドのパラメータ](#)
- [F-94 `srvctl modify vm`コマンドのパラメータ](#)
- [F-95 `srvctl relocate vm`コマンドのパラメータ](#)
- [F-96 `srvctl start vm`コマンドのパラメータ](#)
- [F-97 `srvctl status vm`コマンドのパラメータ](#)
- [F-98 `srvctl stop vm`コマンドのパラメータ](#)
- [G-1 エージェント・フレームワークで使用するOracleデータ型](#)
- [G-2 エントリ・ポイント・タイプ](#)
- [G-3 非推奨のエージェント・フレームワークAPI](#)
- [H-1 非推奨のCLSCRS API](#)
- [H-2 Oracle Clusterwareの対話形式CLSCRS APIのサマリー](#)

- [H-3 非対話形式CLSCRS API](#)
- [I-1 Oracle Cluster Registryの一般的な問題および解決策](#)
- [I-2 OCRDUMPのオプションおよびオプションの説明](#)
- [J-1 crsctl query calogコマンドのパラメータ](#)
- [J-2 クラスタ・リソース・アクティビティ・ログのフィールド](#)
- [J-3 diagcollection.plスクリプトのパラメータ](#)

はじめに

『Oracle Clusterware管理およびデプロイメント・ガイド』では、Oracle Clusterwareのアーキテクチャと、この製品の概要について説明します。また、Oracle Clusterwareの管理とデプロイメントについても説明します。

このマニュアルの情報は、特に指定がないかぎり、すべてのプラットフォーム上で動作するOracle Clusterwareに適用されます。このマニュアルの内容は、他のOracleマニュアルにおけるシングル・インスタンスのOracle Databaseの管理およびデプロイメントに関する項目の説明を補足するものです。必要に応じて、プラットフォーム固有のマニュアルも参照してください。ここでは、次の内容について説明します。

- [対象読者](#)
- [ドキュメントのアクセシビリティ](#)
- [関連ドキュメント](#)
- [表記規則](#)

対象読者

Oracle Clusterware管理およびデプロイメント・ガイドでは、データベース管理者、ネットワーク管理者およびシステム管理者を対象とし、Oracle Clusterwareのアーキテクチャの説明、および管理とデプロイメントに関するトピックを提供しています。

このガイドは、Oracle Real Application Clusters (Oracle RAC)データベースを管理する管理者を対象としています。

ドキュメントのアクセシビリティ

オラクルのアクセシビリティについての詳細情報は、Oracle Accessibility ProgramのWebサイト (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>)を参照してください。

Oracle Supportへのアクセス

サポートをご契約のお客様には、My Oracle Supportを通して電子支援サービスを提供しています。詳細情報は (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>)か、聴覚に障害のあるお客様は (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>)を参照してください。

関連ドキュメント

詳細は、次のOracleドキュメントを参照してください。

- プラットフォーム固有のOracle ClusterwareおよびOracle RACのインストール・ガイド
Oracle Database 11gのプラットフォーム固有の各インストール・メディアには、プラットフォーム固有のOracle ClusterwareおよびOracle RACのインストールおよび構成ガイド(HTML形式およびPDF形式)が含まれています。これらのインストール・ガイドには、Oracle ClusterwareおよびOracle RACが動作するUNIX、LinuxおよびWindowsの様々なプラットフォームに対するインストール前、インストールおよびインストール後の情報が記載されています。
- [Oracle TopLink Java APIリファレンス](#)
このドキュメントは重要であり、インスタンスの管理、チューニング、バックアップとリカバリ、サービスの管理などについて説明しています。

- [Oracle Database 2日でデータベース管理者](#)
- [Oracle Database管理者ガイド](#)
- [Oracle Database Net Services管理者ガイド](#)
- [Oracle Database管理者リファレンス for Linux and UNIX-Based Operating Systems](#)
- [Oracle Databaseエラー・メッセージ](#)

表記規則

このドキュメントでは、次のテキスト表記規則が使用されます：

規則	意味
boldface	太字体は、アクションに関連付けられたグラフィカル・ユーザー・インタフェース要素や、本文または用語集で定義されている用語を示します。
<i>italic</i>	イタリック体は、ブック・タイトル、強調、またはユーザーが特定の値を指定するプレースホルダー変数を示します。
<code>monospace</code>	等幅体は、段落内のコマンド、URL、サンプル内のコード、画面に表示されるテキスト、またはユーザーが入力するテキストを示します。

『Oracle Clusterware管理およびデプロイメント・ガイド』に関するこのリリースにおける変更

この章では、Oracle Database 19cおよび18c用のOracle Clusterwareの新機能について説明します。

Oracle Clusterwareリリース19cの変更点

Oracle Clusterware 19cのOracle Clusterware管理およびデプロイメント・ガイドの新機能は次のとおりです。

Oracle Clusterware 19cのSRVCTLの変更

Oracle Clusterware 19cには、既存のコマンドに対する構文の変更やOracle Automatic Storage Management (Oracle ASM)を管理するコマンドなど、サーバー制御ユーティリティ(SRVCTL)への変更が含まれます。

SRVCTLは、Oracle Real Application Clusters (Oracle RAC)およびOracle Clusterwareの管理に使用するツールの1つです。

関連項目:

[サーバー制御\(SRVCTL\)コマンド・リファレンス](#)

高速ホーム・プロビジョニングの名称変更

このリリースでは、これまで高速ホーム・プロビジョニングと呼ばれていた機能の名前が、フリート・パッチ適用およびプロビジョニングに変更されています。機能に対する変更はなく、RHPCTLユーティリティは、フリート・パッチ適用およびプロビジョニング操作の管理に使用するツールのままです。

PDBの自動的なパッチ適用および再配置

統合されたOracleマルチテナント環境では個々のプラガブル・データベースにパッチを適用できるため、コンテナ・データベース全体ではなく、特定のプラガブル・データベースにのみバグ修正のパッチを適用できます。ファイングレイン・シングルインスタンスのプラガブル・データベース・パッチ適用によって、広範な変更の採用(バグ修正など)で発生する可能性のあるリスクが減少し、必要な場所のみ変更を行う影響も軽減されます。

フリート・パッチ適用およびプロビジョニングを使用した停止時間0のOracle Grid Infrastructureパッチ適用

停止時間0のOracle Grid Infrastructureパッチ適用では、Oracle Real Application Clusters (Oracle RAC)データベース・インスタンスに影響を与えることなく、Oracle Grid Infrastructureの個別パッチを適用できます。フリート・パッチ適用およびプロビジョニングを使用して、クラスタのノードごとに1つずつパッチを適用します。この機能は、2つ以上のノードを持つすべてのOracle RACクラスタで使用できますが、現在は個別パッチにのみ適用されます(リリース更新またはリリース更新リビジョンには適用されない)。

フリート・パッチ適用およびプロビジョニングを使用して、データベース・インスタンスの停止時間を0にしてOracle Grid Infrastructureパッチを適用すると、ユーザーに対する影響と、Oracle RACデータベース・インスタンスからのサービスが中断なくなります。以前のデータベース・リリースでは、Oracle Grid Infrastructureパッチを適用する前にデータベース・インスタンスを停止する必要があり、エンタープライズ操作に明らかに影響していました。

Oracle Grid Infrastructureのアップグレードの自動トランザクション・ドレイン

Oracle Grid Infrastructureアップグレードの自動トランザクション・ドレインでは、データベース・サービス構成に従って、一度

に1ノードずつ、データベース・インスタンスに対するトランザクションの自動ドレインが行われます。トランザクション・ドレイン機能は、データベース・サービス設計の不可欠な部分であり、Oracle Grid Infrastructureのローリング・パッチ適用に自動的に統合されています。

フリー・パッチ適用およびプロビジョニングを使用した、ローリング・パッチ適用中のデータベース・トランザクションの自動かつ協調的なドレインにより、パッチ適用操作の影響が軽減されます。ユーザー・トランザクションがドレインされると、クラスタ上の特定のノードに対するパッチ適用操作を完了でき、その後でインスタンスおよびサービスがローカルで再起動し、新しい接続が確立されます。接続は、パッチ適用操作の前にクラスタ内の次のノードに進みます。

Oracle Restartのパッチ適用およびアップグレード

フリー・パッチ適用およびプロビジョニングを使用して、Oracle Restartにパッチを適用しアップグレードします。以前のリリースのOracle Restart環境は、ユーザーがパッチ適用およびアップグレードの操作を実行する必要があり、多くの場合、手動操作が必要でした。フリー・パッチ適用およびプロビジョニングでは、これらの手順が自動化されます。

フリー・パッチ適用およびプロビジョニングを使用してOracle Restartにパッチ適用とアップグレードを行うと、Oracle RACデータベース・インストールで実装されるプロセスが自動化および標準化されます。これにより、Oracle Restartデプロイメントが多い場合は特に、操作の必要性和リスクも低減されます。

CRSCTLを使用したTLS暗号の仕様のサポート

CRSCTLユーティリティの拡張機能によって、トランスポート層セキュリティ(TLS)暗号の仕様のサポートが追加されます。

関連項目:

[トランスポート層セキュリティ暗号スイート管理](#)

セキュアなクラスタ通信

セキュアなクラスタ通信を単一ネットワーク・サポートと併用すると、一般的なセキュリティに対する脅威からクラスタ・インターコネクトが保護されます。セキュアなクラスタ通信には、メッセージ・ダイジェスト・メカニズム、ファジングに対する保護、およびトランスポート層セキュリティ(TLS)を使用したクラスタ・メンバー間のプライバシーとデータ整合性の提供が含まれます。

クラスタ・インターコネクトのセキュリティ強化は、新しいOracle Grid Infrastructure 19cデプロイメント、またはOracle Grid Infrastructure 19cへのアップグレードの一部として、自動的に呼び出されます。データベース管理者やクラスタ管理者は、この機能の構成を変更する必要はありません。

OCRおよび投票ディスクの直接ファイル配置の再サポート

Oracle Grid Infrastructure 19c以上では、共有ファイル・システム上の直接OCRおよび投票ディスク・ファイルの配置のサポート終了は、Oracleスタンドアロン・クラスタに対して廃棄されます。Oracleドメイン・サービス・クラスタの場合、共有ファイル・システムでホストされているファイルに加えて、OCRおよび投票ファイルをOracle Automatic Storage Management (Oracle ASM)に配置し、ASMディスクとして使用するという要件は変わりません。

Oracle Grid Infrastructure 12cリリース2 (12.2)では、直接、共有ファイル・システム上でOracle Grid InfrastructureのOracle Cluster Registry (OCR)および投票ファイルの配置がサポートされなくなることが発表されました。このサポート終了は現在破棄されました。Oracle Grid Infrastructure 19c (19.3)以上では、Oracleスタンドアロン・クラスタとともに、OCRおよび投票ディスク・ファイルを共有ファイル・システムに直接配置できます。

関連項目:

グリッド・インフラストラクチャ管理リポジトリのオプションのインストール

Oracle Grid Infrastructure 19c以上では、グリッド・インフラストラクチャ管理リポジトリ(GIMR)は、Oracleスタンドアロン・クラスタの新規インストールでオプションです。Oracleドメイン・サービス・クラスタでは、GIMRをサービス・コンポーネントとしてインストールする必要があります。

GIMRに含まれているデータは、適用されている機械学習に基づく予防診断の基礎となり、Oracle Real Application Clusters (RAC)データベースの可用性の向上に役立ちます。GIMRをオプションでインストールすると、特にテスト・システムや開発システムのインストール時に、記憶領域の管理および高速デプロイメントをより柔軟に行うことができます。

関連項目:

[グリッド・インフラストラクチャ管理リポジトリの概要](#)

Oracle Clusterware 19cで非推奨となった機能

次の機能は、Oracle Clusterware 19cでは非推奨であり、将来のリリースではサポートされなくなる可能性があります。

Addnodeスクリプトの非推奨

addnodeスクリプトは、Oracle Grid Infrastructure 19cでは非推奨です。クラスタにノードを追加する機能は、インストーラ・ウィザードで使用できます。

addnodeスクリプトは今後のリリースで削除される可能性があります。addnodeスクリプト(addnode.shまたはaddnode.bat)を使用するかわりに、インストーラ・ウィザードを使用してノードを追加してください。インストーラ・ウィザードでは、addnodeスクリプト以上に多くの機能が拡張されています。インストーラ・ウィザードを使用すると、すべてのソフトウェア・ライフサイクル操作を1つのツールに統合することで管理が簡略化されます。

clone.plスクリプトの非推奨

clone.plスクリプトは今後のリリースで削除される可能性があります。clone.plスクリプトを使用するかわりに、インストーラ・ウィザードを使用して、抽出されたゴールド・イメージをホームとしてインストールすることをお勧めします。

Oracle Clusterwareとベンダー・クラスタウェアの統合の非推奨

ベンダーまたはサード・パーティのクラスタウェアとOracle Clusterwareの統合は、Oracle Database 19cで非推奨になりました。

Oracle Clusterwareとベンダー・クラスタウェアの統合は非推奨であり、将来のリリースでサポートされなくなる可能性があります。特定のクラスタ機能を非推奨にし、制限付きの採用とすることにより、オラクル社では、すべての機能の中核となるスケーリング、可用性および管理性の向上に注力できます。異なるクラスタ・ソリューション間の統合がない場合、システムでクラスタ・ソリューションの競合の問題が生じることがあります。これは、独立した各クラスタ・ソリューションが、特定の障害が発生した場合に実行する必要がある修正処理を個別に決定する可能性があるという事実を示しています。このため、いつの時点においても1つのソリューションのみをアクティブにすることをお勧めします。次のソフトウェアまたはハードウェアのアップグレードでは、この理由により、ベンダー・クラスタ・ソリューションを使用しない構成に移行することをお勧めします。

Oracle Clusterwareを使用したブラック・ボックス仮想マシン管理の非推奨

Oracle Clusterwareを使用した仮想マシン(VM)リソースの直接管理は、Oracle Database 19cで非推奨となり、将来のリリースで削除される可能性があります。

Oracle Clusterware 19cを使用した物理ハードウェア上のブラックボックスOracle Grid Infrastructure仮想マシン (GIVM)のOracle Clusterware管理の使用が引き続きサポートされているため、可用性が向上し、仮想マシンの管理が容易になります。

Oracle Clusterware 19cでサポートが終了した機能

Oracle Clusterware 19cでサポート対象外になった機能は、次のとおりです。

フレックス・クラスタ・アーキテクチャでのリーフ・ノードのサポート終了

リーフ・ノードはOracle Grid Infrastructure 19cのOracle Flex Clusterアーキテクチャでサポートされなくなりました。

Oracle Grid Infrastructure 19c (19.1)以上のリリースでは、Oracle Flex Clusterのすべてのノードはハブ・ノードとして機能します。Oracle Flex Clusterアーキテクチャの元の実装でリーフ・ノードによって提供された機能は、ハブ・ノードで容易に提供できます。このため、リーフ・ノードはサポートされなくなりました。

Standard Edition 2 (SE2)データベース・エディション用のOracle Real Application Clustersのサポート終了

Oracle Database 19c以上では、Oracle Real Application Clusters (Oracle RAC)はOracle Database Standard Edition 2 (SE2)でサポートされなくなりました。

Oracle Real Application Clusters (Oracle RAC)機能を使用するOracle Database Standard Editionデータベースを以前のリリースからOracle Database 19cにアップグレードすることはできません。これらのデータベースをOracle Database 19cにアップグレードするには、アップグレードを開始する前にOracle RAC機能を削除するか、Oracle Database Standard EditionからOracle Database Enterprise Editionにアップグレードします。

アップグレード後のシステムの再構成方法を含む各ステップの詳細は、[My Oracle Support](#)ノート2504078.1「Oracle Database Standard Edition 19cでのOracle Real Application Clusters(RAC)のサポート終了」を参照してください。

1 Oracle Clusterwareの概要

Oracle Clusterwareの概念とコンポーネントです。

Oracle Clusterwareを使用すると、複数のサーバーが相互に通信することにより、1つの集合単位として機能するように見えます。このようなサーバーの組合せは、一般にクラスタと呼ばれます。これらのサーバーはスタンドアロン・サーバーですが、各サーバーには他のサーバーと通信する追加のプロセスがあります。このように、個々のサーバーはアプリケーションおよびエンド・ユーザーから1つのシステムとして認識されます。

この章の内容は次のとおりです。

- [Oracle Clusterwareの概要](#)
- [Oracle Clusterwareのシステム要件の理解](#)
- [Oracle Clusterwareのプラットフォーム固有のソフトウェア・コンポーネントの概要](#)
- [Oracle Databaseの高可用性オプション](#)
- [Oracle Clusterwareのインストールの概要](#)
- [Oracle Clusterwareのアップグレードおよびパッチ適用の概要](#)
- [グリッド・インフラストラクチャ管理リポジトリの概要](#)
- [ドメイン・サービス・クラスタの概要](#)
- [Oracle Clusterware環境の管理の概要](#)
- [コマンドの評価の概要](#)
- [グリッド環境でのOracle Clusterwareのクローニングおよび拡張の概要](#)
- [Oracle Clusterwareの高可用性フレームワークおよびAPIの概要](#)
- [クラスタの時間管理の概要](#)

Oracle Clusterwareの概要

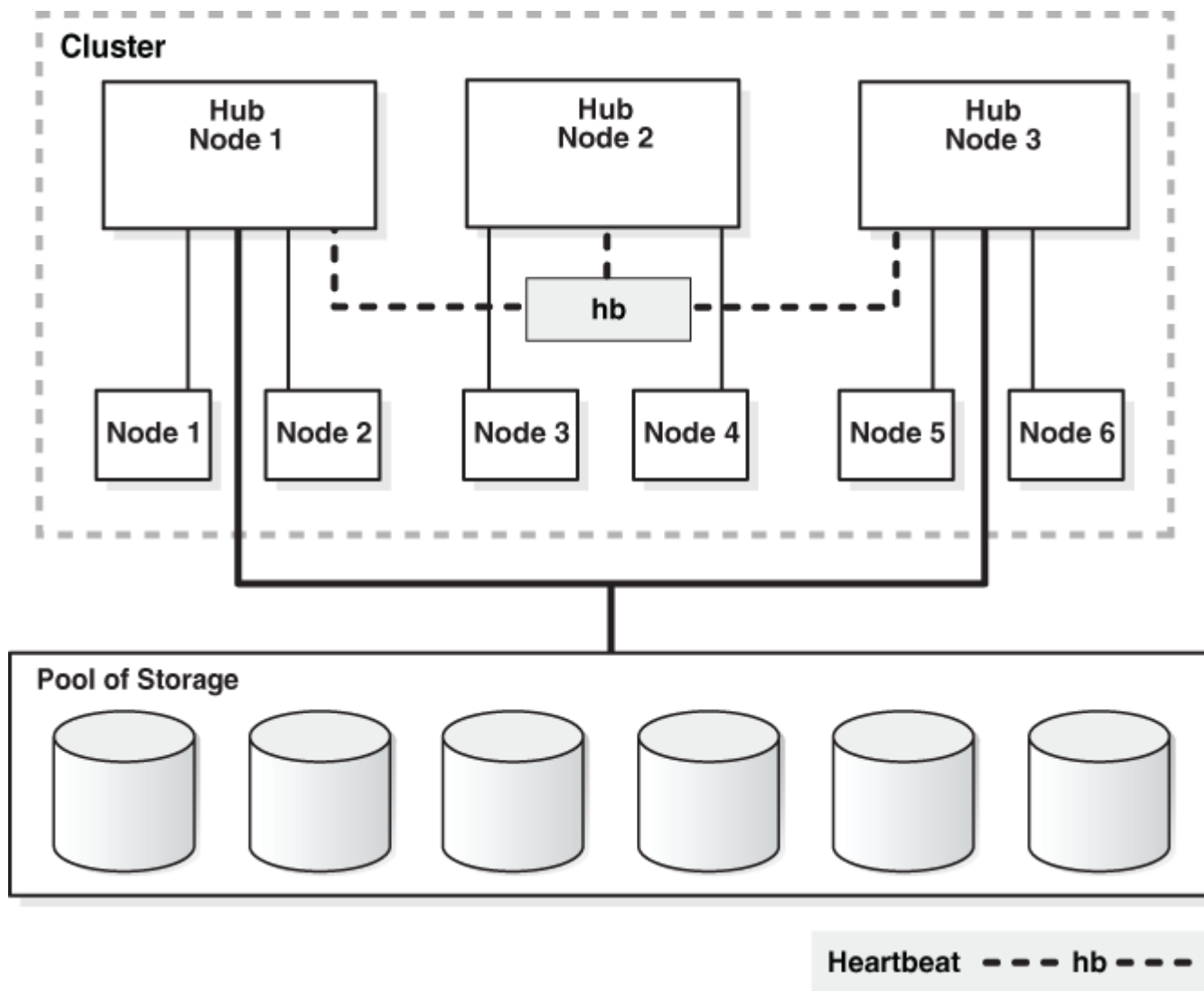
Oracle Clusterwareは、統合環境に包括的な複数層の高可用性およびリソース管理を提供するポータブル・クラスタ・ソフトウェアです。独立したサーバーのクラスタリングが単一システムとして連携するようにサポートします。

Oracle Clusterwareは、Oracle Real Application Clusters (Oracle RAC)の統合された基盤であり、任意の主要なプラットフォーム上のすべてのアプリケーションに対する高可用性およびリソース管理フレームワークです。Oracle Clusterwareは最初、Oracle複数インスタンス・データベース(Oracle RAC)に必要なクラスタ技術として、Oracle Database 10gリリース1 (10.1)とともにリリースされました。目的は、クラウドでOracle Clusterwareを活用して、必要な状況でエンタープライズ・クラスの回復性を提供すること、および必要な場所と時間に計算リソースの動的なオンライン割当てを提供することです。

Oracle Flex Cluster

Oracle Clusterware 12c リリース2 (12.2)では、すべてのクラスタがOracle Flex Clusterとして構成されます。つまり、クラスタは1つ以上の[ハブ・ノード](#)を使用して構成され、多数のノードをサポートできます。以前のバージョンのOracle Clusterwareで現在構成されているクラスタは、Oracle Flex ASM (Oracle Flex Clusterの要件)のアクティブ化を含むアップグレード・プロセスの一部として、適切に変換されます。

図1-1 Oracle Clusterwareの構成



ハブ・ノードは共有記憶域と緊密に接続し、直接アクセスします。従来より、これらはOracle RACまたはOracle RAC One データベース・インスタンスのホストとしてデプロイされます。クラスタの他のノードは、共有記憶域への直接アクセスが必要なく、かわりにハブ・ノードを介してデータにアクセスする点が、ハブ・ノードと異なります。

Oracle Flex Cluster内のすべてのノードは、単一のOracle Grid Infrastructureクラスタに属します。このアーキテクチャでは、様々なサービス・レベル、負荷、障害のレスポンス、およびリカバリに対処するために、アプリケーション・ニーズに基づいてリソースのデプロイメントに対するポリシー決定が集中管理されます。

Oracle Flex Clusterは、クラスタの他のノードをサポートするハブ・ノードで構成されます。Oracle Flex Cluster内のハブ・ノードの最小数は1、最大数は64で、サポートされる他のノードの数はさらに多くできます。ハブ・ノードは、異なるタイプのアプリケーションをホストできます。

Oracle Flex Clusterは、1つ以上のハブ・ノードと連携できますが、他のノードはオプションで、1つ以上のハブ・ノードを含むクラスタのメンバーとしてのみ存在できます。

クラスタの使用によるメリットは、次のとおりです。

- アプリケーションのスケーラビリティ(Oracle RACおよびOracle RAC Oneデータベースを含む)
- スケーラブルなシステムを低コストの汎用ハードウェアと組み合わせることで、インフラストラクチャの総所有コストを削減できます。
- フェイルオーバーの機能

- サーバーをクラスタに追加してクラスタ・リソースを増やすことによって、クラスタ対応アプリケーションのスループットが必要に応じて向上します。
- クラスタ内のすべてのノードでアプリケーションを実行できるため、クラスタ対応アプリケーションのスループットが向上します。
- 依存プロセスが適切な順序で確実に起動するように計画した順序でアプリケーションの起動をプログラムする機能
- プロセスの監視、およびプロセスが停止した場合に再起動する機能
- ハードウェアやソフトウェアの障害に起因する計画外停止時間を解消できます。
- ソフトウェア・メンテナンスのための計画停止時間を削減または解消できます。

Oracle Clusterwareを構成して、ユーザー・アプリケーションとOracle Databaseの可用性を管理できます。Oracle RAC環境では、Oracle Clusterwareによってすべてのリソースが自動的に管理されます。Oracle Clusterwareが管理するすべてのアプリケーションおよびプロセスは、[クラスタ・リソース](#)または[ローカル・リソース](#)のいずれかです。

Oracle Clusterwareは、Oracle RACを使用するために必要であり、Oracle RACが動作するプラットフォームで必要とされる唯一のクラスタウェアです。Oracle RACでは、引き続き特定のプラットフォームにおいて多くのサード・パーティ製クラスタウェア製品がサポートされますが、Oracle Clusterwareもインストールして使用する必要があります。Oracle Clusterwareをインストールして実行する各サーバーでは、同じオペレーティング・システムを実行している必要があります。Oracle Database 19c以降では、Oracle Clusterwareとベンダー・クラスタウェアの統合は非推奨であり、将来のリリースでサポートされなくなる可能性があります。このため、次のソフトウェアまたはハードウェアのアップグレードでは、ベンダー・クラスタ・ソリューションを使用しないように移行することをお勧めします。

Oracle Clusterwareの使用により、ベンダー固有のクラスタウェアが不要になり、Oracleソフトウェアのみを使用するメリットが発揮されます。オラクル社では、Oracle Automatic Storage Management(Oracle ASM)によるディスク管理から、Oracle DatabaseおよびOracle RACによるデータ管理までのあらゆるソフトウェア・ソリューションを提供しています。さらに、基礎となるOracle Clusterwareの高可用性フレームワークとともに使用するとき、OracleサービスなどのOracle Database機能によって高度な機能が提供されます。

Oracle Clusterwareには、バイナリ以外に、ノードのメンバーシップ情報を記録する投票ファイルと、クラスタ構成情報を記録するOracle Cluster Registry(OCR)の2つのストア・コンポーネントがあります。投票ファイルおよびOCRは、すべてのクラスタ・メンバー・ノードに使用可能な共有記憶域に存在する必要があります。

Clusterwareのアーキテクチャ

Oracle Clusterwareでは、インストール・プロセス中に、新しいクラスタに対する2つの異なるデプロイメント・アーキテクチャの選択肢が提供されます。ドメイン・サービス・クラスタと、アプリケーションおよびデータベースをホストするために使用されるメンバー・クラスタのいずれかを選択できます。

ドメイン・サービス・クラスタは、1つ以上のハブ・ノード(データベース・インスタンス用)とゼロ個以上の他のノードを持つOracle Flex Clusterです。共有記憶域は各ハブ・ノードにローカルにマウントされ、Oracle ASMインスタンスはすべてのハブ・ノードで使用可能です。さらに、管理データベースはクラスタ内でローカルに格納およびアクセスされます。このデプロイメントは、アップグレードされた既存のクラスタにも使用されます。

メンバー・クラスタにより、管理目的で複数のクラスタ構成がグループ化され、そのクラスタ・ドメイン内で使用可能な共有サービスが使用されます。そのクラスタ・ドメイン内のクラスタ構成は、次のとおりです。

- ドメイン・サービス・クラスタ: クラスタ・ドメイン内の他のクラスタに一元化されたサービスを提供するクラスタ。サービスには、一元化されたグリッド・インフラストラクチャ管理リポジトリ(ここには、クラスタ・ドメイン内の各クラスタの管理データベースが存在します)、トレース・ファイル・アナライザ・サービス、オプションのフリー・パッチ適用およびプロビジョニング・サービス、および統合されたOracle ASM記憶域管理サービス(多くの場合、提供されます)を含めることができます。

- データベース・メンバー・クラスタ: Oracle RACまたはOracle RAC Oneデータベース・インスタンスをサポートすることを目的とし、その管理データベースはドメイン・サービス・クラスタにオフロードされ、ローカルのOracle ASM記憶域管理で構成することも、ドメイン・サービス・クラスタで提供される統合されたOracle ASM記憶域管理サービスを使用することもできるクラスタ。
- アプリケーション・メンバー・クラスタ: Oracle RACまたはOracle RAC Oneデータベース・インスタンスのサポートに必要なリソースを使用せずにアプリケーションをサポートするために構成されるクラスタ。このクラスタ・タイプではローカルの共有記憶域は構成されず、実行されるアプリケーション・プロセスのために可用性が高くスケーラブルなプラットフォームを提供することを意図しています。

Oracle Clusterwareのシステム要件の理解

Oracle Clusterwareのハードウェアとソフトウェアの概念および要件。

Oracle Clusterwareを使用するには、ハードウェアとソフトウェアの概念および要件を理解する必要があります。

Oracle Clusterwareのハードウェアの概念および要件

ハードウェアの概念および要件の理解は、Oracle Clusterwareの正常なデプロイメントに役立ちます。

クラスタは、1つ以上のサーバーで構成されます。外部ネットワークへのアクセスは、クラスタのサーバー(クラスタ・メンバーまたはノードとも呼ばれる)の場合も、スタンドアロン・サーバーの場合と同じです。

ノート:



多くのハードウェア・ベンダーが、1つのクラスタに単一の製品を使用してクラスタ構成を検証しています。クラスタリングを扱うのが初めてのユーザーは、この項の情報を使用することで、クラスタを作成するためのハードウェアを購入する際に、ハードウェアの調達に関する作業を簡素化できます。

クラスタに属するノードには、第2のネットワークが必要です。この第2のネットワークは、インターコネクトと呼ばれます。このため、クラスタ・メンバー・ノードには、2つ以上のネットワーク・インタフェース・カードが必要です(1つはパブリック・ネットワーク用、もう1つはプライベート・ネットワーク用)。インターコネクト・ネットワークは、クラスタ内のノードのみがアクセスできる単一のスイッチ(または複数のスイッチ)を使用するプライベート・ネットワークです。[脚注1](#)

ノート:



Oracle では、Oracle Clusterware のインターコネクトとして、クロスオーバー・ケーブルの使用はサポートされていません。

クラスタのサイズは、クラスタで実行するワークロードの要件と、クラスタで構成したノードの数によって決まります。高可用性対応のクラスタを実装する場合は、インフラストラクチャのすべてのコンポーネントで次の冗長構成を使用します。

- パブリック・ネットワーク用で、1つのアドレスを提供するために搭載される2つ以上のネットワーク・インタフェース
- プライベート・インターコネクト・ネットワーク用の2つ以上のネットワーク・インタフェース

クラスタには、クラスタ内の各サーバーの記憶域への共有接続が必要です。Oracle Clusterwareは、ネットワーク・ファイル・システム(NFS)、iSCSI、Direct Attached Storage (DAS)、Storage Area Network (SAN)ストレージおよび

Network Attached Storage (NAS)をサポートしています。

ストレージの冗長性を確保するには、通常、各サーバーからクラスタ対応ストレージに対して最低2つの接続を用意します。実際のI/O要件によっては、さらに多くの接続を用意することもあります。ストレージ・サブシステムを選択する際に、クラスタ全体のI/O要件を考慮することは重要です。

ほとんどのサーバーには、内部に1つ以上のローカル・ディスクが含まれます。通常、このディスクは、オペレーティング・システム・バイナリ用に使用されますが、ユーザーはこのディスクをOracleソフトウェアのバイナリ用としても使用できます。各サーバーでOracleバイナリの独自コピーを保持することのメリットは、高可用性が向上することであり、このため1つのバイナリの破損が同時にクラスタ内のすべてのノードに影響を及ぼすことがなくなります。また、これによってローリング・アップグレードも可能になり、停止時間を削減できます。

Oracle Clusterwareのオペレーティング・システムの概念および要件

Oracle Clusterwareをインストールする前に、まず、オペレーティング・システムをインストールして、検証する必要があります。

各サーバーには、インストールするOracle Clusterwareのリリースの動作が保証されたオペレーティング・システムが存在する必要があります。詳細は、使用するプラットフォームの『Oracle Grid Infrastructureインストレーションおよびアップグレード・ガイド』またはMy Oracle Support (以前のOracle MetaLink)の保証マトリックスを参照してください。

オペレーティング・システムをインストールして実行したら、Oracle Clusterwareをインストールしてクラスタを作成できます。Oracle Clusterwareは、Oracle Databaseとは別にインストールされます。Oracle Clusterwareのインストール後に、Oracle DatabaseまたはOracle RACをクラスタ内の任意のノードにインストールできます。

関連トピック

- [UNIXプラットフォームに関するOracle RACテクノロジーの動作保証マトリックス](#)
- [Oracle Grid Infrastructureインストレーションおよびアップグレード・ガイドfor Linux](#)

Oracle Clusterwareのソフトウェアの概念および要件

Oracle Clusterwareでは、投票ファイルを使用して、フェンシングの提供およびクラスタ・ノードのメンバーシップの判断が行われます。Oracle Cluster Registry (OCR)ではクラスタ構成情報が提供されます。投票ファイルとOCRはまとめてOracle Clusterwareファイルと呼ばれます。

Oracle ClusterwareファイルはOracle ASMに格納する必要があります。Oracle ASMディスクの基礎となる記憶域が、RAIDのようにハードウェアによって保護されていない場合は、OCRおよび投票ファイル用に複数の場所を構成することをお勧めします。次に、投票ファイルおよびOCRについて説明します。

● 投票ファイル

Oracle Clusterwareでは、投票ファイルを使用して、クラスタのメンバーであるノードを判断します。投票ファイルはOracle ASMまたは共有記憶域で構成できます。

投票ファイルをOracle ASMで構成する場合は、投票ファイルを手動で構成する必要はありません。ディスク・グループの冗長性に応じて、適切な数の投票ファイルが作成されます。

投票ファイルをOracle ASMで構成しない場合は、高可用性を確保するために、物理的に異なるストレージに3つ以上の投票ファイルを配置することをお勧めします。これによって、シングル・ポイント障害を回避できます。単一の投票ファイルを構成した場合、冗長性のために外部のミラー化を使用することをお勧めします。

最大15の投票ファイルがサポートされていても、6つ以上の投票ファイルを使用しないことをお勧めします。

- Oracle Cluster Registry

Oracle Clusterwareでは、Oracle Cluster Registry(OCR)を使用して、Oracle Clusterwareで制御するOracle RACデータベース、リスナー、仮想IPアドレス(VIP)、サービスと任意のアプリケーションなどのコンポーネントの情報を格納および管理します。OCRには、一連のキーと値のペアからなる構成情報がツリー構造に格納されます。クラスタの高可用性を確保するために、複数のOCRの場所を定義することをお勧めします。さらに、次のような特徴があります。

- 最大5つのOCRの場所を使用できます。
- 各OCRの場所は、クラスタ内のすべてのノードがアクセスできる共有記憶域に存在している必要があります。
- 障害が発生したOCRの場所がOCRの唯一の場所でないかぎり、オンラインで交換できます。
- OCRは、Oracle Enterprise Manager、Oracle Clusterware制御ユーティリティ(CRSCTL)、サーバー制御ユーティリティ(SRVCTL)、OCR構成ユーティリティ(OCRCONFIG)、Oracle Database Configuration Assistant (Oracle DBCA)などのサポートされているユーティリティを介して更新する必要があります

関連トピック

- [Oracle Clusterwareの構成と管理](#)

Oracle Clusterwareのネットワーク構成の概念

Oracle Clusterwareを使用すると、クラスタのネットワーク要件の自己管理により、動的なOracle Grid Infrastructureが可能になります。

Oracle Clusterwareは、[動的ホスト構成プロトコル\(DHCP\)](#)またはVIPアドレス用のステートレス・アドレス自動構成と[単一クライアント・アクセス名\(SCAN\)](#)アドレスの使用をサポートしていますが、パブリック・アドレスの使用はサポートしていません。DHCPはIPv4 VIPアドレスの動的割当てを行い、ステートレス・アドレス自動構成はIPv6 VIPアドレスの動的割当てを行います。

ノードVIPの使用は、クラスタ・デプロイメントではオプションです。デフォルトでは、クラスタ環境をデプロイするとノードVIPが組み込まれます。

Oracle RACを使用する場合、すべてのクライアントはデータベースにアクセスする必要があります。これは、クライアントは、VIP名およびSCAN名をすべてのVIPアドレスおよびSCANアドレスにそれぞれ解決する必要があることを意味しています。この問題は、[グリッド・ネーミング・サービス\(GNS\)](#)をクラスタに追加することによって解決します。GNSは会社のドメイン名サービス(DNS)にリンクされるため、クライアントは、ホスト名をこれらの動的アドレスに解決し、クラスタおよびデータベースに透過的に接続できます。Oracle Clusterware 12c以降のリリースでは、DHCPまたはゾーン委任なしのGNSの使用がサポートされています(ゾーン委任または動的ネットワークなしで構成できるOracle Flex ASMサーバーと同様です)。

ノート:



Windows 上に DHCP またはゾーン委任がない場合、GNS の使用はサポートされていません。

Oracle Clusterware 12c以降では、GNSインスタンスが1つのクラスタのみではなく複数のクラスタにサービスを提供できるように拡張されたため、DNSのGNSに委任する必要があるのは1つのドメインのみです。GNSは引き続き、以前のバージョンのOracle Clusterwareと同じサービスを提供します。

GNSサーバーが実行されるクラスタは、[サーバー・クラスタ](#)と呼ばれます。[クライアント・クラスタ](#)は、サーバー・クラスタを使用して名前を通知します。サーバー・クラスタでは1つのGNSデーモン・プロセスのみを実行できます。Oracle Clusterwareは、可用性を維持するために、GNSデーモン・プロセスをクラスタ内のノードの1つの上に置きます。

以前の単一クラスタ・バージョンのGNSでは、単一クラスタは、その内部でGNSサービス・プロバイダを簡単に探すことができました。ただし、マルチクラスタ環境ではクライアント・クラスタは、サーバー・クラスタのGNSアドレスを知る必要があります。このアドレスが与えられている場合、クライアント・クラスタは、サーバー・クラスタで実行されているGNSサーバーを見つけることができます。

サーバー・クラスタでGNSを動作させるには、次のことが必要です。

- DNS管理者がゾーンをGNSで使用されるように委任すること
- GNSインスタンスがネットワーク上のどこかで稼働しており、ファイアウォールでブロックされていないこと
- GNSが提供するクラスタのセット内のすべてのノード名は一意である必要があります。

関連トピック

- [Oracle Automatic Storage Management管理者ガイド](#)
- [グリッド・ネーミング・サービスの概要](#)

単一クライアント・アクセス名(SCAN)

Oracle Clusterwareでは、動的VIPアドレス構成に単一クライアント・アクセス名(SCAN)を使用でき、手動でサーバー構成を実行する必要がありません。

SCANは、DNSまたはGNSのIPアドレス(1つ以上3つ以下)に登録されているドメイン名です。GNSおよびDHCPを使用している場合、Oracle Clusterwareでは、クラスタの構成時に指定されるSCAN名のVIPアドレスが構成されます。

ノードVIPおよび3つのSCAN VIPは、GNSを使用時している場合、DHCPサーバーから取得されます。新しいサーバーがクラスタに追加されると、Oracle Clusterwareでは、必要なVIPアドレスはDHCPサーバーから動的に取得されてクラスタ・リソースが更新され、GNSを介してサーバーにアクセスできるようになります。

関連トピック

- [SCANアドレスおよびクライアント・サービス接続の理解](#)

アドレスの手動構成

動的な構成にGNSおよびDHCPを使用するかわりに、手動でアドレスを構成することもできます。

アドレスの手動構成では、次のものを構成します。

- ノードごとに1つのパブリック・アドレスとホスト名。
- ノードごとに1つのVIPアドレス。

クラスタ内の各ノードにVIPアドレスを割り当てる必要があります。各VIPアドレスは、ノードのパブリックIPアドレスと同じサブネット上に存在し、DNSで名前を割り当てられているアドレスである必要があります。また、Oracle Clusterwareのインストール前は、各VIPアドレスがネットワーク内から使用されず、pingの実行も不可能であることが必要です。

- クラスタ全体で最大3つのSCANアドレス。



ノート:

SCAN は、パブリック・ネットワークの 1 つ以上のアドレスに解決される必要があります。高可用性およびスケーラビリティを確保するために、パブリック・ネットワーク上の 3 つのアドレスに解決されるように SCAN を構成することをお勧めします。

関連トピック

- [Oracle Grid Infrastructureインストールおよびアップグレード・ガイドfor Linux](#)

Oracle Clusterwareのプラットフォーム固有のソフトウェア・コンポーネントの概要

稼働しているOracle Clusterware内で、様々なプラットフォーム固有のプロセスまたはサービスが各クラスタ・ノードで実行されます。

この項では、これらのプロセスおよびサービスについて説明します。

Oracle Clusterware技術スタック

Oracle Clusterwareは、クラスタ・レディ・サービス(CRS)・デーモン(CRSD)によってアンカーされている上位技術スタックと、Oracle高可用性サービス・デーモン(OHASD)によってアンカーされている下位技術スタックという2つの別個の技術スタックで構成されています。

これらの2つの技術スタックには、クラスタ操作を円滑化する複数のプロセスがあります。次の項では、これらの技術スタックについて詳細に説明します。

クラスタ・レディ・サービス技術スタック

クラスタ・レディ・サービス(CRS)技術スタックは、様々なサービスを管理するために複数のプロセスを利用します。

次のリストでは、これらのプロセスについて説明します。

- クラスタ・レディ・サービス(CRS): クラスタで高可用性操作を管理する主要プログラム。
CRSDは、各リソースのOCRに格納されている構成情報に基づいて、クラスタ・リソースを管理します。これには、起動、停止、監視およびフェイルオーバー操作が含まれます。CRSDプロセスは、リソースのステータスが変更されるとイベントを生成します。Oracle RACをインストールしている場合、CRSDプロセスはOracle Databaseインスタンスやリスナーなどを監視し、障害が発生した場合にこれらのコンポーネントを自動的に再起動します。
- クラスタ同期サービス(CSS): クラスタのメンバーシップを管理し、ノードがクラスタに対して追加または削除された際にメンバーに通知することによって、クラスタ構成を管理します。保証されているサード・パーティ製クラスタウェアを使用している場合、CSSプロセスは、クラスタウェアとともに動作して、ノードのメンバーシップに関する情報を管理します。
cssdagentプロセスは、クラスタの監視およびI/Oフェンシングを実行します。このサービスは、以前はOracle Process Monitor Daemon(oprocd)(WindowsではOraFenceServiceとも呼ばれる)によって提供されていました。cssdagentで障害が発生した場合、Oracle Clusterwareによってノードが再起動されることがあります。
- Oracle ASM: Oracle ClusterwareおよびOracle Databaseのディスク管理が提供されます。
- クラスタ時刻同期化サービス(CTSS): Oracle Clusterwareのクラスタで時間管理が提供されます。
- イベント・マネージメント(EVM): Oracle Clusterwareによって作成されたイベントを発行するバックグラウンド・プロセス。

- グリッド・ネーミング・サービス(GNS): 外部DNSサーバーから送信されるリクエストを処理し、クラスタで定義されている名前の名前解決を実行します。
- Oracle Agent (oraagent): クラスタウェアを拡張して、Oracle固有の要件および複雑なリソースをサポートします。このプロセスは、FANイベントが発生した場合にサーバー・コールアウト・スクリプトを実行します。このプロセスは、Oracle Clusterware 11gリリース1(11.1)ではRACGと呼ばれていました。
- Oracle Notification Service (ONS): 高速アプリケーション通知(FAN)イベントの通信用のパブリッシュおよびサブスクライブ・サービス。
- Oracle Root Agent(orarootagent): rootが所有しているネットワークやグリッド仮想IPアドレスなどのリソースをCRSDが管理するのを支援する専用のoraagentプロセス。

クラスタ同期サービス(CSS)、イベント・マネージメント(EVM)およびOracle Notification Service (ONS)コンポーネントは、同じクラスタ・データベース環境で、他のノードのクラスタ・コンポーネント・レイヤーと通信します。また、これらのコンポーネントは、Oracle Database、アプリケーションおよびOracle Clusterwareの高可用性コンポーネント間における主要通信リンクです。さらに、これらのバックグラウンド・プロセスは、データベース操作を監視および管理します。

Oracle高可用性サービス技術スタック

Oracle高可用性サービス技術スタックは、Oracle Clusterwareの高可用性を提供するために複数のプロセスを使用します。次のリストでは、Oracle高可用性サービス技術スタック内のプロセスについて説明します。

- appagent: 以前のバージョンのOracle Clusterwareで使用されていたapplicationリソース・タイプのリソースを保護します。
- Cluster Logger Service (ologgerd): クラスタ内のすべてのノードから情報を受け取り、Oracle Grid Infrastructure管理リポジトリベースのデータベース内で持続します。このサービスは、クラスタ内の2つのノードのみで実行されます。
- グリッド・プロセス間通信(GIPC): 冗長なインターコネクトの使用を可能にするサポート・デーモン。
- グリッド・プラグ・アンド・プレイ(GPNPD): グリッド・プラグ・アンド・プレイ・プロファイルへのアクセスを提供し、クラスタのノード間でプロファイルの更新を調整して、すべてのノードで最新のプロファイルが保持されるようにします。
- マルチキャスト・ドメイン名サービス(mDNS): クラスタ内のプロファイルを特定するためにグリッド・プラグ・アンド・プレイによって使用され、名前解決を実行するためにGNSによって使用されます。mDNSプロセスは、Linux、UNIXおよびWindowsのバックグラウンド・プロセスです。
- Oracle Agent (oraagent): クラスタウェアを拡張して、Oracle固有の要件および複雑なリソースをサポートします。このプロセスは、Oracle Clusterware所有者として実行されるGIPC、GPNPD、GIPCなどのデーモンを管理します。



ノート:

このプロセスは、Cluster Ready Services テクノロジ・スタックで実行する同じ名前のプロセスとはまったく異なります。

- Oracle Root Agent (orarootagent): rootが所有するCluster Health Monitor (CHM)などのリソースをCRSDが管理するのを支援する専用のoraagentプロセス。

ノート:



このプロセスは、Cluster Ready Services テクノロジ・スタックで実行する同じ名前のプロセスとはまったく異なります。

- scriptagent: シェル・スクリプトまたはバッチ・スクリプトを使用してアプリケーションを保護する場合に、application以外のリソース・タイプのリソースを保護します。
- システム監視サービス(osysmond): 監視して、データをクラスタ・ログ出力サービスに送信するオペレーティング・システム・メトリック収集サービス。このサービスは、クラスタ内のすべてのノードで実行されます。

表1-1に、Oracle Clusterwareコンポーネントに関連付けられたプロセスおよびサービスを示します。表1-1で、UNIXまたはLinuxシステムのプロセスに(r)とある場合、そのプロセスがrootユーザーとして実行されることを示します。

ノート:



Oracle ASM は単なる 1 つのプロセスではなく、インスタンスでもあります。Oracle Flex ASM の場合、Oracle ASM はすべてのクラスタ・ノードで必ずしも実行されるわけではなく、一部のクラスタ・ノードでのみ実行されます。

表1-1 Oracle Clusterwareコンポーネントに関連付けられたプロセスおよびサービスのリスト

Oracle Clusterwareコンポーネント	LinuxおよびUNIXのプロセス	Windowsのプロセス
CRS	crsd.bin (r)	crsd.exe
CSS	ocssd.bin, cssdmonitor, cssdagent	cssdagent.exe, cssdmonitor.exe, ocssd.exe
CTSS	octssd.bin (r)	octssd.exe
EVM	evmd.bin, evmlogger.bin	evmd.exe
GIPC	gipcd.bin	
GNS	gnsd (r)	gnsd.exe
グリッドのプラグ・アンド・プレイ	gpnpd.bin	gpnpd.exe
LOGGER	ologgerd.bin (r)	ologgerd.exe
マスターDiskmon	diskmon.bin	

Oracle Clusterwareコンポーネント		
	LinuxおよびUNIXのプロセス	Windowsのプロセス
mDNS	mdnsd.bin	mDNSResponder.exe
Oracle Agent	oraagent.bin (Oracle Clusterware 12c リリース 1 (12.1)以上のリリース)	oraagent.exe
Oracle 高可用性サービス	ohasd.bin (r)	ohasd.exe
ONS	ons	ons.exe
Oracle Root Agent	orarootagent (r)	orarootagent.exe
SYSMON	osysmond.bin (r)	osysmond.exe

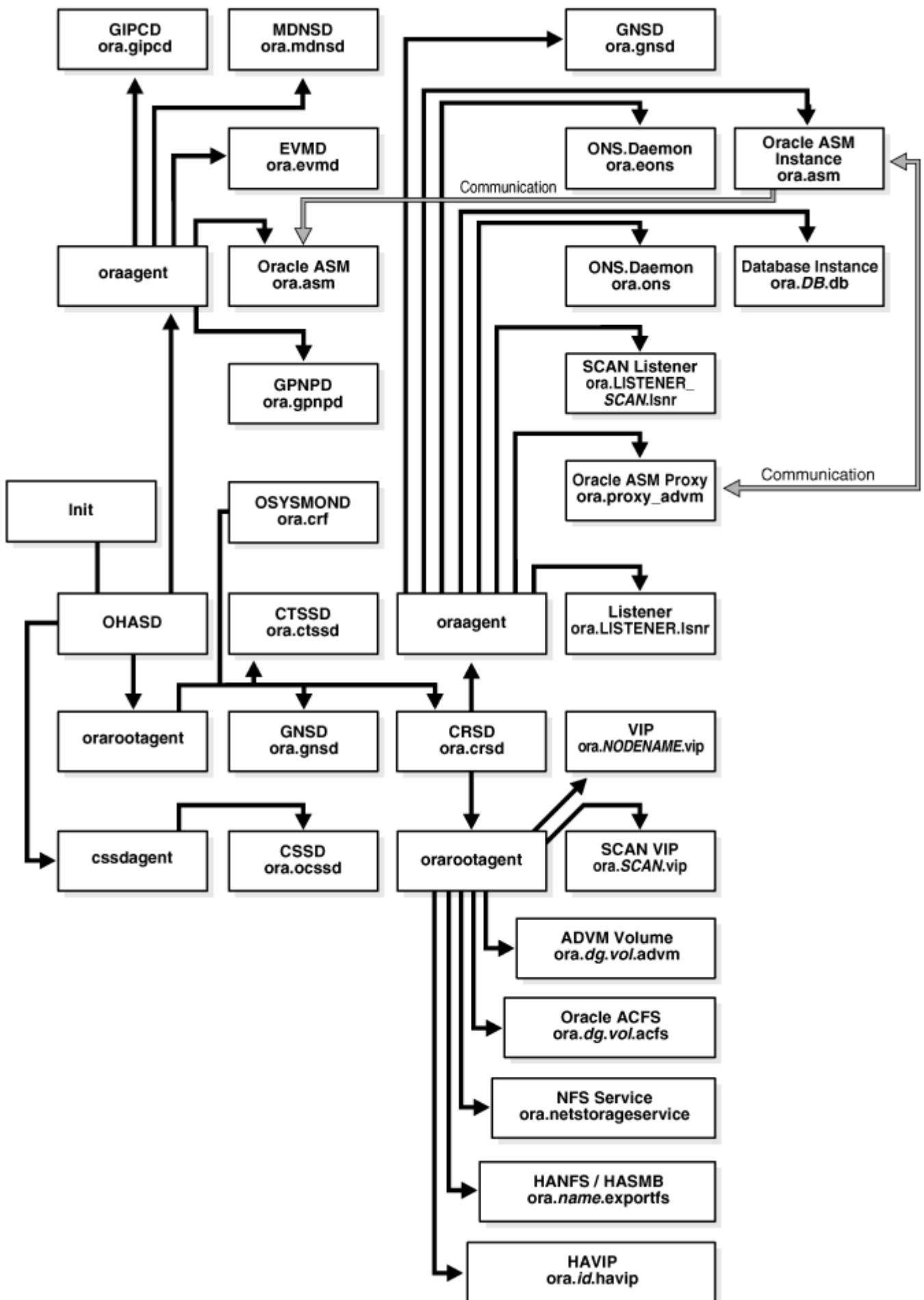
ノート:



Linux プラットフォームの Oracle Clusterware では、複数のスレッドが、一意のプロセス識別子を持つ個別のプロセスとして表示される場合があります。

[図1-2](#)は、クラスタの起動を表しています。

図1-2 クラスタの起動



関連トピック

- [Oracle Clusterwareのリソース](#)

- [Oracle Clusterwareの診断およびアラート・ログ・データ](#)

トランスポート層セキュリティ暗号スイート管理

Oracle Clusterwareでは、特定の暗号スイートを無効化するCRSCTLコマンドが提供されており、無効化された暗号スイートの詳細がOracle Local RegistryおよびOracle Cluster Registryに格納されます。それにより、無効化リストに含まれている暗号スイートがトランスポート層セキュリティのネゴシエーションに使用されないようにします。

Oracle Clusterware 19c以降のテクノロジ・スタックでは、ノード間通信とノード内通信の両方にGIPCライブラリが使用されます。ノード間通信チャネルを保護するために、GIPCライブラリはトランスポート層セキュリティを使用します。Oracle Clusterwareのどのリリースでも、GIPCライブラリはプリコンパイルされた暗号スイートのセットをサポートします。時間の経過とともに、暗号スイートが影響されることがあります。Oracle Clusterware 19cより前は、今後新しい接続で使用されないように、セットに含まれる特定の暗号スイートを無効にすることはできませんでした。

暗号リストの問合せ

使用可能な暗号スイートのリストを取得するには:

```
crsctl get cluster tlsciphersuite
```

無効化されたリストへの暗号スイートの追加

無効化されたリストに暗号スイートを追加するには:

```
crsctl set cluster disabledtlsciphersuite add cipher_suite_name
```

無効化されたリストからの暗号スイートの削除

無効化されたリストから暗号スイートを削除するには:

```
crsctl set cluster disabledtlsciphersuite delete cipher_suite_name
```

関連トピック

- [crsctl get cluster tlsciphersuite](#)
- [crsctl set cluster disabledtlsciphersuite](#)

WindowsシステムでのOracle Clusterwareプロセス

Oracle Clusterwareは、Microsoft Windowsシステムでの操作用に様々なMicrosoft Windowsプロセスを使用します。

これらには、次のプロセスが含まれます。

- mDNSResponder.exe: 所属するサブネット内で名前解決とサービス検出を管理します
- OracleOHService: すべてOracle Clusterwareデーモンを起動します

Oracle Databaseの高可用性オプション

Standard Edition High Availability、Oracle Restart、Oracle Real Application Clusters (Oracle RAC)およびOracle RAC One Nodeを使用して、Oracle Databaseで使用可能な高可用性オプションを確認します。

次に、Oracle Databaseで使用可能な高可用性オプションの概要を示します。

Standard Edition High Availability

- クラスターベースのアクティブ/パッシブOracle Databaseフェイルオーバー・ソリューションです。
- シングル・インスタンスのStandard Edition Oracle Database用に設計されています。
- Oracle Database 19cリリース更新(RU) 19.7以降で使用可能です。
- Oracle Grid Infrastructure 19c RU 19.7以降がスタンドアロン・クラスタとしてインストールされている必要があります。

Oracle Restart

- スタンドアロン・サーバー・デプロイメントの場合は、Oracle Databaseインスタンスにより、Oracle Automatic Storage Management (Oracle ASM)の機能および基盤のみが再起動されます。
- シングル・インスタンスのOracle Databasesの場合
- スタンドアロン・サーバー(クラスタなし)の場合はOracle Grid Infrastructureが必要です。

Oracle Real Application Clusters (Oracle RAC) One Node

- クラスターベースのアクティブ/パッシブOracle Databaseフェイルオーバー・ソリューションおよびオンライン・データベース再配置ソリューションを提供します。
- Oracle RAC対応Oracle Databaseで使用可能です。
- 通常操作ではOracle RAC対応Oracle Databaseのインスタンスが1つのみ実行されています。
- アクティブなインスタンスをオンライン方式でクラスタ内の別のサーバーに再配置できます。アクティブなインスタンスを再配置するには、宛先サーバーで2つ目のインスタンスを一時的に起動し、ワークロードを再配置します。
- ローリング・アップグレード(パッチ・セット、データベースおよびオペレーティング・システム)がサポートされています。
- アプリケーション・コンティニューイティがサポートされています。
- Oracle Grid Infrastructureをスタンドアロン・クラスタとしてインストールする必要があります。

Oracle Real Application Clusters(Oracle RAC)

- アクティブ/アクティブのOracle Database高可用性ソリューションおよびスケーラビリティ・ソリューションを提供します。
- 複数のサーバーが同じOracle Database上で複数のトランザクションを同時実行できます。
- 次のような高可用性を提供します。データベース・インスタンスまたはサーバーに障害が発生しても、他のインスタンスやそれらのサーバーが稼働しているため、データベース・サービス全体が中断されることはありません。
- ローリング・アップグレード(パッチ・セット、データベースおよびオペレーティング・システム)がサポートされています。
- アプリケーション・コンティニューイティがサポートされています。
- Oracle Grid Infrastructureをスタンドアロン・クラスタとしてインストールする必要があります。

Oracle Clusterwareのインストールの概要

インストールおよびデプロイメントの概念を理解していれば、Oracle Clusterwareを正常にデプロイメントできる可能性が高くなります。

ノート:



Oracle Clusterware は、Oracle Universal Installer を使用してインストールします。

Oracle Clusterwareのリリースの互換性

クラスタには、異なるリリースのOracle Clusterware、Oracle ASMおよびOracle Databaseをインストールできます。ただし、互換性に関する考慮事項に注意する必要があります。

クラスタに異なるリリースのソフトウェアをインストールする場合、次のガイドラインに従ってください。

- クラスタ内で実行できるOracle Clusterwareのインストールは1つのみであり、Oracle Clusterwareは自身のソフトウェア・ホーム(Grid_home)にインストールする必要があります。使用するOracle Clusterwareのリリースは、クラスタ内で実行するOracle ASMおよびOracle RACのバージョン以上である必要があります。クラスタ内で実行するOracle Clusterwareのバージョンより後に公開されたOracle RACのバージョンをインストールすることはできません。たとえば：

- Oracle Clusterware 19cは、Oracle ASM 19cのみをサポートしていますが、これは、Oracle ASMがOracle Grid Infrastructureホームにあり、これもOracle Clusterwareを含んでいるためです
- Oracle Clusterware 19cでは、Oracle Database 19cおよびOracle Database 12.1リリース1 (12.1以上)がサポートされます
- Oracle ASM 19cでは、Oracle Clusterware 19cが必要で、Oracle Database 19cおよびOracle Database 12cリリース1 (12.1以上)がサポートされます
- Oracle Database 19cには、Oracle Clusterware 19cが必要です。

たとえば：

- Oracle Clusterware 19cをクラスタウェアとしてインストールしている場合、1つのノードでOracle Database 12cリリース1 (12.1)単一インスタンス・データベースを実行し、またクラスタで別個のOracle Real Application Clusters 12cリリース2 (12.2)およびOracle Real Application Clusters 18cデータベースも実行できます。
- 異なるリリースのOracle ASMおよびOracle Databaseを使用する場合、各製品の機能は、より古いリリースのソフトウェア機能に依存します。したがって、Oracle Clusterware 19cをインストールし、後でOracle ASMを構成し、Oracle Clusterwareを使用して既存のOracle Database 12cリリース2 (12.2)インストールをサポートする場合、Oracle ASM機能は、12cリリース2 (12.2)リリース・バージョンで使用可能な機能にのみ相当します。ディスク・グループのcompatible属性は、使用されているソフトウェアの適切なリリースに設定します。
- クラスタでは、Oracle Databaseソフトウェア(シングル・インスタンスおよびOracle RACの両方)の複数のOracle ホームを使用できます。Oracle RACデータベースのすべてのノードのOracleホームは同じである必要があります。
- Oracle ClusterwareとOracle Databaseのホームでは、属しているプライマリ・グループが同じであれば異なるユーザーを使用できます。
- Oracle Clusterware 12c以降では、クラスタ内で実行できるOracle ASMのインストールは1つのみです。Oracle ASMは常にOracle Clusterwareと同じバージョンであり、Oracle Clusterwareは、Oracle Databaseのリリースと同じか、それ以上のリリースである必要があります。
- Oracle RAC 10gをインストールするには、Oracle Clusterwareもインストールする必要があります。
- 連携して動作する保証がないかぎり、同じサーバーでは異なるクラスタ・ソフトウェアを実行しないことをお勧めします。クラスタの一部であるサーバーにOracle RACを追加する場合は、Oracle Clusterwareに移行するか、次のことを確認します。
 - 実行するクラスタウェアは、Oracle RAC 18cで動作するようにサポートされています。
 - Oracle Clusterwareと他のベンダーのクラスタウェアが連携して動作するための適切なオプションをインストール済であることを確認します。

ノート:



Oracle Database 19c 以降では、Oracle Clusterware とベンダー・クラスタウェアの統合は非推奨であり、将来のリリースでサポートされなくなる可能性があります。このため、次のソフトウェアまたはハードウェアのアップグレードでは、ベンダー・クラスタ・ソリューションを使用しないように移行することをお勧めします。

関連トピック

- [Oracle Automatic Storage Management 管理者ガイド](#)
- [Oracle Grid Infrastructure のインストール・ガイド](#)

Oracle Clusterware のアップグレードおよびパッチ適用の概要

インプレース・パッチ適用によって、Oracle Clusterware ソフトウェアは同じ Grid ホームの新しいバージョンに置き換えられます。アウトオブプレース・アップグレードの場合は、同じソフトウェアの両方のバージョンがノードに同時に存在しますが (Grid ホームは異なる)、アクティブなバージョンは 1 つのみです。

Oracle Clusterware 12c 以降のリリースの場合、Oracle では、インプレースまたはアウトオブプレース・パッチ適用がサポートされます。ホーム外のアップグレードのみがサポートされます。

Oracle では、インプレース・パッチ適用についてはパッチ・バンドルおよび個別パッチがサポートされていますが、アウトオブプレース・アップグレードについてはパッチ・セットおよびメジャー・ポイント・リリースのみがサポートされています。

ローリング・アップグレードでは、停止時間が回避され、ソフトウェアが新しいバージョンにアップグレードされる間、Oracle Clusterware の連続的な可用性が確保されます。Oracle Clusterware 12c 以降のリリースにアップグレードすると、Oracle Clusterware および Oracle ASM バイナリは、Oracle Grid Infrastructure という単一バイナリとしてインストールされます。

Oracle では、クラスタの一部のノードが停止している場合の強制アップグレードがサポートされます。

関連トピック

- [Oracle Grid Infrastructure インストールおよびアップグレード・ガイド for Linux](#)

グリッド・インフラストラクチャ管理リポジトリの概要

グリッド・インフラストラクチャ管理リポジトリは、様々なクラスタ・クライアントが収集および必要とするリアルタイム・パフォーマンス・データやメタデータなど、クラスタに関する情報を格納します。

以前のバージョンの Oracle Grid Infrastructure では、グリッド・インフラストラクチャ管理リポジトリは Grid Infrastructure ホームのスタンドアロン・データベースでした。これには、作成された最初のディスク・グループのディスク領域が必要で、その領域を Oracle Cluster Registry (OCR) および投票ディスクと共有していました。スタンドアロン・クラスタの場合、Oracle Grid Infrastructure のインストール時に、OCR バックアップ・ファイルとともにグリッド・インフラストラクチャ管理リポジトリを独自の MGMT ディスク・グループにインストールできます。

Oracle メンバー・クラスタでは、そのグリッド・インフラストラクチャ管理リポジトリがドメイン・サービス・クラスタのグリッド・インフラストラクチャ管理リポジトリ・コンテナ・データベース (CDB) 内のプラガブル・データベース (PDB) である、そのドメイン・サービス・クラスタにあるグリッド・インフラストラクチャ管理リポジトリ・サービスが使用されます。これにより、ローカルのグリッド・インフラストラクチャ管理リポジトリを実行して、メンバー・クラスタのディスク・グループにそのデータ・ファイルを配置するための要件がなくなります。

さらに、Oracle Grid Infrastructure のインストール時に、ローカルのグリッド・インフラストラクチャ管理リポジトリとドメイン・サー

ビス・クラスタのグリッド・インフラストラクチャ管理リポジトリの両方をMGMTディスク・グループにインストールできます。ドメイン・サービス・クラスタは、OCRおよび投票ディスクとは別にディスク・グループに配置する必要があります。

グリッド・インフラストラクチャ管理リポジトリについて、次に説明します。

- クラスタ状態モニターによって収集されたリアルタイムのオペレーティング・システム・メトリックを格納する、Oracle Databaseです。クラスタ上でOracle Clusterware 12cのインストールまたはこれへのアップグレード中に、グリッド・インフラストラクチャ管理リポジトリを構成します。

ノート:



Oracle Clusterware を Oracle Clusterware 12c にアップグレード中で、OCR および投票ファイルが RAW またはブロック・デバイス上に格納されている場合、ソフトウェアをアップグレードする前に、これらを Oracle ASM に移動する必要があります。

- ハブ・ノードで実行し、ノードまたは記憶域の障害が発生したときの他のノードへのフェイルオーバーをサポートする必要があります。
- プライベート・ネットワークを介して、あらゆるクラスタ・クライアント(クラスタ状態アドバイザ、クラスタ・リソース・アクティビティ・ログ、フリート・パッチ適用およびプロビジョニング・サーバー、OLOGGERDおよびOCLUMONなど)との通信を行います。グリッド・インフラストラクチャ管理リポジトリは、パブリック・ネットワークを介して外部クライアント、ドメイン・サービス・クラスタおよびメンバー・クラスタのみと通信します。
- Oracle Grid Infrastructureをインストールする際に作成できる独自のOracle ASMディスク・グループMGMTにデータ・ファイルを格納することをお勧めします。インストール中にこのディスク・グループを作成しなかった場合、データ・ファイルはOCRおよび投票ファイルと同じ場所に配置されます。

オラクル社では、グリッド・インフラストラクチャ管理リポジトリ(ネットワーク・ファイル・システム(NFS)、クラスタ・ファイル・システム、Oracle ASMディスク・グループなど)に対応するために、Oracle Clusterwareの共有ストレージ要件を増加しました。

- mgmt dbとmgmt lsnrの名詞を使用したSRVCTLコマンドで管理されます。
- 保存については、その各クライアントのコマンドの1つのユーティリティで管理されます。

ノート:



Oracle Grid Infrastructure 19c 以上では、グリッド・インフラストラクチャ管理リポジトリ(GIMR)は、Oracle スタンドアロン・クラスタの新規インストールでオプションです。Oracle ドメイン・サービス・クラスタでは、GIMR をサービス・コンポーネントとしてインストールする必要があります。

関連トピック

- [Oracle Autonomous Health Frameworkユーザーズ・ガイド](#)

ドメイン・サービス・クラスタの概要

ドメイン・サービス・クラスタは、単一の管理ドメイン内にクラスタ・システムをデプロイするためのサービス指向インフラストラクチャのベースです。ドメイン・サービス・クラスタは、その管理ドメイン内のメンバー・クラスタにサービスを提供するために単独でデプロイさ

れるOracleクラスタです。

メンバー・クラスタでは、必要なときに構成済サービスが使用されます。サービスには、一元化されたドメイン管理リポジトリ、記憶域管理サービスおよびフリー・パッチ適用およびプロビジョニングが含まれます。

ドメイン・サービス・クラスタを構成すると、次のような多くのメリットがあります。

- デプロイメントとプロビジョニングが簡単
 - 構成した共有サービスは、何回でも再利用できます。
 - メンバー・クラスタごとに共有記憶域をプロビジョニングして構成する必要はありません。
 - 共有記憶域管理サービスまたはローカルに構成された記憶域を使用するオプションでデータベース・メンバー・クラスタをデプロイします。
 - データベースのサポートなしでアプリケーション・メンバー・クラスタをデプロイして、消費リソースを抑え、共有記憶域にOracle ACFS Remoteを使用します。
- 一元化された共有サービス
 - 記憶域管理をドメイン・サービス・クラスタに統合すると、次のメリットがあります。
 - スケーリングの効率(記憶域と労力の節約)
 - すべての記憶域の同時管理
 - ローカル・クラスタからドメイン管理リポジトリにデータベース管理が遷移します。
 - メンバー・クラスタは、Oracle ACFS Remoteを使用してドメイン・サービス・クラスタのOracle ACFSファイル・システムにアクセスできます。

クラスタ・ドメインは、ドメイン・サービス・クラスタと、提供されたサービスを使用するメンバー・クラスタで構成されます。すべてのクラスタはOracle Universal Installerを使用してデプロイされ、同じ標準のOracle Clusterwareを使用します。最初にドメイン・サービス・クラスタをデプロイして、その後それを使用して、デプロイされて必要なサービスに対して構成されたメンバー・クラスタにサービスを提供する必要があります。

アプリケーション・メンバー・クラスタは、次のように構成されます。

- Oracleデータベースとデータベース・リソースをサポートしないため、消費するメモリー・リソースが非常に少なくなります。
- Oracle ACFS Remoteを使用して、ドメイン・サービス・クラスタからプロビジョニングされた共有記憶域にアクセスします。
- Oracle Cluster Registryの管理リポジトリと記憶域および一元化された記憶域で提供される投票ファイル。

データベース・メンバー・クラスタは、記憶域へのアクセス方法に応じて3つの異なる方法で構成できます。すべてのケースで、管理リポジトリはドメイン管理リポジトリに格納されます。データベース・メンバー・クラスタの記憶域へのアクセスのための3つの構成は、次のとおりです。

- 共有記憶域をローカルで構成し、クラスタでローカルのOracle ASMインスタンスを使用します。
- ドメイン・サービス・クラスタ上のリモートのOracle ASMを介して管理される記憶域に直接アクセスします。
- Oracle ACFS RemoteおよびOracle ASM IOサービスを介して、ドメイン・サービス・クラスタのOracle ASM記憶域に間接的にアクセスします。

関連トピック

- [クラスタ間依存性プロキシ](#)
- [Oracle Automatic Storage Management管理者ガイド](#)

Oracle Clusterware環境の管理の概要

Oracle Clusterwareでは、環境を管理するためのいくつかの異なるユーティリティを用意しています。

次のリストに、Oracle Clusterware環境を管理するユーティリティを示します。

- Cluster Health Monitor (CHM): [Cluster Health Monitor](#)は、オペレーティング・システムおよびクラスタ・リソースに関連した低下および障害を検出および分析して、ノードの削除など、Oracle ClusterwareおよびOracle RACの数多くの問題について詳細情報をユーザーに提供します。このツールは、オペレーティング・システム・リソースの消費状況を、ノード、プロセスおよびデバイス・レベルで継続的に追跡します。また、クラスタ全体のデータを収集し分析します。リアルタイム・モードでは、しきい値に達すると、ツールによってユーザーにアラートが表示されます。根本原因分析のため、履歴データを再生し、障害発生時に何が起きていたのかを理解できます。

- クラスタ検証ユーティリティ(CVU): CVUは、クラスタとOracle RAC固有の様々なコンポーネントの検証に使用するコマンドライン・ユーティリティです。CVUを使用して、共有ストレージ・デバイス、ネットワーク構成、システム要件、Oracle Clusterware、およびオペレーティング・システムのグループやユーザーを検証します。

CVUをインストールして使用し、インストール前およびインストール後のクラスタ環境をチェックします。CVUは、Oracle ClusterwareおよびOracle RACコンポーネントのインストール前およびインストール時に特に役立ち、これによって、現在の構成がインストールの最小要件を満たしていることを確認できます。また、CVUを使用して、ノードの追加や削除などの管理タスクの完了後にその構成を検証できます。

- Oracle Cluster Registryコンフィギュレーション・ツール(OCRCONFIG): OCRCONFIGは、OCR管理のためのコマンドライン・ツールです。また、OCRCHECKおよびOCRDUMPユーティリティを使用して、OCRに影響を及ぼす構成の問題のトラブルシューティングを実行できます。

- Oracle Clusterware制御(CRSCTL): CRSCTLは、Oracle Clusterwareを管理するためのコマンドライン・ツールです。一般的なクラスタウェア管理、データベース以外のアプリケーション用の個々のリソース、構成ポリシーおよびサーバー・プールの管理には、CRSCTLを使用します。

Oracle Clusterware 12cには、クラスタ対応コマンドが導入されています。これらのコマンドを使用すると、操作に応じて、クラスタ内の別のノードに対して、またはクラスタ内のすべてのノードに対して、クラスタ内の任意のノードから操作を実行できます。

`crsctl`コマンドを使用すると、クラスタ・リソースを監視したり(`crsctl status resource`)、サーバー、および接頭辞`ora.*`を持つ名前のサーバー・プール以外のサーバー・プールを監視および管理できます(`crsctl status server`、`crsctl status serverpool`、`crsctl modify serverpool`、`crsctl relocate server`など)。ノード固有のオプション引数`-n`または`-all`を使用して、クラスタ全体でOracle高可用性サービスを管理することもできます(`crsctl start | stop | enable | disable | config crs`)。また、CRSCTLを使用して個々のノードでOracle Clusterwareを管理できます(`crsctl start | stop | enable | disable | config crs`)。

- Oracle Enterprise Manager: Oracle Enterprise Managerには、単一インスタンス環境とOracle RACデータベース環境の両方を管理するために、Cloud ControlインタフェースとGrid Control GUIインタフェースの両方があります。また、Oracle Clusterware、およびOracle Grid Infrastructureのインストールで構成したすべてのコンポーネントを管理するGUIインタフェースもあります。Oracle Enterprise Managerを使用して管理タスクを実行することをお勧めします。

- Oracle Interface Configurationツール(OIFCFG):OIFCFGシングル・インスタンスのOracle DatabaseとOracle RAC環境の両方のコマンドライン・ツールです。OIFCFGを使用して、ネットワーク・インタフェースのコンポーネントへの割当ておよび割当て解除を行います。また、OIFCFGを使用して、特定のネットワーク・インタフェースを使用するようにコンポーネントを設定したり、コンポーネントの構成情報を取得できます。
- サーバー制御ユーティリティ(SRVCTL): SRVCTLは、クラスタ内のデータベース、サービス、リスナーなどのOracleリソースを管理するためのコマンドライン・インタフェースです。



ノート:

接頭辞 ora.*を持つ名前のサーバー・プールのみ、SRVCTL を使用して管理できます。

関連トピック

- [Oracle Autonomous Health Frameworkユーザーズ・ガイド](#)
- [クラスタ検証ユーティリティのリファレンス](#)
- [Oracle Clusterwareの構成と管理](#)
- [Oracle Clusterware制御\(CRSCTL\)ユーティリティ・リファレンス](#)
- [Oracle Interface Configurationツール\(OIFCFG\)のコマンド・リファレンス](#)
- [サーバー制御\(SRVCTL\)コマンド・リファレンス](#)

コマンドの評価の概要

Oracle Clusterware制御(CRSCTL)ユーティリティを使用すると、Oracle Clusterware環境内でサーバー、サーバー・プールおよびポリシーを管理するためにCRSCTLコマンドを使用した場合に、どのような理由で(妥当なコマンドの評価)何が発生するか(コマンドの評価)について、実際に変更することなく評価できます。

コマンド評価は、計画済または計画外のイベントの結果を示す以外に、ポリシー管理環境で、Oracle Clusterwareのポリシー決定に関する仮定を検証する機能が役立ちます。妥当なコマンド評価の機能は、様々なCRSCTLコマンドの冗長出力を介してOracle Clusterwareで特定のアクションが実行される理由を示すことによって拡張されます。

サーバー・プールからサーバーを削除した場合に発生することの例を次に示します。

```
$ crsctl eval delete server mjk-node2-3 -explain
```

```
Stage Group 1:
```

```
-----
Stage Required      Action
-----
```

```
1 E Server 'mjk-node2-3' is removed from server pool 'sp1'.
   E Server pool 'sp1' is below the MIN_SIZE value of 2 with 1
     servers.
   E Looking at other server pools to see whether MIN_SIZE value 2 of
     server pool 'sp1' can be met.
   E Scanning server pools with MIN_SIZE or fewer servers in
     ascending order of IMPORTANCE.
   E Considering server pools (IMPORTANCE): sp2(2) for suitable
     servers.
   E Considering server pool 'sp2' because its MIN_SIZE is 2 and it
     has 0 servers above MIN_SIZE.
   E Relocating server 'mjk-node2-0' to server pool 'sp1'.
   Y Server 'mjk-node2-3' will be removed from pools 'sp1'.
   Y Server 'mjk-node2-0' will be moved from pools 'sp2' to
     pools 'sp1'
=====
```


前述の例に含まれる情報は、コマンド評価によって戻される内容です。アクション計画ごとに、最終決定に到達するために使用されるOracle Clusterwareの属性と基準が説明されます。

ノート:



CRSCTL はサード・パーティ・リソースのみを評価できます。ora.orcl.db など、ora という接頭辞が付いたリソースは、SRVCTL コマンドを使用して評価する必要があります。

関連トピック

- [コマンド評価API](#)
- [Oracle Clusterware制御\(CRSCTL\)ユーティリティ・リファレンス](#)

グリッド環境でのOracle Clusterwareのクローニングおよび拡張の概要

ノードのクローニングは、新しいクラスタを作成する1つの方法です。クローニングを使用して、同じ構成で複数のクラスタを迅速に作成します。

クローニング・プロセスでは、Oracle Clusterwareソフトウェアのイメージが同様のハードウェアおよびソフトウェアを持つ他のノードにコピーされます。クローニングを使用する前に、プラットフォーム固有のOracle Clusterwareのインストール・ガイドに記載された手順を使用して、1つ以上のノードにOracle Clusterwareホームを正常にインストールする必要があります。

新規インストールの場合またはインストールするクラスタが1つのみの場合は、Oracle Universal InstallerやOracle Enterprise ManagerのProvisioning Pack機能など、自動化された対話型のインストール方法を使用することをお勧めします。これらの方法ではインストールのチェックが実行されるため、正常にインストールできます。クラスタ内のノードに対してOracle Clusterwareを追加または削除するには、gridsetup.shスクリプトを使用します。

関連トピック

- [Oracle Clusterwareのクローニング](#)
- [クラスタ・ノードの追加および削除](#)

Oracle Clusterwareの高可用性フレームワークおよびAPIの概要

Oracle Clusterwareでは、クラスタで実行されるアプリケーションまたはプロセスを管理できる、CLSCRS APIという多くの高可用性Application Program Interfaceが提供されます。このCLSCRS APIによって、すべてのアプリケーションで高可用性を実現できます。

アプリケーションが実行されているクラスタ内のノードとは関係なくユーザーがアプリケーションにアクセスできるように、アプリケーションのVIPアドレスを定義できます。これは、アプリケーションVIPとも呼ばれます。複数のアプリケーションVIPを定義できます。通常、実行中のアプリケーションごとに1つのアプリケーションVIPを定義します。アプリケーションVIPは、Oracle Clusterwareにより定義されたアプリケーション・リソースに依存することでアプリケーションに関連付けられます。

Oracle Clusterwareコンポーネントは、高可用性を維持するために、定義されている高可用性規則に従い、ステータスの変化に応じてアプリケーションおよびプロセスを再起動できます。アプリケーションをOracle Clusterwareに登録し、クラスタウェアがアプリケーション・プロセスを起動、停止または再配置できるように構成することによって、Oracle Clusterwareの高可用性フレームワークを使用できます。これを行うには、Oracle Clusterwareを使用して、アプリケーションを監視、再配置および再起動するプロファイルを作成し、カスタム・アプリケーションの高可用性を確保します。

関連トピック

クラスタの時間管理の概要

クラスタ時刻同期化サービス(CTSS)によって、クラスタ内のノード間の時刻同期に関する問題を検出できます。

CTSSはOracle Clusterwareの一部としてインストールされます。時刻同期化サービス(NTPやChronyなど)または時刻同期化サービス構成が、有効でも無効でもシステム上で検知されれば、これはオブザーバ・モードで実行されます。たとえば、`etc/ntp.conf`ファイルがクラスタ内の任意のノードに存在する場合、時刻同期化ソフトウェアが実行されていないときでも、CTSSはオブザーバ・モードで実行されます。

時刻同期化サービスまたは時刻同期化サービス構成がクラスタ内のどのノードにもないことをCTSSが検出した場合、CTSSはアクティブ・モードになり、クラスタの時刻管理を引き継ぎます。

CTSSがアクティブ・モードで実行されていて、NTP以外の別の時刻同期化ソフトウェアが実行されている場合、`etc/ntp.conf`というファイルを作成することで、CTSSがオブザーバ・モードで実行されるように変更することができます。CTSSは、オブザーバ・モードの変更にに関するアラート・ログにエントリを記録します。

ノードがクラスタに追加されると、CTSSではアクティブ・モードの場合、これらのノードの時刻がクラスタ内の1つのノードにある参照クロックと比較されます。2つの時刻の間に不一致があり、かつ、特定のステップ制限内に不一致がある場合、CTSSは、ステップ時間の同期を実行します。これは、基準と同期させるために、クラスタを結合するノードの時刻を前または後に移動させることです。

クラスタ内のノードのクロックは、様々な理由のために定期的に参照クロック(時刻CTSSが基準として使用し、クラスタ内で最初に起動するノードのクロック)と非同期になります。この場合、CTSSは時刻の同期化(Slew)を実行し、参照システムの時刻と同期するまで、ノードのシステムの時刻を速めたり、遅くします。この時刻の同期方法では、CTSSによって時刻が戻されることがないため、システムの時刻における単調増加が保証されます。

Oracle Clusterwareを起動したときに、CTSSがアクティブ・モードで実行されており、時刻の不一致が段階制限(制限は24時間)を超えている場合、CTSSはアラート・ログにアラートを生成して終了し、Oracle Clusterwareの起動は失敗します。クラスタに追加したノードの時刻をクラスタと同期するように手動で調整する必要があり、その後、Oracle Clusterwareを起動してCTSSでノードの時刻を管理することができます。

時刻の同期化(Slew)を実行する場合、CTSSでは参照クロックと同期するために時刻が戻されることはありません。CTSSによって定期的にアラート・ログにアラートが書き込まれ、これには、ノードが参照クロックと同期し続けるようにCTSSによって時刻が調整される頻度に関する情報が含まれます。

CTSSは、次の場合にOracle Clusterwareアラート・ログおよびsys logにエントリを書き込みます。

- 時刻の変更を検出したとき
- 参照ノードとの大きな時間の違いを検出したとき
- モードがオブザーバからアクティブに切り替わるとき、またはその逆のとき

クラスタ内の時刻を同期させるためにCTSSを実行すると、Oracle Clusterwareの問題のトラブルシューティングに役立ちます。これは、様々なノードでのイベントの順序の時間オフセットを考慮に入れる必要がないためです。

クラスタの時間管理のアクティブ化および非アクティブ化

クラスタの時間管理サービスを引き受けるために、CTSSをアクティブ化できます。別のクラスタ時刻同期化サービスを使用する場合は、これを非アクティブ化することもできます。

クラスタのCTSSをアクティブ化するには、クラスタ内のすべてのノードでベンダーの時刻同期サービスを停止し、構成を解除する必要があります。CTSSは、この操作を検出し、クラスタの時間管理を引き受けます。

たとえば、NTPの構成を解除するには、`etc/ntp.conf`ファイルを削除するか、名前を変更する必要があります。

同様に、クラスタのCTSSを非アクティブ化するには、次の手順を実行します。

1. クラスタ内のすべてのノードでベンダーの時刻同期サービスを構成します。CTSSは、この変更を検出し、オブザーバ・モードに戻ります。
2. `crsctl check ctss`コマンドを使用して、CTSSがオブザーバ・モードで動作していることを確認します。
3. クラスタ内のすべてのノードでベンダーの時刻同期サービスを起動します。
4. `cluvfy comp clocksync -n all`コマンドを使用して、時刻同期サービスが動作していることを検証します。

関連トピック

- [crsctl check ctss](#)
 - [cluvfy comp clocksync](#)
 - [Oracle Grid Infrastructureインストールおよびアップグレード・ガイドfor Linux](#)
-

脚注の凡例

脚注1:

Oracle Clusterwareでは、Oracle Database 10g リリース2 (10.2)以上のリリースが稼働する構成において、クラスタ内で最大100のノードがサポートされます。

2 Oracle Clusterwareの構成と管理

Oracle Clusterwareとその様々なコンポーネントの構成と管理には、アプリケーションとデータベースの管理およびクラスタ内のネットワーク構成が含まれます。

クラスタを構成および管理するには2つの方法があり、いずれかを選択できます。従来型の管理者管理アプローチを使用して、クラスタ・リソースとワークロードを手動で管理するか、ポリシー管理アプローチを使用して、様々な程度の自動化管理を起動できます。

管理者管理クラスタの場合、クラスタ・リソースをどのようにデプロイするか、およびワークロードをどこで管理するかを手動で構成する必要があります。通常、これは、どのクラスタ・ノードでどのデータベース・インスタンスを優先して実行するか、および障害発生時にどこでそれらのインスタンスを再起動するかを構成する必要があるということです。データベース・インスタンスの配置場所を構成することにより、クラスタ全体のワークロードを構成します。

ポリシー管理クラスタでは、ワークロードをサーバー・プール内で実行するように構成します。これには、各サーバー・プール内でのこれらのワークロードの管理方法を指示するポリシー・セットを構成します。サーバー・プールとポリシー・セットの管理は自分で行い、データベース・インスタンスの場所およびワークロード配置の詳細は、設定したポリシーに委任します。このアプローチを使用する際は、Oracle Quality of Service Management (Oracle QoS Management)を使用することにより、クラスタの管理をさらに自動化する追加オプションを選択できます。

ロール別管理

ロール別管理は、リソースの競合と不足のリスクを軽減する目的でクラスタ・リソースおよびワークロードを協調的に管理するためのアプローチです。

ロール別管理では、オペレーティング・システムのセキュリティとロール定義、およびユーザーのロールに従ってリソースとワークロード管理を分離するためのOracle Clusterwareアクセス権限を使用します。統合環境においては、コンピューティング・リソースの競合が存在する可能性があり、リソース・コンシューマおよびそれらのリソースの管理にはある程度の分離が必要となるため、このような統合環境で作業するユーザーにとっては特にこれが重要になります。デフォルトでは、この機能はインストール時に実装されません。

ロール別管理を構成する場合、意図されたロールに従って(データベースなどの)クラスタ・リソースを管理するオペレーティング・システムのユーザーおよびグループを設定し、必要に応じてアクセス制御リスト(ACL)を介してクラスタ・リソースおよびサーバー・プールに対する権限を追加します。また、Oracle Automatic Storage Management (Oracle ASM)では、これらのロール別構造をストレージ管理機能にまで拡張できます。

ロール別管理の原則は、管理者管理システムとポリシー管理システムの両方に等しく適用されます。管理者管理の場合は、クラスタ・リソースおよびロールをノード・レベルで管理するように構成しますが、ポリシー管理システムの場合は、クラスタ・リソースおよびロールをサーバー・プール内で管理するように構成します。

Oracle Clusterwareのロール別管理は、現在は[クラスタ管理者](#)に依存していません(下位互換性は維持されています)。デフォルトでは、Oracle Grid Infrastructureホーム(Gridホーム)にOracle Clusterwareをインストールしたユーザーおよびrootは、永続的なクラスタ管理者となります。プライマリ・グループ権限(デフォルトではoinstall)により、データベース管理者はDatabase Configuration Assistant (DBCA)を使用して新しく作成したサーバー・プールにデータベースを作成できますが、ロール区分は有効になりません。

ノート:



クラスタに最初にサーバー・プールを作成する前にロール区分を有効にすることをお勧めします。構成ポリシーおよび各ポリシー・セットを使用してサーバー・プールを作成および管理します。ACL 属性に格納されている各サーバー・プールのアクセス権限の詳細は、[表 3-1](#)を参照してください。

関連トピック

- [クラスタ構成ポリシーおよびポリシー・セットの概要](#)

クラスタ管理者の管理

アクセス制御リスト(ACL)を使用して、クラスタの管理者を定義できます。

クラスタにサーバー・プールを作成する機能は、クラスタ管理者に制限されています。以前のリリースでは、デフォルトでは、登録されているすべてのオペレーティング・システム・ユーザーがクラスタ管理者とみなされましたが、このデフォルトは必要に応じて `crsctl add | delete crs administrator` コマンドを使用して変更できました。ただし、このリリースではこれらのコマンドの使用は非推奨であり、かわりに、ポリシー・セットのACLを使用してサーバー・プールの作成機能を制御する必要があります。

原則として、サーバー・プールまたはクラスタ・リソースの作成権限を設定するには、オペレーティング・システム・ユーザー、またはそのユーザーがメンバーになっているオペレーティング・システム・グループの読取り、書込みおよび実行権限がACL属性内で設定されている必要があります。

ロール区分の構成

ロール区分は、必要なロール、ロールが管理するリソースおよびサーバー・プール、およびロールのアクセス権限の決定です。これらを決定した後、ACLおよびCRSCTLユーティリティを使用して、グループ権限(`oinstall`や`grid`など)用のオペレーティング・システム・ユーザー・アカウントを作成または変更できます。

最も基本的には、2つのオペレーティング・システム・ユーザーを`oinstall`グループの一部として作成した後、クラスタを作成し、2つのサーバー・プールを作成します。サーバー・プールごとに、そのサーバー・プールを管理するオペレーティング・システム・ユーザーの1つを割り当て、それ以外のユーザーには、そのサーバー・プールへの読取りアクセス権のみを付与します。

このことは、慎重に計画し、一貫性を維持しながら細部を重視して実行することが必要ですが、実装後、誤りの修正や時間の経過に伴う調整のために構成を変更できます。

ノート:



ロール別手法を `ora.*` リソース(Oracle RAC データベース・リソース)に適用することはできません。これらの手法は、サーバー・プールとユーザー定義のクラスタ・リソースおよびタイプにのみ適用できます。

サーバー・プールまたはリソースは、`root`または`grid`アカウントの下に作成します。指定されたオペレーティング・システム・ユーザーがこれらのサーバー・プールまたはリソースを管理するためには、それらのユーザーに各自のロールを実行するための正しい権限を付与する必要があります。

`crsctl setperm` コマンドを使用して、サーバー・プール、リソースまたはその両方に割り当てられるACLを使用した水平ロール区分を構成します。CRSCTLユーティリティは、パス`Grid_home/bin`にあります(`Grid_home`は、クラスタ・ホームのOracle Grid Infrastructureです)。

コマンドには次の構文を使用し、アクセス制御(ACL)文字列はイタリックで示します。

```
crsctl setperm {resource | type | serverpool} name {-u acl_string |  
-x acl_string | -o user_name | -g group_name}
```

フラグ・オプションは、次のとおりです。

- -u: エンティティACLの更新
- -x: エンティティACLの削除
- -o: エンティティ所有者の変更
- -g: エンティティ・プライマリ・グループの変更

ACL文字列は、次のとおりです。

```
{user:user_name[:readPermwritePermexecPerm] |  
group:group_name[:readPermwritePermexecPerm] |  
other[:readPermwritePermexecPerm] }
```

前述の構文例では、次のとおりです。

- user: ユーザーACL(指定したユーザーに付与されるアクセス権限)を指定します。
- group: グループACL(指定したグループ・メンバーに付与される権限)を指定します。
- other: 他のACL(特定のアクセス権限は付与されないユーザーまたはグループに付与されるアクセス権)を指定します。
- readperm: 読取り権限の場所(rは権限を付与、-は権限を禁止)
- writeperm: 書込み権限の場所(wは権限を付与、-は権限を禁止)
- execperm: 実行権限の場所(xは権限を付与、-は権限を禁止)

たとえば、グループpersonnelに対して、psftというサーバー・プール上の権限を設定し、管理ユーザーが読取り/書込み/実行権限を持ち、personnelグループのメンバーが読取り/書込み権限を持ち、グループ以外のユーザーにはアクセス権が付与されないようにするには、次のコマンドをrootユーザーとして入力します。

```
# crsctl setperm serverpool psft -u user:personadmin:rw-,group:personnel:rw-,  
other::---
```

クラスタ・リソースの場合、グループcrsadminに対して(Maynardにより管理される) MyProgramという名前のアプリケーション(リソース)に対する権限を設定するとき、管理ユーザーに読取り、書込みおよび実行権限を付与し、crsadminグループのメンバーに読取り権限と実行権限を付与し、グループ外のユーザーに(ステータスおよび構成チェックのために)読取りアクセス権のみを付与するには、最初にリソースを作成したユーザー(rootまたはgrid所有者)として、次のコマンドを入力します。

```
# crsctl setperm resource MyProgram -u user:Maynard:r-x,group:crsadmin:rw-,other:---  
:r--
```

関連トピック

- [Oracle Clusterwareのリソース・リファレンス](#)

グリッド設定ウィザードを使用したOracle Grid Infrastructureの構成

構成ウィザードでは、1つ以上のノードに新しいOracle Grid Infrastructureを構成するか、またはアップグレードされたOracle Grid Infrastructureを構成することができます。また、グリッド設定ウィザードはサイレント・モードでも実行できます。

Oracle Grid Infrastructureのソフトウェアのみのインストールを実行した後、グリッド設定ウィザードを使用してソフトウェアを構成できます。このウィザードでは、ウィザードの実行前と実行後にグリッド・ホームおよび入力の様々な検証が実行されます。

ノート:

- グリッド設定ウィザードを実行する前に、Oracle Grid Infrastructure ホームがカレントであり、必要なすべてのパッチが適用されていることを確認してください。
- グリッド設定ウィザードを一連の手順で起動するには:

Linux および UNIX の場合は、次のコマンドを実行します。

```
Oracle_home/gridSetup.sh
```

Windows の場合は、次のコマンドを実行します。

```
Oracle_home¥gridSetup.bat
```

単一ノードの構成

構成ウィザードを使用して、単一ノードを構成できます。

単一ノードを構成するには:

1. 次のように、構成ウィザードを起動します。

```
$ Oracle_home/gridSetup.sh
```

2. 「インストール・オプションの選択」ページで、「クラスタ用のOracle Grid Infrastructureの構成」を選択します。
3. 「クラスタ・ノードの情報」ページで、ローカル・ノードと、それに対応するVIP名のみを選択します。
4. 残りのウィザード・ページで、引き続き情報を追加します。
5. 「サマリー」ページで入力を確認して、「終了」をクリックします。
6. 構成ウィザードの指示に従って、root.shスクリプトを実行します。

複数ノードの構成

構成ウィザードを使用して、クラスタ内に複数ノードを構成できます。

構成ウィザードを使用して構成するノードへのOracle Grid Infrastructureソフトウェアのインストールは、必須ではありません。

ノート:

構成ウィザードを起動する前に、次の点を確認します。

すべてのノードにソフトウェアをインストールする必要はありませんが、インストールする場合は、ソフトウェアを同じGrid_homeパスにインストールし、すべてのノードで同一レベルに配置する必要があります。

構成ウィザードを使用して複数ノードを構成するには:

1. 次のように、構成ウィザードを起動します。

```
$ Oracle_home/gridSetup.sh
```

2. 「インストール・オプションの選択」ページで、「クラスタ用のOracle Grid Infrastructureの構成」を選択します。
3. 「クラスタ・ノードの情報」ページで、構成するノードと、それに対応するVIP名を選択します。構成ウィザードは、選択されたノードを検証し、その準備が整っていることを確認します。
4. 残りのウィザード・ページで、引き続き情報を追加します。
5. 「サマリー」ページで入力を確認して、「終了」をクリックします。
6. 構成ウィザードの指示に従って、root.shスクリプトを実行します。

Oracle Grid Infrastructureのアップグレード

グリッド設定ウィザードを使用して、クラスタのOracle Grid Infrastructureをアップグレードします。

クラスタ用のOracle Grid Infrastructureをアップグレードするには:

1. グリッド設定ウィザードを起動します。

```
$ Oracle_home/gridSetup.sh
```

2. 「インストール・オプションの選択」ページで、「Oracle Grid Infrastructureのアップグレード」を選択します。
3. Oracle Grid Infrastructureノード選択ページで、アップグレードするノードを確認します。また、停止しているノードをアップグレードしないように選択できます。
4. 残りのウィザード・ページで、引き続き情報を追加します。
5. 「サマリー」ページで入力を確認して、「終了」をクリックします。
6. 構成ウィザードの指示に従って、rootupgrade.shスクリプトを実行します。

関連項目:

Oracle Restartの手順の詳細は、ご使用のプラットフォームの『Oracle Databaseインストレーション・ガイド』を参照してください。

サイレント・モードでの構成ウィザードの実行

—silentパラメータを指定して、構成ウィザードをサイレント・モードで実行できます。

構成ウィザードをサイレント・モードで使用してノードを構成またはアップグレードするには:

1. 次のようにして、コマンドラインから構成ウィザードを起動します。

```
$ $ORACLE_HOME/gridSetup.sh -silent -responseFile file_name
```

構成ウィザードはレスポンス・ファイルを検証してから、構成に進みます。レスポンス・ファイルで無効な入力を検出すると、構成ウィザードはエラーを表示して終了します。

2. 指示に従って、rootスクリプトおよびGrid_home/gridSetup -executeConfigToolsスクリプトを実行します。

Oracle Grid Infrastructureホームの移動およびパッチ適用

グリッド設定ウィザードを使用して、Oracle Grid Infrastructureホームを移動しそれにパッチを適用します。

Oracleインストーラ・スクリプトgridSetup.shでは、この目的のために、新しいスイッチ-switchGridHomeがサポートされています。この機能では、同じかより新しいパッチ・レベルになるように、Oracle Grid Infrastructureホームを移動してそれにパッチを適用できます。

関連項目:

- パッチ適用後のOracle Grid Infrastructureホームの切替えの詳細は、[Oracle Grid Infrastructure Linuxインストールおよびアップグレードガイドfor Linux](#)

サーバーの重みベースのノード削除

Oracle Clusterwareの障害リカバリ・メカニズムを、プライベート・ネットワーク(クラスタ・インターコネクト)障害のイベントで終了または削除するノードを選択するように構成できます。

クラスタでネットワーク分離が発生し、クラスタが非結合コホートにパーティション化されるスプリット・ブレイン状況においては、Oracle Clusterwareは特定のルールを適用して存続コホートを選択し、場合によってはクリティカルなシングルトン・リソースを実行しているノードを削除することがあります。

これらの決定の結果を反映させるには、値をデータベース・インスタンスまたはノードに追加します。これにより、Oracle Clusterwareは、削除が終了かを決定する必要があるとき、これらの要因を考慮し、すべてのクリティカル・コンポーネントを使用可能な状態に保持しようとし、重み関数を構成して、クラスタ内のクリティカル・コンポーネントに重みを追加できます。これにより、スプリット・ブレイン状況を解決するときどのノードを削除するかを決定する際に、Oracle Clusterwareに付加的な入力提供されます。

同じレベルのノードから選択するプロセスにおいては、特定のハードウェア特性を考慮して特性のノードを存続させたり、特定のデータベースまたはサービスを考慮して特定のリソースを存続させることもできます。次の基準に基づいて、特定のノード、リソースまたはサービスに重みを割り当てることができます。

- 重みは管理者管理ノードにのみ割り当てることができます。
- 重みは、登録済のOracle Clusterwareリソースであるサーバーまたはアプリケーションに割り当てることができます。

重みは、コンポーネントの重要度の一因となり、分離ブレイン状況の管理中にOracle Clusterwareが行う選択に影響します。様々なコホート間で他のクリティカルな因子が同等であれば、Oracle Clusterwareは最も重いコホートを存続させるように選択します。

様々なコンポーネントに重みを割り当てるには、次のようにします。

- データベース・インスタンスまたはサービスに重みを割り当てるには、データベース・インスタンスまたはサービスを追加するときに、`srvctl add database`または`srvctl add service`コマンドで`-css_critical yes`パラメータを使用します。また、`srvctl modify database`および`srvctl modify service`コマンドでこのパラメータを使用することもできます。
- `ora.*`でないリソースに重みを割り当てるには、リソースを追加または変更するときに、`crsctl add resource`および`crsctl modify resource`コマンドで`-attr "CSS_CRITICAL=yes"`パラメータを使用します。
- サーバーに重みを割り当てるには、`crsctl set server`コマンドで`-css_critical yes`パラメータを使用します。

ノート:



- 値を有効にするには、ノードで Oracle Clusterware スタックを再起動する必要があります。リソースは再起動しなくても変更が有効になるため、これはリソースには適用されません。
- 管理者管理環境をポリシー管理環境または両者混合の環境に変更した場合、割り当てた重みはすべて保存されますが、考慮はされません。つまり、クラスタを元の管理者管理に再構成しないかぎり、適用も考慮もされません。

Oracle Database Quality of Service Managementの概要

Oracle Database Quality of Service Management (Oracle Database QoS Management)は、システム全体のワークロード・リクエストを監視する自動化されたポリシーベース製品です。

Oracle Database QoS Managementは、アプリケーション間で共有されるリソースを管理し、システム構成を調整して、アプリケーションの実行をビジネスに必要なパフォーマンス・レベルに維持します。Oracle Database QoS Managementは、システム構成および要求で発生した変更に対応し、アプリケーションのパフォーマンス・レベルがそれ以上変動することを回避します。

Oracle Database QoS Managementは、Oracle RACデータベース・ワークロード・パフォーマンスの目標を監視および管理するもので、これらの目標に影響を与えているボトルネックのリソースを特定し、パフォーマンスを回復するためのアクションを推奨し、実行します。管理者管理のデプロイメントではデータベース・インスタンスがノードにバインドされますが、ポリシー管理のデプロイメントではバインドされないため、Oracle Database QoS Managementサーバー・プール・サイズ・リソース制御はポリシー管理のデプロイメントでのみ使用可能です。これ以外のすべてのリソース管理制御は、どちらのデプロイメントでも使用可能です。

Oracle Database QoS Managementでは、管理者管理のOracle RACおよびOracle RAC One Nodeデータベースを、測定のみモード、監視モードおよび管理モードでサポートしています。このため、管理者管理のOracle RACデータベース内でパフォーマンス・クラスのCPU共有を調整することにより、データベース内でのスキーマ統合サポートが可能になります。また、同じ物理サーバー上でホストされるデータベースのCPU数を調整することにより、データベース統合もサポートされます。

ポリシー管理データベース・デプロイメントではサーバー・プール・サイズの変更によるインスタンス数の拡張または縮小機能がサポートされていますが、管理者管理のデータベースはサーバー・プール内で実行されないため、管理者管理のデータベースではこの機能を使用できません。この新しいデプロイメント・サポートは、Oracle Enterprise Manager Cloud ControlのOracle QoS Managementページに統合されています。

グリッド・ネーミング・サービスの概要

Oracle Clusterwareでは、シングルクラスタおよびマルチクラスタの環境のアドレス解決にグリッド・ネーミング・サービス(GNS)が使用されます。単一のプライマリGNSインスタンスを使用してクラスタを構成し、さらに異なるロールを持つ1つ以上のセカンダリGNSインスタンスを構成して、高可用性アドレス参照などのサービスをクライアントに提供することもできます。

- [GNSおよびGNS仮想IPアドレスのネットワーク管理タスク](#)
- [グリッド・ネーミング・サービス構成オプションの理解](#)

GNSおよびGNS仮想IPアドレスのネットワーク管理タスク

GNSを実装するには、ネットワーク管理者がDNSを構成してクラスタのドメインを設定し、そのドメイン解決をGNS VIPに委譲

する必要があります。別のドメインを使用するか、クラスタに既存のドメインのサブドメインを作成できます。

GNSは、そのクラスタ・ノードのホスト名の一部であるクラスタ名と個別のノード識別子を使用してノードを区別するため、クラスタAのクラスタ・ノード123とクラスタBのクラスタ・ノード123を区別することができます。

ただし、手動でホスト名を構成する場合、GNSに委譲するサブドメインにサブドメインがあってはなりません。たとえば、サブドメインmydomain.example.comをGNSに委譲して解決する場合、other.mydomain.example.comドメインがあってはなりません。GNSが排他的に使用するGNSにサブドメインを委譲することをお勧めします。

ノート:



Oracle Flex ASM または Oracle Flex Cluster など静的なアドレス変換を行う構成では、DNS に委譲せずに GNS を使用できます。ただし、DHCP を使用して住所を割り当てる場合、GNS には委譲先となるドメインが必要です。

[例2-1](#)は、myclustergns.example.comというドメインをGNS VIPアドレス10.9.8.7に委譲するために必要なDNS エントリを示しています。

GNSデーモンおよびGNS VIPは、サーバー・クラスタの1つのノードに対して実行されます。GNSデーモンは、ポート53を使用しているGNS VIPでDNSリクエストをリスニングします。Oracle Clusterwareによって、GNSデーモンおよびGNS VIPはいつでも使用できるように管理されます。GNSデーモンを実行しているサーバーに障害が発生した場合、Oracle ClusterwareはGNSサーバーおよびGNS VIPを、正常に動作するクラスタ・メンバー・ノードにフェイルオーバーします。クラスタがOracle Flex Cluster構成である場合、Oracle Clusterwareは、GNSデーモンおよびGNS VIPをハブ・ノードにフェイルオーバーします。

ノート:



Oracle Clusterware は、GNS アドレスを他のクラスタにフェイルオーバーしません。フェイルオーバーは、同じクラスタのメンバーのみに実行されます。

例2-1 DNSエントリ

```
# Delegate to gns on mycluster
mycluster.example.com NS myclustergns.example.com
#Let the world know to go to the GNS vip
myclustergns.example.com. 10.9.8.7
```

関連トピック

- [Oracle Grid Infrastructureのインストール・ガイド](#)
- [Oracle Flex Clusters](#)

グリッド・ネーミング・サービス構成オプションの理解

GNSは、自動または標準のクラスタ・アドレス構成モードのいずれかで実行できます。自動構成は、IPv4アドレスのDynamic Host Configuration Protocol (DHCP)またはIPv6アドレスのStateless Address Autoconfiguration Protocol (autoconfig) (RFC 2462およびRFC 4862)のいずれかを使用します。

この項には次のトピックが含まれます:

- [高可用性グリッド・ネーミング・サービス](#)

- [アドレスの自動構成オプション](#)
- [アドレスの静的な自動構成オプション](#)
- [アドレスの共有GNSオプション](#)

高可用性グリッド・ネーミング・サービス

高可用性GNSは、1つのプライマリGNSインスタンスおよび0以上のセカンダリGNSインスタンスで構成されています。

プライマリGNSインスタンスでクライアントからのすべての更新を処理し、プライマリGNSインスタンスとセカンダリGNSインスタンスの両方で参照問合せを処理します。また、セカンダリGNSインスタンスはプライマリGNSインスタンスのバックアップとしても機能します。既存のプライマリGNSインスタンスが失敗するか、クラスタ管理者によって削除されたときでも、セカンダリGNSインスタンスをプライマリ・ロールに昇格できます。

さらに、高可用性GNSでは、ゾーン転送を使用してセカンダリGNSインスタンスでデータ・バックアップを取ることでフォルト・トレランスを実現します。セカンダリGNSインスタンスは、インストール中にプライマリGNSインスタンスからデータのコピーを取得します。これ以降、プライマリGNSインスタンス上の更新は、セカンダリGNSインスタンスにレプリケートされます。

プライマリGNSインスタンスは[ゾーン・データ](#)を管理し、委任されたドメインですべてのレコードを保持します。ゾーン・データおよびその変更履歴はOracle Cluster Registry (OCR)に格納されます。セカンダリGNSインスタンス上でゾーン・データが更新されると、次の2つのいずれかの方法でゾーン転送が行われます。

- 完全ゾーン転送: プライマリGNSインスタンスは、すべてのゾーン・データをセカンダリGNSインスタンスにレプリケートします。
- 増分ゾーン転送: プライマリGNSインスタンスは、変更されたデータのみをセカンダリGNSインスタンスにレプリケートします。GNSでは、次のシナリオにおいてこの転送メカニズムを使用します。
 - プライマリGNSインスタンスにゾーン・データの更新がある場合、このインスタンスはデータ転送を開始するようにセカンダリ・インスタンスに通知します。セカンダリGNSインスタンスは、プライマリGNSインスタンスのOCR内のデータのシリアル番号がセカンダリGNSインスタンスのデータのシリアル番号よりも大きいときにのみ、データ転送を要求します。
 - セカンダリGNSインスタンスのリフレッシュ時間が満了すると、インスタンスは、そのデータ・シリアル番号を含む問合せをプライマリGNSインスタンスに送信します。セカンダリGNSインスタンスのシリアル番号がプライマリGNSインスタンスのシリアル番号よりも小さい場合、GNSによってゾーン転送が開始されます。

ノート:



セカンダリ GNS インスタンスに返答してもプライマリ・インスタンスの動作に支障がないように、リフレッシュ時間は、プライマリ GNS インスタンスのロードを減らすために十分な長さにする必要があります。デフォルトのリフレッシュ時間は 1 時間ですが、この値はクラスタ管理者がクラスタ・サイズに基づいて変更できます。

まずプライマリGNSインスタンスを構成してから、セカンダリGNSインスタンスを構成する必要があります。プライマリGNSインスタンスを正常に構成した後、クライアント用のクライアント・データおよびセカンダリGNSインスタンスをエクスポートします。セカンダリGNSインスタンスを構成するときに、エクスポートしたクライアント・データを提供します。すべてのセカンダリGNSインスタンスはプライマリGNSインスタンスに自身を登録し、ゾーン・データのコピーを取得します。セカンダリGNSインスタンスは、セカンダリGNSインスタンスのリフレッシュ時間が満了したときまたは通知に応答するとき、ゾーン転送メカニズムを使用して、プライマリGNSインスタンスにデータ更新について問い合わせます。

関連トピック

- [高可用性GNSの構成](#)
- [プライマリGNSインスタンスおよびセカンダリGNSインスタンスの削除](#)

アドレスの自動構成オプション

自動構成の場合、DNS管理者はGNSサブドメインを使用して解決するDNS上のドメインを委譲します。Oracle Universal Installerは、インストールまたは構成中にOracle Grid Infrastructureを使用できるよう、指定された各クラスタ・メンバー・ノード・インタフェースに名前を割り当てます。SCANおよびその他すべてのクラスタ名とアドレスは、DNS上ではなくクラスタ内で解決されます。

自動構成は次の方法のいずれかで実行されます。

- IPv4アドレスの場合、Oracle Clusterwareは、Oracle Grid Infrastructureに割り当てられた各クラスタ・メンバー・ノード・インタフェースに一意的識別子を割り当て、GNSに委譲されたサブドメイン内のこれらの識別子を使用して名前を生成します。DHCPサーバーはこれらのインタフェースにアドレスを割り当て、GNSは、IPv4 DHCP プールから借用したIPv4アドレスに関連する名前とアドレスを維持します。
- IPv6アドレスの場合、Oracle Clusterwareは、autoconfigを使用して自動的にアドレスを生成します。

アドレスの静的な自動構成オプション

静的な構成では、委譲されるサブドメインはありません。DNS管理者はGNS VIPを構成して、DNS上で構成された名前とアドレスに解決し、また、DNS管理者はSCAN名を構成して、クラスタのこれらの静的なアドレスに解決します。

また、DNS管理者は、各クラスタ・メンバー・ノードに静的なパブリックIP名とアドレスおよび仮想IP名とアドレスを構成します。DNS管理者は、クラスタに追加された各ノードに新しいパブリックおよび仮想IP名とアドレスを構成する必要があります。すべての名前とアドレスはDNSによって解決されます。

静的なVIPアドレスおよびSCANを使用したサブドメインの委譲がないGNSでは、クラスタ内に名前解決情報が必要なOracle Flex ClusterおよびCloudFS機能を使用できます。ただし、ノードを追加したり変更する場合は、手動で管理タスクを実行する必要があります。

アドレスの共有GNSオプション

動的な構成の場合、GNSを構成して、あるクラスタに名前解決をしたり、複数のクラスタに名前を通知できるため、単一のGNSインスタンスで、登録された複数のクラスタに対して名前解決を実行できます。このオプションは共有GNSと呼ばれます。

共有GNSは標準的なGNSと同じサービスを提供し、名前解決を受け取るクライアント側でも同様に表示されます。相違点は、1つのクラスタで動作するGNSデーモンは、GNSに名前解決を委譲したドメイン内のすべてのクラスタに対して名前解決を提供するように構成される点、およびGNSをSRVCTLコマンドを使用して一元的に管理できる点です。共有GNS構成を使用すると、Oracle Grid Infrastructureクラスタのエンタープライズ全体に対するネットワーク管理者タスクを最小限に抑えることができます。

マルチクラスタ環境でのアドレス解決に共有GNSを使用するクラスタでは、静的なアドレス構成は使用できません。共有GNSには自動アドレス構成が必要であり、DHCPまたはIPv6ステートレス・アドレス自動構成のいずれかによって割り当てられるアドレスを使用します。



ノート:

GNS が提供するクラスタのセット内のすべてのノード名は一意である必要があります。

Oracle Universal Installerを使用すると、共有GNSクライアントまたはサーバーのGNSを使用して、あるいは検出に使用するGNSを使用して、静的アドレスを構成できます。

グリッド・ネーミング・サービスの管理

SRVCTLを使用して、シングルクラスタおよびマルチクラスタの両方の環境でグリッド・ネーミング・サービス(GNS)を管理します。

ノート:



GNS サーバーおよびクライアントは、同じオペレーティング・システムおよびプロセッサ・アーキテクチャを使用しているコンピュータで実行する必要があります。コンピュータ上のオペレーティング・システムが異なる、プロセッサ・アーキテクチャが異なる、またはその両方が異なる場合、GNS は実行できません。

この項には次のトピックが含まれます:

- [高可用性GNSの構成](#)
- [プライマリGNSインスタンスおよびセカンダリGNSインスタンスの削除](#)
- [SRVCTLを使用したGNSの起動と停止](#)
- [GNSサーバーまたはGNSクライアント・クラスタへのクラスタの変換](#)
- [別のクラスタへのGNSの移動](#)
- [IPv4からIPv6ネットワークへの移行時のGNSサブドメインの変更](#)
- [DNSによるクラスタの名前解決をGNSにローリング変換](#)

高可用性GNSの構成

高可用性GNSの構成では、プライマリGNSインスタンスとセカンダリGNSインスタンスを構成します。GNSはOracle Clusterwareのインストール時にOracle Universal Installerを使用して構成できますが、セカンダリGNSインスタンスはOracle Clusterwareのインストール後にしか構成できないため、高可用性GNSはOracle Clusterwareのインストール後にしか構成できません。

高可用性GNSでは、マルチクラスタ環境における複数のGNSインスタンスを様々なロールで実行できます。1つのプライマリGNSインスタンスと、0以上のセカンダリ・インスタンスが存在します。プライマリ・インスタンスは更新および解決の操作を管理します。ただし、セカンダリ・インスタンスは解決操作のみを管理します。セカンダリGNSインスタンスを使用するように選択する場合は、まずプライマリGNSインスタンスを構成するか、共有GNS環境内の既存のGNSをプライマリGNSインスタンスに変更する必要があります。

1. クラスタ管理者として、次のようにして、既存のクラスタ内の任意のノードにプライマリGNSインスタンスを構成します。

```
# srvctl add gns -vip gns_vip -domain gns_subdomain
```

2. 次のようにして、プライマリGNSインスタンスを起動します。

```
# srvctl start gns
```

マルチクラスタ環境内に存在できるプライマリGNSインスタンスは1つのみです。GNSによって複数のプライマリGNSイン

スタンスが検出されると、startコマンドを終了します。この場合、ステップ5の説明に従って、このインスタンスをセカンダリGNSインスタンスとして構成できます。

3. 次のようにして、セカンダリGNSインスタンスのクライアント・データを作成します。

```
# srvctl export gns -clientdata file_name -role secondary
```

GNSにより、このゾーン・データ情報がOCRに格納され、構成後にセカンダリGNSインスタンスが更新されます。

4. 前述のステップで作成したクライアント・データ・ファイルをセカンダリGNSインスタンスにコピーします。
5. 次のようにして、セカンダリGNSインスタンスを構成します。

```
# srvctl add gns -vip gns_vip -clientdata file_name
```

6. 次のようにして、セカンダリGNSインスタンスを起動します。

```
# srvctl start gns
```

セカンダリGNSインスタンスを正常に構成および起動すると、プライマリGNSインスタンスにゾーン・データについての問合せが行われます。

プライマリGNSインスタンスおよびセカンダリGNSインスタンスの削除

プライマリGNSインスタンスおよびセカンダリGNSインスタンスをクラスタから削除できます。

プライマリGNSインスタンスは、別のプライマリGNSインスタンスを構成するまでは削除できません。置き換わるインスタンスがないのにプライマリGNSインスタンスを削除した場合、GNSは機能しなくなります。

クラスタ管理者として、次のようにして、セカンダリGNSインスタンスを選択し、それをプライマリGNSインスタンスに昇格します。

```
# srvctl modify gns -role primary
```

1. プライマリGNSインスタンスかセカンダリGNSインスタンスのどちらかを削除するかに応じて、次のようにしてインスタンスを停止します。

```
# srvctl stop gns
```

2. 次のようにして、インスタンスを削除します。

```
# srvctl remove gns
```

 ノート:

DNS 管理者は、クライアントが使用不能になったインスタンスの問合せを試行することがないように、インスタンスのネーム・サーバーおよび A/AAAA レコード・エントリを DNS ゾーン・データから削除できます。

SRVCTLを使用したGNSの起動と停止

srvctlコマンドを使用してGNSを起動および停止します。

サーバー・クラスタでのGNSの起動と停止は、次のそれぞれのコマンドをrootとして実行します。

```
# srvctl start gns  
# srvctl stop gns
```

ノート:



クライアント・クラスタでは GNS を起動したり停止できません。

GNSサーバーまたはGNSクライアント・クラスタへのクラスタの変換

GNSを実行していないクラスタをGNSサーバーまたはクライアント・クラスタに変換したり、サーバーおよびクライアント・クラスタのGNSクラスタ・タイプの構成を変更できます。

この項でのクラスタ・コンバージョンの使用例は、次のとおりです。

- [非GNSクラスタのGNSサーバー・クラスタへの変換](#)
- [非GNSクラスタのクライアント・クラスタへの変換](#)
- [GNSを実行しているシングル・クラスタのサーバー・クラスタへの変換](#)
- [GNSを実行しているシングル・クラスタのGNSクライアント・クラスタへの変換](#)

非GNSクラスタのGNSサーバー・クラスタへの変換

srvctlコマンドを使用して、GNSを実行していないクラスタをGNSサーバー・クラスタに変換できます。

1. rootとして次のコマンドを実行して、GNSをクラスタに追加し、有効なIPアドレスおよびドメインを指定します。

```
# srvctl add gns -vip IP_address -domain domain
```

2. GNSインスタンスを起動します。

```
# srvctl start gns -node node_name
```

ノート:



- GNS VIP を追加するときに、ドメインを指定する必要はありません。
- 現在別の GNS インスタンスで使用されている IP アドレスを指定することはできません。
- 構成済クラスタが GNS サーバー・クラスタとなるためには、DNS 委任が必要です。

非GNSクラスタのクライアント・クラスタへの変換

GNSを実行していないクラスタをクライアント・クラスタに変換するには、サーバー・クラスタから資格証明ファイルをインポートする必要があります。

次のようにして、GNSを実行していないクラスタをGNSクライアント・クラスタに変換します。

1. サーバー・クラスタで次のコマンドを実行して、GNSインスタンス・クライアント・データ構成をファイルにエクスポートします。

```
$ srvctl export gns -clientdata path_to_file -role client
```

ファイルの完全修飾パスを指定する必要があります。

ノート:



Oracle Universal Installer で生成する GNS 構成クライアント・データ・ファイルを、共有 GNS クライアントを作成する入力ファイルとして使用できます。

2. 次のコマンドをrootとして実行して、前述のステップで作成したファイルをクラスタ内のノードにインポートして、そのクラスタをクライアント・クラスタにします。

```
# srvctl add gns -clientdata path_to_file
```

ノート:



GNS データを含むファイルを、サーバー・クラスタからこのコマンドを実行するクラスタのノードにコピーします。

3. 次のようにSCAN名を変更します。

```
$ srvctl modify scan -scanname  
scan_name.client_cluster_name.server_GNS_subdomain
```

GNSを実行しているシングル・クラスタのサーバー・クラスタへの変換

GNSを実行しているシングル・クラスタをGNSサーバー・クラスタに変換するためには、何もする必要がありません。クライアント・クラスタが追加されると、自動的にサーバー・クラスタであるとみなされます。

GNSを実行しているシングル・クラスタのGNSクライアント・クラスタへの変換

srvctlコマンドを使用して、GNSを実行している単一クラスタをGNSクライアント・クラスタに変換できます。この変換が処理されている間、現在のGNSに接続した状態を維持する必要があるため、手順は、単一のクラスタをサーバー・クラスタに変換する場合の手順よりも複雑になります。

GNSを実行しているシングル・クラスタをGNSクライアント・クラスタに変換するには:

1. サーバー・クラスタで次のコマンドを実行して、GNSクライアント情報をファイルにエクスポートします。

```
$ srvctl export gns -clientdata path_to_client_data_file
```

ファイルの完全修飾パスを指定する必要があります。

2. クライアント・クラスタに変換するクラスタのGNSを停止します。

```
# srvctl stop gns
```

ノート:



変換の進行中は、GNS を使用した名前解決は使用できません。

3. サーバー・クラスタで次のコマンドを実行して、GNSインスタンスをエクスポートします。

```
$ srvctl export gns -instance path_to_file
```

ファイルの完全修飾パスを指定する必要があります。

4. GNSインスタンス・ファイルをインポートするには、サーバー・クラスタでrootとして次のコマンドを実行します。

```
# srvctl import gns -instance path_to_file
```

ファイルの完全修飾パスを指定する必要があります。

5. GNSインスタンスを開始するには、GNSインスタンス・ファイルをインポートしたノードで、rootとして次のコマンドを実行します。

```
# srvctl start gns
```

GNSインスタンスを起動するノードの名前を指定しないと、インスタンスはランダム・モードで開始します。

6. 次のコマンドを使用してGNSクライアント・クラスタからGNSを削除します。

```
# srvctl remove gns
```

7. 次のように、前のクラスタをクライアント・クラスタにします。

```
# srvctl add gns -clientdata path_to_client_data_file
```

ノート:



GNS データを含むファイルを、サーバー・クラスタからこのコマンドを実行するクラスタのノードにコピーします。

8. GNSクライアント・クラスタのSCANを変更して、次のようにクライアント・クラスタ名で修飾されたGNSサブドメインを使用するようにします。

```
$ srvctl modify scan -scanname scan_name.gns_domain
```

前述のコマンドでは、gns_domainの形式はclient_cluster_name.server GNS subdomainです。

別のクラスタへのGNSの移動

クラスタ障害または管理計画のいずれかが理由で、別のクラスタをGNSサーバー・クラスタにする必要が出た場合、srvctlコマンドを使用してGNSを別のクラスタに移動できます。

ノート:



この手順には、サーバー・クラスタおよびクライアント・クラスタのダウンタイムが必要です。また、GNS クライアント・データを新しいサーバー・クラスタから任意の Oracle Flex ASM とフリート・パッチ適用およびプロビジョニング・サーバーおよびクライアントにインポートする必要があります。

GNSを別のクラスタに移動するには:

1. 現在のサーバー・クラスタのGNSインスタンスを停止します。

```
# srvctl stop gns
```

2. GNSインスタンス構成をファイルにエクスポートします。

```
# srvctl export gns -instance path_to_file
```

ファイルへの完全修飾パスを指定します。

3. 以前のサーバー・クラスタからGNS構成を削除します。

```
# srvctl remove gns
```

4. GNSを新しいクラスタに追加します。

```
# srvctl add gns -domain domain_name -vip vip_name
```

別の方法として、VIPのIPアドレスを指定できます。

5. ステップ2で作成したファイルに格納されているインスタンス情報を使用して、次のように新しいサーバー・クラスタのGNSインスタンスを構成します。

```
# srvctl import gns -instance path_to_file
```

ノート:



以前のサーバー・クラスタの GNS データを含むファイルが、`srvctl import gns` コマンドを実行するクラスタのノードに存在する必要があります。

6. 新しいサーバー・クラスタのGNSインスタンスを起動します。

```
# srvctl start gns
```

IPv4からIPv6ネットワークへの移行時のGNSサブドメインの変更

IPv4ネットワークからIPv6ネットワークに移行するとき、GNSサブドメインを変更する必要があります。

GNSサブドメインを変更するには、次のようにして、IPv6ネットワークを追加し、GNSドメインを更新して、SCANを更新します。

1. 次のようにして、`srvctl modify network`コマンドを使用してIPv6サブネットを追加します。

```
$ srvctl modify network -subnet ipv6_subnet/ipv6_prefix_length[/interface] -  
nettype autoconfig
```

2. 次のようにして、GNSドメインを更新します。

```
$ srvctl stop gns -force  
$ srvctl stop scan -force  
$ srvctl remove gns -force  
$ srvctl add gns -vip gns_vip -domain gns_subdomain  
$ srvctl start gns
```

3. 次のようにして、新しいドメインでSCAN名を更新します。

```
$ srvctl remove scan -force  
$ srvctl add scan -scanname new_domain  
$ srvctl start scan
```

4. 次のようにして、ネットワークIPタイプをIPv4からIPv4 DHCPとIPv6両方のautoconfigに変換します。

```
$ srvctl modify network -iptype both
```

5. 次のようにして、ネットワークの使用プロトコルを、両方のプロトコルからIPv6 autoconfigのみに移行します。

```
$ srvctl modify network -iptype ipv6
```

DNSによるクラスタの名前解決をGNSにローリング変換

名前解決にDNSを使用しているOracle Grid Infrastructureクラスタ・ネットワークを、GNSで名前解決を取得するグリッド・ネーミング・サービス(GNS)を使用したクラスタ・ネットワークに変換できます。

次の手順を使用して、標準的なDNS名前解決ネットワークからGNS名前解決ネットワークにダウンタイムなしで変換します。

1. グリッド・ユーザー(grid)としてログインし、次のCluster Verification Utilityを使用して、クラスタをGNSに移動させるためのステータスを確認します(ここで、nodeListは、クラスタ・メンバー・ノードのカンマ区切りのリストです)。

```
$ cluvfy stage -pre crsinst -n nodeList
```

2. グリッド・ユーザーとして、次のコマンドを使用してGNS構成の整合性を確認します(ここで、domainは、解決をGNSに委譲されるドメインで、gns_vipはGNS VIPです)。

```
$ cluvfy comp gns -precrsinst -domain domain -vip gns_vip
```

3. rootユーザーとしてログインし、次のSRVCTLコマンドを使用してGNSリソースを構成します(ここで、domain_nameは、ネットワーク管理者が解決をGNSに委譲するよう構成したドメインで、ip_addressは、GNSがDNSリクエストをリスニングするIPアドレスです)。

```
# srvctl add gns -domain domain_name -vip ip_address
```

4. GNSを起動するには、次のコマンドを使用します。

```
# srvctl start gns
```

GNSが起動され、VIPおよびSCAN名を登録します。

5. 静的およびDHCPネットワーク・アドレスが混在するモードをサポートするために、rootとして、次のコマンドを使用してネットワークCRSリソースを変更します。

```
# srvctl modify network -nettype MIXED
```

必要なVIPアドレスがDHCPサーバーから取得され、決定されます。

6. グリッド・ユーザーとして、次のコマンドを入力して、Oracle Clusterwareが新しいGNS、動的アドレスおよびリスナー・エンドポイントを使用していることを確認します。

```
cluvfy stage -post crsinst -n all
```

7. 検証が正常に終了した後、GNSで解決されるSCANおよびVIPを使用するために、DNSで解決されるSCANまたはVIPを以前に使用したリモート・エンドポイントを変更します。

SCANを使用する各クライアントで、クライアントが使用するSCANを変更して、クライアントがGNSに委譲されたドメイン内のSCANを使用するようにします。

VIP名を使用している各クライアントでは、クライアントが同じサーバーVIP名およびGNSに移譲されたドメインのドメイン名を使用するように、各クライアントのVIP名を変更します。

8. rootとして次のコマンドを入力して、GNSサブドメインのSCAN名でシステムを更新します。

```
# srvctl modify scan -scanname scan_name.gns_domain
```

このコマンド構文のgns_domainは、この手順のステップ3で入力したドメイン名です。

9. すべてのクライアントで動的アドレスが使用されたら、次のように静的アドレスを無効化します。

```
$ srvctl modify network -nettype DHCP
```

関連トピック

- [Oracle Grid Infrastructureのインストール・ガイド](#)

ノード障害分離

障害分離は、障害が発生したノードをクラスタの残りのノードから分離させ、障害が発生したノードでデータが失われるのを防ぐプロセスです。

障害が発生したノードの分離には、Oracle Clusterwareやそのノードで実行されているオペレーティング・システムとの連携なく問題のノードを再起動できる外部メカニズムが関与します。この機能を提供するために、Oracle Clusterware 12cでは、業界標準の管理プロトコルであるIntelligent Platform Management Interface仕様(IPMI)(Baseboard Management Controller(BMC)とも呼ばれる)をサポートしています。

通常、IPMIを使用する障害分離は、Oracle Grid Infrastructureのインストール時の「障害の分離のサポート」画面でIPMI構成のオプションが提示されたときに構成します。インストール中にIPMIを構成しない場合は、インストール後にOracle Clusterware Controlユーティリティ(CRSCTL)を使用して構成できます(後続の項を参照)。

障害分離用にIPMIを使用するには、各クラスタ・メンバー・ノードに、IPMIバージョン1.5(Local Area Network (LAN)を介してIPMIがサポートされます)に対応するファームウェアが動作するIPMIデバイスを装備する必要があります。BMCに加えて、各ノードにipmiutilがインストールされている必要があります。ipmiutilユーティリティは、Oracle Grid Infrastructureインストールとともに配布されないため、ダウンロードする必要があります。ipmiutilは、ipmiutil.sourceforge.netまたは他のリポジトリからダウンロードできます。バージョン番号はヘルプ・プロンプトでも表示されますが、ipmiutilの実行によっても表示できます。

DHCPを使用してIPMIの動的IPアドレス割当てをサポートするには、クラスタ同期サービス・デーモンは、クラスタ同期サービスの起動時にローカルのIPMIデバイスと直接通信を行いIPMIデバイスのIPアドレスを取得する必要があります。(ただし、これはHP-UXおよびSolarisプラットフォームにはあてはまりません。これらのプラットフォームでは、IPMIデバイスに静的IPアドレスを割り当てる必要があります。)これは、各クラスタ・システムにインストールする必要があるIPMIドライバを介してIPMIデバイスと通信するIPMIプローブ・コマンド(OSD)を使用して行われます。

IPMIデバイスに静的IPアドレスを割り当てた場合、IPMIドライバは、クラスタ同期サービス・デーモンにとって必須ではありません。ただし、ipmitoolまたはipmiutilを使用してIPMIデバイスを構成するにはドライバが必要ですが、一部のプラットフォームでは管理コンソールを使用してこれを行うこともできます。

関連トピック

- [CRSCTLを使用したIPMIベース障害分離のインストール後の構成](#)

IPMI用のサーバー・ハードウェア構成

最初に、ipmiutilバイナリをインストールし、IPMIドライバをインストールして有効にし、IPMIデバイスを構成する必要があります(ご使用のプラットフォームの『Oracle Grid Infrastructureインストールおよびアップグレード・ガイド』を参照)。

関連トピック

- [Oracle Grid Infrastructureのインストール・ガイド](#)

CRSCTLを使用したIPMIベース障害分離のインストール後の構成

Oracle Clusterwareをインストールした後、crsctlコマンドを使用してIPMIベースの障害分離を構成します。このコマンドを使用して、IPMI構成を変更または削除することもできます。

これは、次のトピックで説明します。

- [Oracle ClusterwareでのIPMIインストール後の構成](#)
- [CRSCTLを使用したIPMI構成の変更](#)
- [CRSCTLを使用したIPMI構成の削除](#)

Oracle ClusterwareでのIPMIインストール後の構成

ipmiutilバイナリをインストールし、IPMIドライバをインストールして有効化し、IPMIデバイスを構成し、サーバー構成を完了した後、CRSCTLコマンドを使用してIPMI構成を完了できます。

インストールを開始する前に、ipmiutilバイナリをインストールし、サーバー・オペレーティング・システムにIPMIドライバをインストールして有効にし、各ノードでIPMIハードウェア(IPアドレス・モード、管理資格証明など)を構成しました([『Oracle Grid Infrastructureインストレーション・ガイド』](#)を参照)。Oracle Clusterwareをインストールすると、インストーラによって、IPMI管理者のユーザーIDおよびパスワードが収集され、ノードのローカル記憶域の(OLRにある)Oracle Walletに格納されます。

サーバーの構成が完了したら、各クラスタ・ノードで次の手順を実行し、ノードにIPMI管理者およびパスワードを登録します。

ノート:



IPMI が DHCP を使用して IP アドレスを取得するように構成されている場合は、IPMI をリセットするか、またはノードを再起動して、アドレスを取得するようにする必要がある可能性があります。

1. Oracle Clusterwareを起動します(起動すると、IPMIから現在のIPアドレスを取得できます)。これにより、起動時に必要となるクラスタウェアとIPMIの通信が可能であることが確認されます。

IPMIが構成される前にOracle Clusterwareが実行されていた場合は、Oracle Clusterwareを停止して再起動できます。または、IPMI管理ユーティリティを使用してIPMIのIPアドレスを取得してから、CRSCTLを使用して次のようなコマンドを実行することで、OLRにIPアドレスを格納します。

```
crsctl set css ipmiaddr 192.168.10.45
```

2. CRSCTLを使用して、crsctl set css ipmiadminコマンドを実行し、プロンプトにパスワードを指定することで、装備されているIPMI用に以前に作成したユーザーIDおよびパスワードをOLRに格納します。たとえば:

```
crsctl set css ipmiadmin administrator_name  
IPMI BMC password: password
```

このコマンドでは、指定した資格証明が検証され、これを使用して別のクラスタ・ノードがローカルIPMIにアクセスできない場合は失敗します。

ハードウェアおよびオペレーティング・システムの構成を完了し、Oracle ClusterwareにIPMI管理者を登録した後、IPMIベースの障害分離は完全に機能します。

インストールを開始する前に、ipmiutilをダウンロードしてクラスタ内の各ノードにインストールし、サーバー・オペレーティング・システムにIPMIドライバをインストールして有効にし、各ノードでIPMIハードウェア(IPアドレス・モード、管理資

格証明など)を構成します([『Oracle Grid Infrastructureインストール・ガイド』](#)を参照)。Oracle Clusterwareをインストールすると、インストーラによって、IPMI管理者のユーザーIDおよびパスワードが収集され、ノードのローカル記憶域の(OLRにある)Oracle Walletに格納されます。

CRSCTLを使用したIPMI構成の変更

既存のIPMIベースの障害分離構成を変更して、IPMIパスワードを変更するか、既存のインストールの障害分離に対するIPMIを構成する必要がある場合があります。ipmiutilバイナリ・ファイルへのパスの変更が必要になる場合もあります。

ご使用のプラットフォームに適したIPMI構成ツールでCRSCTLを使用して、これらの変更を行います。

たとえば、IPMIの管理者パスワードを変更するには、まず『Oracle Grid Infrastructureインストールおよびアップグレード・ガイド』で説明しているようにIPMI構成を変更してから、CRSCTLを使用してOLR内のパスワードを変更します。

Oracle Clusterwareで必要なIPMIの構成データはOLRのOracle Walletに保存されています。構成情報はセキュアなストアに保持されるため、この情報はOracle Clusterwareインストールの所有者アカウント(グリッド・ユーザー)が書き込む必要があります。このインストールのユーザーとしてログインする必要があります。

既存のIPMI構成を変更するには、次の手順を実行します。

1. `crsctl set css ipmiadmin administrator_name`コマンドを入力します。たとえば、ユーザーIPMIadmの場合は、次のようになります。

```
$ crsctl set css ipmiadmin IPMIadm
```

管理者パスワードを指定します。Oracle Clusterwareでは、ローカルIPMIの管理者の名前およびパスワードがOLRに格納されます。

新しい資格証明が格納されると、Oracle Clusterwareは新しい資格証明を取得し、必要に応じて配布することができます。

2. `crsctl set css ipmiaddr bmc_ip_address`コマンドを入力します。たとえば:

```
$ crsctl set css ipmiaddr 192.0.2.244
```

このコマンドでは、ローカルIPMIの新しいIPMI IPアドレスがOLRに格納され、IPアドレスの格納後、Oracle Clusterwareは、新しい構成を取得し、必要に応じて配布することができます。

3. `crsctl get css ipmiaddr`コマンドを入力します。たとえば:

```
$ crsctl get css ipmiaddr
```

このコマンドでは、ローカルIPMIのIPアドレスがOLRから取得され、コンソールに表示されます。

4. 次のように、OLRからローカルIPMIのIPMI構成情報を削除して、レジストリ・エントリを削除します。

```
$ crsctl unset css ipmiconfig
```

関連トピック

- [Oracle Grid Infrastructureのインストール・ガイド](#)
- [Oracle RAC環境のCRSCTLコマンド](#)

CRSCTLを使用したIPMI構成の削除

IPMIの使用を完全に停止する場合、またはIPMIの最初の構成がOracle Clusterwareをインストールしたユーザー以外のユーザーによって行われていた場合は、CRSCTLを使用してクラスタからIPMI構成を削除できます。

後者に該当する場合、Oracle ClusterwareはIPMI構成データにアクセスできず、Oracle ClusterwareソフトウェアはIPMIを使用できないため、Oracle ClusterwareをインストールしたユーザーとしてIPMIを再構成する必要があります。

IPMIを完全に削除するには、次のステップを実行します。Oracle ClusterwareをインストールしたユーザーとしてIPMIを再構成するには、ステップ3および4を実行し、前の項のステップ3および4を繰り返します。

1. 次のように、IPMIドライバを無効にして、起動時のインストールを回避します。

```
/sbin/modprobe -r
```

2. LAN経由でのアクセスを防ぐためにipmitoolまたはipmiutilを使用してローカルIPMIのIPMI-over-LANを無効にするか、またはIPMI管理者のユーザーIDとパスワードを変更します。
3. Oracle Clusterwareが実行中であることを確認し、次のコマンドでCRSCTLを使用し、OLRからIPMI構成データを削除します。

```
$ crsctl unset css ipmiconfig
```

4. rootとして次のコマンドを実行し、Oracle Clusterwareを再起動して、IPMI構成なしで実行するようにします。

```
# crsctl stop crs  
# crsctl start crs
```

関連トピック

- [CRSCTLを使用したIPMI構成の変更](#)
- [Oracle Grid Infrastructureインストールおよびアップグレードガイドfor Linux](#)

手動構成ネットワーク上のネットワーク・アドレスの理解

手動構成ネットワーク上のネットワーク・アドレスの概念および要件を理解することは役立ちます。

この項では、次の項目について説明します。

- [ネットワーク・アドレスの構成要件の理解](#)
- [SCANアドレスおよびクライアント・サービス接続の理解](#)
- [有効なノード・チェックにおけるSCANリスナーおよびサービス登録制限](#)

ネットワーク・アドレスの構成要件の理解

Oracle Clusterware構成には、少なくとも1つのパブリック・ネットワーク・インタフェースと1つのプライベート・ネットワーク・インタフェースが必要です。

- パブリック・ネットワーク・インタフェースは、データベース・サーバー上のデータにアクセスするためにユーザーとアプリケーション・サーバーを接続します。
- プライベート・ネットワーク・インタフェースはノード間の通信用で、Oracle Clusterwareによって排他的に使用されます。

パブリック・ネットワーク・インタフェースは、特定のネットワーク上の[IPv4](#)、[IPv6](#)または両方のタイプのアドレスに対して構成できます。冗長なネットワーク・インタフェース(ボンディングまたはチーミングされたインタフェース)を使用する場合、Oracleでは、1つのインタフェースがIPv4アドレスをサポートしていますが、別のインタフェースがIPv6アドレスをサポートするような構成はサポートしていないので注意してください。冗長なインタフェースのネットワーク・インタフェースは、同じIPプロトコルを使用して構成する必要があります。

1つ以上のプライベート・ネットワーク・インタフェースを、すべてのネットワーク・アダプタに対してIPv4アドレスかIPv6アドレスのどちらかを使用して構成できます。プライベート・ネットワーク・インタフェースに対してIPv4アドレスとIPv6アドレスを組み合わせることはできません。

ノート:



クラスタ内のプライベート・ネットワークに対して IPv6 を使用できるのは、Oracle Clusterware 12c リリース 2 (12.2)以上を使用する場合のみです。

クラスタ内のすべてのノードには、同じIPプロトコル構成を使用する必要があります。すべてのノードがIPv4を使用するか、すべてのノードがIPv6を使用するか、すべてのノードがIPv4およびIPv6の両方を使用するかのいずれかです。クラスタ内の一部のノードがIPv6アドレスのみをサポートするように構成し、その他のノードがIPv4アドレスのみをサポートするように構成することはできません。

VIPエージェントは、ステートレス・アドレス自動構成(RFC 2462)を使用したIPv6の生成をサポートしており、これらのアドレスをGNSで通知します。srvctl config networkコマンドを実行して、DHCPまたはステートレス・アドレス自動構成が使用されているかを識別します。

この項には次のトピックが含まれます:

- [IPv6アドレスのフォーマットについて](#)
- [名前解決およびネットワーク・リソースのアドレス・タイプ](#)

IPv6アドレスのフォーマットについて

Oracle Grid Infrastructureクラスタの各ノードは、同一ネットワーク上でIPv4アドレスとIPv6アドレスの両方をサポートできます。推奨されるIPv6アドレス書式は次のとおりです(各xは、16進文字を表します)。

```
xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx
```

IPv6アドレス・フォーマットはRFC 2460によって定義されます。Oracle Grid Infrastructureでは次のIPv6アドレスがサポートされます。

- RFC 4193によって定義されるグローバルIPv6アドレスとサイトローカルのIPv6アドレス。

ノート:



RFC 1884 で定義されているリンクローカルとサイトローカルの IPv6 アドレスはサポートされません。

- IPアドレスの各フィールドの先頭に圧縮されたゼロを使用。
- 圧縮されて、「::」セパレータで表現された空のフィールド。たとえば、IPv6アドレス 2001:0db8:0000:0000:0000:8a2e:0370:7334を2001:db8::8a2e:370:7334と表記できます。
- 8ビットの断片を含む下位4つのフィールド(標準的なIPv4アドレス・フォーマット)。たとえば、2001:db8:122:344::192.0.2.33です。

名前解決およびネットワーク・リソースのアドレス・タイプ

ネットワーク構成の確認にはsrvctl config networkコマンド(構成の確認用)を、ネットワーク・アドレス・タイプの制御にはsrvctl modify network -iptypeコマンドをそれぞれ使用します。

srvctl modify network -nettypeコマンドを使用してアドレスの取得方法を構成できます。-nettypeパラメータの値をdhcpまたはstaticに設定して、IPv4ネットワーク・アドレスの取得方法を制御します。または、-nettypeパラメータの値をautoconfigまたはstaticを設定して、IPv6アドレスの生成方法を制御します。

-nettypeと-iptypeパラメータは直接関連していませんが、-iptype ipv4と-nettype dhcpを同時に使用したり、-iptype ipv6と-nettype autoconfigを同時に使用できます。

ノート:

ネットワークが IPv4 サブネットと IPv6 サブネットの両方で構成されている場合、-nettype が mixed に設定されていると両方のサブネットがサポートされません。



-nettype を mixed に設定した場合、IPv4 から IPv6 への移行はサポートされません。最初に static から dhcp への移行が完了してから、IPv6 をサブネットに追加する必要があります。

同様に、-nettype を mixed に設定した場合、IPv6 から IPv4 への移行の開始はサポートされません。最初に autoconfig から static への移行が完了してから、IPv4 をサブネットに追加する必要があります。

関連トピック

- [クラスタ内のネットワーク・アドレス構成](#)

SCANアドレスおよびクライアント・サービス接続の理解

パブリック・ネットワーク・アドレスはクライアントにサービスを提供するために使用します。

クライアントで単一クライアント・アクセス名(SCAN)アドレスに接続している場合は、クラスタに対してノードを追加または削除する際にパブリックおよび仮想IPアドレスを変更する必要がある場合がありますが、クライアントを新しいクラスタ・アドレスで更新する必要はありません。

ノート:



listener.ora ファイルを編集して、SCAN およびノード・リスナーの Oracle Net リスナー・パラメータを変更できます。たとえば、TRACE_LEVEL_listener_name を設定できます。ただし、リスナー・エージェントは動的にリスナー・エンドポイントを管理するため、プロトコル・アドレス・パラメータを設定して、リスニング・エンドポイントを定義することはできません。

SCANはクラスタの別名のように機能します。ただし、SCANはクラスタ内の任意のノードで解決され、ノードのVIPアドレスとは異なり、SCANに接続するクライアントでは、クラスタに対してノードを追加または削除する際に、更新されたVIPアドレスは必要なくなります。SCANアドレスはクラスタ内のノード・アドレスではなくクラスタに解決されるため、SCANアドレス構成に影響を与えることなく、クラスタでノードを追加または削除できます。

SCANは完全修飾名(ホスト名およびドメイン)で、SCANに割り当てられたすべてのアドレスに解決するように構成されています。SCANは、DNSサーバー上のSCAN名に構成された3つすべてのアドレスに解決するか、GNS構成のクラスタ内に解決します。SCANリスナーはクラスタ内の任意のノードで実行できます。SCANは、クライアント構成が特定のデータベースを実行するノードに依存する必要がないように、データベースに位置の独立性を提供します。

Oracle Database 11gリリース2 (11.2)以上のインスタンスは、SCANリスナーにのみリモート・リスナーとして登録されます。アップグレードしたデータベースは、リモート・リスナーとしてSCANリスナーに登録されるとともに、引き続きすべてのノード・リスナーにも登録されています。

ノート:



インストール時に SCAN 名を指定するという Oracle Clusterware のインストール要件により、サーバーの /etc/hosts ファイルを使用して 1 つ以上の IP アドレスを解決してこのインストール要件を回避し、SCAN に必要なインフラストラクチャがない場合は、インストール後、SCAN を無視し、VIP を使用してクラスタ内のデータベースに接続できます。

Oracle では、SCAN アドレスの削除はサポートされていません。

関連トピック

- [『Oracle Database Net Servicesリファレンス・ガイド』](#)

有効なノード・チェックにおけるSCANリスナーおよびサービス登録制限

有効なノード・チェックを使用して、SCANリスナーが登録を受け入れるノードおよびサブネットを指定できます。SRVCTLを使用してノードおよびサブネット情報を指定できます。

SRVCTLは、ノードおよびサブネット情報をSCANリスナー・プロファイルに格納します。SCANリスナー・エージェントは、リソース・プロファイルからその情報を読み出し、listener.oraファイルに書き込みます。

リスナーへのデータベース・インスタンスの登録は、リクエスト元が有効なノードである場合にのみ成功します。ネットワーク管理者は、有効なノードおよび除外ノードのリストを指定したり、有効なノードの確認を完全に無効にすることができます。有効なノードのリストでは、データベースに登録できるノードやサブネットを明示的にリストします。除外ノードのリストでは、データベースに登録できないノードを明示的にリストします。動的登録を制御することによって、Oracle RACデプロイメントの管理性およびセキュリティが向上します。

デフォルトでは、SCANリスナー・エージェントはREMOTE_ADDRESS_REGISTRATION_listener_nameをプライベートIPエンドポイントに設定します。SCANリスナーは、プライベート・ネットワークからの登録要求のみを受け入れます。SCANリスナーのプライベート・ネットワークにアクセスできないリモート・ノードは、listener.oraファイルのregistration_invited_nodes_aliasパラメータを使用して、またはコマンドライン・インタフェースのSRVCTLを使用してSCANリスナーを変更して、有効なノードのリストに含める必要があります。

ノート:



Oracle Grid Infrastructure 12c 以降、SCAN リスナーについて、VALID_NODE_CHECKING_REGISTRATION_listener_name および REGISTRATION_INVITED_NODES_listener_name パラメータが listener.ora ファイルに設定されている場合、リスナー・エージェントはこれらのパラメータを上書きします。

SRVCTLユーティリティを使用してinvitednodes値とinvitedsubnets値を設定すると、リスナー・エージェントは自動的にVALID_NODE_CHECKING_REGISTRATION_listener_nameをSUBNETに設定し、REGISTRATION_INVITED_NODES_listener_nameをlistener.oraファイルで指定されたリストに設定します。

CRSによって管理されるその他のリスナーの場合、リスナー・エージェントは、listener.oraファイルでまだ設定されていない場合にのみ、listener.oraファイルでVALID_NODE_CHECKING_REGISTRATION_listener_nameをSUBNETに設定します。SRVCTLユーティリティでは、SCAN以外のリスナーについてinvitednodes値とinvitedsubnets値の設定はサポートされていません。リスナー・エージェントは、SCAN以外のリスナーについてlistener.oraファイルのREGISTRATION_INVITED_NODES_listener_nameを更新しません。

関連トピック

- [Oracle Real Application Clusters管理およびデプロイメント・ガイド](#)
- [Oracle Database Net Services管理者ガイド](#)

共有単一クライアント・アクセス名の構成

共有単一クライアント・アクセス名(SCAN)により、専用クラスタと他のクラスタで、一連のSCAN仮想IP(VIP)とリスナーを共有できます。

共有単一クライアント・アクセス名の構成について

データベース・サーバーとデータベース・クライアントの両方で共有単一クライアント・アクセス名(SCAN)を構成する必要があります。

共有SCANを使用すると、クラスタごとにSCAN VIPセットをデプロイするのではなく、複数のクラスタでSCAN仮想IP(VIP)アドレスの単一共通セットを使用してユーザー接続を管理できます。たとえば、10のクラスタで、クラスタごとに3つのSCAN VIPをデプロイする場合、合計30のIPアドレスを使用しますが、共有SCANデプロイメントでは、それらの同じ10のクラスタに対して3つのSCAN VIPのみデプロイするため、必要なIPアドレスは3つのみです。

SCAN VIP(共有またはそれ以外)は、Oracle Real Application Cluster (Oracle RAC)データベース・クラスタに必要ですが、アプリケーション・メンバー・クラスタやドメイン・サービス・クラスタには必要ないので注意してください。

共有SCANを構成する一般的な手順として、最初にサーバー(つまり、共有SCANをホストするクラスタ)でsrvctlユーティリティを使用して構成を行い、次に、クライアント(この共有SCANを使用するOracle RACクラスタ)でこのユーティリティを使用して構成を行います。サーバーでは、srvctlを使用した構成に加えて、環境変数の設定、資格証明ファイルの作成が必要であり、またSCANクラスタ固有のOracle Notification Service (ONS)プロセスが、ONS構成を作成および管理する専用の構成ディレクトリにアクセス可能であることを保証する必要があります。

共有SCANの使用の構成

他の必要な構成タスクの実行に加えて、SRVCTLを使用して、専用クラスタをホストするサーバーで共有SCANを構成します。

1. 共有SCANを構成するサーバー・クラスタにログインします。
2. 次のように、この共有SCANクラスタ専用のSCANリスナーを作成します。

```
$ srvctl add scan_listener -clientcluster cluster_name
```

3. サーバー・クラスタに固有の新しいOracle Notification Service (ONS)リソースを作成します。

```
$ srvctl add ons -clientcluster cluster_name
```

srvctl add onsコマンドにより、SCANにIDを割り当てます。

4. 次のようにSCANリスナーをクライアント・クラスタにエクスポートします。

```
$ srvctl export scan_listener -clientcluster cluster_name -clientdata file_name
```

5. 次のようにONSリソースをクライアント・クラスタにエクスポートします。

```
$ srvctl export ons -clientcluster cluster_name -clientdata file_name
```

ノート:



SCAN リスナーと ONS の両方に同じ資格証明ファイル名を使用できます。これらのオブジェクトをクライアント・クラスタに追加すると、SRVCTL により、使用する資格証明ファイルが作成されます。

6. このサービスを使用する各クラスタで共有SCANを構成します。

- 共有SCANを構成するクライアント・クラスタにログインします。
- 次のようにクライアント・クラスタにSCANを追加します。

```
$ srvctl add scan -clientdata file_name
```

- 次のように、このクライアント・クラスタ専用のSCANリスナーを作成します。

```
$ srvctl add scan_listener -clientdata file_name
```

- 次のように、このクラスタのONSリソースを作成します。

```
$ srvctl add ons -clientdata file_name
```

ノート:



前述の各コマンドでは、前のステップで作成した資格証明ファイルの名前を指定します。

手動構成システム上のネットワーク・アドレスの変更

手動構成システム上でネットワーク・アドレス・メンテナンスを実行できます。

これは、次のトピックで説明します。

- [SRVCTLを使用した仮想IPアドレスの変更](#)
- [Oracle Clusterwareのプライベート・ネットワーク構成の変更](#)
- [SRVCTLを使用したネットワークの作成](#)
- [クラスタ内のネットワーク・アドレス構成](#)
- [SRVCTLを使用した、静的IPv4アドレスの静的IPv6アドレスへの変更](#)
- [SRVCTLを使用した、動的IPv4アドレスの動的IPv6アドレスへの変更](#)
- [IPv4ネットワークのIPv4およびIPv6ネットワークへの変更](#)
- [SRVCTLを使用したVIPアドレスのネットワークのIPv4からIPv6への移行](#)

SRVCTLを使用した仮想IPアドレスの変更

SRVCTLを使用して、仮想IPアドレスを変更できます。

Oracle Database 11gリリース2(11.2)より前のOracle DatabaseリリースのパブリックVIPアドレスを使用するように構成されたクライアントは、既存の接続アドレスを引き続き使用できます。SCANを使用するようにクライアントを構成することをお薦

めしますが、ユーザーがSCANを使用する必要はありません。以前のバージョンのOracle Databaseは、アップグレードされるとSCANに登録され、クライアントでは、SCANを使用してこのデータベースへの接続を開始するか、接続用のVIPアドレスの使用を継続することができます。

クライアント接続用のVIPアドレスの使用を継続する場合、Oracle DatabaseおよびOracle ASMの実行が継続している間にVIPアドレスを変更できます。ただし、アドレスを変更する間は、サービスを停止する必要があります。VIPアドレスを再起動すると、サービスもノードで再起動します。

この手順を使用して、静的パブリック・サブネットでDHCPを使用するように変更することはできません。`srvctl add network -subnet`コマンドは、DHCPネットワークを作成するだけです。

ノート:



次の説明は、VIP アドレスのみを変更する方法を示すもので、VIP アドレスに関連付けられたホスト名は変更しないことを想定しています。GNS を使用していて、VIP が DHCP を使用して割り当てられている場合は、VIP アドレスを手動で更新する必要はありません。

VIP アドレスのみを変更する場合、DNS およびクライアントの hosts ファイルを更新します。また、サーバーの hosts ファイルが VIP アドレスに対して使用されている場合は、それも更新します。

VIPアドレスを変更するには、次のステップを実行します。

1. 次のコマンド構文を使用して、変更するVIPアドレスのノードで実行されているすべてのサービスを停止します。

database_nameはデータベースの名前、service_name_listは停止するサービスのリスト、およびmy_nodeは変更するVIPアドレスのノードの名前です。

```
srvctl stop service -db database_name -service "service_name_list" -node node_name
```

次の例では、-dbオプションでデータベース名(grid)を指定し、適切なノード(mynode)上のサービス(sales,oltp)を指定しています。

```
$ srvctl stop service -db grid -service "sales,oltp" -node mynode
```

2. `srvctl config vip`コマンドを実行して、VIPアドレスの現在のIPアドレスを確認します。このコマンドにより、ネットワーク・インタフェースの1つにバインドされた現在のVIPアドレスが表示されます。node03-vipという名前のVIPに構成されたVIPアドレスの例を次に示します。

```
$ srvctl config vip -vipname node03-vip
VIP exists: /node03-vip/192.168.2.20/255.255.255.0/eth0
```

3. `srvctl stop vip`コマンドを使用してVIPリソースを停止します。

```
$ srvctl stop vip -node node_name
```

4. LinuxまたはUNIXシステムで`ifconfig -a`コマンドを実行(Windowsシステムの場合は`ipconfig /all`コマンドを発行)して、VIPリソースが実行されていないことを確認し、インタフェース(この例ではeth0:1)が出力に表示されていないことを確認します。
5. LinuxおよびUNIXシステムではすべてのノードの/etc/hostsファイルに、Windowsシステムでは%windir%\system32\drivers\etc\hostsファイルに、必要なすべての変更を加え、DNSを適切に変更して、新しいIPアドレスを古いホスト名に関連付けます。

6. VIPリソースを変更する前に、デフォルト・ネットワークに異なるサブネットまたはネットワーク・インタフェース・カードを使用するには、`srvctl modify network -subnet subnet/netmask/interface`コマンドをrootとして使用して、ネットワーク・リソースを変更する必要があります(subnetは新しいサブネット・アドレス、netmaskは新しいネットマスク、interfaceは新しいインタフェースです)。サブネットの変更後、次のステップで説明するとおり、各ノードのVIPを新しいサブネットのIPアドレスに変更する必要があります。
7. 次の`srvctl modify nodeapps`構文を使用して、ノード・アプリケーションを変更し、新しいVIPアドレスを指定します。

```
$ srvctl modify nodeapps -node node_name -address new_vip_address
```

コマンドには次のフラグおよび値が含まれます。

- `-n node_name`はノード名です。
- `-A new_vip_address`はノード・レベルのVIPアドレスで、次のようになります。
`name|ip/netmask/[if1[|if2|...]]`

たとえば、rootユーザーとして次のコマンドを実行します。

```
# srvctl modify nodeapps -node mynode -address  
192.168.2.125/255.255.255.0/eth0
```

インストール所有者アカウントとしてこのコマンドを実行しようとすると、エラーが発生する場合があります。たとえば、インストール所有者がoracleの場合は、「PRCN-2018: 現在のユーザーoracleは、権限のあるユーザーではありません」というエラーが表示される可能性があります。このエラーを回避するには、rootまたはシステム管理者アカウントとしてコマンドを実行します。

8. 次のようにして、`srvctl start vip`コマンドを実行し、ノードVIPを起動します。

```
$ srvctl start vip -node node_name
```

次のコマンドの例では、mynodeという名前のノードでVIPを起動します。

```
$ srvctl start vip -node mynode
```

9. クラスタ内の各ノードでこのステップを繰り返します。

SRVCTLユーティリティはクラスタ全体の管理ツールであるため、各クラスタ・ノードにログインせずに、クラスタ内の任意のノードから特定のノードに対し、これらのタスクを完了することができます。

10. 次のコマンドを実行して、クラスタが構成されているすべてのノード間の接続性を確認します。このコマンドは、クラスタ・ノード上で使用可能なすべてのネットワーク・インタフェースを検出し、検出されたインタフェースからすべてのノードへの接続性を確認します。また、ノードで使用可能な、VIPアドレスとしての使用に適したすべてのインタフェースを表示します。

```
$ cluvfy comp nodecon -n all -verbose
```

関連トピック

- [Oracle Real Application Clusters管理およびデプロイメント・ガイド](#)

Oracle Clusterwareのプライベート・ネットワーク構成の変更

Oracle Clusterwareプライベート・ネットワーク構成を変更できます。

この節では、以下のトピックについて説明します。

- [プライベート・ネットワークおよびネットワーク・インタフェースについて](#)
- [冗長インターコネクトの使用](#)
- [OIFCFGを使用したインタフェース名の変更結果](#)
- [ネットワーク・インタフェースの変更](#)

プライベート・ネットワークおよびネットワーク・インタフェースについて

Oracle Clusterwareでは、各ノードが、(パブリック・ネットワークに加えて)プライベート・ネットワークに接続されている必要があります。プライベート・ネットワーク接続は、クラスタ・インターコネクトとも呼ばれます。

[表2-1](#)に、ネットワーク・インタフェース・カードおよびプライベートIPアドレスがどのように格納されるかを示します。

すべてのノードが、同一のサブネットに接続された同一のネットワーク・インタフェース(oifcfgコマンドによりグローバル・インタフェースとして定義済)を使用しているクラスタのみをサポートしています。ノードごとに異なるネットワーク・インタフェース(ノード固有のインタフェース)を使用することはできません。

表2-1 ネットワーク・インタフェース、プライベートIPアドレスおよびプライベート・ホスト名の記憶域

エンティティ	格納先	コメント
ネットワーク・インタフェース名	オペレーティング・システム たとえば、eth1 など	ネットワーク・インタフェース名を指定する場合、ワイルドカードを使用できます。 たとえば、eth* など
プライベート・ネットワーク・インタフェース	Oracle Clusterware のグリッド・プラグ・アンド・プレイ(GPnP)・プロファイル内	インストール時にインタフェースを「プライベート」とすることによって、使用するインタフェースをプライベート・インタフェースとして構成するか、oifcfg setif コマンドを使用して、インタフェースをプライベート・インタフェースに指定します。

関連トピック

- [Oracle Interface Configurationツール\(OIFCFG\)のコマンド・リファレンス](#)

冗長インターコネクトの使用

インストール時、またはインストール後にoifcfg setifコマンドを使用してインタフェースのロールをプライベートとして分類することにより、冗長なインターコネクトの使用に対して複数のインタフェースを定義できます。

このようにすると、定義したインタフェースの数に応じて1つから4つの高可用性IP(HAIP)アドレスがOracle Clusterwareによって作成され、Oracle DatabaseおよびOracle ASMインスタンスは、これらのアドレスを使用して高可用性のロード・バランスされた通信を実現します。

デフォルトでは、Oracleソフトウェア(11gリリース2(11.2.0.2)以上のOracle RAC、Oracle ASMおよびOracle ACFSなど)は、そのすべてのトラフィックのHAIPアドレスとして、「プライベート」ロールを指定したインタフェースのHAIPアドレスを使用するので、提供されたクラスタ・インターコネクト・インタフェースのセット全体でロード・バランシングが可能になります。定義済のクラスタ・インターコネクト・インタフェースのいずれかに障害が発生するか、または通信できなくなった場合、Oracle Clusterwareは、機能している残りのインタフェースのいずれかに対応するHAIPアドレスを透過的に移動します。

たとえば、インストール後に新しいインタフェースを、サブネット番号が172.16.2.0のeth3というサーバーに追加する場合は、次のコマンドを使用して、このインタフェースをプライベート・インタフェースとしてOracle Clusterwareで使用できるようにします。

```
$ oifcfg setif -global eth3/172.16.2.0:cluster_interconnect
```

Oracle Clusterwareではeth3で169.254.*.*というHAIPアドレス(HAIP用の予約済サブネット)が使用され、データベースOracle ASMおよびOracle ACFSはそのアドレスを通信に使用しますが、Oracle Clusterwareは自身の通信に172.16.2.0アドレスも使用します。

注意:



HAIP サブネット(169.254.*.*)を分類するために OIFCFG を使用しないでください。OIFCFG は、Oracle Clusterware のインタフェース名、サブネットおよびタイプ(パブリック、クラスタ・インターコネクトまたは Oracle ASM)を記録するために使用できます。ただし、OIFCFG を使用して各インタフェースの実際の IP アドレスを変更することはできません。

ノート:



Oracle Clusterware が使用するインタフェースは、定義されているインタフェースの数に関係なく、常に最大で 4 つです。あるインタフェースに障害が発生すると、HAIP アドレスは、定義されたセット内の別の構成済インタフェースに移動します。

HAIP アドレスが 1 つのみで、選択するインタフェースが複数ある場合、HAIP アドレスの移動先のインタフェースは、そのアドレスが構成された元のインタフェースではなくなります。Oracle Clusterware は、HAIP アドレスの追加先として、数が最も小さいサブネットのインタフェースを選択します。

関連トピック

- [Oracle Grid Infrastructureのインストール・ガイド](#)

OIFCFGを使用したインタフェース名の変更結果

インタフェース名の変更結果は、どの名前を変更するか、またIPアドレスも変更するかどうかによって異なります。

インタフェース名のみを変更する場合、結果は、それほど重要ではありません。OCR内に格納されたパブリック・インタフェースの名前を変更する場合、クラスタのノード・アプリケーションも変更する必要があります。したがって、この変更を有効にするために、ノード・アプリケーションを停止する必要があります。

ネットワーク・インタフェースの変更

OIFCFGコマンドを使用して、ネットワーク・インタフェースおよびそれに関連するサブネット・アドレスを変更できます。

この手順では、Oracle ClusterwareおよびOracle Databaseが以前使用していたクラスタ内の各ノードで、ネットワーク・インタフェースおよびIPアドレスを変更します。

注意:



Oracle RAC(RDBMS)インターコネクトで使用するインタフェースは、Oracle Clusterware でホスト名に使用されるインタフェースと同じである必要があります。Oracle Clusterware で監視されない別のインタフェースで、Oracle RAC のプライベート・インターコネクトを構成しないでください。

1. 次のコマンドを実行し、Oracle Clusterwareがすべてのクラスタ・ノードで実行されていることを確認します。

```
$ olsnodes -s
```

このコマンドでは、次のような出力が戻され、Oracle Clusterwareがクラスタ内のすべてのノードで実行されていることが表示されます。

```
./olsnodes -s  
myclustera Active  
myclusterc Active  
myclusterb Active
```

2. 代替インタフェースが構成され、すべてのノードのオペレーティング・システムで操作可能であることを確認します。ご使用のプラットフォームのifconfigコマンド(Windowsではipconfig)を使用します。たとえば、Linuxでは次のように使用します。

```
$ /sbin/ifconfig
```

3. 次のコマンドを使用して、新しいインタフェースの名前およびサブネット・アドレスを指定して、新しいインタフェースをクラスタに追加します。

```
$ oifcfg setif -global if_name/subnet:cluster_interconnect
```

インタフェース名にはワイルドカードを使用できます。たとえば、oifcfg setif -global "eth*/192.168.0.0:cluster_interconnect"は有効な構文です。ただし、他のクラスタ・インタフェースで使用している他のアドレスやマスクとまぎらわしくならないように注意してください。ワイルドカードを使用すると、次のような警告が表示されます。

```
eth*/192.168.0.0 global cluster_interconnect  
PRIF-29: Warning: wildcard in network parameters can cause mismatch  
among GPnP profile, OCR, and system
```

ノート:



従来型ネットワーク構成ではワイルドカードがサポートされないため、ワイルドカードは更新の時点で現行のノード構成を使用して解決されます。

4. Oracle ASMネットワークを変更した場合は、次のようにしてOracle ASMリスナーを更新します。

```
$ srvctl update listener -listener listener_name -asm -remove -force  
$ srvctl add listener -listener listener_name -asmlistener -subnet subnet
```

5. 前のステップが完了したら、次のように、以前のインタフェースの名前およびサブネット・アドレスを指定して、以前のサブネットを削除できます。

```
oifcfg delif -global if_name/subnet
```

たとえば:

```
$ oifcfg delif -global eth1/10.10.0.0
```

注意:



このステップは、代替インターフェイスがグリッド・プラグ・アンド・プレイ構成に取り込まれた後でのみ実行する必要があります。有効な代替インターフェイスを準備せずにクラスタのインターフェイスを削除すると、無効なクラスタ構成になる可能性があります。

6. 次のコマンドを使用して現行の構成を検証します。

```
oifcfg getif
```

たとえば:

```
$ oifcfg getif
eth2 10.220.52.0 global cluster_interconnect
eth0 10.220.16.0 global public
```

7. プライベート・ネットワークを変更した場合は、各ノードでrootとして次のコマンドを実行して、すべてのノードでOracle Clusterwareを停止します。

```
# crsctl stop crs
```

ノート:



eth0 および eth1 で HAIP を構成したときに、eth1 を eth3 に置換する場合、Oracle Clusterware を停止する必要はありません。ただし、eth0 と eth1 をすでに構成している HAIP 構成に、インターフェイスの別セット(eth2 と eth3 など)を追加する場合は、Oracle Clusterware を停止する必要があります。

8. Oracle Clusterwareが停止したら、ifconfigコマンドを使用して、削除したネットワーク・インターフェイスをオペレーティング・システムで構成解除できます。たとえば:

```
$ ifconfig down
```

この時点で、ネットワーク・インターフェイスからの古いサブネットのIPアドレスは、Oracle Clusterwareから構成解除されます。このコマンドは、オペレーティング・システムのIPアドレスの構成に影響しません。

ifconfigによる変更は、永続的ではないため、オペレーティング・システムの構成変更を更新する必要があります。

9. クラスタの各ノードでrootユーザーとして次のコマンドを実行し、Oracle Clusterwareを再起動します。

```
# crsctl start crs
```

変更は、Oracle Clusterwareの再起動時に有効になります。

CLUSTER_INTERCONNECTS初期化パラメータを使用している場合は、変更を反映するためにそれを更新する必要があります。

関連トピック

- [Oracle Interface Configurationツール\(OIFCFG\)のコマンド・リファレンス](#)

SRVCTLを使用したネットワークの作成

SRVCTLを使用して、クラスタ・メンバー・ノードにネットワークを作成し、アプリケーション構成情報を追加します。

次のように、クラスタ・メンバー・ノード用のネットワークを作成します。

1. rootとしてログインします。
2. 次の構文を使用してノードにノード・アプリケーションを追加します。

```
srvctl add nodeapps -node node_name -address {vip |  
addr}/netmask[/if1[|if2|...]] [-pingtarget "ping_target_list"]
```

前の構文では:

- node_nameはノードの名前です。
- vipはVIP名で、addrはIPアドレスです。
- netmaskはネットマスクです。
- if1[|if2|...]は、アプリケーションで使用するために結合されたインタフェースの、パイプ文字(|)区切りのリストです
- ping_target_listは、pingの対象のIPアドレスまたはホスト名のカンマ区切りリストです

ノート:



- リンク・ステータスの監視が仮想マシン環境と同様に機能しない場合は-pingtarget パラメータを使用します。
- srvctl add nodeapps -help コマンドを入力して、他の構文オプションを確認します。

次の例では、srvctl add nodeappsを使用してIPv4ノード・アプリケーションを構成しており、ノード名はnode1で、ネットマスクは255.255.252.0で、インタフェースはeth0です。

```
# srvctl add nodeapps -node node1 -address node1-  
vip.mycluster.example.com/255.255.252.0/eth0
```

関連トピック

- [Oracle Real Application Clusters管理およびデプロイメント・ガイド](#)

クラスタ内のネットワーク・アドレス構成

IPv4、IPv6または指定されたネットワーク上での両タイプのアドレスのいずれかにネットワーク・インタフェースを構成できます。

サード・パーティーの技術を使用して冗長なネットワーク・インタフェースを構成する場合、Oracleでは、1つのインタフェースがIPv4アドレスをサポートし、別のインタフェースがIPv6アドレスをサポートするような構成はサポートしていません。冗長なインタフェースのネットワーク・インタフェースは、同じIPアドレス・タイプを使用して構成する必要があります。Oracle Clusterwareの冗長インターコネクト機能を使用する場合は、インタフェースにIPv4アドレスを使用する必要があります。

クラスタ内のすべてのノードには、同じIPプロトコル構成を使用する必要があります。すべてのノードがIPv4を使用するか、すべてのノードがIPv6を使用するか、すべてのノードがIPv4およびIPv6の両方を使用するかのいずれかです。クラスタ内の一部のノード

ドがIPv6アドレスのみをサポートするように構成し、その他のノードがIPv4アドレスのみをサポートするように構成することはできません。

ローカル・リスナーは、ネットワーク・リソースに構成されたサブネットのアドレス・タイプに基づくエンドポイントでリスニングします。IPv4、IPv6または両方のタイプを使用できます。

SRVCTLを使用した、静的IPv4アドレスの静的IPv6アドレスへの変更

IPv4静的アドレスからIPv6静的アドレスに変更する場合は、静的IPv6アドレスのみを使用するように切り替える前に、IPv6アドレスを追加し、IPv4アドレスとIPv6アドレスの両方を短期間受け入れるようにネットワークを変更します。

ノート:



IPv4 ネットワークが、静的および動的アドレスの両方が混合したモードである場合、この手順は実行できません。最初にすべてのアドレスを静的に移行する必要があります。

静的IPv4アドレスを静的IPv6アドレスに変更するには:

1. ネットワーク全体に対して次のコマンドをrootとして一度使用してIPv6サブネットを追加します。

```
# srvctl modify network -subnet ipv6_subnet/prefix_length
```

前述の構文のipv6_subnet/prefix_lengthは、変更後のIPv6アドレスのサブネットと、接頭辞の長さです(3001::/64など)。

2. 各ノードで、次のコマンドをrootとして使用してIPv6 VIPを一度追加します。

```
# srvctl modify vip -node node_name -netnum network_number -address vip_name/netmask
```

前の構文では:

- node_nameはノードの名前です。
- network_numberは、ネットワークの番号です。
- vip_name/netmaskは、IPv4アドレスとIPv6アドレスの両方に解決されるローカルVIPの名前です
VIP名の後に続くIPv4ネットマスクまたはIPv6の接頭辞の長さは、次の2つの要件を満たす必要があります。
 - ネットマスクをIPv4フォーマット(255.255.255.0など)で指定すると、VIP名はIPv4アドレスに解決されます(IPv6アドレスに解決することもできます)。同様に、IPv6の接頭辞の長さを指定すると(64など)、VIP名はIPv6アドレスに解決されます(IPv4アドレスに解決することもできます)。
 - IPv4ネットマスクを指定する場合は、ネットワークの-iptypeがIPv6であるかどうかに関係なく、登録されているIPv4ネットワーク・サブネット番号のネットマスクに一致する必要があります。同様に、IPv6の接頭辞の長さを指定する場合は、ネットワークの-iptypeがIPv4であるかどうかに関係なく、登録されているIPv6ネットワーク・サブネット番号の接頭辞の長さに一致する必要があります。

3. 次のコマンドを使用してIPv6ネットワーク・リソースをOCRに追加します。

```
$ oifcfg setif -global if_name/subnet:public
```

4. IPv4アドレスと同じ数のIPv6アドレスを持つようDNSのSCANを更新します。ネットワーク全体に対して次のコマンドを

rootとして一度使用して、IPv6アドレスをSCAN VIPに追加します。

```
# srvctl modify scan -scanname scan_name
```

scan_nameは、IPv4アドレスとIPv6アドレスの両方に解決するSCANの名前です。

5. ネットワーク全体に対して次のコマンドをrootとして一度使用して、ネットワークIPタイプをIPv4から、IPv4とIPv6の両方に変換します。

```
srvctl modify network -netnum network_number -iptype both
```

このコマンドにより、IPv6静的アドレスが決定します。

6. クラスタが提供するすべてのクライアントをIPv4のネットワークおよびアドレスからIPv6のネットワークおよびアドレスに変更します。
7. 次のコマンドを使用して、ネットワークの使用プロトコルを、両方のプロトコルからIPv6のみに移行します。

```
# srvctl modify network -iptype ipv6
```

8. 次のコマンドをrootとして実行して、IPv6に解決するVIP名でVIPを変更します。

```
# srvctl modify vip -node node_name -address vip_name -netnum network_number
```

これを各ノードに一度実行します。

9. 次のコマンドを実行して、IPv6に解決するSCAN名でSCANを変更します。

```
# srvctl modify scan -scanname scan_name
```

これをクラスタ全体に一度実行します。

関連トピック

- [Oracle Real Application Clusters管理およびデプロイメント・ガイド](#)
- [OIFCFGコマンドの書式](#)

SRVCTLを使用した、動的IPv4アドレスの動的IPv6アドレスへの変更

SRVCTLコマンドによって、動的IPv4アドレスを動的IPv6アドレスに変更します。

ノート:



IPv4 ネットワークが、静的および動的アドレスの両方が混合したモードである場合、この手順は実行できません。最初にすべてのアドレスを動的に移行する必要があります。

動的IPv4アドレスを動的IPv6アドレスに変更するには:

1. `srvctl modify network`コマンドを使用してIPv6サブネットを追加します。

IPv6サブネットを追加するには、rootとしてログインし、次のコマンド構文を使用します。

```
# srvctl modify network -netnum network_number -subnet ipv6_subnet/  
ipv6_prefix_length[/interface] -nettype autoconfig
```

前の構文では:

- network_numberは、ネットワークの番号です。

- `ipv6_subnet`は、変更先のIPv6アドレスのサブネットです(たとえば、`2001:db8:122:344:c0:2:2100::`)
- `ipv6_prefix_length`は、IPv6ネットワーク・アドレスを指定する接頭辞です(たとえば、`64`)

たとえば、次のコマンドは、IPv6サブネット(`2001:db8:122:344:c0:2:2100::`)および接頭辞長(`64`)を追加することで、ネットワーク3を変更します。

```
# srvctl modify network -netnum 3 -subnet 2001:db8:122:344:c0:2:2100::/64
-nettype autoconfig
```

2. 次のコマンドを使用してIPv6ネットワーク・リソースをOCRに追加します。

```
$ oifcfg setif -global if_name/subnet:public
```

3. 次のように、IPv6動的アドレスを開始します。

```
# srvctl modify network -netnum network_number -iptype both
```

たとえば、ネットワーク番号が3の場合は次のようになります。

```
# srvctl modify network -netnum 3 -iptype both
```

4. クラスタが提供するすべてのクライアントをIPv4のネットワークおよびアドレスからIPv6のネットワークおよびアドレスに変更します。

この時点で、GNSに委譲されたドメイン`scan_name.gns_domain`のSCANは、3つのIPv4と3つのIPv6アドレスに解決されます。

5. 次のコマンドを使用して、クラスタの動的アドレスのIPv4の部分を無効にします。

```
# srvctl modify network -iptype ipv6
```

前述のコマンドを実行した後は、SCAN (`scan_name.gns_domain`)は、3つのIPv6アドレスのみに解決されます。

関連トピック

- [OIFCFGコマンドの書式](#)

IPv4ネットワークのIPv4およびIPv6ネットワークへの変更

IPv6ネットワークを既存のIPv4ネットワークに追加して、IPv4ネットワークをIPv4およびIPv6ネットワークに変更できます。

このプロセスは、[SRVCTLを使用した、静的IPv4アドレスの静的IPv6アドレスへの変更](#)のステップ1から5に説明されています。

これらのステップを完了した後、グリッド・ユーザーとしてログインし、次のコマンドを実行します。

```
$ srvctl status scan
```

出力を確認して、SCAN VIPの変更を確認します。

SRVCTLを使用した、VIPアドレスのIPv4からIPv6ネットワークへの移行

SRVCTLコマンドを使用して、結合されたIPv4およびIPv6ネットワークからIPv4アドレス・タイプを削除します。

次のコマンドを入力します。

```
# srvctl modify network -iptype ipv6
```

このコマンドにより、クラスタに構成されたIPv4アドレスの削除プロセスが開始されます。

クラスタ間依存性プロキシ

クラスタ間依存性プロキシは、ドメイン・サービス・クラスタでリソースを稼働させるための、メンバ・クラスタ上の軽量の耐障害性の高いプロキシ・リソースです。

メンバー・クラスタにより、インフラストラクチャ・リソース使用のオーバーヘッドが削減されるため、ドメイン・サービス・クラスタでは、Oracle Automatic Storage Management(Oracle ASM)などの共有インフラストラクチャ・リソースの状態を効率的に監視できるようにすることが重要です。これにより、メンバー・クラスタ上のリソースは、その状態を適切に調整できます。

クラスタ間依存性プロキシは、ドメイン・サービス・クラスタ・リソースにこの機能を提供します。具体的およびより一般的に言うと、一方のクラスタで稼働しているリソースの状態が、他方のクラスタに反映されます。ドメイン・サービス・クラスタでは、クラスタ間依存性プロキシがデフォルトで構成されます。

3 ポリシーベースでのクラスタおよび容量の管理

Oracle Clusterwareは、Oracle Databaseまたはアプリケーションで使用されるサーバーおよびリソースのポリシーベース管理を使用します。

この章の内容は次のとおりです。

- [サーバー・プールおよびポリシー・ベース管理の概要](#)
- [デフォルト・サーバー・プール](#)
- [サーバーのカテゴリ化の概要](#)
- [クラスタ構成ポリシーおよびポリシー・セットの概要](#)
- [サーバーの重みベースのノード削除](#)
- [ロード対応のリソース配置](#)
- [サーバー構成およびサーバー状態の属性](#)
- [サーバー・カテゴリ属性](#)
- [ポリシー・セット構成の例](#)

サーバー・プールおよびポリシー・ベース管理の概要

Oracle Clusterwareによって管理されるリソースは、サーバー・プールというサーバーの論理グループに含まれます。これは、Oracle Clusterware 11gリリース2 (11.2)で導入されました。

[リソース](#)は、共有インフラストラクチャでホスティングされ、[サーバー・プール](#)内に含まれます。Oracle Clusterwareが管理するリソースの例は、データベース・インスタンス、データベース・サービス、アプリケーションVIPおよびアプリケーション・コンポーネントです。

Oracle Clusterでは、サーバー・プールを使用してクラスタ・メンバー・ノードで特定のタイプのワークロードを実行できるようになり、その一方で管理オプションが簡素化されています。[クラスタ構成ポリシー・セット](#)を使用して、クラスタ全体にわたってクラスタ・ポリシーを動的に管理できます。

Oracle Clusterware 11g リリース2 (11.2)のサーバー・プール・モデルを使用して、Oracle Clusterwareの標準クラスタでリソースの管理を継続することも、サーバー・プールを使用しない従来どおりの方法でリソースを手動で管理することもできます。

この項には次のトピックが含まれます：

- [サーバー・プールおよびサーバーのカテゴリ化](#)
- [サーバー・プールおよびポリシーベース管理](#)
- [サーバー・プールの動作](#)
- [デフォルト・サーバー・プール](#)
- [サーバー・プール属性](#)
- [サーバー・プールを使用した、Oracle Clusterwareでの新しいサーバーの割当て方法](#)
- [デフォルト属性を使用したサーバー・プールの管理](#)

サーバー・プールおよびサーバーのカテゴリ化

管理者は、特定の属性で区別されるサーバーを識別することによって([サーバーのカテゴリ化](#)というプロセス)、サーバー・プールを使用してサーバーを動的にデプロイおよび管理できます。このようにして、異機種環境のノードで構成されるクラスタを作成できます。

関連トピック

- [サーバーのカテゴリ化の概要](#)

サーバー・プールおよびポリシーベース管理

管理者は、ポリシーベース管理で、サーバーを実行するサーバー・プールを指定します(汎用または空きプールを除く)。

たとえば、データベース管理者はSRVCTLを使用して、データベースまたはデータベース・サービスをホスティングしているサーバーのサーバー・プールを作成します。クラスタウェア管理者はCRSCTLを使用して、アプリケーションをホストするサーバーのサーバー・プールの作成など、データベース以外の用途のサーバー・プールを作成します。

ポリシーベース管理:

- 定義済のポリシーに基づいたオンライン・サーバーの再割当てを有効にして、ワークロード容量の要件を満たします。
- ポリシーの定義どおりに、重要な作業に必要なリソースの割当てが保証されます。
- 必要時には分離が保証され、アプリケーションとデータベースについて、クラスタの専用サーバーを指定できます。
- ビジネス・ニーズまたはアプリケーションの要求に応じてプールが変更されるようにポリシーを構成して、適切なときに必要な容量がプールから提供されるようになります。

サーバー・プールはリソースを分離して、1つのサーバー・プールで実行されているアプリケーションが、別のサーバー・プールで実行されているリソースにアクセスできないようにします。Oracle Clusterwareでは、サーバー・プール間のロール区分を詳細に設定できます。この機能では、組織で別々のグループによって管理される環境をクラスタ化する場合に、これらのグループ間で必要となる管理ロール区分が維持されます。

Oracle Clusterwareは、クラスタ内のサーバーを効率的に割り当てます。サーバー・プールの作成時に定義されるサーバー・プール属性は、サーバーの配置および優先順位付けを、IMPORTANCEサーバー・プール属性に基づいて決定します。

関連トピック

- [Oracle Clusterwareのリソース・リファレンス](#)
- [クラスタ構成ポリシーおよびポリシー・セットの概要](#)

サーバー・プールの動作

サーバー・プールは、クラスタをシングルトン・アプリケーションおよび共通アプリケーションの両方をホスティングしているサーバーの論理グループに分割します。アプリケーションは、データベース・サービスまたはデータベース以外のアプリケーションに分類できます。アプリケーション・ワークロードをサーバー・プールのすべてのサーバーに分散させる場合、アプリケーションは共通です。アプリケーションがサーバー・プール内の1つのサーバーで実行される場合、アプリケーションはシングルトンです。Oracle Clusterwareのロール別管理では、サーバー・プールへのアクセスおよびサーバー・プールの使用が決定されます。

Oracle RACデータベースを含むサーバー・プールは、サーバー制御(SRVCTL)ユーティリティを使用して管理します。他のすべてのサーバー・プールを管理するには、Oracle Clusterware制御(CRSCTL)ユーティリティを使用します。最上位のサーバー・プールを作成する権限を所有しているのは、クラスタ管理者のみです。

データベース管理者は、サーバー制御(SRVCTL)ユーティリティを使用して、Oracle RACデータベースを含むサーバー・プールを作成して管理します。クラスタ管理者は、Oracle Clusterware制御(CRSCTL)ユーティリティを使用して、データベース以外のアプリケーションのサーバー・プールなど、他のサーバー・プールすべてを作成して管理します。最上位のサーバー・プールを作成する権限を所有しているのは、クラスタ管理者のみです。

最上位のサーバー・プールの特徴は、次のとおりです。

- クラスタを論理的に分割します。
- 常に排他的です。これは、1つのサーバーが特定の時期に1つの特定のサーバー・プールにのみ存在できることを意味します。

デフォルト・サーバー・プール

Oracle Clusterwareがインストールされると、汎用および空きという2つの内部サーバー・プールが自動的に作成されます。新規インストールのすべてのサーバーは、最初、空きサーバー・プールに割り当てられます。空きサーバー・プールにあるサーバーは、新しく定義したサーバー・プールに自動的に移動します。

空きサーバー・プール

空きサーバー・プールには、他のサーバー・プールに割り当てられていないサーバーが含まれています。空きサーバー・プールの属性は、次のように制限されます。

- SERVER_NAMES、MIN_SIZEおよびMAX_SIZEは、ユーザーが編集することはできません。
- IMPORTANCEおよびACLは、ユーザーが編集することができます。

汎用サーバー・プール

汎用サーバー・プールには、最上位のサーバー・プールには含まれておらず、ポリシー管理されていないサーバーが格納されます。ポリシー管理されていないアプリケーション(管理者管理データベースなど)をホスティングするサーバーは、汎用サーバー・プールに静的に割り当てられます。

汎用サーバー・プールの属性は、次のように制限されています。

- 汎用サーバー・プールの構成属性は誰も変更できません(すべての属性は読み取り専用です)。
- 管理者管理データベースは、データベースの作成先のサーバーが次のいずれかである場合にのみ、汎用サーバー・プールに作成できます。
 - オンラインで汎用サーバー・プールに存在する。
 - オンラインで空きサーバー・プールに存在する。この場合、Oracle Clusterwareによってサーバーが汎用サーバー・プールに移動されます。
 - オンラインで、他のサーバー・プールに存在し、ユーザーがクラスタ管理者であるか、またはサーバー・プールのサーバーの使用が許可されている場合(この場合、サーバーは汎用サーバー・プールに移動されます)。
 - オフラインで、ユーザーがクラスタ管理者の場合

サーバー・プール属性

SRVCTLまたはCRSCTLを使用して、データベースおよびその他のアプリケーションのそれぞれにサーバー・プールを作成できます。SRVCTLを使用してサーバー・プールを作成する場合、この項で説明されているサーバー・プール属性のサブセットのみを使用で

きます。CRSCTLを使用してサーバー・プールを作成する場合、一連のサーバー・プール属性すべてを使用できます。

サーバー・プール属性は、サーバー・プールを作成および管理するために定義する属性です。

使用するユーティリティは、サーバー・プールでホスティングされているリソースのタイプによって決まります。Oracle Databaseをホスティングするサーバー・プールを作成するには、SRVCTLを使用する必要があります。中間層および中間アプリケーションなどのデータベース以外のリソースをホスティングするサーバー・プールを作成するには、CRSCTLを使用する必要があります。

SRVCTLを使用してサーバー・プールを作成するときに利用できるサーバー・プール属性は、次のとおりです。

- -category
- -importance
- -min
- -max
- -serverpool
- -servers

SRVCTLは、サーバー・プールの名前にora.を付加します。

CRSCTLを使用してサーバー・プールを作成する場合、次の表のリストで説明されているすべての属性を利用できます。

表3-1 サーバー・プール属性

属性	説明
ACL	次の書式の文字列を使用します。 <pre>owner:user:rwx,pgrp:group:rwx,other::r-</pre> サーバー・プールの所有者、および様々なオペレーティング・システムのユーザーとグループに付与する権限を定義します。サーバー・プールの所有者には、所有者になるオペレーティング・システム・ユーザーとそのユーザーに付与する権限を定義します。 このオプション属性の値は、明示的に上書きされないかぎり、サーバー・プールを作成するユーザーの ACL に基づいてサーバー・プールの作成時に移入されます。その後、サーバー・プールの既存の権限に基づいて変更を行うことができる場合、この値は変更できます。 文字列は次のように指定します。 ● owner: サーバー・プール所有者になるオペレーティング・システム・ユーザーです。続いてその所有者の権限を指定します。 ● pgrp: サーバー・プール所有者のプライマリ・グループであるオペレーティング・システム・グループです。続いてプライマリ・グループのメンバーの権限を指定します。 ● other: 続いて他のメンバーの権限を指定します。 ● r: 読取り専用です

属性	説明
	● w: プールの属性を変更または削除します。 ● x: このプールにリソースを割り当てます。 デフォルトでは、サーバー・プールを作成するクライアントの ID は owner です。また、デフォルトでは、root、および owner に指定された user は完全な権限を持ちます。ACL 属性に次の行を追加することで、必要なオペレーティング・システム・ユーザーおよびオペレーティング・システム・グループに権限を付与できます。 <pre>user:user_name:rwxgroup:group_name:rwx</pre> サーバー・プールを作成するオペレーティング・システム・ユーザーはサーバー・プールの所有者であり、このサーバー・プールの ACL 属性には、そのユーザー、プライマリ・グループ、グループなどの UNIX 系の読取り、書込みおよび実行 ACL 定義が反映されます。
ACTIVE_SERVERS	サーバー名の文字列には次の書式を使用します。 <pre>server_name1 server_name2 ...</pre> Oracle Clusterware は、この属性を自動的に管理します。この属性には、現在サーバー・プールに割り当てられているサーバーの空白区切りのリストが含まれています。
EXCLUSIVE_POOLS	このオプション属性は、このサーバー・プールに割り当てられたサーバーが他のサーバー・プールと共有されるかどうかを示します。サーバー・プールは、この属性に同じ値を持つその他のサーバー・プールに対して相互に排他的であることを明示的に示すことができます。サーバー・プールに割り当てられたサーバーのセットに共通の単一サーバーがない場合、2 つ以上のサーバー・プールは相互に排他的です。たとえば、サーバー・プール A および B の両方でこの属性の値が pools_A_B などの同じ文字列に設定されている場合、サーバー・プール A および B は相互に排他的となります。 最上位のサーバー・プールは、デフォルトでは、相互に排他的です。
IMPORTANCE	サーバー・プールの相対的な重要度で、0 から 1000 までの整数で表されます。0 が最低レベルの重要度、1000 が最高レベルの重要度です。このオプション属性は、ノードがクラスタに追加されるか、またはクラスタから削除された場合にサーバー・プールを再構成する方法を判別するために使用されます。デフォルトの値は 0。
MIN_SIZE	サーバー・プールの最小サイズで、負でない整数で表されます。サーバー・プールに含まれるサーバーの数がこの属性で指定する数を下回る場合、Oracle Clusterware は、この数が満たされるまで他のサーバー・プールからこのサーバー・プールにサーバーを自動的に移動します。

属性	説明
	ノート: このオプション属性の値によって、ハード制限は設定されません。クラスタが再構成されるたびにサーバー割当ての優先順位が制御されます。デフォルト値は 0 です。
MAX_SIZE	サーバー・プールに含めることができるサーバーの最大数で、負でない整数で表されます。この属性はオプションで、デフォルトでは、-1(制限なし)に設定されています。 ノート: この属性の -1 という値は、クラスタ全体に適用されます。
NAME	サーバー・プールの名前で、これはサーバー・プールの作成時に指定する必要があります。サーバー・プール名は、ユーザーが作成したエンティティ(リソース、タイプ、サーバーなど)の名前のドメイン内で一意である必要があります。サーバー・プール名には、254 文字の制限があり、プラットフォームでサポートされているすべての文字(感嘆符(!)、チルダ(~)および空白を除く)を使用できます。サーバー・プール名の先頭をピリオドまたは ora にすることはできません。 ノート: SRVCTL を使用してサーバー・プールを作成すると、ユーティリティによってサーバー・プールの名前の先頭に「ora.」が追加されます。
PARENT_POOLS	この属性を使用して、ネストされたサーバー・プールを作成することができます。この属性内に空白区切りのサーバー・プール名の文字列として示されたサーバー・プールは、親サーバー・プールと呼ばれます。親サーバー・プールに含まれているサーバー・プールは、子サーバー・プールと呼ばれます。 ノート: SRVCTL を使用してサーバー・プールを作成する場合、この属性は指定できません。
SERVER_CATEGORY	登録されているサーバー・カテゴリの名前で、サーバーのカテゴリ化の一部として使用されます。Oracle Clusterware の標準クラスタおよび Oracle Flex Cluster には、デフォルトで hub というカテゴリがあります。サーバー・プールの作成時に SERVER_CATEGORY に値を設定すると、SERVER_NAMES には値を設定できません。これらのパラメータの 1 つにのみ、空でない値を設定できます。 サーバー・プールに割り当てられるサーバーを、サーバー属性に基づいて分類するには、SERVER_CATEGORY 属性を使用します。特定のワークロードをサーバーおよびサーバー・プールと釣り合わせるために、定義したサーバー属性に基づいてクラスタ内のサーバーおよびサーバー・プールを編成できます。
SERVER_NAMES	サーバー・プールとの関連付けが可能な候補ノード名のリストで、空白区切りのサーバー名の文字列として表されます。このオプションの属性に対して一連のサーバー名を指定しない場合、PARENT_POOLS などの他の属性の値で許容される範囲で任意のサーバが任意のサーバー・プールに割り当てられるように、Oracle Clusterware は構成されま

属性	説明
	す。 候補ノード名として識別されるサーバー名は、現在アクティブなクラスタ・メンバーであることを確認する検証が行われません。この属性を使用して、クラスタにまだ追加されていない候補としてサーバーを定義します。 SERVER_NAMES の値を設定する場合、SERVER_CATEGORY の値は設定できません。これらの属性の 1 つのみに、空でない値を指定できます。 ノート: SERVER_CATEGORY 属性を設定する場合、および個々のサーバーを指定する必要がある場合、EXPRESSION サーバー・カテゴリ属性を使用して名前でサーバーをリストできます。

関連トピック

- [crsctl add serverpool](#)
- [crsctl status server](#)
- [サーバー・カテゴリ属性](#)
- [Oracle Real Application Clusters管理およびデプロイメント・ガイド](#)

サーバー・プールを使用した、Oracle Clusterwareでの新しいサーバーの割当て方法

Oracle Clusterwareは、次の順序でサーバー・プールに新しいサーバーを割り当てます。

1. 汎用サーバー・プール
2. ユーザー作成のサーバー・プール
3. 空きサーバー・プール

次の条件が満たされるまで、Oracle Clusterwareはサーバー・プールへのサーバーの割当てを続行します。

1. 重要度順にすべてのサーバー・プールが最小値まで一杯になるまで(MIN_SIZE)。
2. 重要度順にすべてのサーバー・プールが最大値まで一杯になるまで(MAX_SIZE)。
3. デフォルトでは、サーバー・プールに配置されていないサーバーは、空きサーバー・プールに割り当てられます。

空きサーバー・プールのIMPORTANCE属性は変更できます。空きサーバー・プールのIMPORTANCE属性の値が、その他の1つ以上のサーバー・プールよりも大きく、これらのMIN_SIZE属性の値が満たされた時点で、残りのサーバーは空きサーバー・プールに割り当てられます。

サーバーをクラスタに追加する場合、いくつかの問題が発生します。

[表3-2](#)で構成されているサーバー・プールについて考えてみます。

表3-2 サーバー・プール属性の構成例

NAME	IMPORTANCE	MIN_SIZE	MAX_SIZE	PARENT_POOLS	EXCLUSIVE_POOLS

NAME	IMPORTANCE	MIN_SIZE	MAX_SIZE	PARENT_POOLS	EXCLUSIVE_POOLS
sp1	1	1	10		
sp2	3	1	6		
sp3	2	1	2		
sp2_1	2	1	5	sp2	S123
sp2_2	1	1	5	sp2	S123

たとえば、クラスタ内にサーバーがなく、すべてのサーバー・プールが空であるとします。

server1というサーバーをクラスタに追加するには、次の手順を実行します。

1. サーバーからプールへの割当てを開始します。
2. Oracle Clusterwareは、最上位のサーバー・プール(親サーバー・プールがないサーバー・プール)のみを最初に処理します。この例では、最上位のサーバー・プールはsp1、sp2およびsp3です。
3. Oracle Clusterwareは、IMPORTANCEの順位に従ってsp2、sp3、sp1のようにサーバー・プールを表示します。
4. sp2のIMPORTANCEが最大値で、MIN_SIZE値は満たされていないため、Oracle Clusterwareはserver1をsp2に割り当てます。
5. Oracle Clusterwareは、残り2つのサーバー・プール(sp2_1およびsp2_2)を処理します。両方のサーバー・プールのサイズはMIN_SIZE属性の値(サーバー・プールは両方とも空で、MIN_SIZE値は1)を下回っています。
6. Oracle Clusterwareは、IMPORTANCEの順位に従ってsp2_1、sp2_2のように残り2つのサーバー・プールを表示します。
7. Oracle Clusterwareは、server1をsp2_1に割り当てますが、sp2_1はsp2_2に対して排他的であるように構成されているため、server1をsp2_2に割り当てることはできません。

処理後、クラスタ構成は次のように表示されます。

表3-3 処理後のサーバー・プール構成

サーバー・プール名	割り当てられたサーバー
sp1	
sp2	server1
sp3	
sp2_1	server1

サーバー・プールからサーバー・プールへのサーバーの移動

サーバー・プール内のサーバーの数がサーバー・プールのMIN_SIZE属性の値を下回る場合(サーバーで障害が発生した場合など)、Oracle Clusterwareは、すべてのサーバー・プールのMIN_SIZEおよびIMPORTANCE属性に設定した値に基づいて、他のサーバー・プールのサーバーをサーバーの数がMIN_SIZE値を下回るサーバー・プールに移動できます。Oracle Clusterwareは、他のサーバー・プールからサーバーを選択して、次の基準に該当する不十分なサーバー・プールに移動します。

- 不十分なサーバー・プールよりIMPORTANCEの値が低いサーバー・プールに対して、Oracle Clusterwareは、サーバーの数がMIN_SIZE属性の値を下回る場合でもそのサーバー・プールからサーバーを取得します。
- IMPORTANCEの値が同一以上のサーバー・プールに対して、Oracle Clusterwareは、サーバー・プール内のサーバーの数がMIN_SIZE属性の値より大きい場合にのみそのサーバー・プールからサーバーを取得します。

デフォルト属性を使用したサーバー・プールの管理

デフォルトでは各サーバー・プールは、サーバー・プールを管理するための次の属性オプションを使用して構成されます。

- MIN_SIZE: サーバー・プールに含まれるサーバーの最小数。
サーバー・プールのサーバーの数がこの属性の値を下回る場合、サーバーの数が属性の値になるまで、Oracle Clusterwareによって自動的に他の場所からサーバー・プールにサーバーが移動されます。
- MAX_SIZE: サーバー・プールに含まれるサーバーの最大数。
- IMPORTANCE: クラスタ内の他のすべてのサーバー・プール間でサーバー・プールをランク付けする0から1000(0の重要度が最低)の数値。

また、クラスタ構成ポリシーの一部として、サーバー・プールをより詳細に管理するために追加の属性を割り当てることもできます。EXCLUSIVE_POOLSやSERVER_CATEGORYなどの属性は、パフォーマンスを向上させるサーバー・プール用のポリシーを作成し、チューニング設計管理をサーバー・プールに組み込む場合に役立ちます。

サーバーのカテゴリ化の概要

サーバーのカテゴリ化では、プロセッサ・タイプ、メモリー、他の顕著なシステム機能などの属性を使用して、サーバーを特定のカテゴリに分類できます。

プールの適格なメンバーを、特定の一連の属性を共有するサーバーのカテゴリに制限するようにサーバー・プールを構成できます。元々、サーバー・プールは、特定のプールに属するようにサーバーを特徴付ける一連の基本属性に制限されており、サーバーのタイプを区別する方法はなく、すべてのサーバーは、プロセッサ、物理メモリーおよび他の特性に関して等しいとみなされていました。サーバーのカテゴリ化によって、これらのサーバーを区別する方法が提供されるようになりました。

ノート:



Oracle Database Quality of Service Management (Oracle Database QoS Management)でポリシーを作成する場合、サーバー・プールのディレクティブ・オーバーライドを設定することでサーバーを分類し、policy および policyset という語を使用する CRSCTL コマンドは無効になります。また、Oracle Clusterware ポリ

シーの使用から Oracle Database QoS Management ポリシーの使用に切り替える場合、Oracle Clusterware ポリシーは置き換えられます。2 つのポリシー・タイプは共存できないためです。ポリシーを切り替える前に、`crsctl status policyset -file file_name` を使用してバックアップを作成することをお勧めします。

関連トピック

- [Oracle Clusterware制御\(CRSCTL\)ユーティリティ・リファレンス](#)

クラスタ構成ポリシーおよびポリシー・セットの概要

クラスタ構成ポリシーとは、クラスタ構成ポリシー・セットが管理する各サーバー・プールに1つのみの定義を含むドキュメントです。クラスタ構成ポリシー・セットとは、クラスタ用に構成されているすべてのサーバー・プールの名前と、すべてのポリシーの定義を定義しているドキュメントです。

Oracle Clusterware 12c以降のリリースでは、サーバー・プールの指定および管理用の[クラスタ構成ポリシー・セット](#)で定義されているポリシーを使用し、Oracle Clusterwareは、ポリシー・セットのポリシーに従ってサーバー・プールを管理します。クラスタ構成ポリシー・セットの場合、たとえば、電子メールの要求に対応するため、週の業務時間中により多くのサーバーをOLTPワークロードに割り当てることができ、週末および夜間により多くのサーバーをバッチ・ワークロードに割り当てることができ、サーバー・プール構成またはサーバー割当てのトランジションをアトミックに実行できます。

いつの時点においても、クラスタに対して有効なポリシーは1つのみです。ただし、いくつかの異なるポリシーを作成できます。このため、ビジネスのニーズや要求に基づいて、または暦日や時刻に基づいてクラスタの要件に違いを反映させるように、パラメータを使用してサーバーのプールを構成できます。

関連トピック

- [ポリシー・セット構成の例](#)

ロード対応のリソース配置

Oracle ClusterwareがデータベースのCPU要件および制限を認識するようにそのデータベースを構成できます。

Oracle Clusterwareでは、この情報を使用して、十分なCPU数またはメモリー量(あるいはその両方)があるサーバーにのみデータベース・リソースを配置します。

Oracle ClusterwareでリソースをCPU要件またはメモリー要件とともに構成した場合、Oracle Clusterwareは、これらの要件を満たすサーバーでのみこれらのリソースの起動を試行します。特にデータベース・リソースの場合は、CPUの値をCPU_COUNTインスタンス・パラメータに、メモリーの値をMEMORY_TARGETインスタンス・パラメータ内に構成できます。

ノート:

インスタンス・パラメータを構成する場合、次のことが必要です。



- CPU (インスタンス・ケーシング)について、リソース・マネージャが有効になっている必要があります。
- メモリーについては、データベースに自動メモリー管理が使用されている必要があります。

データベース・インスタンスを追加または変更するとき、次のようにして、ロード対応のリソース配置を構成できます。

```
$ srvctl modify database -db db_unique_name -cpucount cpu_count -memorytarget memory_target
```

前述の例において、cpu_countはワークロードCPUの数を表し、memory_targetは、リソースにより使用されるターゲット・メモリー(MB)を表します。

データベースでリソース・マネージャが有効になっている場合、Oracle ClusterwareはCPU_COUNTパラメータを-cpucountの値に設定します。また、データベースで自動メモリー管理が有効になっている場合、Oracle ClusterwareはMEMORY_TARGETデータベース・パラメータを-memorytargetの値に設定します。

関連トピック

- [Oracle Databaseリファレンス](#)
- [Oracle Databaseリファレンス](#)
- [Oracle Database管理者ガイド](#)

サーバー構成およびサーバー状態の属性

サーバーをクラスタに追加するとすぐに、Oracle Clusterwareは、各サーバーに属性セットを割り当てます。

サーバーの物理特性を記述する属性もあれば、サーバーの状態を記述する属性もあります。また、サーバーをさらに分類するために役立つ変更可能な他のサーバー属性もあります。クラスタからサーバーを削除すると、Oracle Clusterwareはサーバー・オブジェクトを削除します。

サーバーのカテゴリ化管理ポリシーの一部として、サーバー構成属性を使用してサーバー进行分类します。

次の表に、サーバー構成属性を示します。

表3-4 サーバー構成属性

属性	説明
ACTIVE_CSS_ROLE	サーバーによって実行されるロール。サーバーに hub のロールがある場合もあります。これは、Oracle Flex Cluster 内のサーバーに指定されるロール、または Oracle Clusterware の標準クラスタ内のノードに指定されるロールです。 ノート: この属性は構成できません。
CONFIGURED_CSS_ROLE	サーバー用に構成されたロール。サーバーに hub のロールがある場合もあります。これは、Oracle Flex Cluster 内のサーバーに指定されるロール、または Oracle Clusterware の標準クラスタ内のノードに指定されるロールです。 ノート: この属性は構成できません。
CPU_CLOCK_RATE	メガヘルツ(MHz)単位の CPU クロック速度。
CPU_COUNT	プロセッサ数。
CPU_EQUIVALENCY	サーバーで提供される CPU パワーが、ベースライン(1000 など)による物理的表現から正または負の方向に逸脱する可能性があることを説明するために、Oracle Clusterware

属性	説明
	が使用する相対値(1 以上の正の整数として表される)。値が 1000 より小さい場合は、CPU_COUNT および CPU_CLOCK_RATE パラメータの特定の値にもかかわらず、このサーバーで提供され相当パワーがそれぞれ小さいことを説明します。 ローカル・サーバーで次のコマンドを使用して、この属性をそれぞれ表示または変更します。 <pre>crsctl get cpu equivalency crsctl set cpu equivalency</pre>
CPU_HYPERTHREADING	CPU のハイパー・スレッドのステータス。値 0 は、ハイパー・スレッドが使用されていないことを示しています。値 1 は、ハイパー・スレッドが使用されていることを示しています。
MEMORY_SIZE	メガバイト(MB)単位のメモリー・サイズ。
NAME	サーバーの名前。
RESOURCE_USE_ENABLED	サーバー・プール・リソース管理パラメータ。この属性の値が 1 の場合(デフォルト)、サーバーをリソースの配置に使用できます。値が 0 の場合、Oracle Clusterware によってサーバーでのサーバー・プール・リソースの起動が禁止されます。サーバーが空きプールに残っています。 crsctl get resource use コマンドおよび crsctl set resource use コマンドを使用すると、各クラスタ・メンバー・ノードの設定を見直し、この属性を制御できます。
SERVER_LABEL	サーバーにラベルを付けるために使用できる任意の値。サーバー・カテゴリを設定する場合にこの属性を使用できます。たとえば、場所(building_A、building_B など)を指定することが可能であり、これにより、場所が要件になっているプールにサーバーを配置することができます(適切なサーバー・カテゴリを作成し、それをサーバー・プールに使用します)。 ローカル・サーバーで次のコマンドを使用して、この属性をそれぞれ表示または変更します。 <pre>crsctl get server label crsctl set server label</pre>

次の表に、読取り専用のサーバー状態および構成属性を示します。

表3-5 サーバー状態の属性

属性	説明
ACTIVE_POOLS	サーバーが属しているサーバー・プールの名前の空白区切りのリストです。Oracle Clusterware はこのリストを自動的に管理します。

属性	説明
STATE	サーバーは、次のいずれかの状態になります。 ● ONLINE: サーバーは、クラスタのメンバーであり、リソースの配置に使用できます。 ● OFFLINE: サーバーは、現在クラスタのメンバーではありません。したがって、リソースの配置に使用できません。 ● JOINING: サーバーがクラスタに追加されると、Oracle Clusterware は、サーバーを処理し、リソースの配置に有効であることを確認します。また、Oracle Clusterware は、サーバーで実行されるように構成されているリソースの状態をチェックします。サーバーの妥当性およびリソースの状態が確認されると、サーバーはこの状態から遷移します。 ● LEAVING: サーバーの計画停止が開始すると、サーバーの状態は LEAVING に遷移し、サーバーをリソースの配置に使用できなくなります。 ● VISIBLE: Oracle Clusterware が実行されていて、クラスタ・レディ・サービス・デーモン(CRSD)が実行されてないサーバーは、VISIBLE 状態になります。通常、これは、断続的な問題または障害が発生し、Oracle Clusterware がデーモンのリカバリ(再起動)を試行していることを示します。サーバーがこの状態である間、Oracle Clusterware はサーバーのリソースを管理できません。 ● RECONFIGURING: サーバー・プールの再構成のためにサーバーがサーバー・プール間で移動する場合、現行のサーバー・プールのそのサーバー上で実行されるリソースを停止および再配置する必要があると、サーバーはこの状態になります。これは、サーバーで実行されるリソースは、サーバーの移動先サーバー・プールで実行されるように構成されない可能性があるためです。リソースが正常に再配置されるとすぐに、サーバーは ONLINE 状態に戻されます。 <code>crsctl status server</code> コマンドを使用して、サーバー情報を取得します。
STATE_DETAILS	これは、Oracle Clusterware で管理される読取り専用の属性です。属性ではサーバーの状態に関する追加の詳細が提供されます。サーバーの状態に関する追加の詳細には、次のものが含まれる場合があります。 サーバーの状態: ONLINE: ● AUTOSTARTING RESOURCES (サーバーが再起動する場合または Oracle Clusterware スタックが再起動される場合に実行される)リソース自動起動手順がサーバーに対して進行中である

属性	説明
	ことを示します。 ● AUTOSTART QUEUED サーバーはリソース自動起動が開始するのを待機します。これが実行されると、属性値は AUTOSTARTING RESOURCES に変わります。 サーバーの状態: RECONFIGURING: ● STOPPING RESOURCES 新しいサーバー・プールでの実行が制限されるリソースが停止中です。 ● STARTING RESOURCES 新しいサーバー・プールで実行できるリソースが開始中です。 ● RECONFIG FAILED 1 つ以上のリソースが停止しなかったため、サーバーは ONLINE 状態に遷移できません。この時点で、手動による介入は必須です。停止しなかったリソースを停止または登録解除する必要があります。その後、サーバーは自動的に ONLINE 状態に遷移します。 サーバーの状態: JOINING: ● CHECKING RESOURCES サーバーが再起動、Oracle Clusterware スタックが再起動、またはサーバーの CRSD が再起動するたびに、ポリシー・エンジンではサーバー上のリソースの現在の状態を確認する必要があります。この手順が進行中の間は、この値が戻されます。

データベース・サーバーのメモリー不足の管理

オープン・セッションまたはランナウェイ・ワークロードが多すぎるため、エンタープライズ・データベース・サーバーが使用可能メモリーをすべて使用することがあります。メモリーが不足すると、トランザクションが失敗することがあります。極端な場合には、サーバーが再起動し、アプリケーションの貴重なリソースが失われることもあります。Oracle Database QoS Managementは、サーバー上のメモリー不足をリアルタイムで検出し、新しいセッションを他のサーバーにリダイレクトして、メモリーが不足しているサーバー上のすべての使用可能メモリーが使用されることを防ぎます。

Oracle Database QoS Managementが有効になっていて、Oracle Clusterwareサーバー・プールを管理している場合、Cluster Health Monitorは、クラスタ・サーバーのメモリー・リソースに関するリアルタイム情報を提供するメトリック・ストリームをOracle Database QoS Managementに送信します。この情報の内容は次のとおりです。

- 使用可能メモリーの量

- 現在使用されているメモリーの量

ノードのメモリーが不足しているとOracle Database QoS Managementが判断した場合は、新しい接続が作成されないように、そのノード上のOracle Clusterwareで管理されているデータベース・サービスが停止されます。メモリー不足が緩和されると、そのノードのサービスが自動的に再開し、リスナーはそのサーバーへの新規接続の送信を開始します。メモリー不足は、いくつかの方法で緩和できます(たとえば、既存のセッションの終了やユーザーの介入など)。

新しいセッションを別のサーバーに再ルートすると、メモリーが不足しているサーバー上の既存のワークロードが保護され、サーバーを使用可能な状態に保つことができます。サーバーのメモリー不足の管理によって、Oracle RACデータベースにホストされているアプリケーションのサービス・レベルの管理に新しいリソース保護機能が追加されます。

サーバー・カテゴリ属性

サーバーを名前付きカテゴリに定義し、サーバーをカテゴリのメンバーとして定義する属性を割り当てます。

カテゴリのメンバーを定義するために使用可能で、サーバーの状態を記述する属性もあれば、サーバーの物理特性を記述する属性もあります。独自の特性を作成して、特定カテゴリのメンバーとしてサーバーを定義することもできます。

ノート:



EXPRESSION サーバー・カテゴリ属性に表示されるサーバー属性の値を変更する場合、新しい値を有効にする前に影響を受けるサーバーの Oracle Clusterware 技術スタックを再起動する必要があります。

次の表に、使用可能なサーバー・カテゴリ属性を示します。

表3-6 サーバー・カテゴリ属性

属性	説明
NAME	サーバー・カテゴリの名前で、これはサーバー・カテゴリの作成時に指定する必要があります。サーバー・カテゴリ名は、ユーザーが作成したエンティティ(リソース、タイプ、サーバーなど)の名前のドメイン内で一意である必要があります。サーバー・プール名には、254 文字の制限があり、プラットフォームでサポートされているすべての文字(感嘆符(!)およびチルダ(~)を除く)を使用できます。サーバー・プール名の先頭をピリオドまたは ora にすることはできません。
ACTIVE_CSS_ROLE	サーバーのアクティブ・ロール。これは、Oracle Flex Cluster または Oracle Clusterware 標準クラスタでハブ・ノードになっているサーバーの hub です。いずれの場合も、これがデフォルト値です。
EXPRESSION	サーバーがカテゴリに属するかどうかを判断するために、各サーバーに対して評価できる一連のサーバー属性名、値および状態。 表 3-4 に、サーバー属性を示します。 受け入れ可能な比較演算子は、次のとおりです。 ● =: 等しい ● eqi: 等しい、大/小文字の区別なし

属性	説明
	• >: より大きい • <: より小さい • !=: 等しくない • co: 次を含む • coi: 次を含む、大/小文字の区別なし • st: 次で始まる • en: 次で終わる • nc: 含まない • nci: 含まない、大/小文字の区別なし
	受け入れ可能なブール演算子は、次のとおりです。
	• AND • OR
	ノート: EXPRESSION 文字列で使用する演算子の前後には空白を入力する必要があります。
	たとえば:
	EXPRESSION='((NAME = server1) OR (NAME = server2))'

ポリシー・セット構成の例

この例では、3つの異なるアプリケーションapp1、app2およびapp3によって使用される4ノードのクラスタがあり、3つのサーバー・プールpool1、pool2およびpool3を作成しました。各アプリケーションは専用のサーバー・プールで動作するように割り当て、app1は2つのサーバーを希望し、app2とapp3はそれぞれ1つのサーバーを希望するように、サーバー・プールを構成します。

サーバー・プールの構成は、次のようになります。

```
$ crsctl status serverpool pool1 -p
NAME=pool1
IMPORTANCE=0
MIN_SIZE=2
MAX_SIZE=2
SERVER_NAMES=
PARENT_POOLS=
EXCLUSIVE_POOLS=
ACL=owner:mjk:rw, pgrp:g900:rw, other::r--
SERVER_CATEGORY=
$ crsctl status serverpool pool2 -p
NAME=pool2
IMPORTANCE=0
MIN_SIZE=1
MAX_SIZE=1
SERVER_NAMES=
PARENT_POOLS=
EXCLUSIVE_POOLS=
ACL=owner:mjk:rw, pgrp:g900:rw, other::r--
SERVER_CATEGORY=
$ crsctl status serverpool pool3 -p
```

```
NAME=pool3
IMPORTANCE=0
MIN_SIZE=1
MAX_SIZE=1
SERVER_NAMES=
PARENT_POOLS=
EXCLUSIVE_POOLS=
ACL=owner:mjk:rwX,pgrp:g900:rwX,other::r--
SERVER_CATEGORY=
```

ノート:



前述の例に示す `crsctl status serverpool` コマンドは、CRSCTL を使用してサーバー・プールを作成した場合にのみ機能します。

ただし、この構成では、一部のアプリケーションは日、週または月の様々なときにサーバー時間を必要とする事実を考慮しません。たとえば、電子メール・アプリケーションは通常、業務時間中はより多くのリソースを使用して、夜間および週末は使用するリソースは少なく済みます。

また、app1は、業務時間中は2つのサーバーを必要とし、夜間は1つのみのサーバーを必要とし、週末はサーバーを必要としないと想定します。同時に、app2とapp3はそれぞれ、業務時間中は1つのサーバーを必要とし、夜間はapp2は2つのサーバーを必要とし、app3は1つのサーバーを必要とします。週末は、app2は1つのサーバーを必要とし、app3は3つのサーバーを必要とします。このシナリオは、クラスタに対して構成する必要がある3つの構成を示しています。

1. 昼間:

- app1は2つのサーバーを使用します。
- app2とapp3はそれぞれ1つのサーバーを使用します。

2. 夜間:

- app1は1つのサーバーを使用します。
- app2は2つのサーバーを使用します。
- app3は1つのサーバーを使用します。

3. 週末:

- app1は実行されません(0のサーバー)。
- app2は1つのサーバーを使用します。
- app3は3つのサーバーを使用します。

ポリシー・セットの作成

これらの前提を考えた場合、`crsctl create policyset` コマンドを実行して、Defaultという単一ポリシーを使用してポリシー・セットを作成します。これは、`crsctl status serverpool` コマンドによって表示される構成を反映します。Defaultポリシーを使用すると、この例で想定されたニーズに対応するために他のポリシーを作成できます。`crsctl create policyset` コマンドによって、[例3-1](#)のようなテキスト・ファイルが作成されます。

例3-1 ポリシー・セットのテキスト・ファイル

```
SERVER_POOL_NAMES=Free pool1 pool2 pool3
POLICY
  NAME=Default
  SERVERPOOL
    NAME=pool1
    IMPORTANCE=0
```

```

MAX_SIZE=2
MIN_SIZE=2
SERVER_CATEGORY=
SERVERPOOL
NAME=pool2
IMPORTANCE=0
MAX_SIZE=1
MIN_SIZE=1
SERVER_CATEGORY=
SERVERPOOL
NAME=pool3
IMPORTANCE=0
MAX_SIZE=1
MIN_SIZE=1
SERVER_CATEGORY=

```

ポリシーの変更

この例で想定されているニーズに対応するように前述のポリシー・セットを変更するには、ポリシーの名前をDefaultからDayTimeに変更することによってテキスト・ファイルを編集し、すでに説明した3つのシナリオに対してポリシーを定義します。次に、ポリシーをコピーして2回貼付けを行って、2つのポリシーを作成します。これらには、[例3-2](#)に示すように、NightTimeとWeekendという名前を付けます。

例3-2 変更されたポリシー・セットのテキスト・ファイル

```

SERVER_POOL_NAMES=Free pool1 pool2 pool3
POLICY
NAME=DayTime
SERVERPOOL
NAME=pool1
IMPORTANCE=0
MAX_SIZE=2
MIN_SIZE=2
SERVER_CATEGORY=
SERVERPOOL
NAME=pool2
IMPORTANCE=0
MAX_SIZE=1
MIN_SIZE=1
SERVER_CATEGORY=
SERVERPOOL
NAME=pool3
IMPORTANCE=0
MAX_SIZE=1
MIN_SIZE=1
SERVER_CATEGORY=
POLICY
NAME=NightTime
SERVERPOOL
NAME=pool1
IMPORTANCE=0
MAX_SIZE=1
MIN_SIZE=1
SERVER_CATEGORY=
SERVERPOOL
NAME=pool2
IMPORTANCE=0
MAX_SIZE=2
MIN_SIZE=2
SERVER_CATEGORY=
SERVERPOOL
NAME=pool3
IMPORTANCE=0
MAX_SIZE=1
MIN_SIZE=1

```



```

    SERVER_CATEGORY=
POLICY
    NAME=Weekend
    SERVERPOOL
        NAME=pool1
        IMPORTANCE=0
        MAX_SIZE=0
        MIN_SIZE=0
        SERVER_CATEGORY=
    SERVERPOOL
        NAME=pool2
        IMPORTANCE=0
        MAX_SIZE=1
        MIN_SIZE=1
        SERVER_CATEGORY=
    SERVERPOOL
        NAME=pool3
        IMPORTANCE=0
        MAX_SIZE=3
        MIN_SIZE=3
        SERVER_CATEGORY=

```

個々のポリシーの名前の変更に加えて、各ポリシーの各サーバー・プールのMAX_SIZEポリシー属性とMIN_SIZEポリシー属性も、アプリケーションのニーズに従って変更されたことに注意してください。

次のコマンドによって、ファイルに格納されているポリシー・セットがOracle Clusterwareに登録されます。

```
$ crsctl modify policyset -file file_name
```

前述の例に示すのと同じ結果を得るには、crsctl modify policysetコマンドを使用してDefaultポリシー・セット全体を編集し、crsctl modify serverpoolコマンドを使用して、特定のポリシーに対する個々のサーバー・プール属性を変更します。

次のコマンドでは、3つのサーバー・プールを管理するようにDefaultポリシー・セットを変更します。

```
$ crsctl modify policyset -attr "SERVER_POOL_NAMES=Free pool1 pool2 pool3"
```

次のコマンドでは、3つのポリシーを追加します。

```

$ crsctl add policy DayTime
$ crsctl add policy NightTime
$ crsctl add policy Weekend

```

次のコマンドでは、ポリシーの要件に従って3つのサーバー・プールを構成します。

```

$ crsctl modify serverpool pool1 -attr "MIN_SIZE=2,MAX_SIZE=2" -policy DayTime
$ crsctl modify serverpool pool1 -attr "MIN_SIZE=1,MAX_SIZE=1" -policy NightTime
$ crsctl modify serverpool pool1 -attr "MIN_SIZE=0,MAX_SIZE=0" -policy Weekend
$ crsctl modify serverpool pool2 -attr "MIN_SIZE=1,MAX_SIZE=1" -policy DayTime
$ crsctl modify serverpool pool2 -attr "MIN_SIZE=2,MAX_SIZE=2" -policy NightTime
$ crsctl modify serverpool pool2 -attr "MIN_SIZE=1,MAX_SIZE=1" -policy Weekend
$ crsctl modify serverpool pool3 -attr "MIN_SIZE=1,MAX_SIZE=1" -policy DayTime
$ crsctl modify serverpool pool3 -attr "MIN_SIZE=1,MAX_SIZE=1" -policy NightTime
$ crsctl modify serverpool pool3 -attr "MIN_SIZE=3,MAX_SIZE=3" -policy Weekend

```

これで、3つのアプリケーションの要件に対応するためにサーバー・プールを管理する3つの個々のポリシーができます。

ポリシーのアクティブ化

これでポリシー・セットが構成され、3つの異なるポリシーを使用して3つのサーバー・プールを制御します。必要に応じてポリシーをアクティブ化でき、各ポリシーの構成に従ってOracle Clusterwareでサーバー・プールを再構成するように指示することができます。

次のコマンドでは、DayTimeポリシーをアクティブ化します。

```
$ crsctl modify policyset -attr "LAST_ACTIVATED_POLICY=DayTime"
```

リソースの現在のステータスは、次のとおりです。

```
$ crsctl status resource -t
```

Name	Target	State	Server	State details
Cluster Resources				

app1				
1	ONLINE	ONLINE	mjk_has3_2	STABLE
2	ONLINE	ONLINE	mjk_has3_0	STABLE
app2				
1	ONLINE	ONLINE	mjk_has3_1	STABLE
app3				
1	ONLINE	ONLINE	mjk_has3_3	STABLE

サーバー・プールのステータスは、次のとおりです。

```
$ crsctl stat serverpool
NAME=Free
ACTIVE_SERVERS=
NAME=Generic
ACTIVE_SERVERS=
NAME=pool1
ACTIVE_SERVERS=mjk_has3_0 mjk_has3_2
NAME=pool2
ACTIVE_SERVERS=mjk_has3_1
NAME=pool3
ACTIVE_SERVERS=mjk_has3_3
```

サーバーはDayTimeポリシーに従って割り当てられ、アプリケーションはそれぞれのサーバーで実行されます。

次のコマンドでは、Weekendポリシーをアクティブ化します(サーバー・プールには様々なサイズがあるため、サーバーがサーバー・プール間を移動するにつれて、停止するアプリケーションもあれば、起動するアプリケーションもあります)。

```
$ crsctl modify policyset -attr "LAST_ACTIVATED_POLICY=Weekend"
CRS-2673: Attempting to stop 'app1' on 'mjk_has3_2'
CRS-2673: Attempting to stop 'app1' on 'mjk_has3_0'
CRS-2677: Stop of 'app1' on 'mjk_has3_0' succeeded
CRS-2672: Attempting to start 'app3' on 'mjk_has3_0'
CRS-2677: Stop of 'app1' on 'mjk_has3_2' succeeded
CRS-2672: Attempting to start 'app3' on 'mjk_has3_2'
CRS-2676: Start of 'app3' on 'mjk_has3_2' succeeded
CRS-2676: Start of 'app3' on 'mjk_has3_0' succeeded
```

リソースの現在のステータスは、次のとおりです。

```
$ crsctl status resource -t
```

Name	Target	State	Server	State details
Cluster Resources				

app1				
1	ONLINE	OFFLINE		STABLE
2	ONLINE	OFFLINE		STABLE
app2				
1	ONLINE	ONLINE	mjk_has3_1	STABLE
app3				
1	ONLINE	ONLINE	mjk_has3_0	STABLE
2	ONLINE	ONLINE	mjk_has3_2	STABLE

サーバー・プールのステータスは、次のとおりです。

```
$ crsctl status serverpool
NAME=Free
ACTIVE_SERVERS=
NAME=Generic
ACTIVE_SERVERS=
NAME=pool1
ACTIVE_SERVERS=
NAME=pool2
ACTIVE_SERVERS=mjk_has3_1
NAME=pool3
ACTIVE_SERVERS=mjk_has3_0 mjk_has3_2 mjk_has3_3
```

crsctl modify policysetコマンドを使用して、Oracle Clusterwareは、サーバー・プール構成を変更し、ポリシーの要件に従ってサーバーを移動し、アプリケーションの停止と起動を行いました。

関連トピック

- [Oracle Clusterware制御\(CRSCTL\)ユーティリティ・リファレンス](#)

4 Oracle Flex Cluster

Oracle Flex Clusterは、Oracle Clusterwareを多数のノードに拡張します。

この章の内容は次のとおりです。

- [Oracle Flex Clusterの概要](#)
- [Oracle Flex Clusterの管理](#)
- [Oracle拡張クラスタ](#)

Oracle Flex Clusterの概要

Oracle Flex Cluster構成でインストールされるOracle Grid Infrastructureは、スケーラブルで動的、強固なノード・ネットワークです。

Oracle Flex Clusterは、多数のノードを持つOracle Real Application Clusters (Oracle RAC)データベースなどの様々なアプリケーションにプラットフォームを提供します。Oracle Flex Clusterでは、高可用性のために調整および自動化が必要な他のサービス・デプロイメントのプラットフォームも提供されます。

Oracle Flex Cluster内のすべてのノードは、単一のOracle Grid Infrastructureクラスタに属します。このアーキテクチャでは、様々なサービス・レベル、負荷、障害のレスポンス、およびリカバリに対処するために、アプリケーション・ニーズに基づいてリソースのデプロイメントに対するポリシー決定が集中管理されます。

Oracle Flex Clusterには、[ハブ・ノード](#)とサポートされている他のノードが多数含まれています。Oracle Flex Cluster内のハブ・ノードの最大数は64です。他のノードの数は、さらに多くできます。ハブ・ノードは、異なるタイプのアプリケーションをホストできます。

ハブ・ノードは、接続が強固であり、共有ストレージに直接アクセスできる点で、Oracle Clusterwareの標準クラスタ構成におけるOracle Grid Infrastructureノードに類似しています。ハブ・ノードを使用して、読取り/書き込みデータベース・インスタンスをホスティングします。

Oracle Flex Clusterでサポートされる他のノードは、共有記憶域への直接アクセスが不要なく、かわりにハブ・ノードを介してデータを要求する点が、標準のOracle Grid Infrastructureノードと異なります。他のノードを使用して、読取り専用データベース・インスタンスをホスティングします。

ノート:



同じプライマリ・データベースの読取り/書き込みデータベース・インスタンスと読取り専用データベース・インスタンスは、Oracle Flex Cluster で共存できます。

ハブ・ノードは、クラスタ・メンバー・ノードとして他のノードを一切必要とせずにOracle Flex Cluster構成で実行できますが、他のノードは最低1つのハブ・ノードを含むクラスタのメンバーである必要があります。

ノート:



Oracle Flex Cluster をアップグレードする場合、最初にハブ・ノードをアップグレードし、アップグレード・プロセスの

一環としてアップグレード済のハブ・ノードを起動しておくことをお勧めします。

リーダー・ノード

読み取り専用モードで実行されるOracle RACデータベース・インスタンスをホスティングするために他のノードを使用でき、これらのリーフ・ノードは[リーダー・ノード](#)になります。これらのノードをパラレル問合せ操作に最適化するには、大量のメモリーをノードにプロビジョニングして、データがホスティング・ノードにキャッシュされるようにします。

ホスティング・ノードでは、ハブ・ノード間で発生するハートビート・メッセージとは異なる定期的なハートビート・メッセージを、関連付けられているハブ・ノードに送信します。ハブ・ノードの計画停止中、ホスティング・ノードは、1つのハブ・ノードにのみ接続されている場合を除き、別のハブ・ノードへの接続を試行します。ハブ・ノードが削除されると、ホスティング・ノードもクラスタから削除されます。

Oracle Flex Clusterの管理

CRSCTLを使用して、小さいクラスタまたは大きいクラスタのいずれかにOracle Grid Infrastructureを正常にインストールした後のOracle Flex Clusterを管理します。

この項には次のトピックが含まれます：

- [クラスタ・モードの変更](#)

関連トピック

- [Oracle Grid Infrastructureのインストレーション・ガイド](#)
- [Oracle Clusterware制御\(CRSCTL\)ユーティリティ・リファレンス](#)

クラスタ・モードの変更

既存のOracle Clusterwareの標準クラスタをOracle Flex Clusterに変更できます。

ノート：



- クラスタ・モードを変更する場合、ダウンタイムが発生します。
- Oracle Flex Cluster からの Oracle Clusterware の標準クラスタへの変更をサポートしていません。
- Oracle Flex Cluster ではグリッド・ネーミング・サービス(GNS)が必要です。
- ゾーン委任は必要ありません。

Oracle Clusterwareの標準クラスタからのOracle Flex Clusterへの変更

CRSCTLを使用して、既存のOracle Clusterwareの標準クラスタからOracle Flex Clusterに変更します。

次のステップを実行します。

1. 次のコマンドを実行して、クラスタの現在のモードを判断します。

```
$ crsctl get cluster mode status
```

2. 次のコマンドを実行して、グリッド・ネーミング・サービス(GNS)が固定VIPで構成されているかを確認します。

```
$ srvctl config gns
```

この手順は、GNSが固定のVIPで構成されていない場合は失敗します。GNSがない場合、次のようにrootとして1つ作成します。

```
# srvctl add gns -vip vip_name | ip_address
```

次のコマンドをrootとして実行して、GNSを起動します。

```
# srvctl start gns
```

3. Oracle Automatic Storage Management Configuration Assistant (ASMCA)を使用して、クラスタで Oracle Flex ASMを有効にしてからクラスタ・モードを変更します。

4. rootとして次のコマンドを実行して、クラスタのモードがOracle Flex Clusterになるよう変更します。

```
# crsctl set cluster mode flex
```

5. クラスタの各ノードでrootとして次のコマンドを実行し、Oracle Clusterwareを停止します。

```
# crsctl stop crs
```

6. クラスタの各ノードでrootとして次のコマンドを実行し、Oracle Clusterwareを起動します。

```
# crsctl start crs -wait
```



ノート:

進捗状況およびステータス・メッセージを表示するには、-wait オプションを使用します。

関連トピック

- [Oracle Automatic Storage Management管理者ガイド](#)

Oracle拡張クラスタ

1つのOracle RACクラスタを、地理的に離れ、それぞれ独自の記憶域を備えた2つ以上のサイトにわたって拡張できます。いずれかのサイトに障害が発生すると、他のサイトがアクティブ・スタンバイとして機能します。

Oracle ASMとOracle Databaseスタックは、いずれも、通常、データ・センターのエンタープライズ・クラスの共有記憶域を使用するように設計されています。ただし、ファイバ・チャネル・テクノロジーにより、計算リソースと記憶域リソースを2つ以上のデータ・センターにわたって分散でき、それらを計算および記憶域のそれぞれのニーズに従ってイーサネット・ケーブルやファイバ・チャネルを介して接続できます。

Oracle拡張クラスタは、Oracle Grid Infrastructureをインストールするときに構成できますが、インストール後も、ConvertToExtendedスクリプトを使用して構成できます。CRSCTLを使用して、Oracle拡張クラスタを管理します。

Oracle拡張クラスタの構成

この手順は、Oracle Grid Infrastructure 12c リリース2 (12.2)以上とともにインストールされたか、後でこのリリースに更新され、通常は1つのサイト(デフォルト・サイト)のみで構成されるクラスタの場合にのみサポートされます。

ノート:



この手順では、クラスタ内のすべてのノードがアクセス可能である必要があります。データベース・アクセスが中断すると、クラスタも停止します。1 つまたは複数のディスク・グループおよび複数の障害グループを含む Oracle 拡張クラスタを構成できます。ConvertToExtended スクリプトを使用して、複数のデータ・サイトを作成し、1 つのノードを各データ・サイトに関連付けることができます。すべての Oracle Flex ASM 記憶域は、既存のディスク・グループを拡張ディスク・グループに変換するメカニズムがないため、デフォルトのクラスタ・サイトに関連付けられたままとなります。クラスタを Oracle 拡張クラスタに変換した後、投票ファイル・メンバーシップは階層化されず、フラットなままとなります。

サイト固有の階層的な投票ファイル・アルゴリズムを利用するには、拡張ディスク・グループを追加し、投票ファイルを拡張ディスク・グループに移行することも必要となります。

次のように、CRSCTLを使用してクラスタを問い合わせ、その拡張ステータスを判別します。

```
$ crsctl get cluster extended
CRS-6579: The cluster is 'NOT EXTENDED'

$ crsctl query cluster site -all
Site 'crsclus' identified by '7b7b3bef4c1f5ff9ff8765bceb45433a' in state 'ENABLED',
and contains nodes 'node1,node2,node3,node4', and disks ''.
```

前述の例では、4つのノード(node1、node2、node3およびnode4)と1つのディスク・グループ(datadg)が含まれる crsclusというクラスタを識別します。このクラスタには構成済のサイトが1つ含まれます。

一部のOracle Flex ASMインスタンス、cssdおよびcrsdがクラスタ・モードを拡張ではないとみなす一方で、他のコンポーネントがクラスタ・モードを拡張であるとみなすことを回避するために、Oracle Clusterwareスタックを停止します。グリッド・プラグ・アンド・プレイ・デーモン(gpnpd)をオンラインのままにする利点は、プロファイルがリモート・ノードで更新されることです。Oracle Clusterwareスタックを次回起動したときに、クラスタ・モードが拡張されます。

```
# crsctl stop cluster -all
```

1. クラスタ内の最初のノードで、次のコマンドを実行します。

```
$ rootcrs.sh -converttoextended -firstnode -sites site_list -site site_name
```

2. crsctl check cssを実行して、クラスタ同期サービス(CSS)がいずれのリモート・ノードでも実行されていないことを確認します。
3. 次のコマンドを使用して、サイト名とサイトGUIDを生成します。

```
$ crsctl query cluster site -all
```

前述のコマンドでは次のような出力が表示されます。

```
Site 'SiteA' identified by GUID '7b7b3bef4c1f5ff9ff8765bceb45433a' in state
'ONLINE' contains nodes 'node1,node2,node3', and disks 'disk1, disk2, disk3'.
Site 'SiteB' identified by GUID '23453bef4c1f5ff9ff8765bceb45433a' in state
'QUARANTINED' contains nodes 'node4, node5, node6', and disks 'disk4, disk5,
disk6'.
Site 'SiteQ' identified by GUID '98763bef4c1f5ff9ff8765bceb45433a' in state
'ONLINE' contains no nodes and disk 'disk7'.
Site 'SiteD' identified by GUID '4abcd453c6bc6fc1ffd3a58849d5ba41' in state
'ONLINE' contains nodes 'cuj1234' and no disk
```

チェックポイントをこの情報で更新し、クラスタ内の残りのノードにコピーします。

4. 次のコマンドを使用して、グリッド・プラグ・アンド・プレイ・プロファイルを更新し、ext_modeをTRUEに設定します。

```
$ cd $Grid_home/gpnp/host_name/profiles/peer
$ gpnpntool getpval -p=profile.xml -prf_sq -o=temp_file //
run_gpnpntool_getpval()
$ gpnpntool edit -p=profile.xml -asm_ext="" prf_sq=seq+1 -o=profile_ext.xml
```

ノート:



prf_sq の値は、ProfileSequence キーの現在のプロファイル内の値よりも大きい必要があります。

```
$ gpnpntool sign -p=profile_ext.xml -o=profile_ext_sign.xml
$ mv profile_ext_sign.xml profile.xml $ gpnpntool put -p=profile.xml
```

5. 次のように、サイトをローカル構成に追加します。

```
$ crsctl add crs site site_name -guid site_guid -local
```

6. 次のように、サイトをグローバル構成に追加します(そのためにはOCRが実行中である必要があります)。

```
$ crsctl add crs site site_name -guid site_guid
```

7. 次のように、このノードのローカル構成でサイトへのマッピングを更新します。

```
$ crsctl modify cluster site site_name -n local_host -local
```

8. 次のように、ノードからサイトへのマッピングを更新します。

```
$ crsctl modify cluster site site_name -n host1, host2, ...
```

9. 次のように、Oracle高可用性サービス・スタックを停止してから起動します。

```
# crsctl stop crs
# crsctl start crs
```

クラスタ内の残りのノードで、次を実行します。

1. 次のように、クラスタを特定のノードに拡張します。

```
$ rootcrs.sh -converttoextended -site site_name
```

2. CSSがいずれのリモート・ノードでも実行されていないことを確認します。

3. 前述のチェックポイント情報を使用して、新しいサイトおよびサイトGUIDを参照します。

4. 次のように、サイトをローカル構成に追加します。

```
$ crsctl add crs site site_name -guid site_guid -local
```

5. 次のように、このノードのローカル構成でサイトへのマッピングを更新します。

```
$ crsctl modify cluster site site_name -n local_host -local
```

6. 次のように、Oracle高可用性サービス・スタックを停止してから起動します。

```
# crsctl stop crs
# crsctl start crs
```

Oracle拡張クラスタの構成が完了したら、次のコマンドを実行して、構成を検証します。

```
$ crsctl get cluster extended
CRS-XXXX: The cluster is 'EXTENDED'
$ crsctl query cluster site -all
Site 'crsclus' identified by '7b7b3bef4c1f5ff9ff8765bceb45433a' is 'ONLINE', and
contains nodes '', and disks ''.
Site 'ny' identified by '888b3bef4c1f5ff9ff8765bceb45433a' is 'ONLINE', and ¥
contains nodes 'node1,node2', and disks ''.
Site 'nj' identified by '999b3bef4c1f5ff9ff8765bceb45433a' is 'ONLINE', and ¥
contains nodes 'node3,node4', and disks ''.
```

前述のコマンド例の出力は、コマンドを実行したときにCRSCTLによって表示される内容と似ています。

5 Oracle Cluster Registryおよび投票ファイルの管理

Oracle Clusterwareには、構成およびノード・メンバーシップを管理する2つの重要なコンポーネントが含まれます(ローカル・コンポーネントのOracle Local Registry(OLR)も含むOracle Cluster Registry(OCR)および投票ファイル)。

- OCRは、Oracle ClusterwareおよびOracle RACデータベースの構成情報を格納します
- OLRは、クラスタ内の各ノードに存在し、特定のノードごとにOracle Clusterwareの構成情報を管理します。
- 投票ファイルはノード・メンバーシップに関する情報を格納します。ノードがクラスタのメンバーになるためには、各投票ファイルに、クラスタ内のすべてのノードがアクセスできる必要があります。

ノート:

- Oracle Clusterware 12c では、RAW またはブロック・デバイスの使用をサポートしていません。従来の Oracle Clusterware リリースで RAW デバイスまたはブロック・デバイスを使用していて、Oracle Clusterware 12c へのアップグレードを行うには、アップグレード前に、OCR および投票ファイルを Oracle Automatic Storage Management (Oracle ASM)または共有ファイル・システムに移行する必要があります。
- スタンドアロン・クラスタでは、OCR および投票ファイルを Oracle ASM または共有ファイル・システムに格納できます。

Oracle Database 12c リリース 2 (12.2)では、共有ファイル・システムに OCR および投票ディスク・ファイルを直接配置することがサポートされなくなりました。Oracle Database 19c (19.3)以降のスタンドアロン・クラスタでは、このサポートの終了が撤回されます。Oracle ドメイン・サービス・クラスタの場合は、Oracle ASM で管理されるディスク・グループに OCR および投票ディスク・ファイルを引き続き配置する必要があります。

可用性を向上させるために、Oracle Clusterwareのインストール時に複数の投票ファイルを構成することをお勧めします。投票ファイルをOracle ASMディスク・グループに配置することを選択すると、標準の、または高い冗長性のディスク・グループを使用する場合に、Oracle ASMによって複数の投票ファイルが確実に構成されます。投票ファイルを共有ファイル・システムに格納することを選択する場合は、複数の投票ファイルを構成するオプションを選択します(この場合、異なるディスクに基づく3つの異なるファイル・システムを指定する必要があります)。

必要に応じて、Oracle Clusterwareのインストール・プロセス完了後に、クラスタを停止することなく投票ファイルを動的に追加または置換することもできます。

この章の内容は次のとおりです。

- [Oracle Cluster RegistryおよびOracle Local Registryの管理](#)
- [投票ファイルの管理](#)

Oracle Cluster RegistryおよびOracle Local Registryの管理

OCRおよびOracle Local Registry (OLR)を管理するには、OCRCONFIG、OCRDUMPおよびOCRCHECKを使用します。

OCRには、クラスタ内のすべてのOracleリソースに関する情報が含まれます。

OLRは、クラスタ内の各ノードに配置されている、OCRに類似したレジストリですが、各ノードに固有の情報が含まれます。様々なサービス間の依存性など、Oracle Clusterwareに関する管理性情報が含まれます。Oracle高可用性サービスでこの情報が使用されます。OLRは、クラスタ内の各ノードのローカル記憶域に配置されています。デフォルトの場所はパス `Grid_base/crsdata/host_name/olr/` にあります。ここで、`host_name` はノードのホスト名です。

OCRの管理については、次のトピックで説明しています。

- [Oracle Automatic Storage ManagementへのOracle Cluster Registryの移行](#)
- [Oracle Cluster Registryの場所の追加、置換、修復および削除](#)
- [Oracle Cluster Registryのバックアップ](#)
- [Oracle Cluster Registryのリストア](#)
- [Oracle Cluster Registryの問題の診断](#)
- [エクスポート・コマンドおよびインポート・コマンドを使用したOracle Cluster Registryの管理](#)
- [Oracle Local Registry](#)

関連トピック

- [OCRCONFIGについて](#)
- [Oracle Cluster Registryのトラブルシューティング](#)

Oracle Automatic Storage ManagementへのOracle Cluster Registryの移行

OCRを移行してOracle ASMに配置して、Oracle Clusterware記憶域の管理で改善点を利用できます。

ノート:



以前のバージョンの Oracle Clusterware を 19c にアップグレードして Oracle ASM ディスク・グループに OCR を格納する場合は、「ASM の互換性」の互換性属性を 11.2.0.2 以降に設定する必要があります。

OCRCONFIGを使用してOCRをOracle ASMに移行するには:

1. Oracle Clusterware 19cへのアップグレードが完了していることを確認します。次のコマンドを実行し、現在実行中のバージョンを検証します。

```
$ crsctl query crs activeversion
```
2. Oracle ASM Configuration Assistant(ASMCA)を使用し、クラスタ内のすべてのノードでOracle ASMを構成して起動します。
3. ASMCAを使用して、既存のOCRのサイズと同じかそれ以上で、かつ通常以上の冗長性があるOracle ASMディスク・グループを作成します。

ノート:



- OCR が外部冗長性で Oracle ASM ディスク・グループに格納されている場合に、ディスク・グルー

ブ内のディスクに障害が発生したときは、OCR の損失を回避するために別の OCR 場所を別のディスク・グループに追加することをお勧めします。

Oracle では、OCR を Oracle ASM および共有ファイル・システムの両方に格納するなど、OCR を異なるストレージ・タイプに同時に格納することはできません(ただし、移行中はこのかぎりではありません)。

- あるノードで Oracle ASM インスタンスに障害が発生すると、OCR はそのノードで使用できなくなります。

Oracle ASM インスタンス障害の影響を受けるノードで動作している CRSD プロセスが [OCR ライター](#) であり、過半数の OCR の場所が Oracle ASM に格納されている場合、このノードの Oracle ASM インスタンスの停止中に OCR に対して I/O を試行すると、CRSD は停止して、操作不能になります。この時点で、このノードでのクラスタ管理に影響が及びます。

どのような場合であっても、1 つのノードの 1 つの Oracle ASM インスタンスの障害が、クラスタ全体に影響を及ぼすことはありません。

- 作成する Oracle ASM ディスク・グループが、クラスタ内のすべてのノードにマウントされることを確認します。

4. OCRをOracle ASMディスク・グループに追加するには、[Oracle Clusterwareスタック](#)が実行されていることを確認し、rootとして次のコマンドを実行します。

```
# ocrconfig -add +new_disk_group
```

複数のOCRの場所を追加すると、このコマンドを複数回実行できます。OCRの場所は、5つまで保持できます。ただし、連続する各実行では、異なるディスク・グループを指す必要があります。

5. 使用されなくなったストレージ構成を削除するには、rootとして次のコマンドを実行します。

```
# ocrconfig -delete old_storage_location
```

構成したすべてのOCRに、このコマンドを実行します。

次の例は、OCRCONFIGを使用して2つのOCRをOracle ASMに移行する方法を示しています。

```
# ocrconfig -add +new_disk_group
# ocrconfig -delete /ocrdata/ocr_1
# ocrconfig -delete /ocrdata/ocr_2
```

ノート:



- OCR ではディスク・グループの冗長性が継承されます。OCR に高い冗長性が必要な場合は、作成時にディスク・グループを高い冗長性で構成する必要があります。
- この新規に作成された OCR の場所に Oracle ASM の SPFILE ファイルを置くことを、お勧めします。

関連トピック

- [Oracle Automatic Storage Management管理者ガイド](#)
- [Oracle Automatic Storage Management管理者ガイド](#)
- [Oracle Automatic Storage Management管理者ガイド](#)
- [Oracle Grid Infrastructureのインストール・ガイド](#)

Oracle ASMから他のタイプの記憶域へのOracle Cluster Registryの移行

Oracle ASMから別の記憶域タイプにOCRを移行するには、次の手順を実行します。

1. Oracle Clusterware 12cへのアップグレードが完了していることを確認します。次のコマンドを実行し、現在実行中のバージョンを検証します。

```
$ crsctl query crs activeversion
```

2. ファイルを共有ファイル・システムまたはクラスタ・ファイル・システムにroot、oinstall、640の権限で作成します。



ノート:

OCR のシングル・ポイント障害を排除するには、プライマリ・ストレージの場所の 2 つ以上のミラーを作成します。OCR は、最大 5 つの場所をサポートします。

3. マウント・パーティションに500MB以上の領域があることを確認します。
4. 作成したファイルがクラスタ内のすべてのノードから認識できることを確認します。
5. ファイルをOCRの場所として追加するには、Oracle Clusterwareスタックが実行されていることを確認し、rootとして次のコマンドを実行します。

```
# ocrconfig -add file_location
```

複数のOCRの場所を追加する場合は、このコマンドを複数回実行できます。このコマンドの後続の実行ではそれぞれ異なるファイルの場所を指定する必要があります。

6. OCRバックアップ場所がOracle ASMディスク・グループ上に構成されていないことを、確認します。

OCRバックアップ場所をファイル・システムに移行するには、次のコマンドをrootとして実行します。

```
# ocrconfig -backuploc file_location
```

7. 使用されなくなったストレージ構成を削除するには、rootとして次のコマンドを実行します。

```
# ocrconfig -delete +asm_disk_group
```

複数のOCRの場所が構成されている場合は、このコマンドを複数回実行できます。

次の例は、OCRCONFIGを使用してOCRをOracle ASMからNASに移行する方法を示しています。

```
# ocrconfig -add /nas/ocr
```

Oracle ASMに格納されていないOCRの場合は、異なるデバイスにOCRをミラー化することをお勧めします。

Oracle Cluster Registryの場所の追加、置換、修復および削除

OCRの場所を管理するには、OCRCONFIGを使用します。

Oracle ClusterwareでのOracleインストール・プロセスでは、OCRを自動的にミラー化するかどうかを選択できます。ミラー化

されたOCRは、共有ネットワーク・ファイル・システム(NFS)、またはオラクル社が動作保証している任意のクラスタ・ファイル・システムに手動で配置できます。または、OCRをOracle ASMに配置し、選択する冗長性オプションに応じて、自動的にミラーを作成できるようにすることができます。

また、次のような場合は、OCRを手動でミラー化できます(「Oracle Cluster Registryの場所の追加」を参照)。

- Oracle Clusterware 12cへのアップグレード時に、OCRをミラー化することを選択しなかった場合
- Oracle Clusterwareのインストール時に、1つのみのOCRの場所を作成した場合

次のように構成することを薦めます。

- OCRがミラー化されていない、または冗長化されていない記憶域に構成されている場合は、3つ以上のOCRの場所。基礎となる記憶域がRAIDでない場合は、OCRをミラー化することをお薦めします。ミラー化することで、OCRがシングル・ポイント障害になることを防止できます。
- OCRがOracle ASMディスク・グループに構成されている場合は、2つ以上のOCRの場所。OCRは2つの独立したディスク・グループに構成する必要があります。通常、これは作業領域およびリカバリ領域です。
- OCRがミラー化されているハードウェアまたはサード・パーティのミラー化されたボリュームに構成されている場合は、2つ以上のOCRの場所。

ノート:



- 元の OCR の場所が存在しない場合は、`ocrconfig -add` または `ocrconfig -replace` コマンドを実行する前に、適切な権限で空(0 バイト)の OCR の場所を作成する必要があります。
- OCR 構成で指定した OCR デバイスが存在し、これらの OCR デバイスが有効であることを確認します。
- 指定した Oracle ASM ディスク・グループが存在し、マウントされていることを確認します。
- 新しい OCR ファイル、デバイスまたはディスク・グループは、クラスタ内のすべてのアクティブ・ノードからアクセスする必要があります。

OCRの場所のミラー化に加えて、OCRの場所の管理には次の操作があります。

- [Oracle Cluster Registryの場所の追加](#)
- [Oracle Cluster Registryの場所の削除](#)
- [Oracle Cluster Registryの場所の置換](#)
- [ローカル・ノードのOracle Cluster Registry構成の修復](#)
- [Oracle Cluster Registryのデータ損失保護メカニズムの無効化](#)

ノート:



この項の操作を実行すると、OCR がクラスタ全体にわたって影響を受け、`ocr.loc` ファイル(Linux および UNIX システムの場合)およびレジストリ・キー(Windows システムの場合)の OCR 構成情報が変更されます。ただし、`ocrconfig` コマンドを実行しても、停止しているノードや Oracle Clusterware が実行されていないノードの

OCR 構成情報は変更されません。

関連トピック

- [Oracle Grid Infrastructureのインストール・ガイド](#)
- [Oracle Automatic Storage Management管理者ガイド](#)

Oracle Cluster Registryの場所の追加

OCRの場所を追加するには、この項の手順を実行します。Oracle Clusterwareでは、最大5つの冗長なOCRの場所を管理できます。

ノート:



OCR がクラスタ・ファイル・システムのファイルまたはネットワーク・ファイル・システムに存在している場合は、この項の手順を実行する前に、空(0 バイト)の OCR の場所ファイルを作成してください。

Oracle ASMまたは他のストレージ・デバイスのいずれかにOCRの場所を追加するには、rootユーザーとして次のコマンドを実行します。

```
# ocrconfig -add +asm_disk_group | file_name
```

ノート:



Linux および UNIX システムの場合、ocrconfig コマンドは、root で実行する必要があります。Windows システムの場合、ユーザーは管理者グループのメンバーである必要があります。

Oracle Cluster Registryの場所の削除

OCRの場所または障害が発生したOCRの場所を削除するには、それ以外の1つ以上のOCRがオンライン状態である必要があります。OCRの場所の削除は、OCR関連のオーバーヘッドを軽減する場合や、OCRをRAIDなどの冗長な記憶域に移動してOCRのミラー化を中止する場合に実行します。

Oracle Clusterware環境からOCRの場所を削除するには、rootユーザーとして次の手順を実行します。

1. 削除するOCRの場所の他に、1つ以上のOCRの場所がオンライン状態になっていることを確認します。

注意:



他の 1 つ以上の OCR の場所がアクティブでオンラインになっていない場合は、OCR の削除手順は実行しないでください。

2. クラスタ内の任意のノードで次のコマンドを実行し、Oracle ASMまたは他の場所のいずれかからOCRの場所を削除します。

```
# ocrconfig -delete +ASM_disk_group | file_name
```

file_name変数は、デバイス名またはファイル名です。このコマンドを実行すると、Oracle Clusterwareが実行中

のすべてのノードでOCR構成が更新されます。

ノート:



構成済みの唯一の Oracle ASM OCR の場所は、Oracle ASM OCR バックアップの場所とは異なる Oracle ASM ディスク・グループにあり、削除することはできません。

Oracle Cluster Registryの場所の置換

既存のOCRの場所を変更する必要があるか、または障害のあるOCRの場所を正常に機能する場所に変更する必要がある場合は、残りのすべてのOCRファイルの場所がオンライン状態のままであれば、次の手順を使用できます。ocrconfig - replaceコマンドでは、2つ以上のOCRの場所が構成されている必要があります。

Oracle Cluster Registryの場所を変更するには:

次の手順を実行します。

1. OCRCHECKユーティリティを使用して、置換しようとしているOCR以外のOCRのコピーがオンラインであるかどうかを検証します(使用するコマンドは次のとおりです)。

```
$ ocrcheck
```

OCRCHECKでは、登録されているすべてのOCRの場所、およびそれらが使用可能(オンライン)かどうかが表示されます。OCRの場所が突然使用できなくなった場合は、Oracle Clusterwareがステータスの変更を示すまでにしばらく時間がかかる場合があります。

ノート:



置換する OCR の場所は、オンラインまたはオフラインのいずれの状態でもかまいません。

2. 次のコマンドを使用して、置換操作を実行しようとしているノードでOracle Clusterwareが実行されていることを確認します。

```
$ crsctl check crs
```

3. rootとして次のコマンドを実行し、destination_fileまたは+ASM_disk_groupのいずれかを使用して、現行のOCRの場所を、現行のターゲットのOCRの場所を示すように置換します。

```
# ocrconfig -replace current_OCR_location -replacement new_OCR_location
```

前述のコマンドは、オンラインである構成済みのOCRの場所が2つ未満の場合は失敗します。

構成済でオンラインのOCRの場所が1つのみである場合は、次のように、最初に新しい場所を追加してから、障害が発生した場所を削除する必要があります。

```
# ocrconfig -add new_OCR_location  
# ocrconfig -delete current_OCR_location
```

ノート:



OCR が存在するノードが停止し、Oracle Clusterware スタックが他のノードで実行中であるときにクラスタ構成が変更されると、OCR は構成変更を検出して、ocr.loc ファイルの内容を変更して構成を自己修正します。

関連トピック

- [Oracle Automatic Storage Management管理者ガイド](#)

ローカル・ノードのOracle Cluster Registry構成の修復

ノードが停止し、このノードがクラスタの唯一のメンバーであるときにクラスタ構成を変更した場合は、OCRを修復する必要がある場合があります。

OCRの修復は、OCRの場所の追加、削除または置換のいずれかに関連します。たとえば、現行のOracle RACクラスタの一部であるノードのいずれかが停止している場合、停止しているノードでOCR構成を更新し、そのノードが再起動された後に再びクラスタに参加できるようにする必要があります。再起動されたノードでrootとして次のコマンド構文を使用し、destination_fileまたは+ASM_disk_groupのいずれかを使用して、現行のターゲットのOCRの場所を示します。

```
# ocrconfig -repair -replace current_OCR_location -replacement target_OCR_location
```

この操作では、このコマンドを実行したノードのOCRのみが変更されます。たとえば、OCRの場所が/dev/sde1である場合、ノードでコマンド構文ocrconfig -repair -add /dev/sde1を実行すると、そのノードのOCRが修復されます。

ノート:

- Oracle クラスタ・レディ・サービス・デーモンが実行されているノードでは、OCR 構成を修復できません。
- ocrconfig -repair を使用して、停止しているノードの OCR を修復する場合、他のノードの OCR ファイル名と同様の(大/小文字が区別される)OCR ファイル名を指定する必要があります。
- ocrconfig -add | -repair | -replace コマンドを実行する場合は、追加するデバイス、ファイルまたは Oracle ASM ディスク・グループにアクセスする必要があります。これはデバイスが存在する必要があることを意味します。空(0 バイト)の OCR の場所を作成するか、または Oracle ASM ディスク・グループが存在し、マウントされている必要があります。

関連トピック

- [OCRCONFIGコマンド・リファレンス](#)
- [Oracle Automatic Storage Management管理者ガイド](#)

Oracle Cluster Registryのデータ損失保護メカニズムの無効化

OCRには、不測の上書きによるデータの損失を防止するメカニズムが存在します。ミラー化されたOCRが構成されている場合、Oracle Clusterwareでは、ミラー化されたOCRの場所にアクセスできず、使用可能なOCRの場所に最新の構成が含まれていることが確認できなかった場合、使用可能なOCRの場所に対してそれ以上の変更は行われません。また、この処理により、1つのOCRのみが使用可能なノードでOracle Clusterwareの起動が禁止され、上書きが行われなくなります。このような場合には、Oracle Enterprise ManagerまたはOracle Clusterwareのアラート・ログ・ファイル(あるいはその両方)で、アラート・メッセージが表示されます。ある1つのノードに限って問題が発生している場合は、他のノードでクラスタ・データベースを起動できます。

ただし、環境内のすべてのクラスタ・ノードが起動できず、OCRの修復もすべてのOCRの場所へのアクセスのリストアもできない場合は、この保護メカニズムを無効にできます。次のリストに示した手順により、使用可能なOCRの場所を使用してクラスタを起動できます。ただし、保護メカニズムを無効にすると、以前の正常な状態のときに使用可能でなかったデータは失われる場合があります。

注意:



次の手順を実行して OCR を無効にした場合、現在アクセス可能な OCR に対して最後に正常な OCR の更新が実行された時点から、この上書きを実行するまでの間に行われた OCR への更新は失われる可能性があります。すなわち、上書きを行う OCR の場所にクラスタ環境に対する構成の最新の更新が含まれていない場合は、`ocrconfig -overwrite` コマンドを実行すると、データが失われる場合があります。

ノードが起動できず、アラート・ログにCLSD-1009メッセージおよびCLSD-1011メッセージが含まれている場合は、次の手順を実行して、OCRを上書きします。

1. CLSD-1009メッセージおよびCLSD-1011メッセージの原因の解決を試行します。

ノードのOCR構成(LinuxおよびUNIXシステムの`ocr.loc`およびWindowsシステムのレジストリ)をOracle Clusterwareが実行中の他のノードと比較します。

- 構成の内容が一致しない場合は、`ocrconfig -repair`を実行します。
- 構成の内容が一致している場合は、LinuxおよびUNIXシステムの`ls`コマンドを実行して、ノードから構成済のすべてのOCRにアクセスできることを確認します。Windowsでは、OCRの場所がファイルの場合は`dir`コマンドを使用して、`GuiOracleObjectManager.exe`を実行し、名前付きのクラスタの一部が存在することを確認します。

2. 最新のOCRに、OCRの最新の更新が含まれていることを確認します。

`ocrdump`コマンドの出力を確認して、最新の更新が含まれているかどうかを確認します。

3. CLSDメッセージの原因となっている問題を解決できない場合は、`ocrconfig -overwrite`コマンドを実行してノードを起動します。

Oracle Cluster Registryのバックアップ

この項では、OCRの内容をバックアップし、それをリカバリに使用する方法について説明します。第1の方法では自動的に生成されたOCRのコピーを使用し、第2の方法では手動でバックアップ・コマンドを発行できます。

- 自動バックアップ: OCRのバックアップは、Oracle Clusterwareによって、4時間ごとに自動的に作成されます。バックアップが作成されるごとに、OCRの最後の3つのバックアップのコピーが常に保存されます。また、バックアップを作成するCRSDプロセスによって、毎日および各週の終わりにもOCRのバックアップが作成され、保存されます。バックアップの頻度またはOracle Databaseで保存されるファイルの数はカスタマイズできません。
- 手動バックアップ: Oracle Clusterwareスタックが稼働しているノードで`ocrconfig -manualbackup`コマンドを実行して、自動バックアップを待たずに、Oracle Clusterwareによって任意の時刻にOCRのバックアップが実行されるようにします。このコマンドは、管理権限を持つユーザーとして実行する必要があります。`-manualbackup`オプションは、OCRを変更する前など、必要に応じてバイナリ・バックアップを取得する場合に、特に役立ちます。

クラスタウェア・スタックがクラスタ内のすべてのノードで停止している場合、`ocrconfig -showbackup`コマンドによって示さ

れるバックアップは、ノードごとに異なる場合があります。

ノート:



ノードで Oracle Clusterware をインストールまたはアップグレードした後、または、クラスタにノードを追加した後、root.sh スクリプトが終了すると、OLR がバックアップされます。

この項には次のトピックが含まれます:

- [バックアップ・ファイルのリスト](#)
- [バックアップ場所の変更](#)

バックアップ・ファイルのリスト

次のコマンドを実行し、バックアップ・ファイルをリストします。

```
ocrconfig -showbackup
```

ocrconfig -showbackup コマンドにより、バックアップ場所、タイムスタンプおよび Oracle Clusterware で作成されたバックアップ・ファイルの元のノード名が表示されます。デフォルトでは、-showbackup オプションによって、自動および手動の両方のバックアップに関する情報が表示されますが、auto フラグまたは manual フラグを含めると、自動バックアップまたは手動バックアップのいずれかの情報のみを表示することができます。

次のコマンドを実行し、内容を検査してバックアップ・ファイルの整合性を検証します。

```
ocrdump -backupfile backup_file_name
```

1日に1つ以上のバックアップ・ファイルが自動的に生成されますが、これらのファイルは、任意のバックアップ・ソフトウェアを使用して、プライマリ OCR が存在するデバイスとは別のデバイスにコピーすることもできます。

Oracle Clusterware 18c以降では、Oracle ASM または共有ファイル・システムを使用してバックアップを生成できますが、ローカル・ファイル・システムに変更することはできません。OCRCONFIG ユーティリティで作成したバックアップ・ファイルを、標準的なオペレーティング・システムまたはサード・パーティのツールを使用してオペレーティング・システム・バックアップの一部として含めることをお勧めします。

バックアップ場所の変更

次のコマンドを実行して、OCR がバックアップを作成する場所を変更します。

```
# ocrconfig -backuploc file_name
```

前述のコマンドの file_name 変数には、すべてのノードからアクセス可能なディレクトリのフル・パス名またはすべてのノードにマウントされる Oracle ASM ディスク・グループを指定できます。OCR バックアップの場所を Oracle ASM ディスク・グループに変更する前に、OCR を Oracle ASM に移行する必要があります。別個のディスク・グループに1つ以上の Oracle ASM OCR の場所がある場合のみ、OCR バックアップの場所を Oracle ASM ディスク・グループに変更できます。

たとえば、ディレクトリ内の OCR のバックアップ場所を指定するには、次のコマンドを実行します。

```
# ocrconfig -backuploc Grid_home/cdata/cluster3
```

OCR バックアップの場所を Oracle ASM ディスク・グループに指定するには、次のようにします。

```
# ocrconfig -backuploc +bkupdg
```

ノート:



Linux および UNIX システムでは、ocrconfig コマンド・オプションのほとんど(すべてではない)を実行するために root ユーザーである必要があります。Windows システムの場合、ユーザーは管理者グループのメンバーである必要があります。

関連トピック

- [エクスポート・コマンドおよびインポート・コマンドを使用したOracle Cluster Registryの管理](#)
- [OCRCONFIGコマンド・リファレンス](#)

Oracle Cluster Registryのリストア

Oracle Cluster Registry (OCR)の問題を確認する方法と、それらの問題を解決する方法を学習します。

- [Oracle Cluster Registryの問題の確認方法](#)
- [LinuxまたはUnixシステムでのOracle Cluster Registryのリストア](#)
- [WindowsシステムでのOracle Cluster Registryのリストア](#)
- [Oracle Restart環境でのOracle Cluster Registryのリストア](#)

Oracle Cluster Registryの問題の確認方法

Oracle Cluster Registry (OCR)の問題が発生した場合は、これらのチェックおよびガイドラインを確認してください。

リソースで問題が発生した場合、OCRのリストアを実行する前にリソースを再起動します。OCRが失敗したことを明確に検証するために、ocrcheckを実行します。コマンドによって失敗メッセージが返された場合、プライマリOCRとOCRミラーの両方に障害が発生しています。

ノート:



- -import オプションを使用して OCR バックアップ・ファイルから構成をリストアすることはできません(「エクスポート・コマンドおよびインポート・コマンドを使用した Oracle Cluster Registry の管理」を参照)。かわりに、次の項で説明するとおり、-restore オプションを使用する必要があります。
- OCR を Oracle ASM ディスク・グループに格納していて、ディスク・グループが使用できない場合は、Oracle ASM ディスク・グループをリカバリし、マウントする必要があります。

LinuxまたはUnixシステムでのOracle Cluster Registryのリストア

LinuxまたはUnixシステムでOCRをリストアするには、この手順を使用します。

OCRをOracle ASMディスク・グループに格納していて、そのディスク・グループが破損している場合は、Oracle ASMユーティリティを使用してOracle ASMディスク・グループをリストアし、OCRをリカバリする前にこのディスク・グループを再びマウントする必要があります。次の手順の指示に従って、ocrconfig -restoreコマンドを実行し、OCRをリカバリします。

ノート:



元の OCR の場所が存在しない場合は、`ocrconfig -restore` コマンドを実行する前に、元の OCR の場所と同じ名前の空(0 バイト)の OCR の場所を作成する必要があります。

LinuxまたはUnixシステムでOCRをリストアするには、次の手順を実行します。

1. 1つのノードで次のコマンドを実行して、クラスタのノードをリストします。

```
$ olsnodes
```

2. OCRがOracle ASMディスク・グループまたはNetwork Attached Storage (NAS)のどちらにあるかに応じて、すべてのノードでrootとして次のいずれかのコマンドを実行し、Oracle Clusterwareを停止します。

OCRがOracle ASMディスク・グループにある場合は、Oracle Clusterwareデーモンを次のコマンドで停止します。

```
# crsctl stop crs
```

前述のコマンドでOCRの破損のためエラーが戻される場合は、rootとして、すべてのノードで次のコマンドを実行し、Oracle Clusterwareを強制的に停止します。

```
# crsctl stop crs -f
```

3. OCRをクラスタ・ファイル・システムまたはネットワーク・ファイル・システムにリストアする場合は、rootとして次のコマンドを実行して、「バックアップ・ファイルのリスト」で特定できるOCRバックアップでOCRをリストアします。

```
# ocrconfig -restore file_name
```

ノート:



OCR バックアップ・ファイルが Oracle ASM ディスク・グループにある場合、ディスク・グループが存在し、マウントされていることを確認します。

このステップの完了後、ステップ11に進みます。

OCRをクラスタ・ファイル・システムまたはネットワーク・ファイル・システムにリストアしていない場合は、次のステップに進みます。

4. rootとして次のコマンドを実行し、1つのノードでOracle Clusterwareスタックを排他モードで起動します。

```
# crsctl start crs -excl -nocrs
```

-nocrsオプションは、残りのOracle ClusterwareスタックでCRSDプロセスおよびOCRが起動されないようにします。表示されるエラーは無視します。

5. 次のコマンドを実行して、CRSDが実行しているかどうかを確認します。

```
$ crsctl status resource ora.crsd -init
```

CRSDが実行している場合は、rootとして次のコマンドを実行して停止します。

```
# crsctl stop resource ora.crsd -init
```


注意:



My Oracle Support で指示されないかぎり、他のすべてのコマンドでは -init フラグを使用しないでください。

6. OCRをOracle ASMディスク・グループにリストアするには、SQL*Plusを使用して、リストアするディスク・グループと同じ名前のディスク・グループを最初に作成し、これをローカル・ノードにマウントする必要があります。

ディスク・グループをローカルでマウントできない場合は、次のSQL*Plusコマンドを実行します。

```
SQL> drop diskgroup disk_group_name force including contents;
```

7. rootとして次のコマンドを実行して、「バックアップ・ファイルのリスト」で特定できるOCRバックアップでOCRをリストアします。

```
# ocrconfig -restore file_name
```

ノート:



- 元の OCR の場所が存在しない場合は、ocrconfig -restore コマンドを実行する前に空(0 バイト)の OCR の場所を作成する必要があります。
- OCR 構成で指定した OCR デバイスが存在し、これらの OCR デバイスが有効であることを確認します。
- OCR を Oracle ASM ディスク・グループに構成した場合は、Oracle ASM ディスク・グループが存在し、マウントされていることを確認します。
- OCR バックアップ・ファイルが Oracle ASM ディスク・グループにある場合、ディスク・グループが存在し、マウントされていることを確認します。

8. 次のコマンドを実行して、OCRの整合性を検証します。

```
# ocrcheck
```

9. 次のコマンドを実行して、ノードで排他モードで実行されているOracle Clusterwareを停止します。

```
# crsctl stop crs -f
```

10. ocrconfig -restoreコマンドを実行しなかった、クラスタ内のすべてのノードでrootとして ocrconfig -repair -replaceコマンドを実行します。たとえば、4ノードのクラスタのノード1でocrconfig -restoreコマンドを実行した場合は、ノード2、3および4でocrconfig -repair -replaceコマンドを実行する必要があります。

11. rootとして、すべてのノードで次のコマンドを実行し、Oracle Clusterwareを起動します。

```
# crsctl start crs
```

12. 次のCVUコマンドを実行して、クラスタの一部として構成されているすべてのクラスタ・ノードのOCRの整合性を検証します。

```
$ cluvfy comp ocr -n all -verbose
```

WindowsシステムでのOracle Cluster Registryのリストア

WindowsシステムでOCRをリストアするには、この手順を使用します。

OCRをOracle ASMディスク・グループに格納していて、そのディスク・グループが破損している場合は、Oracle ASMユーティリティを使用してOracle ASMディスク・グループをリストアし、OCRをリカバリする前にこのディスク・グループを再びマウントする必要があります。ocrconfig -restoreコマンドを実行して、OCRをリカバリします。

ノート:



元の OCR の場所が存在しない場合は、ocrconfig -restore コマンドを実行する前に、元の OCR の場所と同じ名前の空(0 バイト)の OCR の場所を作成する必要があります。

WindowsシステムでOCRをリストアするには、次の手順を実行します。

1. 1つのノードで次のコマンドを実行して、クラスタのノードをリストします。

```
C:¥>olsnodes
```

2. 管理者グループのメンバーとして、すべてのノードで次のコマンドを実行し、Oracle Clusterwareを停止します。

```
C:¥>crsctl stop crs
```

前述のコマンドでOCRの破損のためエラーが戻される場合は、管理者グループのメンバーとして、すべてのノードで次のコマンドを実行し、Oracle Clusterwareを停止します。

```
C:¥>crsctl stop crs -f
```

3. 管理者グループのメンバーとして次のコマンドを実行し、1つのノードでOracle Clusterwareスタックを排他モードで起動します。

```
C:¥>crsctl start crs -excl -nocrs
```

-nocrsオプションは、残りのOracle ClusterwareスタックでCRSDプロセスおよびOCRが起動されないようにします。
表示されるエラーは無視します。

4. 管理者グループのメンバーとして次のコマンドを実行して、「バックアップ・ファイルのリスト」で特定したOCRバックアップ・ファイルでOCRをリストアします。

```
C:¥>ocrconfig -restore file_name
```

OCR構成で指定したOCRデバイスが存在し、これらのOCRデバイスが有効であることを確認します。

ノート:



- OCR 構成で指定した OCR デバイスが存在し、これらの OCR デバイスが有効であることを確認します。
- 指定した Oracle ASM ディスク・グループが存在し、マウントされていることを確認します。

関連項目:

OCR作成の詳細は、『*Oracle Grid Infrastructure*インストール・ガイド』を参照してください。

5. 次のコマンドを実行して、OCRの整合性を検証します。

```
C:¥>ocrcheck
```

6. 次のコマンドを実行して、ノードで排他モードで実行されているOracle Clusterwareを停止します。

```
C:¥>crsctl stop crs -f
```

7. 管理者グループのメンバーとして、すべてのノードで次のコマンドを実行し、Oracle Clusterwareを起動します。

```
C:¥>crsctl start crs
```

8. 次のクラスタ検証ユーティリティ(CVU)・コマンドを実行して、クラスタ・データベース内のすべてのノードのOCRの整合性を検証します。

```
C:¥>cluvfy comp ocr -n all -verbose
```

関連トピック

- [バックアップ・ファイルのリスト](#)
- [Oracle Grid Infrastructureのインストール・ガイド](#)
- [Oracle Automatic Storage Management管理者ガイド](#)
- [クラスタ検証ユーティリティのリファレンス](#)

Oracle Restart環境でのOracle Cluster Registryのリストア

Oracle Restart環境でOCRをリストアするには、この手順を使用します。

ノート:



- OCR は、下位互換性のために存在します。
- 作成された OCR の場所は、Oracle Restart 環境では更新されません。
- Oracle Restart ホームがバックアップされている場合、障害が発生したときに、Oracle Restart ホームをリストアすると OCR がリストアされます。

1. rootとして、すべてのノードで次のコマンドを実行し、Oracle高可用性サービスを停止します。

```
# crsctl stop has [-f]
```

2. ocrcheck -configコマンドを実行して、OCRの場所を特定し、適切な権限を使用してその場所で空(0バイト)のOCRの場所を作成します。

3. rootとして次のコマンドを実行し、OCRをリストアします。

```
# crsctl pin css -n host_name
```



ノート:

OCR 構成で指定した OCR デバイスが存在し、これらの OCR デバイスが有効であることを確認します。

関連項目:

OCR作成の詳細は、『*Oracle Grid Infrastructure*インストール・ガイド』を参照してください。

4. ocrcheckコマンドを実行して、OCRの整合性を検証します。
5. すべてのノードで次のコマンドを実行し、Oracle高可用性サービスを起動します。

```
$ crsctl start has
```

関連トピック

- [Oracle Grid Infrastructureのインストール・ガイド](#)

Oracle Cluster Registryの問題の診断

OCRの問題を診断するには、OCRDUMPおよびOCRCHECKユーティリティを使用します。

関連トピック

- [OCRDUMPユーティリティの構文およびオプション](#)
- [OCRCHECKユーティリティの使用](#)

エクスポート・コマンドおよびインポート・コマンドを使用したOracle Cluster Registryの管理

自動的に作成されたOCRバックアップ・ファイルを使用するのみでなく、環境のノードの追加や削除、Oracle Clusterwareリソースの変更、データベースのアップグレード、ダウングレードまたは作成などの重要な構成変更を行う前と後には、OCRの内容をエクスポートする必要もあります。これは、OCRの内容をファイル形式でエクスポートするocrconfig -exportコマンドを使用して実行します。

注意:

OCR のリストアでは次の制限事項に注意してください。

- ocrconfig -restore で生成されるファイル形式は、ocrconfig -export で生成されるファイル形式と互換性がありません。ocrconfig -export コマンドと ocrconfig -import コマンドには互換性があります。ocrconfig -manualbackup コマンドと ocrconfig -restore コマンドには互換性があります。2 つのファイル形式には互換性がないため、交換して使用しないでください。
- OCR をエクスポートする場合は、名前の文字列に ocr、クラスタ名およびタイムスタンプを含めることをお勧めします。たとえば:

```
ocr_mycluster1_20090521_2130_export
```

また、ocrconfig -exportコマンドを使用すると、構成の変更によってエラーが発生した場合に、-importオプションを使

用してOCRをリストアできるようになります。たとえば、構成に解決できない問題がある場合や、構成の変更後にOracle Clusterwareを再起動できない場合は、プラットフォームに応じた手順を使用して、構成をリストアします。

OCRをリストアするには、次の理由から、自動バックアップまたは手動バックアップのいずれかと`ocrconfig -restore`コマンドを使用し、`ocrconfig -export`コマンドおよび`ocrconfig -import`コマンドは使用しないことをお勧めします。

- バックアップはOCRの一貫したスナップショットですが、エクスポートは、そうではありません。
- バックアップはシステムがオンライン状態の場合に作成されます。`ocrconfig -export`コマンドを使用して一貫したスナップショットを取得するには、クラスタ内のすべてのノードでOracle Clusterwareを停止する必要があります。
- OCRDUMPユーティリティを使用して、バックアップを検査できます。エクスポートの内容は検査できません。
- `ocrconfig -showbackup`コマンドを使用してバックアップをリストできますが、生成されたすべてのエクスポートはユーザーが追跡する必要があります。

この項には次のトピックが含まれます：

- [LinuxまたはUNIXシステムでのOracle Cluster Registryの内容のインポート](#)
- [WindowsシステムでのOracle Cluster Registryの内容のインポート](#)

ノート：



構成を変更すると、ほとんどの場合は、OCR の内容が変更されるのみでなく、ファイルおよびデータベース・オブジェクトが作成されます。OCR をリストアしても、このような変更はリストアされない場合があります。これらの構成の変更のリストアが一部でも失敗した場合は、以前の正しい構成に戻すために OCR をリストアしないでください。実行した場合、OCR の場所の内容がシステムの他の部分の状態と一致なくなる可能性があります。

LinuxまたはUNIXシステムでのOracle Cluster Registryの内容のインポート

LinuxまたはUNIXシステムでOCRをインポートするには、この手順を使用します。

ノート：



この手順では、クラスタ内のすべてのノードでの Oracle Clusterware のデフォルトのインストールを前提としています。この場合、Oracle Clusterware の自動起動は有効になります。

1. 1つのノードで次のコマンドを実行して、クラスタのノードをリストします。

```
$ olsnodes
```

2. rootとして、すべてのノードで次のコマンドを実行し、Oracle Clusterwareを停止します。

```
# crsctl stop crs
```

前述のコマンドでOCRの破損のためエラーが戻される場合は、rootとして、すべてのノードで次のコマンドを実行し、Oracle Clusterwareを停止します。

```
# crsctl stop crs -f
```

3. rootとして次のコマンドを実行し、1つのノードでOracle Clusterwareスタックを排他モードで起動します。

```
# crsctl start crs -excl
```

表示されるエラーは無視します。

CRSDが動作しているかどうかをチェックします。実行されている場合は、rootとして次のコマンドを実行して停止します。

```
# crsctl stop resource ora.crsd -init
```

注意:



-init フラグは他のコマンドでは使用しないでください。

4. rootとして次のコマンドを実行し、OCRをインポートします。

```
# ocrconfig -import file_name
```

クラスタまたはネットワーク・ファイル・システムにOCRをインポートしている場合は、ステップ7に進みます。

ノート:



- 元の OCR の場所が存在しない場合は、ocrconfig -import コマンドを実行する前に空 (0 バイト) の OCR の場所を作成する必要があります。
- OCR 構成で指定した OCR デバイスが存在し、これらの OCR デバイスが有効であることを確認します。
- OCR を Oracle ASM ディスク・グループに構成した場合は、Oracle ASM ディスク・グループが存在し、マウントされていることを確認します。

関連項目:

OCR作成の詳細は、『Oracle Grid Infrastructureインストレーション・ガイド』を参照してください。

5. 次のコマンドを実行して、OCRの整合性を検証します。

```
# ocrcheck
```

6. 次のコマンドを実行して、ノードで排他モードで実行されているOracle Clusterwareを停止します。

```
# crsctl stop crs -f
```

7. rootとして、すべてのノードで次のコマンドを実行し、Oracle Clusterwareを起動します。

```
# crsctl start crs
```

8. 次のCVUコマンドを実行して、クラスタの一部として構成されているすべてのクラスタ・ノードのOCRの整合性を検証します。

```
$ cluvfy comp ocr -n all -verbose
```

ノート:



インポートできるのは、エクスポートした OCR のみです。バックアップから OCR をリストアするには、「Oracle Cluster Registry のバックアップ」で説明するとおり、かわりに -restore オプションを使用する必要があります。

関連トピック

- [Oracle Grid Infrastructureのインストール・ガイド](#)
- [Oracle Automatic Storage Management管理者ガイド](#)
- [Oracle Cluster Registryのバックアップ](#)
- [クラスタ検証ユーティリティのリファレンス](#)

WindowsシステムでのOracle Cluster Registryの内容のインポート

WindowsシステムでOCRをインポートするには、この手順を使用します。

ノート:



この手順では、クラスタ内のすべてのノードでの Oracle Clusterware のデフォルトのインストールを前提としています。この場合、Oracle Clusterware の自動起動は有効になります。

1. 1つのノードで次のコマンドを実行して、クラスタのノードをリストします。

```
C:¥>olsnodes
```

2. 管理者グループのメンバーとして、すべてのノードで次のコマンドを実行し、Oracle Clusterwareを停止します。

```
C:¥>crsctl stop crs
```

前述のコマンドでOCRの破損のためエラーが戻される場合は、管理者グループのメンバーとして、すべてのノードで次のコマンドを実行し、Oracle Clusterwareを停止します。

```
C:¥>crsctl stop crs -f
```

3. 管理者グループのメンバーとして次のコマンドを実行し、1つのノードでOracle Clusterwareスタックを排他モードで起動します。

```
C:¥>crsctl start crs -excl
```

表示されるエラーは無視します。

CRSDが動作しているかどうかをチェックします。実行されている場合は、管理者グループのメンバーとして次のコマンドを実行して停止します。

```
C:¥>crsctl stop resource ora.crsd -init
```

注意:



-init フラグは他のコマンドでは使用しないでください。

4. 管理者グループのメンバーとして次のコマンドを実行してOCRをインポートします。

```
C:¥>ocrconfig -import file_name
```

OCR構成で指定したOCRデバイスが存在し、これらのOCRデバイスが有効であることを確認します。

ノート:



- 元の OCR の場所が存在しない場合は、ocrconfig -import コマンドを実行する前に空 (0 バイト) の OCR の場所を作成する必要があります。
- OCR 構成で指定した OCR デバイスが存在し、これらの OCR デバイスが有効であることを確認します。
- 指定した Oracle ASM ディスク・グループが存在し、マウントされていることを確認します。

5. 次のコマンドを実行して、OCRの整合性を検証します。

```
C:¥>ocrcheck
```

6. 次のコマンドを実行して、ノードで排他モードで実行されているOracle Clusterwareを停止します。

```
C:¥>crsctl stop crs -f
```

7. 管理者グループのメンバーとして、すべてのノードで次のコマンドを実行し、Oracle Clusterwareを起動します。

```
C:¥>crsctl start crs
```

8. 次のクラスタ検証ユーティリティ(CVU)・コマンドを実行して、クラスタ・データベース内のすべてのノードのOCRの整合性を検証します。

```
C:¥>cluvfy comp ocr -n all -verbose
```

関連トピック

- [Oracle Automatic Storage Management管理者ガイド](#)
- [クラスタ検証ユーティリティのリファレンス](#)

Oracle Local Registry

Oracle Clusterware 12cでは、クラスタ内の各ノードに、Oracle ClusterwareによるOCRのインストール時にインストールおよび構成される、Oracle Local Registry (OLR)というノード固有のリソースのローカル・レジストリがあります。Oracle Clusterwareが実行されているかどうか、または完全に機能しているかどうかに関係なく、各ノードの複数のプロセスには、それらのプロセスが存在するノードに固有のOLRに対する同時読取りおよび書き込みアクセス権があります。

デフォルトでは、OLRは各ノードのGrid_base/crsdata/host_name/olr/hostname_release.olrに格納されています。

OLRは、-localオプションを指定したOCRCHECK、OCRDUMPおよびOCRCONFIGユーティリティをrootとして使用して管理します。

- OCRCHECKユーティリティを次のように使用すると、ローカル・ノードのOLRのステータスをチェックできます。

```
# ocrcheck -local
```

```
Status of Oracle Local Registry is as follows :
Version                : 4
Total space (kbytes)   : 491684
Used space (kbytes)    : 83120
Available space (kbytes) : 408564
ID                     : 1260380057
Device/File Name       : Grid_base/crsdata/dglnx6/olr/dglnx6_19.olr
                        Device/File integrity check succeeded
Local registry integrity check succeeded
Logical corruption check succeeded
```

- OCRDUMPユーティリティを次のように使用すると、プログラムを起動したテキスト端末にローカル・ノードのOLRの内容を表示できます。

```
# ocrdump -local -stdout
```

- OCRCONFIGユーティリティを使用すると、ローカル・ノードのOLRで管理タスクを実行できます。

- OLRをファイルにエクスポートするには、次のようにユーティリティを使用します。

```
# ocrconfig -local -export file_name
```

ノート:

- -manualbackup コマンドおよび -restore コマンドを使用し、-import コマンドおよび -export コマンドは使用しないことをお勧めします。

- OLR をエクスポートする場合は、名前の文字列に olr、ホスト名およびタイムスタンプを含めることをお勧めします。たとえば:

```
olr_myhost1_20090603_0130_export
```

- 指定したファイルをOLRにインポートするには、次のようにユーティリティを使用します。

```
# ocrconfig -local -import file_name
```

- OLRを手動でバックアップするには、次のようにユーティリティを使用します。

```
# ocrconfig -local -manualbackup
```

ノート:

Oracle Clusterware は、デフォルトでは、インストールまたはアップグレード後に OLR をバックアップし、その後も定期的に OLR をバックアップします。初期バックアップの後はいつでも、OLR を手動でバックアップできます。

OCR を Oracle ASM から他の記憶域に移行する場合、または OCR を他の記憶域から Oracle ASM に移行する場合は、新しいバックアップを作成することをお勧めします。

OLR のデフォルトのバックアップ場所は、パス Grid_base/crsdata/host_name/olr/にあります。

- OLRバックアップ・ファイルの内容を表示するには、次のようにユーティリティを使用します。

```
ocrdump -local -backupfile olr_backup_file_name
```

- OLRのバックアップの場所を変更するには、次のようにユーティリティを使用します。

```
ocrconfig -local -backuploc new_olr_backup_path
```

- OLRをリストアするには、次のようにします。

```
# crsctl stop crs
# ocrconfig -local -restore file_name
# ocrcheck -local
# crsctl start crs
$ cluvfy comp olr
```

Oracle Cluster Registry構成のアップグレードおよびダウングレード

Oracle Cluster Registry構成を手動でアップグレードおよびダウングレードできます。

Oracle Clusterwareをアップグレードすると、ocrconfig -upgradeコマンドが自動的に実行されます。ダウングレードするには、各コンポーネントのダウングレード手順に従ってください。OCRをアップグレードする場合は、OCRCHECKユーティリティを使用してOCRの整合性を検証できます。

投票ファイルの管理

この項では、クラスタ内の投票ファイルの管理について、次のトピックで説明します。

- [Oracle ASMでの投票ファイルの格納](#)
- [投票ファイルのバックアップ](#)
- [投票ファイルのリストア](#)
- [投票ファイルの追加、削除または移行](#)

注意:



以前のバージョンの Oracle Clusterware で投票ファイルのバックアップおよびリカバリに使用された dd コマンドは、Oracle Clusterware 12c 以降のリリースではサポートされていません。dd コマンドまたは cp コマンドを使用してコピーされた投票ファイルをリストアすると、Oracle Clusterware 12c 以降のリリースのスタックが起動されない場合があります。この章で説明されているバックアップおよびリストア手順を使用して、適切な投票ファイルの機能を確保します。

ノート:



- 投票ファイルの管理には有効かつ動作中の OCR が必要です。投票ファイルを追加、削除、置換またはリストアする前に、ocrcheck コマンドを root として実行します。OCR が使用できない場合、または破損している場合は、「Oracle Cluster Registry のリストア」で説明するとおり、OCR をリストアする必要があります。

- 以前のバージョンの Oracle Clusterware を Oracle Clusterware 12c 以降のリリースにアップグレードして Oracle ASM ディスク・グループに投票ファイルを格納する場合は、「ASM の互換性」(COMPATIBLE.ASM)の互換性属性を 12.1.0.0 に設定する必要があります。

関連トピック

- [Oracle Cluster Registryのリストア](#)
- [Oracle Automatic Storage Management管理者ガイド](#)

Oracle ASMでの投票ファイルの格納

投票ファイルをOracle ASMに格納するようにすると、Oracle ASMは、選択したディスク・グループにクラスタのすべての投票ファイルを格納します。

Oracle ASMでは、投票ファイルは、格納されている他のファイルとは異なる方法で管理されます。Oracle ASMに格納される投票ファイルとOracle ASMに格納されない投票ファイルは、同じクラスタ内で使用できません。

Oracle ASMで投票ファイルを構成した後、投票ファイルの構成に変更を加えることができるのは、`crsctl replace votedisk`コマンドを使用する場合のみです。これは、作業投票ファイルがない場合にもあてはまります。`crsctl query css votedisk`によって使用中の投票ディスクがないと報告されても、Oracle ClusterwareはOracle ASMが使用中だったことを記憶しているため、`replace`が必要になります。`replace`を使用して、Oracle ASM以外の記憶域に投票ファイルを移動した後でのみ、`add css votedisk`および`delete css votedisk`が再び使用可能になります。

特定のOracle ASMディスク・グループに格納できる投票ファイルの数は、ディスク・グループの冗長性に依存します。

デフォルトでは、Oracle ASMが、ディスク・グループ内のそれ自体の障害グループに各投票ファイルを配置します。障害グループは、ディスク・グループ内のディスクのサブセットです。障害グループは、あるディスクに障害が発生し、コンポーネントを共有する他のディスクにも障害が発生した場合などに、コンポーネントを共有するディスクを定義します。障害グループとして定義する可能性のある対象の例として、同じSCSIコントローラを共有しているSCSIディスク・セットがあります。障害グループは、冗長データの格納に使用するOracle ASMディスクを決定するために使用されます。たとえば、ファイルに2方向のミラー化が指定されている場合、ファイル・エクステンツの冗長コピーは別の障害グループに格納される必要があります。

Oracle ASMディスク・グループに選択した冗長レベルによって、Oracle ASMがディスク・グループ内のファイルをミラー化する方法および必要となるディスク数とディスク領域が決まります。投票ファイルがディスク・グループにある場合、Oracle Clusterwareファイル(OCRおよび投票ファイル)があるディスク・グループの障害グループの最小数は、投票ファイルが定数障害グループに格納されているため、他のディスク・グループよりも多くなります。

定数障害グループは、Oracle Clusterware投票ファイルを格納する特殊なタイプの障害グループです。定数障害グループは、指定した障害グループの定数が使用可能であることを確認するために使用されます。Oracle ASMでOracle Clusterwareファイルが含まれるディスク・グループがマウントされるときに、1つ以上の障害グループが失われた場合にこのディスク・グループをマウントできるかどうかを判別するために、クォラム障害グループが使用されます。定数障害グループ内のディスクにはユーザー・データが含まれないため、ユーザー・データを格納するための冗長性要件を決定するときには、このグループは考慮されません。

Oracle ASMフレックス・ディスク・グループは、Oracle ASMファイル・グループおよび割当て制限グループをサポートするディスク・グループ・タイプです。通常、フレックス・ディスク・グループを使用すると、ユーザーはディスク・グループ・レベルだけでなくデータベースの粒度でもストレージを管理できます。

冗長性レベルには次のものがあります。

- 外部冗長: 外部冗長ディスク・グループでは、最小で1台のディスク・デバイスが必要です。外部冗長のディスク・グループで有効なディスク領域は、全デバイスのディスク領域の合計です。

Oracle ASMは外部冗長ディスク・グループ内のデータをミラー化しないため、RAIDなどのストレージ・デバイスによる外部冗長を使用するか、または独自のデータ保護メカニズムを持つ類似デバイスを使用することをお勧めします。

- 標準冗長: 標準冗長ディスク・グループでは、最小で2台のディスク・デバイス(または2つの障害グループ)が必要です。標準冗長のディスク・グループで有効なディスク領域は、すべてのデバイスのディスク領域の合計の半分です。

Oracle Clusterwareファイルの場合、標準冗長のディスク・グループは最小で3台のディスク・デバイス(3台のうち2台は障害グループにより使用され、3台すべてがクォーラム障害グループにより使用されます)を必要とし、3つの投票ファイル、1つのOCRおよびOCRのミラーを提供します。標準冗長のディスク・グループを使用すると、1つの障害グループが失われた後もクラスタは機能したままになります。

- 高冗長: 高冗長ディスク・グループでは、Oracle ASMはデフォルトで3方向のミラー化を使用してパフォーマンスを向上させ、最高レベルの信頼性を提供します。高冗長ディスク・グループでは、最小で3台のディスク・デバイス(または3つの障害グループ)が必要です。高冗長のディスク・グループで有効なディスク領域は、全デバイスのディスク領域の合計の3分の1です。

Oracle Clusterwareファイルの場合、高冗長ディスク・グループは最小で5台のディスク・デバイス(5台のうち3台は障害グループにより使用され、5台すべてがクォーラム障害グループにより使用されます)を必要とし、5つの投票ファイル、1つのOCRおよび2つのOCRミラーを提供します。高い冗長性により、クラスタは2つの障害グループが失われた後も機能を維持できます。

`crsctl replace votedisk`コマンドを使用すると、特定の投票ファイル・セットを、あるOracle ASMディスク・グループから別のOracle ASMディスク・グループに、または動作保証されているファイル・システムに移動できます。あるOracle ASMディスク・グループから別のOracle ASMディスク・グループに投票ファイルを移動する場合は、以前のディスク・グループとは異なる冗長レベルのディスク・グループに投票ファイルを配置することで、投票ファイルの数を変更できます。

ノート:

- 1つのディスク・グループで、投票ファイルの数を直接変更することはできません。
- Oracle ASM ではディスク・グループの冗長性レベルに従って投票ファイルの数が管理されるため、Oracle ASM ディスク・グループに格納されている投票ファイルに対して `crsctl add | delete votedisk` コマンドを使用することはできません。
- 投票ファイルが Oracle ASM ディスク・グループに格納されている場合は、クラスタ・ファイル・システムに投票ファイルを追加できません。Oracle ASM、および同じクラスタのクラスタ・ファイル・システム上に直接、投票ファイルを同時に格納することはサポートされていません。

関連トピック

- [Oracle Automatic Storage Management管理者ガイド](#)
- [投票ファイルの追加、削除または移行](#)

投票ファイルのバックアップ

Oracle Clusterwareは、構成変更の一部として投票ファイルのデータを自動的にOCRにバックアップし、ユーザーが追加した任意の投票ファイルにデータを自動的にリストアします。

ただし、すべての投票ファイルが破損した場合は、「投票ファイルのリストア」で説明するとおり、それをリストアできます。

関連トピック

- [投票ファイルのリストア](#)

投票ファイルのリストア

すべての投票ファイルが破損している場合は、この手順を使用してリストアできます。

1. 必要に応じて、「Oracle Cluster Registryのリストア」で説明するとおり、OCRをリストアします。

このステップが必要なのは、OCRがOracle ASM上に存在し、ASMディスク・グループが使用できなくなっている場合など、OCRも破損しているか、または使用できない場合のみです。

2. 次のコマンドを1つのノードのみからrootとして実行し、Oracle Clusterwareスタックを排他モードで起動します。この場合、投票ファイルが存在したり、使用可能である必要はありません。

```
# crsctl start crs -excl
```

3. コマンドcrsctl query css votediskを実行して、現在定義されている投票ファイルのリストを取得します。

たとえば:

```
$ crsctl query css votedisk
--
## STATE      File Universal Id      File Name Disk group
 1. ONLINE    6f57843d89464c46ea747362e8a3fa43 (/dev/sdb1) [DATA]
 2. ONLINE    7c54856e98474f61bf349401e7c9fb95 (/dev/sdc1) [DATA]
 3. ONLINE    9c46232b76234f61fc934673d5c8ec59 (/dev/sdd1) [DATA]
```

このリストは、すべての投票ファイルが破損していたり、投票ファイルにステータス3またはOFFとマーク付けされているエントリがリストに存在する可能性がある場合は、空であることがあります。

4. 投票ファイルの格納場所に応じて、次のいずれかの方法を実行します。

- 投票ファイルがOracle ASMに格納される場合は、次のコマンドを実行して、指定したOracle ASMディスク・グループに投票ファイルを移行します。

```
crsctl replace votedisk +asm_disk_group
```

投票ファイルの移行先のOracle ASMディスク・グループは、Oracle ASMに存在する必要があります。このコマンドは、投票ファイルがOracle ASMに格納されていても、他のストレージ・デバイスに格納されていても使用できます。

- Oracle ASMに投票ファイルを格納していなかった場合は、前のステップで取得したファイル・ユニバーサル識別子(FUID)を使用して、次のコマンドを実行します。

```
$ crsctl delete css votedisk FUID
```

次のコマンドを実行して、投票ファイルを追加します。

```
$ crsctl add css votedisk path_to_voting_disk
```

5. rootとしてOracle Clusterwareスタックを停止します。

```
# crsctl stop crs
```


ノート:



Oracle Clusterware スタックが排他モードで実行されている場合は、`-f` オプションを使用してスタックを強制的に停止します。

6. rootとしてOracle Clusterwareスタックを通常モードで再起動します。

```
# crsctl start crs
```

関連トピック

- [Oracle Automatic Storage Management管理者ガイド](#)

投票ファイルの追加、削除または移行

Oracle Clusterwareのインストール後に、投票ファイルを追加、削除および移行できます。これを行うために使用するコマンドは、投票ファイルがOracle ASMに存在するか、別の記憶域オプションに存在するかによって異なります。

Oracle ASMに格納されている投票ファイルの変更

- 投票ファイルのFUIDと現在の各投票ファイルのファイル・パスを表示するには、`crsctl query css votedisk` コマンドを実行して、次のような出力を表示します。

```
$ crsctl query css votedisk
--  -----
##  STATE      File Universal Id                        File Name Disk group
  1.  ONLINE    6f57843d89464c46ea747362e8a3fa43      (/dev/sdb1) [DATA]
  2.  ONLINE    7c54856e98474f61bf349401e7c9fb95      (/dev/sdc1) [DATA]
  3.  ONLINE    9c46232b76234f61fc934673d5c8ec59      (/dev/sdd1) [DATA]
```

このコマンドでは、ディスク順序番号、ディスクのステータス、FUID、ディスクのパス、およびディスクが格納されているOracle ASMディスク・グループの名前が戻されます。

- Oracle ASMから代替ストレージ・デバイスに投票ファイルを移行するには、次のコマンドを使用して、Oracle ASMディスク・グループと置き換えるOracle ASM以外のストレージ・デバイスまでのパスを指定します。

```
$ crsctl replace votedisk path_to_voting_disk
```

クラスタ内の任意のノードでこのコマンドを実行できます。

- Oracle ASMに格納されていないすべての投票ファイルを、Oracle ASMディスク・グループでOracle ASMにより管理される投票ファイルに置き換えるには、次のコマンドを実行します。

```
$ crsctl replace votedisk +asm_disk_group
```

Oracle ASMに格納されていない投票ファイルの変更

- 投票ファイルFUIDおよび現在の各投票ファイルのファイル・パスを表示するには、次のコマンドを実行します。

```
$ crsctl query css votedisk
##  STATE      File Universal Id                        File Name Disk group
--  -----
  1.  ONLINE    7c54856e98474f61bf349401e7c9fb95      (/cfs/host09_vd3) []
```

このコマンドでは、ディスク順序番号、ディスクのステータス、FUID、およびディスクのパスが戻され、Oracle ASMディスク・グループの名前は戻されません。

- 1つ以上の投票ファイルを追加するには、次のコマンドを実行します。path_to_voting_disk変数は、追加する投票ファイルの1つ以上の空白区切りの完全パスで置き換えます。

```
$ crsctl add css votedisk path_to_voting_disk [...]
```

- 投票ファイルAを投票ファイルBと置き換えるには、投票ファイルBを追加してから投票ファイルAを削除する必要があります。新しいディスクを追加して既存のディスクを削除するには、次のコマンドを実行します (path_to_voting_diskB変数は投票ファイルBの完全修飾パス名に置き換えます)。

```
$ crsctl add css votedisk path_to_voting_diskB -purge
```

-purgeオプションを使用すると、既存の投票ファイルが削除されます。

- 投票ファイルを削除するには、次のコマンドを実行します。1つ以上の空白区切りの投票ファイルFUIDまたは削除する投票ファイルのカンマ区切りのディレクトリ・パスを指定します。

```
$ crsctl delete css votedisk {FUID | path_to_voting_disk[...]}
```

ノート:

クラスタが停止し、失われた投票ファイルのために再起動できない場合は、root として次のコマンドを実行して CSS を排他モードで起動する必要があります。



```
# crsctl start crs -excl
```

CSS を排他モードで起動したら、次のコマンドを使用して投票ファイルを置換できます。

```
# crsctl replace votedisk path_to_voting_disk
```

Oracle ASMへの投票ファイルの移行

Oracle ASMに投票ファイルを移行するには、次のコマンドにOracle ASMディスク・グループ名を指定します。

```
$ crsctl replace votedisk +asm_disk_group
```

クラスタ内の任意のノードでこのコマンドを実行できます。

投票ファイルの場所の確認

投票ファイルの変更後、次のコマンドを使用して投票ファイルの場所を確認します。

```
$ crsctl query css votedisk
```

関連トピック

- [crsctl query css votedisk](#)

6 クラスタ・ノードの追加および削除

既存のクラスタにノードを追加する方法、およびクラスタからノードを削除する方法について説明します。

この章では、Linux、UNIXおよびWindowsシステムでこれらのタスクを実行する手順を説明しています。

ノート:



- 特に指定がないかぎり、Oracle Clusterware をインストールしたユーザーとしてノードの追加および削除のすべてのステップを実行します。
- フリート・パッチ適用およびプロビジョニングを使用して、クラスタ・ノードを追加および削除できます。
- クラスタを作成する場合は、「Oracle Clusterware のクローニング」に記載されているクローニング手順を使用することをお勧めします。

この章の内容は次のとおりです。

- [クラスタ・ノードを追加するための必須ステップ](#)
- [LinuxおよびUNIXシステムでのクラスタ・ノードの追加および削除](#)
- [Windowsシステムでのクラスタ・ノードの追加および削除](#)

関連トピック

- [Oracle Clusterwareのクローニング](#)

クラスタ・ノードを追加するための必須ステップ

ノート:



クラスタにノードを追加する前に、『Oracle Grid Infrastructure インストレーション・ガイド for Linux』に示す前処理を実行してください。

Oracle Clusterware はインストールしないでください。ソフトウェアはクラスタにノードを追加したときに既存のノードからコピーされます。

次のステップを実行して、クラスタに追加するノードを準備します。

1. 物理的な接続を確立します。

ノードのハードウェアをクラスタのネットワーク・インフラストラクチャに接続します。この作業には、電気的な接続の確立、ネットワーク・インターコネクトの構成、共有ディスク・サブシステム接続の構成などが含まれます。このステップの詳細は、ハードウェア・ベンダーのマニュアルを参照してください。

2. オペレーティング・システムをインストールします。

クラスタの他のノード上のオペレーティング・システムと一致するオペレーティング・システムのクローン・イメージをインストー

ルします。これには、必要なサービス・パッチ、更新およびドライバのインストールも含まれます。この手順の詳細は、オペレーティング・システム・ベンダーのマニュアルを参照してください。

ノート:



クローン・イメージを使用することをお勧めします。ただし、インストール要件が満たされている場合は、ベンダーのドキュメントに従ってオペレーティング・システムをインストールしてください。

3. Oracleユーザーを作成します。

既存ノードに存在すべてのOracleユーザーを新規ノードで作成する必要があります。たとえば、2つのノードを持つクラスタにノードを追加する際、これら2つのノードでOracle Grid InfrastructureホームとOracleホームに異なる所有者が存在する場合は、新規ノードにOracleホームをインストールする予定がなくても、新規ノードでこれらの所有者を作成する必要があります。

ノート:



このステップは、Linux および UNIX システムに対してのみ実行します。

rootとして、既存ノードと同じユーザーIDおよびグループIDを使用してOracleユーザーおよびグループを作成します。

4. SSHがノードに構成されていることを確認します。

ノート:



Oracle Clusterware をインストールすると、SSH が構成されます。SSH が構成されない場合は、SSH の構成について、『Oracle Grid Infrastructure インストレーション・ガイド』を参照してください。

5. クラスタ検証ユーティリティ(CVU)を使用して、ハードウェアおよびオペレーティング・システムのインストールを確認します。

追加対象ノードにハードウェアおよびオペレーティング・システムを構成した後、次のコマンドを実行して、クラスタ内の他のノードから追加対象ノードへの到達可能性を検査できます。また、このコマンドで、ローカル・ノードから指定されたすべてのノードへのユーザー等価関係、指定されたすべてのノード間のノード接続性、指定されたすべてのノードからの共有記憶域へのアクセス可能性などを検査することもできます。

- a. 既存ノードで、Grid_home/binディレクトリからCVUコマンドを実行して、参照ノードのプロパティと、現在のクラスタ環境に含まれるその他のすべてのノードのプロパティとの詳細な比較情報を取得します。ref_nodeを、CVUで追加対象ノードと比較する既存クラスタ内のノードの名前に置き換えます。-nオプションの後に、カンマ区切りのノードのリストを指定します。次の例では、orainventory_groupはOracle Inventoryグループの名前、osdba_groupはOSDBAグループの名前です。

```
$ cluvfy comp peer [-refnode ref_node] -n node_list  
[-orainv orainventory_group] [-osdba osdba_group] [-verbose]
```

- b. グリッド・インフラストラクチャ管理リポジトリに、前述の4つの手順で追加された各ノードに対して500MB以上の追加領域があることを確認します。

```
$ oclumon manage -get repsize
```

必要な場合は、次のように追加領域を追加します。

```
$ oclumon manage -repos changerepossize total_in_MB
```

関連項目:

OCLUMONの使用の詳細は、*Oracle Autonomous Health Framework* ユーザーズ・ガイドを参照してください。

ノート:



参照ノードについては、CVU による比較を行うクラスタ・ノードを選択します。たとえば、-n オプションで指定して追加するノードなどです。

この項の手順を完了すると、ノードをクラスタに追加できるようになります。

ノート:



Oracle Clusterware インストールの完了後は、ホスト名の変更(ドメイン修飾の追加または削除を含む)はしないでください。ホスト名を変更されたノードはクラスタから削除され、新しい名前を追加されます。

関連トピック

- [Oracle Grid Infrastructure インストレーションおよびアップグレード・ガイド for Linux](#)
- [Oracle Autonomous Health Framework ユーザーズ・ガイド](#)

Linux および UNIX システムでのクラスタ・ノードの追加および削除

Linux および UNIX システムでのクラスタ・ノードの追加および削除。

この項のノードの追加の手順は、「クラスタ・ノードを追加するための必須ステップ」の項のステップを完了していることを前提としています。

ノード追加プロセスの最後のステップには、既存のノード上の Oracle Clusterware ホームから追加対象ノードへの Oracle Clusterware ホームの拡張が含まれます。

この項には次のトピックが含まれます:

- [Linux および UNIX システムでのクラスタ・ノードの追加](#)
- [Linux および UNIX システムでのクラスタ・ノードの削除](#)

ノート:



Oracle Clusterware 11g リリース 2 (11.2) 以降では、ノードの追加時、Oracle Universal Installer はデフォルトでサイレント・モードになっています。

関連トピック

- [クラスタ・ノードを追加するための必須ステップ](#)

LinuxおよびUNIXシステムでのクラスタ・ノードの追加

ノードをクラスタに追加する場合は、使用できる方法が2つあります。

フリー・パッチ適用およびプロビジョニングを使用したノードの追加

フリー・パッチ適用およびプロビジョニング・サーバーがある場合は、次の例に示すように、フリー・パッチ適用およびプロビジョニングを使用して、1つのコマンドだけでクラスタにノードを追加できます。

```
$ rhpctl addnode gihome -client rhpclient -newnodes clientnode2:clientnode2-vip -root
```

前述の例では、VIP clientnode2-vipを含むclientnode2というノードを、ルート資格証明を使用して(追加するノードへのログイン)、rhpclientというフリー・パッチ適用およびプロビジョニング・クライアントに追加しています。

Oracle Grid Infrastructureインストーラを使用したノードの追加

クラスタへのノードの追加にフリー・パッチ適用およびプロビジョニングを使用しない場合は、Oracle Grid Infrastructureインストーラを使用してこのタスクを完了できます。

ノート:



\$Oracle_home/install/response/gridSetup.rsp テンプレートを使用すると、非対話型(サイレント・モード)操作の Oracle Grid Infrastructure インストーラを使用してノードを追加するためのレスポンス・ファイルを作成できます。

Oracle Grid Infrastructureインストーラを使用してノードをクラスタに追加するには、次の手順を実行します。

1. ./gridSetup.shを実行して、インストーラを起動します。
2. 「構成オプションの選択」ページで、「クラスタに、さらにノードを追加してください」を選択します。
3. 「クラスタ・ノード情報」ページで、「追加」をクリックして、追加するノードに関する情報を提供します。
4. 検証プロセスが「前提条件チェックの実行」ページで完了したら、サマリーを確認し、「インストール」をクリックします。
5. プロンプトに従い、oraInstRoot.shスクリプトをrootとして実行して、/etc/oraInst.locファイルに中央イベントリの場所を移入します。たとえば:

```
# /opt/oracle/oraInventory/oraInstRoot.sh
```

6. node3でrootとしてGrid_home/root.shスクリプトを実行し、指示に従って後続のスクリプトを実行します。

ノート:



- 前のステップで root.sh スクリプトを実行した場合は、再度実行する必要はありません。
- ポリシー管理データベースを使用している場合、必ず Oracle ホームを新しいノードにクローニングしてから root.sh スクリプトを実行してください。
- クラスタに追加するノード上で管理者管理データベース・インスタンスが構成されている場合は、root.sh スクリプトを実行する前に、Oracle ホームを新規ノードに拡張する必要があります。

または、srvctl remove instance コマンドを使用して管理者管理データベース・インスタ

ンスを削除します。

7. Oracle RACまたはOracle RAC One Nodeデータベースをクラスターで構成した場合で、ローカルのOracleホームを持つ場合は、次を実行してOracle Databaseホームをnode3に拡張します。

- a. node1のOracle_home/addnodeディレクトリに移動し、次の構文を使用してOracle RACをインストールしたユーザーとしてaddnode.shスクリプトを実行します。

```
$ ./addnode.sh "CLUSTER_NEW_NODES={node3}"
```

- b. rootとしてnode3でOracle_home/root.shスクリプトを実行します。Oracle_homeはOracle RACホームです。

- c. SQL*Plusセッションで次のコマンドを使用して、新しく追加したノードでプラグブル・データベース(PDB)をオープンします。

```
SQL> CONNECT / AS SYSDBA
SQL> ALTER PLUGGABLE DATABASE pdb_name OPEN;
```

Oracle Automatic Storage Management Cluster File System (Oracle ACFS)を使用して共有されている共有Oracleホームを持つ場合は、次の手順を実行し、Oracle Databaseホームをnode3に拡張します。

- d. rootとして、node3に対しGrid_home/root.shスクリプトを実行します。Grid_homeはOracle Grid Infrastructureホームです。

- e. Oracle RACをインストールしたユーザーとして、追加したノードのOracle_home/oui/binディレクトリから次のコマンドを実行し、Oracle RACデータベース・ホームを追加します。

```
$ ./runInstaller -attachHome ORACLE_HOME="ORACLE_HOME"
"CLUSTER_NODES={node3}"
LOCAL_NODE="node3" ORACLE_HOME_NAME="home_name" -cfs
```

- f. node1のOracle_home/addnodeディレクトリに移動し、次の構文を使用してOracle RACをインストールしたユーザーとしてaddnode.shスクリプトを実行します。

```
$ ./addnode.sh -noCopy "CLUSTER_NEW_NODES={node3}"
```

ノート:



宛先ノードのOracle ホームはすでにソフトウェアに完全に移入されているため、-noCopy オプションを使用します。

Oracle ACFS以外の共有ファイル・システムで共有Oracleホームを持つ場合は、まずターゲット・ノードでOracle RACデータベース・ホームのマウント・ポイントを作成し、Oracle RACデータベース・ホームをマウントおよびアタッチし、次のようにOracle Inventoryを更新します。

- g. クラスター内の既存ノードでsrvctl config database -db db_nameコマンドを実行して、マウント・ポイント情報を取得します。

- h. 次のコマンドをrootとしてnode3で実行し、マウント・ポイントを作成します。

```
# mkdir -p mount_point_path
```

- i. Oracle RACデータベース・ホームをホストするファイル・システムをマウントします。

j. Oracle RACをインストールしたユーザーとして、追加したノードのOracle_home/oui/binディレクトリから次のコマンドを実行し、Oracle RACデータベース・ホームを追加します。

```
$ ./runInstaller -attachHome ORACLE_HOME="ORACLE_HOME" "CLUSTER  
_NODES={local_node_name}" LOCAL_NODE="node_name"  
ORACLE_HOME_NAME="home_name" -cfs
```

node1のOracle_home/addnodeディレクトリに移動し、次の構文を使用してOracle RACをインストールしたユーザーとしてaddnode.shスクリプトを実行します。

```
$ ./addnode.sh -noCopy "CLUSTER_NEW_NODES={node3}"
```

ノート:



宛先ノードの Oracle ホームはすでにソフトウェアに完全に移入されているため、-noCopy オプションを使用します。

ノート:



addnode.sh の実行後、Grid_home/network/admin/samples ディレクトリの権限が 750 に設定されていることを確認します。

8. rootとしてGrid_home/binディレクトリから次のコマンドを実行して、新しいノードでOracle ACFSリソースを起動します。

```
# srvctl start filesystem -device volume_device_name -node node3
```

ノート:



Oracle ホームが格納されている Oracle ACFS レジストリ・リソースおよび Oracle ACFS ファイル・システム・リソースを含む Oracle ACFS リソースが、新しく追加されたノードでオンラインであることを確認します。

9. Oracle Clusterwareをインストールしたユーザーとして次のCVUコマンドを実行して、クラスタの整合性をチェックします。このコマンドで、指定した任意の数のノードがネットワーク、共有記憶域およびクラスタウェアの各レベルでクラスタに正常に追加されていることを確認します。

```
$ cluvfy stage -post nodeadd -n node3 [-verbose]
```

関連トピック

- [Oracle Grid Infrastructureのインストール・ガイド](#)
- [cluvfy stage nodeadd](#)
- [Oracle Real Application Clusters管理およびデプロイメント・ガイド](#)
- [Oracle Real Application Clusters管理およびデプロイメント・ガイド](#)

LinuxおよびUNIXシステムでのクラスタ・ノードの削除

LinuxおよびUNIXシステムでのクラスタ・ノードの削除。

ノート:

- GNS が提供するクラスタの最後のノードを削除する場合、GNS からそのクラスタのエントリを削除する必要があります。
- 固定解除されているクラスタにノードがある場合、しばらくすると Oracle Clusterware はそれらのノードを無視するため、ユーザーが削除する必要はありません。
- (特定のノードでサービスを無効にしたり、サーバー・プールの候補リストにノードを追加するなどして)あるノードにノード固有の構成を作成すると、ノードがクラスタから削除されても、ノード固有の構成は削除されません。このようなノード固有の構成は、手動で削除する必要があります。
- クラスタに変更を行うと、投票ファイルは自動的に OCR にバックアップされます。
- Oracle Flex Cluster からハブではないノードを削除する場合は、次の手順のステップ 1 から 4 のみを実行します。

クラスタからノードを削除するには:

1. Grid_homeが各ノード上のOracle Clusterwareホームへの完全なディレクトリ・パスを正しく指定していることを確認します。Grid_homeは、Oracle Clusterwareソフトウェアをインストールした場所です。
2. rootとしてまたはOracle Clusterwareをインストールしたユーザーとして次のコマンドを実行し、削除対象のノードがアクティブであるかどうか、および固定されているかどうかを確認します。

```
$ olsnodes -s -t
```

ノードが固定されている場合は、`crsctl unpin css`コマンドを実行します。それ以外の場合は、次のステップに進みます。

3. 削除するノードで、実行中のすべてのOracle Databaseインスタンスを停止します。

```
$ srvctl stop instance -n node_to_be_deleted
```

4. 削除対象のノードで、共有Oracleホームを持つかローカルOracleホームを持つかに応じて、Oracle Clusterwareをインストールしたユーザーとして、次のいずれかの手順を実行します。

- ローカル・ホームの場合は、次のコマンドを実行し、削除対象のノードからOracle Clusterwareホームを削除します。Grid_homeは、Oracle Clusterwareホームに定義されているパスです。

```
$ Grid_home/deinstall/deinstall -local
```

注意:



- -local フラグを指定しない場合は、コマンドによって、クラスタ内のすべてのノードから Oracle Grid Infrastructure ホームが削除されます。
- 前述のコマンドラインをカット・アンド・ペーストする場合は、このドキュメントに含まれている書式設定を削除するために、テキスト・エディタにペーストしてからコマンドラインにペースト

してください。

ノート:



または、Oracle Grid Infrastructure を構成した後でグリッド・インフラストラクチャからノードを削除する場合、Grid_home/gridSetup.sh を実行し、「クラスタからのノードの削除」を選択してプロンプトに従えば削除できます。

- 共有ホームがある場合は、削除するノードで次のコマンドを次の順序で実行します。

Oracle Clusterwareを構成解除するには、次のコマンドを実行します。

```
$ Grid_home/crs/install/rootcrs.sh -deconfig -force
```

Gridホームをデタッチするには、次のコマンドをGrid_home/oui/binディレクトリから実行します。

```
$ ./runInstaller -detachHome ORACLE_HOME=Grid_home -silent -local
```

インストール・ユーティリティの指示に従って、構成ファイルがあれば手動で削除します。

5. 削除対象ではない任意のノードから、rootとしてGrid_home/binディレクトリで次のコマンドを実行して、クラスタからノードを削除します。

```
# crsctl delete node -n node_to_be_deleted
```

6. 指定したノードがクラスタから正常に削除されたことを確認するには、次のCVUコマンドを実行します。

```
$ cluvfy stage -post nodedel -n node_list [-verbose]
```

7. Oracle Clusterwareが停止しているクラスタ・ノードを削除した場合、次のようにして、削除したノードのVIPが削除されずに残っていないかどうか確認します。

```
$ srvctl config vip -node deleted_node_name
```

VIPがまだ残っている場合、次のようにして削除します。

```
$ srvctl stop vip -vip vip_name  
$ srvctl remove vip -vip vip_name
```

フリー・パッチ適用およびプロビジョニングを使用したノードの削除

フリー・パッチ適用およびプロビジョニングを使用して、次の例に示す1つのコマンドでクラスタからノードを削除することもできます。

```
$ rhpctl deletenode gihome -client rhpcclient -node clientnode2 -root
```

前述の例では、clientnode2というノードを、ルート資格証明を使用して(削除するノードへのログイン)、rhpcclientというフリー・パッチ適用およびプロビジョニング・クライアントから削除しています。

関連トピック

- [Oracle Real Application Clusters管理およびデプロイメント・ガイド](#)
- [cluvfy stage nodedel](#)

Windowsシステムでのクラスタ・ノードの追加および削除

この項では、Windowsシステムでのクラスタ・ノードの追加および削除について説明します。この項には次のトピックが含まれます。

す:

- [Windowsシステムでのクラスタへのノードの追加](#)
- [Windowsシステムでのクラスタ・ノードの削除](#)

関連項目:

クラスタ全体を削除する方法の詳細は、『Oracle Grid Infrastructureインストレーションおよびアップグレード・ガイドfor Microsoft Windows』を参照してください

Windowsシステムでのクラスタへのノードの追加

ノードを追加する前に、「クラスタ・ノードを追加するための必須ステップ」にリストされている前提条件を満たしておく必要があります。

この手順では、クラスタにノードを追加する方法について説明します。この手順では、次の項目を前提としています。

- node1およびnode2という2つのノードを持つクラスタが存在する
- node3という名前のノードを追加する
- Oracle Clusterwareがnode1およびnode2のローカル・ホームにインストールされており、Grid_homeは正常にインストールされたホームを表す

ノート:



Windows システムで Oracle Database が Oracle Database 10g リリース 1(10.1)からアップグレードされている構成の場合は、クラスタ・ノードの追加に、この項で説明する手順を使用しないでください。

ノードを追加するには:

1. クラスタおよびnode3の整合性を検証します。

```
C:¥>cluvfy stage -pre nodeadd -n node3 [-fixup] [-verbose]
```

-fixupオプションと、検証が失敗した場合にクラスタまたはノードを修正するための指示がCVUによって出力されるディレクトリを指定できます。

2. node1でGrid_home¥addnodeディレクトリに移動し、addnode.batスクリプトを次のように実行します。

```
C:¥>addnode.bat "CLUSTER_NEW_NODES={node3}"  
"CLUSTER_NEW_VIRTUAL_HOSTNAMES={node3-vip}"
```

3. 新規ノードで次のコマンドを実行します。

```
C:¥>Grid_home¥crs¥config¥gridconfig.bat
```

4. Oracle ACFSを使用するようにデータベース・ホームを構成した場合にのみ、次のステップを実行する必要があります。

- a. Oracle ACFSを使用するように構成した各データベースに対して、Oracle RACデータベース・ホームから次のコマンドを実行します。

```
C:¥>ORACLE_HOME¥bin¥srvctl stop database -db database_unique_name
```

ノート:

Oracle Clusterware で構成されているすべてのデータベースをリストするには、`srvctl config database` コマンドを実行します。データベースの詳細を検索するには、`srvctl config database -db database_unique_name` を使用します。ORACLE_HOME のパスが Oracle ACFS のマウント・パスにつながる場合、データベースは Oracle ACFS を使用します。このコマンドの出力から、新しく追加されたノードで実行するように構成されたデータベース・インスタンス名を見つけます。

b. Windows Server Manager Controlを使用して、サービスを停止して削除します。

c. このステップの最初の部分で収集したデータベースおよびデータベース・ホームのそれぞれに対して、次のコマンドを実行します。

```
C:\> ORACLE_HOME\bin\srvctl start database -db database_unique_name
```

5. 次のコマンドを実行し、構成済のすべてのノード(既存のノードと追加したノードの両方)でOracle Clusterwareコンポーネントの整合性を検証します。

```
C:\>cluvfy stage -post crsinst -n all [-verbose]
```

この項のノードの追加手順を完了してから、必要に応じてOracle RACコンポーネントを含むOracle Databaseを新規ノードに拡張し、新規ノードを既存のOracle RACデータベースのメンバーにすることも可能です。

関連項目:

Oracle RACを含むOracle Databaseの新規ノードへの拡張の詳細は、『*Oracle Real Application Clusters管理およびデプロイメント・ガイド*』を参照してください。

OraMTS Service for Microsoft Transaction Serverの作成

Oracle Services for Microsoft Transaction Server (OraMTS)を使用すると、Microsoftアプリケーションで調整されるトランザクション内で、リソース・マネージャとしてOracle Databaseを使用できます。OraMTSは、Microsoft分散トランザクション・コーディネータ(MSDTC)に対するOracle Databaseのプロキシとして機能します。この結果、OraMTSによってクライアント側の接続プールが提供され、Oracleを利用するクライアント・コンポーネントを昇格可能なトランザクションおよび分散トランザクションに使用できるようになります。また、サービス自体がWindowsで実行される場合、OraMTSは、任意のオペレーティング・システム上で実行されているOracle Databaseと連携して動作できます。

Oracle Database 12cより前のリリースでは、OraMTSはソフトウェアのみのインストールの一部として作成されました。Oracle Database 12c以上では、構成ツールを使用してこのサービスを作成する必要があります。

ノードの追加後、またはOracle RACのソフトウェアのみのインストールを実行した後でOraMTSサービスを作成するには、次の手順を実行します。

1. コマンド・ウィンドウを開きます。
2. ディレクトリを%ORACLE_HOME%\binに変更します。
3. OraMTSctlユーティリティを実行してOraMTSサービスを作成します(ここで、host_nameはサービスが作成されるノードのリストです)。

```
C:\¥..bin> oramtscctl.exe -new -host host_name
```

関連トピック

- [クラスタ・ノードを追加するための必須ステップ](#)
- [Oracle Real Application Clusters管理およびデプロイメント・ガイド](#)
- [Microsoft Windows用Oracle Services for Microsoft Transaction Server開発者ガイド](#)

関連項目:

分散されるトランザクション内で、リソース・マネージャとしてOracle Databaseを使用できるOraMTSの詳細は、『*Oracle Services for Microsoft Transaction Server開発者ガイドfor Microsoft Windows*』を参照してください。

Windowsシステムでのクラスタ・ノードの削除

Windowsシステムでのクラスタ・ノードの削除。

この手順は、Oracle Clusterwareがnode1、node2およびnode3にインストールされており、クラスタからnode3を削除すると想定しています。

ノート:

- Oracle では、Windows システムで Oracle Enterprise Manager を使用してノードを削除することはサポートしていません。
- GNS が提供するクラスタの最後のノードを削除する場合、GNS からそのクラスタのエントリを削除する必要があります。
- クラスタからノードを削除する前にノードから Oracle RAC データベース・インスタンスを削除することができますが、このステップは必須ではありません。インスタンスを削除しない場合、インスタンスは構成されたままですが、実行されることはありません。クラスタからノードを削除しても、クラスタからノードの構成情報が削除されることはありません。残った構成情報がクラスタの操作を妨害することはありません。

関連項目: Oracle RAC データベース・インスタンスの削除の詳細は、『*Oracle Real Application Clusters 管理およびデプロイメント・ガイド*』を参照してください。

Windowsシステムでクラスタ・ノードを削除するには:

1. ローカル・ホームを持つ場合のみ、削除対象のノードで次のコマンドを -local オプションで実行し、ノード・リストを更新します。

```
C:\¥>Grid_home¥oui¥bin¥setup.exe -updateNodeList ORACLE_HOME=Grid_home  
"CLUSTER_NODES=  
{node_to_be_deleted}" CRS=TRUE -local
```

ノート:

Oracle Flex Cluster からハブではないノードを削除する場合、このコマンドを実行する必要はありま

せん。

2. 次のように、削除するノードで削除ツールを実行して、Oracle Clusterwareホームを削除して構成解除します。

```
C:¥Grid_home¥deinstall¥>deinstall.bat -local
```

注意:



- -local フラグを指定しない場合は、コマンドによって、クラスタ内のすべてのノードから Oracle Grid Infrastructure ホームが削除されます。
- 前述のコマンドラインをカット・アンド・ペーストする場合は、このドキュメントに含まれている書式設定を削除するために、テキスト・エディタにペーストしてからコマンドラインにペーストしてください。

3. 削除対象ではないノードで、次のコマンドを実行します。

```
C:¥>Grid_home¥bin¥crsctl delete node -n node_to_be_deleted
```

4. 指定したノードがクラスタから正常に削除されたことを確認するには、次のCVUコマンドを実行します。

```
C:¥>cluvfy stage -post nodedel -n node_list [-verbose]
```

5. Oracle Clusterwareが停止しているクラスタ・ノードを削除した場合、次のようにして、削除したノードのVIPが削除されずに残っていないかどうか確認します。

```
C:¥> ORACLE_HOME¥bin¥srvctl config vip -node deleted_node_name
```

VIPがまだ残っている場合、次のようにして削除します。

```
C:¥> ORACLE_HOME¥bin¥srvctl stop vip -vip vip_name  
C:¥> ORACLE_HOME¥bin¥srvctl remove vip -vip vip_name
```

関連トピック

- [Oracle Real Application Clusters管理およびデプロイメント・ガイド](#)
- [cluvfy stage nodedel](#)

7 Oracle Clusterwareのクローニング

この章では、Oracle Grid Infrastructureホームをクローニングする方法およびクローニングされたホームを使用してクラスタを作成する方法について説明します。この章のクローニング手順を実行するには、サイレント・モードでスクリプトを使用します。クローニング手順は、LinuxおよびUNIXシステムに適用できます。この章の例ではLinuxおよびUNIXのコマンドを使用しますが、通常、クローニングの概念および手順はすべてのプラットフォームに適用されます。

ノート:

この章は、次のように構成された Oracle Clusterware 19c インストールをクローニングすることを前提としています。



- グリッド・ネーミング・サービス(GNS)なし
- Intelligent Platform Management Interface 仕様(IPMI)なし
- 投票ファイルおよび Oracle Cluster Registry(OCR)が Oracle Automatic Storage Management(Oracle ASM)に格納されている
- 単一クライアント・アクセス名(SCAN)は DNS で解決される

この章のトピックは、次のとおりです:

- [Oracle Clusterwareのクローニングの概要](#)
- [Oracle Grid Infrastructureホームをクローニングするための準備](#)
- [Oracle Clusterwareのクローニングによるクラスタの作成](#)
- [クローニングを使用したクラスタへのノードの追加](#)
- [クローニング時に生成されたログ・ファイルの検索および表示](#)

Oracle Clusterwareのクローニングの概要

クローニングとは、既存のOracle Clusterwareインストールを別の場所にコピーし、その後、コピーしたインストールを新しい環境で動作するよう更新するプロセスです。クローニング元のOracle Grid Infrastructureホームに適用済の個別パッチによる変更は、クローニング後も継承されます。クローニング中には、Oracle Grid Infrastructureホームのインストールで実行された操作を再実行するスクリプトを実行します。

クローニングには、正常にインストールされたOracle Grid Infrastructureホームを使用することが必要です。このホームを、Oracle Grid Infrastructureホームを拡張するスクリプト実装のベースにして、元のGridホームに基づくクラスタを作成します。

手動でのクローニング・スクリプトの作成では、入力を検証する対話形式のチェックを使用せずにスクリプトを準備するため、エラーが発生しやすくなる場合があります。ただし、数十または数百のクラスタに対して単一スクリプトを実行するような場合には、この最初の作業は意味があります。インストールするクラスタが1つのみの場合は、Oracle Universal Installer(OUI)またはOracle Enterprise ManagerのProvisioning Pack機能など、従来の自動化された対話形式のインストール方法を使用してください。

ノート:



クローニングは、Provisioning Packに含まれる Oracle Enterprise Manager クローニングにかわるものではありません。Oracle Enterprise Manager によるクローニングでは、プロビジョニング・プロセスによって Oracle ホームの詳細が対話形式で確認されることで、クローニングが単純化されています。対話時の質問では、クローニングされた環境のデプロイ先、Oracle Database ホームの名前、クラスタ内のノードのリストなどが確認されます。

Oracle Enterprise Manager Grid ControlのProvisioning Pack機能では、ノードおよびクラスタのプロビジョニングを自動化するフレームワークが提供されています。多数のクラスタを持つデータ・センターの場合、既存のクラスタに新規クラスタおよび新規ノードをプロビジョニングするためのクローニング手順を作成する作業は、その労力に見合った価値があります。

次のリストに、クローニングが有効ないくつかの状況を示します。

- クローニングにより、Oracle Grid Infrastructureホームを1回準備するだけで、多数のホストに同時にデプロイされます。非対話形式のプロセスとして、サイレント・モードでインストールを完了できます。グラフィカル・ユーザー・インタフェース(GUI)のコンソールを使用する必要がなく、必要に応じて、セキュア・シェル(SSH)・ターミナル・セッションからクローニングを実行できます。
- クローニングにより、すべてのパッチが適用されたインストール(本番、テスト、または開発用インストールのコピー)を1回のステップで作成できます。ベース・インストールを実行してソース・システムにすべてのパッチ・セットとパッチを適用した後は、クローニングにより、それらの個別のステップはすべて単一の手順で実行されます。これにより、クラスタの各ノードですべてのインストール・プロセスを通して行い、インストール、構成およびパッチ適用を個別のステップで実行するのとは、対照的な効果が得られます。
- クローニングによるOracle Clusterwareのインストールでは、非常に短時間で処理を完了できます。たとえば、3つ以上のノードを持つクラスタにOracle Grid Infrastructureホームをクローニングする場合、必要な時間は、Oracleソフトウェアのインストールにかかる数分間と各ノードにかかる数分間を合わせた時間(合計時間はroot.shスクリプトの実行にかかる時間とほぼ等しい)となります。
- クローニングでは、複数のクラスタで同一のOracle Clusterwareインストールを正確に繰り返すための確実な方法が提供されます。

クローニングされたインストールは、クローニング元のインストールと同じ動作をします。たとえば、クローニングされたOracle Grid Infrastructureホームは、OUIを使用して削除したり、OPatchを使用してパッチを適用することができます。また、クローニングされたOracle Grid Infrastructureホームを、別のクローニング操作のソースとして使用できます。コマンドラインのクローニング・スクリプトを使用して、テスト用、開発用または本番用のインストールをクローニングしたコピーを作成できます。

多くの場合、デフォルトのクローニング手順で十分です。ただし、カスタム・ポート割当ての指定、またはカスタム設定の保持など、クローニングのいくつかの側面をカスタマイズすることもできます。

たとえば、リスナーのカスタム・ポートを、次のように指定できます。

```
$ export ORACLE_HOME=/u01/app/12.1.0/grid
$ $ORACLE_HOME/bin/srvctl modify listener -endpoints tcp:12345
```

クローニング・プロセスでは、クローニング元のOracle Grid Infrastructureホームからクローニング先のOracle Grid Infrastructureホームにすべてのファイルをコピーします。Oracle Grid Infrastructureホームは、ローカル(非共有)ホームまたは共有ホームのどちらの場合でもクローニングできます。したがって、ソース・インスタンスで使用されているファイルの中で、クローニング元のOracle Grid Infrastructureホームのディレクトリ構造の外部に置かれているものがある場合、それはクローニ

ング先にコピーされません。

ソースおよび宛先のバイナリ・ファイルのサイズは異なる場合がありますが、これはクローニング操作の一部としてこれらのファイルが再リンクされており、これら2つの位置のオペレーティング・システムのパッチ・レベルが異なっていることがあるためです。また、いくつかのファイルがソースからコピーされることによって、特にインスタンス化されたファイルはクローン操作の一部としてバックアップされるため、クローン・ホームのファイルの数が増える場合があります。

Oracle Grid Infrastructureホームをクローニングするための準備

クローニング元のOracle Grid Infrastructureホームでクローニングを準備するには、インストール済のOracle Grid Infrastructureホームのコピーを作成し、別のノードでクローニング手順を実行する際に使用します。次の段階ごとの手順に従って、Oracle Grid Infrastructureホームのコピーを準備します。

- [ステップ1: Oracle Clusterwareのインストール](#)
- [ステップ2: 実行中のソフトウェアの停止](#)
- [ステップ3: Oracle Grid Infrastructureホームのコピーの作成](#)

ステップ1: Oracle Clusterwareのインストール

『Oracle Grid Infrastructureインストレーションおよびアップグレード・ガイド』の詳細指示に従って、ソース・ノードで、次のステップを実行します。

1. Oracle Clusterwareをインストールします。このインストールによって、Oracle Cluster Registry(OCR)および投票ファイルがOracle Automatic Storage Management(Oracle ASM)に配置されます。

ノート:



クラスタ用の Oracle Grid Infrastructure をインストールして構成するか、Oracle Clusterware ソフトウェアのみをインストールしてください(ご使用のプラットフォーム固有の *Oracle Grid Infrastructure* インストレーションおよびアップグレード・ガイドを参照)。

クラスタ用のOracle Grid Infrastructureをインストールして構成した場合は、クローニング手順を実行する前にOracle Clusterwareを停止する必要があります。ソフトウェアのみのインストールを実行する場合は、Oracle Clusterwareを停止する必要はありません。

2. 必要に応じて、適切なパッチ(Oracle Grid Infrastructureバンドル・パッチなど)をインストールします。
3. 必要に応じて、1回限りのパッチを適用します。

関連トピック

- [Oracle Grid Infrastructureインストレーションおよびアップグレード・ガイドfor Linux](#)

ステップ2: 実行中のソフトウェアの停止

クローニング元のOracle Grid Infrastructureホームをコピーする前に、ノードで実行されているサービス、データベース、リスナー、アプリケーション、Oracle ClusterwareおよびOracle ASMインスタンスをすべて停止します。サーバー制御(SRVCTL)ユーティリティを使用してデータベースを停止してから、Oracle Clusterware制御(CRSCTL)ユーティリティを使用して残りのコンポーネントを停止することをお勧めします。

関連トピック

- [Oracle Real Application Clusters管理およびデプロイメント・ガイド](#)
- [Oracle Clusterware制御\(CRSCTL\)ユーティリティ・リファレンス](#)

ステップ3: Oracle Grid Infrastructureホームのコピーの作成

インストールしたOracle Grid Infrastructureホームを作業ホームとして保持するために、クローニング元のOracle Grid Infrastructureホームの全体コピーを作成します。

ヒント:



コピーの作成時には、ファイル名にリリース番号を含めることをお勧めします。

次のいずれかの方法を使用して、Oracle Grid Infrastructureホームの圧縮コピーを作成できます。

- [方法1: 設定ウィザードを使用してOracle Grid Infrastructureホームのコピーを作成する](#)
- [方法2a: そのコピーから不要なファイルを削除することによりOracle Grid Infrastructureホームのコピーを作成する](#)
- [方法2b: -Xオプションを使用してOracle Grid Infrastructureホームのコピーを作成する](#)

方法1が、Oracle Grid Infrastructureホームのゴールド・イメージを作成するためのお勧めの方法です。

Oracle Grid Infrastructureホームのコピーを手動で作成するには方法2aまたは2bを使用できます(Grid_homeはすべてのファイルを含む元のノードの元のOracle Grid Infrastructureホームであり、copy_pathは不要なファイルが削除されるコピーされたOracle Grid Infrastructureホームへのディレクトリ・パスです)。

方法1: 設定ウィザードを使用してOracle Grid Infrastructureホームのコピーを作成する

グリッド設定ウィザードを使用してOracle Grid Infrastructureホームのゴールド・イメージを作成できます。

1. gridSetup.shを適切なオプションを使用して実行してゴールド・イメージを作成します。
たとえば、指定した宛先にゴールド・イメージを作成します。

```
./gridSetup.sh -createGoldImage -destinationLocation /tmp/my_grid_images
```

2. 詳細を表示するには、-helpオプションを使用してgridSetup.shを実行します。

関連項目:

イメージを作成するためのグリッド設定ウィザードのインストール・オプションの使用の詳細は、[『Oracle Grid Infrastructureインストールおよびアップグレード・ガイド』](#)を参照してください。

方法2a: そのコピーから不要なファイルを削除することによりOracle Grid Infrastructureホームのコピーを作成する

1. ソース・ノードで、Oracle Grid Infrastructureホームのコピーを作成します。インストールしたOracle Grid Infrastructureホームを作業ホームとして保持するために、クローニング元のOracle Grid Infrastructureホームの全体コピーを作成し、そのコピーから不要なファイルを削除します。たとえば、Linuxシステムでは、rootユーザーとしてcpコマンドを実行します。

```
# cp -prf Grid_home copy_path
```

2. コピーから不要なファイルを削除します。

Oracle Grid Infrastructureホームには、ソース・ノードにのみ関連するファイルが含まれているため、log、crs/init、crfおよびcdataディレクトリのOracle Grid Infrastructureホームのコピーから不要なファイルを削除できます。次に、LinuxおよびUNIXシステムでOracle Grid Infrastructureホームのコピーから不要なファイルを削除する場合に実行するコマンドの例を示します。

```
[root@node1 root]# cd copy_path
[root@node1 grid]# rm -rf log/host_name
[root@node1 grid]# rm -rf gnpn/host_name
[root@node1 grid]# find gnpn -type f -exec rm -f {} \;;
[root@node1 grid]# rm -rf cfgtoollogs/*
[root@node1 grid]# rm -rf cdata/*
[root@node1 grid]# rm -rf crf/*
[root@node1 grid]# rm -rf network/admin/*.ora
[root@node1 grid]# rm -rf crs/install/crsconfig_params
[root@node1 grid]# rm -rf crs/install/crsngenconfig_params
[root@node1 grid]# find . -name '*.ouibak' -exec rm {} \;;
[root@node1 grid]# find . -name '*.ouibak.1' -exec rm {} \;;
[root@node1 grid]# rm -rf root.sh*
[root@node1 grid]# rm -rf bin/clsecho/*
[root@node1 grid]# rm -rf rdbms/audit/*
[root@node1 grid]# rm -rf rdbms/log/*
[root@node1 grid]# rm -rf inventory/backup/*
[root@node1 grid]# rm -rf oraInst.loc
```

3. LinuxおよびUNIXシステムではtarまたはgzipを使用して、前述のコピー済Oracle Grid Infrastructureホームの圧縮コピーを作成します。使用するツールによって、権限およびファイル・タイムスタンプが保持されることを確認してください。たとえば:

LinuxおよびUNIXシステムの場合:

```
[root@node1 root]# cd copy_path
[root@node1 grid]# tar -zcvpf /copy_path/gridHome.tgz
```

前述の例で、cdコマンドは、この手順の最初の2つのステップで作成したOracle Grid Infrastructureのホームのコピー(不要なファイルは削除済み)に場所を変更し、tarコマンドはgridHome.tgzという名前のファイルを作成します。tarコマンドで、copy_pathは、Oracle Grid Infrastructureホームのコピーの場所を表します。

AIXまたはHPUNIXシステムの場合は、次のコマンドを発行します。

```
uncompress gridHome.tar.Z
tar xvf gridHome.tar
```

Windowsシステムで、WinZipを使用してzipファイルを作成します。

方法2b: -Xオプションを使用してOracle Grid Infrastructureホームのコピーを作成する

1. Oracle Grid Infrastructureホーム内の不要なファイルを示すファイルを作成します。たとえば、excludeFileListというファイルに、アスタリスク(*)のワイルドカードを使用して次のファイル名を示します。

```
Grid_home/host_name
Grid_home/log/host_name
Grid_home/gnpn/host_name/*
Grid_home/bin/clsecho/*
Grid_home/cdata/*
Grid_home/crf/*
Grid_home/network/admin/*.ora
```

```
Grid_home/root.sh*
Grid_home/cfgtoollogs/*
Grid_home/crs/install/crsconfig_params
Grid_home/crs/install/crsngenconfig_params
Grid_home/rdbms/audit/*
Grid_home/rdbms/log/*
Grid_home/inventory/backup/*
Grid_home/oraInst.loc
*.ouibak
*.ouibak1
```

2. tarコマンドまたはWinzipを使用して、Oracle Grid Infrastructureホームの圧縮コピーを作成します。たとえば、LinuxおよびUNIXシステムでは、次のコマンドを実行して、クローニング元のOracle Grid Infrastructureホームをアーカイブおよび圧縮します。

```
tar cpfX - excludeFileList Grid_home | compress -fv > temp_dir/gridHome.tar.Z
```



ノート:

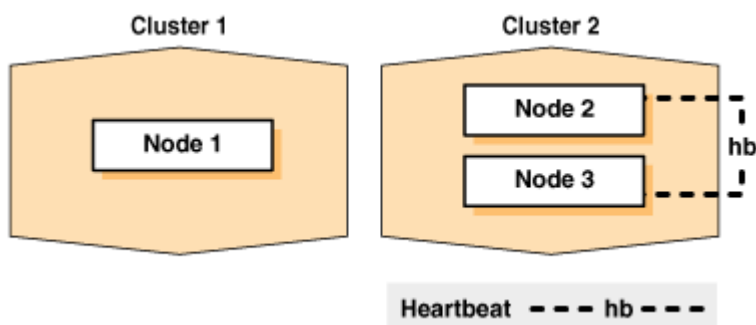
Oracle Grid Infrastructure ホームをコピーおよび圧縮するために、jar ユーティリティは使用しないでください。

Oracle Clusterwareのクローニングによるクラスタの作成

この項では、正常にインストールされたOracle Clusterware環境をクローニングし、それをクローニング先のクラスタ上のノードにコピーすることにより、クラスタを作成する方法を説明します。この項の手順では、Linux、UNIXおよびWindowsシステムでクローニングを使用する方法について説明します。クローニングでクラスタを正常に作成した後、OCRおよび投票ファイルは2つのクラスタ間で共有されません。

たとえば、クローニングを使用して、正常にインストールされたOracle Clusterware環境を複製し、クラスタを作成することができます。[図7-1](#)では、クローニング手順の結果を示しており、ノード1上のOracle Grid Infrastructureホームはクラスタ2上のノード2およびノード3にクローニングされ、クラスタ2が新規の2ノード・クラスタとなっています。

図7-1 クローニングによるOracle Clusterware環境の作成



クローニングでクラスタを作成するステップは次のとおりです。

- [ステップ1: 新規クラスタ・ノードの準備](#)
- [ステップ2: Oracle Grid Infrastructureホームのデプロイ](#)
- [ステップ3: gridSetup.shユーティリティの実行](#)

ステップ1: 新規クラスタ・ノードの準備

Oracle Clusterwareをクローニングする前に、インストール前のステップを完了する必要があります。

各宛先ノードで、次のインストール前のステップを実行します。

1. カーネル・パラメータの指定
2. Oracle Clusterwareデバイス用のブロック・デバイスの構成
3. ブロック・デバイスの権限が正しく設定されていることの確認
4. /etc/hostsファイル内のすべての名前に対する短い非ドメイン修飾の名前の使用
5. pingコマンドによる、インターコネクト・インタフェースが到達可能かどうかのテスト
6. pingコマンドによる、クローニング・プロセスの開始時に仮想IP (VIP)アドレスがアクティブでないことの検証(VIPアドレスのpingコマンドは失敗する必要があります)
7. (ベンダーのクラスタウェアを実行しているIBM AIXシステムおよびOracle Solaris x86-64-bitシステムのみ)クラスタにノードを追加する場合は、クラスタにノードを追加する前に、ノードでrootpre.shスクリプト(DVDからOracle Clusterwareをインストールした場合はマウント・ポイント、ソフトウェアをダウンロードした場合はtarファイルを解凍したディレクトリにあります)を実行する必要があります

ノート:



Oracle Database 19c 以降では、Oracle Clusterware とベンダー・クラスタウェアの統合は非推奨であり、将来のリリースでサポートされなくなる可能性があります。このため、次のソフトウェアまたはハードウェアのアップグレードでは、ベンダー・クラスタ・ソリューションを使用しないように移行することをお勧めします。

8. ハードウェアおよびオペレーティング・システム環境が正しく構成されていることを確認するには、クラスタ検証ユーティリティ (CVU)を実行します

ノート:



続行する前に、CVU を使用してクラスタをチェックしてください。従来のインストール方法とは異なり、クローニング・プロセスでは、準備フェーズ中に入力検証されません。これと比較して、Oracle Universal Installer (OUI)を使用した従来のインストール方法では、インタビュー・フェーズ中に様々なチェックが実行されます。したがって、ハードウェア設定または準備フェーズ中にミスをすると、クローニング・インストールは失敗します。

インストール前の全チェックリストは、ご使用のプラットフォーム固有のOracle Clusterwareインストレーション・ガイドを参照してください。

ステップ2: Oracle Grid Infrastructureホームのデプロイ

この項で説明しているクローニング手順を開始する前に、Oracle Grid Infrastructureホームのコピーを作成するための必須タスクを完了していることを確認してください(「Oracle Grid Infrastructureホームをクローニングするための準備」を参照)。

1. クラスタ内のノードの1つに、ステップ3: Oracle Grid Infrastructureホームのコピーの作成で作成したOracle Grid Infrastructureホームのコピーを次のようにデプロイします。

たとえば、LinuxまたはUNIXシステムで、次のようなコマンドを実行します。

```
[root@node1 root]# mkdir -p location_of_the_copy_of_the_Grid_home
[root@node1 root]# cd location_of_the_copy_of_the_Grid_home
[root@node1 crs]# tar -zxvf /gridHome.tgz
```

この例で、location_of_the_copy_of_the_Grid_homeは、/u01/app/12.1.0/gridなどのOracle Grid Infrastructureホームのインストール先のディレクトリ構造を表します。このプロセスでは、Gridホームの場所を変更できます。

2. Oracle Grid Infrastructureホームから不要なファイルを削除していない場合は、「方法2: -Xオプションを使用してOracle Grid Infrastructureホームの圧縮コピーを作成する」のステップ2を繰り返します。
3. 必要に応じて、Oracle Grid Infrastructureホームのすべてのファイルの所有権をOracle Grid Infrastructureインストール所有者およびOracle Inventory (oinstall権限)グループによって所有されるように変更します。次の例では、Oracle Grid Infrastructureインストール所有者がoracleで、Oracle Inventoryグループがoinstallの場合にLinuxシステムでこれを行うコマンドを示します。

```
[root@node1 crs]# chown -R oracle:oinstall /u01/app
```

Gridホームで前述のコマンドを実行すると、Oracleバイナリからsetuidとsetgidの情報がクリアされます。想定どおり、このコマンドによって、次のバイナリからsetuidもクリアされます。

```
Grid_home/bin/extjob
Grid_home/bin/jssu
Grid_home/bin/oradism
```

setuid情報は、後続のステップで正しく設定されます。

4. 処理を続行する前に、ソースおよび宛先の両方のノードのGrid_homeディレクトリからすべてのOracleネットワーク・ファイルを削除することも重要です。

関連トピック

- [ステップ3: Oracle Grid Infrastructureホームのコピーの作成](#)
- [方法2b: -Xオプションを使用してOracle Grid Infrastructureホームのコピーを作成する](#)
- [Oracle Grid Infrastructureホームをクローニングするための準備](#)
- [クローニング時に生成されたログ・ファイルの検索および表示](#)

ステップ3: gridSetup.shユーティリティの実行

新しいOracle Clusterware環境を設定するには、新しいクラスタ用のOracle Grid Infrastructureをインストールするときと同様に、1つのノードで対話モードまたはサイレント・モードでgridSetup.shユーティリティを実行します。

\$ORACLE_home/gridSetup.shユーティリティを起動すると、対話型のインタフェースを使用して対話モードで実行することも、レスポンス・ファイルを指定してサイレント・モードで実行することもできます。

クローニングを使用したクラスタへのノードの追加

また、クローニングを使用してクラスタにノードを追加することもできます。

[図7-2](#)では、クローニング手順の結果を示しており、ノード1上のOracle Grid Infrastructureホームは同じクラスタ上のノード2にクローニングされ、クラスタが2ノード・クラスタとなっています。クラスタへ新しく追加されたノードは、同じOCRおよび投票ファイルを共有します。

図7-2 クラスタにノードを追加するためのクローニング

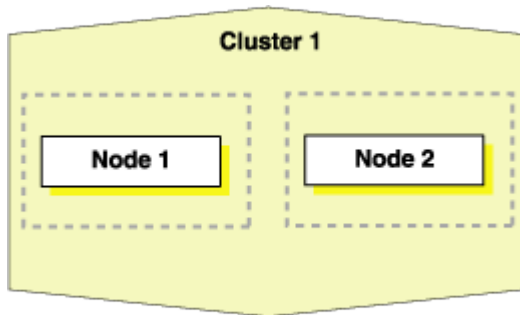


図7-2を例とし、次の手順では、クローニングを使用してクラスタへノードを追加する方法を説明します。この手順では、最初にノード1を作成するために使用したイメージ(クローン)をノード2にコピーします。

「[ステップ1: 新規クラスタ・ノードの準備](#)」で説明されているように、ノード2を準備します。

1. 「[ステップ2: 宛先ノードでのOracle Grid Infrastructureホームのデプロイ](#)」で説明されているように、ノード2でOracle Grid Infrastructureホームをデプロイします。

tarユーティリティを使用してノード1でOracle Grid Infrastructureホームのアーカイブを作成し、それをノード2にコピーします。ノード1のOracle Grid Infrastructureホームの場所が\$ORACLE_HOMEである場合は、これと同じディレクトリをノード2のコピー先として使用する必要があります。

2. gridSetup.shスクリプトを実行して、Software Onlyインストール・オプションを選択します。
3. このステップはWindowsには適用されません。

ノード2の中央インベントリ・ディレクトリで、rootとしてoraInstRoot.shスクリプトを実行します。このスクリプトは、/etc/oraInst.locディレクトリに中央インベントリの場所を移入します。たとえば：

```
[root@node2 root]# /opt/oracle/oraInventory/oraInstRoot.sh
```

複数の宛先ノードで同時にスクリプトを実行することができます。

4. gridSetup.sh -noCopyスクリプトを実行して、Add more nodes to clusterを選択します。
5. ノード2で、Grid_home/root.shスクリプトを実行します。

ノート：



- root.sh または gridconfig.bat スクリプトを実行する前に、すべてのデータベース・ホームを展開したことを確認します。
- この例のクラスタのノードは2つのみです。クラスタに複数のノードを追加する場合、すべてのノードで root.sh を同時に実行することができます。

次の例はLinuxまたはUNIXシステム用の例です。ノード2で、次のコマンドを実行します。

```
[root@node2 root]# Grid_home/root.sh
```

root.shスクリプトは、Oracle Cluster Registry (OCR)に仮想IP (VIP)リソースを自動的に構成します。

Windowsでは、ノード2で次のコマンドを実行します。

```
C:\>Grid_home\crs\config\gridconfig.bat
```

6. ノード1で次のクラスタ検証ユーティリティ(CVU)・コマンドを実行します。

```
$ cluvfy stage -post nodeadd -n destination_node_name [-verbose]
```

関連トピック

- [cluvfy stage nodeadd](#)

クローニング時に生成されたログ・ファイルの配置と表示

クローニング・スクリプトは、それぞれがログ・ファイルを生成できる複数のツールを実行します。gridSetup.shユーティリティの実行が終了した後、これらのログ・ファイルを参照して、クローニング手順のステータスに関する詳細情報を入手することができます。[表7-1](#)では、クローニング時に生成される、診断に重要なログ・ファイルを示します。

ノート:



[表 7-1](#) の Central_inventory は、Oracle Inventory ディレクトリを表します。

表7-1 クローニングのログ・ファイルおよび説明

ログ・ファイルの名前および場所	説明
Central_inventory/logs/cloneActionstamp.log	クローニングの OUI 部分で行われたアクションの詳細ログが含まれます。
Central_inventory/logs/oraInstalltimeamp.err	OUI の実行中に発生したエラーに関する情報が含まれます。
Central_inventory/logs/oraInstalltimeamp.out	その他の情報が含まれます。

[表7-2](#)は、様々なプラットフォーム用のOracle Inventoryディレクトリの場所を示しています。

表7-2 Oracleインベントリ・ディレクトリの位置の確認

システムのタイプ	Oracleインベントリ・ディレクトリの位置
Linux および IBM AIX を除くすべての UNIX コンピュータ	/var/opt/oracle/oraInst.loc
IBM AIX および Linux	/etc/oraInst.loc
Windows	C:\Program Files\Oracle\Inventory

8 Oracle Clusterwareを使用したアプリケーションの高可用性の実現

クラスタ内でアプリケーション、プロセスまたはサーバーに障害が発生した場合、その影響をユーザーにまったく気付かれないようにすることはできないまでも、できるかぎり短時間に抑える必要があります。たとえば、サーバーでアプリケーションに障害が発生した場合、そのアプリケーションをクラスタ内の別のサーバーで再起動して、そのアプリケーションの使用に対する影響を最小限にするか、またはなくしてしまうことができることを意味します。同様に、クラスタ内のサーバーで障害が発生した場合、そのサーバーで実行されているすべてのアプリケーションおよびプロセスを別のサーバーにフェイルオーバーして、ユーザーへのサービスの提供を継続できる必要があります。Oracle Clusterwareでは、組込みのgeneric_applicationリソース・タイプまたはカスタマイズ可能なスクリプトおよびアプリケーション・エージェント・プログラムと、アプリケーションとプロセスに割り当てられるリソース属性を使用することで、これらすべてのエンティティを管理し、高可用性を確保します。

この章では、Oracle Clusterwareを使用して、アプリケーションの起動、停止、監視、再起動および再配置を行う方法について説明します。Oracle Clusterwareは、Oracle Real Application Clusters(Oracle RAC)の基礎となるクラスタ・ソリューションです。Oracle RACデータベースを管理する場合と同じ機能および方針がアプリケーションの管理に適用されます。

この章の内容は次のとおりです。

- [Oracle Clusterwareリソースおよびエージェント](#)
- [Oracle Clusterwareを使用した高可用性の実現の概要](#)
- [リソースとしてのアプリケーションの登録](#)
- [リソースの管理](#)
- [Oracle Clusterwareによるリソースの自動再起動の管理](#)

Oracle Clusterwareリソースおよびエージェント

この項では、アプリケーションの高可用性を確保するために、Oracle Clusterwareで[リソース](#)の監視および管理に使用されるフレームワークについて説明します。

この項には次のトピックが含まれます：

- [Oracle Clusterwareのリソース](#)
- [Oracle Clusterwareのリソース・タイプ](#)
- [Oracle Clusterwareのエージェント](#)
- [アクション・スクリプト](#)
- [エージェントの作成](#)
- [Oracle Clusterwareへのリソースの登録](#)

Oracle Clusterwareのリソース

Oracle Clusterwareでは、アプリケーションおよびプロセスはOracle Clusterwareに登録するリソースとして管理されます。アプリケーションを管理するためにOracle Clusterwareに登録するリソースの数は、アプリケーションによって異なります。1つの

プロセスでのみ構成されるアプリケーションは、ほとんどの場合、1つのリソースでのみ表されます。複数のプロセスまたはコンポーネントに基づいて構築されたより複雑なアプリケーションでは、複数のリソースが必要となる可能性があり、このような場合は[リソース・グループ](#)を作成できます。

各リソースは、テンプレートとして機能するリソース・タイプに基づいています。明示的なサーバー・リストを指定したり、サーバー・プールやポリシーなどの機能を使用することによって、Oracle Clusterwareでのアプリケーションのクラスタ内への配置方法を構成できます。アプリケーションとコンポーネントの関係は、依存性を使用して表されます。Oracle Clusterwareはリソースに対して操作を実行することによりアプリケーションを管理し、リソースの状態はアプリケーションの可用性を表します。

Oracle Clusterwareのリソースとしてアプリケーションを登録する場合は、実際にリソースをシステムに追加するだけでなく、リソースの特徴を示すリソース属性を使用してOracle Clusterwareでのアプリケーションの管理方法を定義します。リソース属性の例としては、リソースのチェック頻度、障害が発生してから別のサーバーで起動が試行される(フェイルオーバー)までの間に同じサーバーでリソースの再起動が試行される回数などがあります。また、登録情報には、アクション・スクリプトへのパス、またはアプリケーションの起動、停止、チェックおよびクリーンアップを実行するためにOracle Clusterwareによってコールされるアプリケーション固有のアクション・プログラムも含まれています。

アクション・スクリプトは、Oracle Clusterwareで提供されている汎用スクリプト・エージェントがコールするシェル・スクリプト(Windowsのバッチ・スクリプト)です。アプリケーション固有のエージェントは、ほとんどの場合、Oracle Clusterwareで提供されているAPIを直接コールするCまたはC++プログラムです。

クリティカル・リソース

リソース・グループとしてモデル化された一部の大規模なエンタープライズ・アプリケーションには、アプリケーションやインフラストラクチャのコンポーネントを表す複数のリソースが含まれることがあります。リソース・グループ内のいずれかのリソースに障害が発生した場合、Oracle Clusterwareはリソース・グループ全体をクラスタ内の別のサーバーにフェイルオーバーする必要があります。

リソース・グループのCRITICAL_RESOURCES属性内にリソースの名前を指定することによって、あるリソースをそのリソース・グループのクリティカル・リソースとしてマークできます。

関連トピック

- [Oracle Clusterwareのリソース・リファレンス](#)
- [リソース・グループ](#)

仮想マシンのリソース

仮想マシンは、ゲスト・オペレーティング・システムと呼ばれる稼働中のオペレーティング・システム用に作成された環境です。仮想マシンは、コンピュータのデスクトップにウィンドウとして表示され、このウィンドウは全画面モードで表示することも、別のコンピュータ上にリモートで表示することもできます。

仮想マシンは基本的に、その動作を決定するパラメータのセットであり、コンピュータ・システム・ハードウェアと似ています。パラメータには、ハードウェア設定(仮想マシンのメモリー量など)や状態情報(仮想マシンが現在実行中かどうかなど)が含まれます。

ブラックボックス仮想マシンは、その内容が管理インタフェースに認識されない仮想マシンです。ブラックボックス仮想マシンについて認識される情報は、それに含まれる仮想ハードウェア、CPUの数、RAMの量、接続されたディスク、および接続されたネットワーク・インタフェースのみです。ただし、ハードウェアの内容は認識されません。たとえば、多数のディスクが接続されていたとしても、それらにどのオペレーティング・システムがインストールされているか、またネットワーク・カードが構成されているかどうかについては認識されません。

Oracle Clusterwareを使用して物理ハードウェア上のブラックボックスOracle仮想マシンを管理すると、可用性が向上し、仮想マシンの管理が容易になります。

ノート:

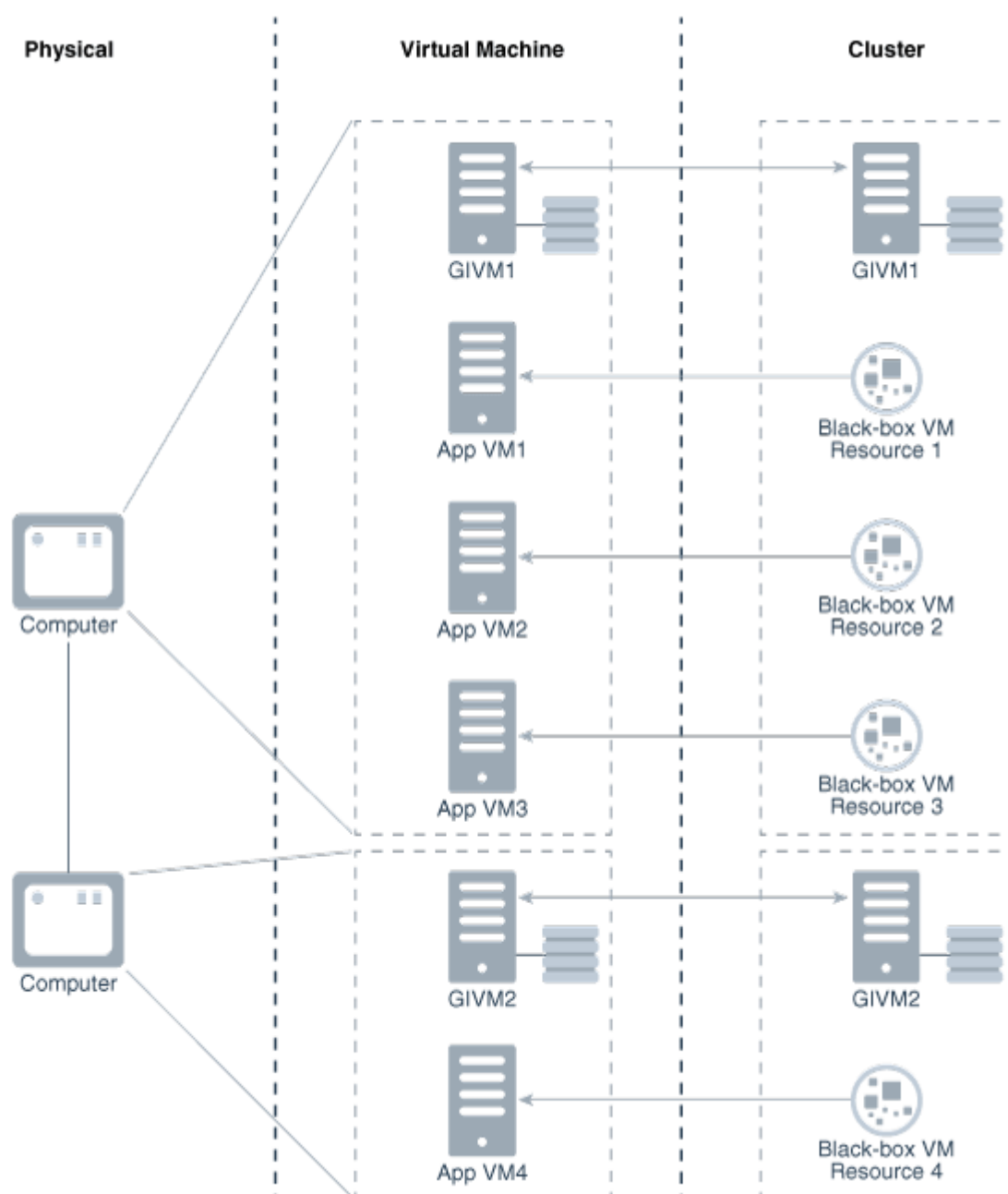


これは仮想マシンに限定されることであり、Oracle VM VirtualBox やその他の Oracle VM 製品には該当しません。

たとえば、次の図では、2つの物理コンピュータがあり、それぞれで複数の仮想マシンが実行されています。各物理ホストに対するコンピュータの1つが、Oracle Grid Infrastructure仮想マシン(GIVM)です。

GIVM自体が1つのOracle Clusterwareクラスタを構成し、このクラスタ内に4つのブラックボックス仮想マシンOracle Clusterwareリソースがあって、それぞれがGIVM以外の仮想マシンの1つを監視しています。監視対象の仮想マシンはブラックボックス仮想マシンであるため、クラスタはこれらの仮想マシンの内容を認識しません。この例では、物理ホストの1つが停止した場合、そのGIVMも停止し、そのGIVMがクラスタから離脱します。これにより、リソースは他のGIVMにフェイルオーバーされ、新しい物理ホスト上でブラックボックス仮想マシンが起動されます。

図8-1 Oracle Database Applianceでの高可用性仮想マシン



仮想マシン・アーキテクチャ

Oracle仮想マシンは、仮想マシン・サーバーと仮想マシン・マネージャの2つの部分で構成されます。仮想マシン・サーバーは、Xenハイパーバイザを使用してゲストを管理するベア・ハードウェアにインストールされた最小オペレーティング・システムです。サーバーには、仲介役となるOracle仮想マシン・エージェントというエージェント・プロセスがあり、仮想マシン・マネージャはこれを介してサーバー上のドメインを操作します。

仮想マシン・マネージャは、仮想マシン・サーバーおよびその仮想マシンの管理に使用されるWebベースの管理コンソールです。仮想マシン・マネージャを実行するにはデータベースおよびOracle WebLogic Serverが必要であり、Oracle仮想マシン・サーバーのOracleサポート管理にはこの仮想マシン・マネージャが必要です。管理ドメインはできるだけ小さくする必要があります。そのため、仮想マシン・マネージャをそこにインストールすることはできません。仮想マシン・マネージャは別のホストにインストールする必要があります。つまり、別の物理コンピュータを用意するか、Xen xmコマンドを使用して一時的な仮想マシンを手動で作成します。

Oracle仮想マシン・マネージャは、仮想マシンを管理するための唯一のインタフェースです。すべてのAPIおよびユーティリティを含むすべてのリクエストは、これを介して転送されます。

仮想マシン・リソース(Oracle Clusterwareリソース)のリソース・タイプは、次のようになります。

```
ATTRIBUTE=DESCRIPTION
TYPE=string
DEFAULT_VALUE="Resource type for VM Agents"
ATTRIBUTE=AGENT_FILENAME
TYPE=string
DEFAULT_VALUE=%CRS_HOME%/bin/oraagent%CRS_EXE_SUFFIX%
ATTRIBUTE=CHECK_INTERVAL
DEFAULT_VALUE=1
TYPE=int
ATTRIBUTE=OVMM_VM_ID
TYPE=string
DEFAULT_VALUE=' '
ATTRIBUTE=OVMM_VM_NAME
TYPE=string
DEFAULT_VALUE=' '
ATTRIBUTE=VM
TYPE=strings
```

リソース・グループ

リソース・グループは、論理的に関連したリソースのグループのコンテナです。

アプリケーションは、アプリケーション・リソースと関連アプリケーション・リソース(WebServerなど)、およびインフラストラクチャ・リソース(ディスク・グループやVIPなど)を含むリソース・グループとしてモデル化されます。リソース・グループは、すべてのアプリケーション・クラスの高可用性モデリングのための論理的で直感的なエンティティを提供します。

CRSCTLを使用してリソース・グループを作成し、リソースをそのリソース・グループに追加します。リソース・グループは、ネーミング、説明および一般的な配置に対応する属性セットと、リソース・グループのメンバーであるリソースのフェイルオーバー・パラメータ値を提供します。

リソース・グループの原則

- リソース・グループは、リソース・グループ・タイプに基づいて作成します。
- リソースがメンバーになれるリソース・グループは、1つのみです。リソースを作成するときに、リソースにリソース・グループを指定できます。

リソースの作成時にリソース・グループを指定しない場合、リソースは、そのリソースに作成された自動リソース・グループのメンバーになります。後でリソースを別のリソース・グループに追加できます。

- リソース・グループはクリティカル・リソースを認識し、リソース・グループの状態はそのクリティカル・リソースの状態によってのみ決まります。

クリティカルでないリソースはリソース・グループから削除でき(依存性チェックに従う)、後で別のリソース・グループに追加することもできます。

- リソース・グループには、クラスタ内で同時に実行できるリソース・グループのインスタンス数を指定するためのカーディナリティがあります。
- 稼働中のリソース・グループ・インスタンスのすべてのメンバー・リソースは、同じサーバーに配置されます。
- Oracle Clusterwareは、障害発生時にリソース・グループを再起動し、ローカルな再起動が失敗した場合にはリソース・グループを別のサーバーに再配置します。

自動リソース・グループ

リソース・グループを指定せずにリソースを作成した場合、Oracle Clusterwareによって暗黙的かつ自動的にリソースと同名のリソース・グループにリソースが追加されます。

自動リソース・グループは、リソース・グループに明示的に追加されていないリソースそれぞれに対して作成されます。リソース・グループを使用せずにリソースを作成しても、問題なくOracle Clusterwareを使用できます。ただし、リソース・グループを使用すると、SRVMや他の既存のユーティリティにより作成された(自動リソース・グループを介する)インフラストラクチャおよびアプリケーション・リソースの関係を定義できます。

リソースに対して作成された自動リソース・グループは、そのリソースによってのみ記述され、管理者が変更することはできません。リソース・グループを指定せずに作成したリソースは、後でリソース・グループに追加できます。リソースが明示的にリソース・グループに追加された場合、Oracle Clusterwareはそのリソースが属する自動リソース・グループを削除します。

リソース・グループ管理

- リソースを作成するときに、リソースをリソース・グループに追加できます。
- 自動リソース・グループに属するリソースを別のグループに明示的に追加できます。ONLINEまたはOFFLINEのどちらのグループにリソースを追加する場合も、リソースはOFFLINEである必要があります。
- クリティカルでないリソースは、リソース・グループ内の他のリソースがそれに依存していないかぎり、リソース・グループ(OFFLINEまたはONLINE)から削除できます。この場合、削除したリソースは、その自動リソース・グループのメンバーとなります。後で、このリソースを別のグループに追加できます。
- クリティカルでないリソースを削除することによって、そのリソースをリソース・グループから削除し、Oracle Clusterwareから削除できます。リソース・グループのクリティカル・リソースは、まずリソース・グループのクリティカル・リソース・リストを更新してリソースのクリティカルマークを解除しないかぎり、削除できません。
- リソース・グループは、作成された当初は空で、グループ内の各リソースが削除されたときにも空になります。空のリソース・グループは起動できず、その状態は常にOFFLINEになります。

リソースの共有

様々なOracle Clusterwareデプロイメントにおいては、複数のアプリケーションにより共有されるコンポーネント(ファイル・システムなど)が存在します。たとえば、1つのOracle ACFSリソースを、ファイルシステムを活用する複数のリソース・グループのメンバーにすることはできません。定義上、リソースがメンバーになれるリソース・グループは1つのみであるためです。

これらのタイプのリソースについては、明示的または自動的にそれぞれ個別のリソース・グループに配置し、アプリケーション・リソース・グループからこれらの共有リソース・グループへの適切な依存性を構成することをお勧めします。この方法で、複数のアプリケーションはコンポーネントを共有できます。

クリティカル・リソース

リソース・グループとしてモデル化された大規模なエンタープライズ・タイプ・アプリケーションには、アプリケーションやインフラストラクチャのコンポーネントに対応する複数のリソースが含まれることがあります。このようなリソース・グループ内のリソースのいずれかに障害が発生した場合、Oracle Clusterwareはリソース・グループ全体をクラスタ内の別のサーバーにフェイルオーバーします。ただし、リソース・グループにはアプリケーションにクリティカルでないリソースも含まれ、そのようなリソースのために、インスタンスの実行を不必要に中断させるようなリソース・グループ全体のフェイルオーバーが必要になるとはかぎりません。

リソース・グループ内の特定のリソースをクリティカルとして定義できます(これには、リソース・グループのCRITICAL_RESOURCESリスト属性内にリソース名を指定します)。これらのリソースのいずれかに障害が発生した場合、Oracle Clusterwareはリソース・グループをクラスタ内の別のサーバーにフェイルオーバーします。

さらに、リソース・グループの状態はそのクリティカル・リソースの状態によってのみ決まります。クリティカルでないリソースがリソース・グループの状態に影響することはない、リソース・グループのフェイルオーバーをトリガーすることはありません。リソース・グループを起動してオンラインにするには、リソース・グループに1つ以上のクリティカル・リソースが含まれている必要があります。

リソース・グループ内の個々のリソースをクリティカルとして指定することも、リソース・タイプをクリティカルとして指定して、その特定タイプのすべてのリソースをクリティカルにすることもできます。たとえば：

```
CRITICAL_RESOURCES="r1 r2 r3"
```

前述の例では、空白で区切られた3つのリソースがクリティカルとしてマークされています。

```
CRITICAL_RESOURCES="appvip type:ora.export.type"
```

前述の例では、特定のリソース・タイプがクリティカルとして指定されているため、このタイプのリソースがすべてクリティカル・リソースとなります。

リソース・グループを作成したとき、またはそのメンバーを削除したとき、リソース・グループは空であり、したがってクリティカル・リソースは存在しません。リソース・グループのCRITICAL_RESOURCES属性内にリソース・タイプを指定していないかぎり、リソース・グループに追加した最初のリソースが、Oracle Clusterwareによって自動的にクリティカルとしてマークされます。Oracle Clusterwareは、リソース・グループを起動する前に、常に、クリティカル・リソースの存在を確認します。

リソース・グループの権限

リソース・グループおよびリソース・グループ・タイプを作成した後、リソースを作成してこれらのグループに追加できます。また、リソース・グループのACL属性を使用して、リソース・グループに対する変更権限および操作の実行権限を定義することもできます。また、リソース・グループ所有者は、リソース・グループのACL属性を適切に設定することにより、他のオペレーティング・システム・ユーザーに権限を割り当てることができます。リソース・グループ内のリソースは、そのACL属性内に指定された自身の権限指定を維持できます。具体的には、次のようになります。

- リソース・グループに対する書込み権限およびリソースに対する書込み権限を持つユーザーは、リソースをグループに追加できます。
- リソース・グループの所有者は常に、グループ内のすべてのリソースに対する実行権限を持ちます。グループに対する実行権限を付与されたユーザーまたはグループはいずれも、そのグループ内のすべてのリソースに対する実行権限を持ちます。

たとえば、リソース・グループ内の特定のインフラストラクチャ・リソースをrootによって管理する必要がある場合、リソースの所有者をrootとして指定し、リソースに対する実行権限をグループ所有者に付与する必要があります。これは、rootユーザーが明示的に行う必要があります。

- ローカル管理ユーザー(UNIXではroot、Windowsでは管理者グループ・ユーザー)は、任意のリソース・グループを変

更、削除、起動および停止できます。

リソース・グループの依存性

リソース・グループ間の依存性を設定することにより、アプリケーションとコンポーネントの間の関係を表すことができます。Oracle Clusterwareには、リソース・グループ間の依存性の様々な順序、場所および強制レベルを指定する修飾子が用意されています。リソース・グループの依存性に関して、次のことを考慮する必要があります。

- リソース・グループは、別のリソース・グループとの依存関係を持つことはできますが、個々のリソースとの依存関係を持つことはできません。
- 明示的に作成されたリソース・グループは、自動リソース・グループとの依存関係を持つことができます。
- グループ内のリソースは、同じグループ内の別のリソース・グループとの依存関係を持つことができます。
- リソース・グループを指定せずに作成した(そのために自動リソース・グループに属する)リソースは、別のリソース・グループとの依存関係を持つことができます。
- リソースがリソース・グループとの依存関係を持ったり、別のリソース・グループ内のリソースとの依存関係を持つことはできません。

Oracle Clusterwareリソースの使用可能な依存性はすべて、リソース・グループでも使用可能です。リソース・グループの依存性を指定するには、リソース・グループのSTART_DEPENDENCIESおよびSTOP_DEPENDENCIES属性を構成します。

表8-1 リソース・グループの依存性のタイプおよび修飾子

依存性のタイプ	説明
hard 起動	このリソース・グループを起動する前に特定の他のリソース・グループが(クラスタ内のどこかで)オンラインになっている必要があるという要件を指定します。 たとえば: <pre>START_DEPENDENCIES=hard([global: intermediate: uniform:] other_resource_group)</pre> いずれかの依存リソース・グループの起動に失敗すると、Oracle Clusterware はこのリソース・グループの起動を取り消します。
weak 起動	このリソース・グループの起動前に特定の他のリソース・グループの起動を試行する必要があるという要件を指定します。特定の他のリソース・グループの起動に失敗した場合でも、Oracle Clusterware によってこのリソース・グループが起動されます。 たとえば: <pre>START_DEPENDENCIES=weak([global: concurrent: uniform:] other_resource_group)</pre>

依存性のタイプ	説明
pullup	依存リソース・グループの起動時にこのリソース・グループを自動的に起動する必要がある場合は、この依存性を使用します。 たとえば: <pre>START_DEPENDENCIES=pullup([intermediate: always:]other_resource_group])</pre> リソース・グループ間に停止依存性が存在する場合には、この依存性を使用することをお勧めします。
hard 停止	この依存性は、特定の他のリソース・グループが実行を停止したときにこのリソース・グループを停止するという必須要件を指定します。 たとえば: <pre>STOP_DEPENDENCIES=hard([intermediate: global: shutdown:]other_resource_group)</pre>
attraction	特定の他のリソース・グループとの共存を優先することを指定します。 たとえば: <pre>START_DEPENDENCIES=attraction([intermediate:]:]other_resource_group)</pre> Oracle Clusterware は、特定の他のリソース・グループがすでにオンラインになっている同じサーバー上で、このリソース・グループの起動を試行します。
dispersion	特定の他のリソース・グループとの非共存を優先することを指定します。Oracle Clusterware は、dispersion 依存性を持つオンライン・リソース・グループの数が最も少ないサーバー上で、このリソース・グループの起動を試行します。 たとえば: <pre>START_DEPENDENCIES=dispersion([intermediate: active: pool:]:]other_resource_group)</pre>
exclusion	このリソース・グループを特定の他のリソース・グループと同じサーバー

依存性のタイプ	説明
	上で実行しないことを必須要件として指定します。Oracle Clusterware は、このリソース・グループの起動を拒否するか、依存リソース・グループを停止して別のサーバー上で再起動します。 たとえば: <pre>START_DEPENDENCIES=exclusion[(preempt_pre: preempt_post)] other_resource_group)</pre>

リソース・グループの障害とリカバリ

前述のように、クリティカル・リソースによってリソース・グループの状態とフェイルオーバーが決まります。

クリティカル・リソースの障害とリカバリ

- リソース・グループのクリティカル・リソースに障害が発生すると、リソース・グループは即時にOFFLINE状態に移行します。
- Oracle Clusterwareは、RESTART_ATTEMPTSおよびUPTIME_THRESHOLDリソース属性に従って、障害の発生したクリティカル・リソースのローカル再起動を試行します。
- Oracle Clusterwareは、障害が発生したリソースに対する停止依存性を持つ同グループ内の他のリソースに対して即時チェック・アクションを開始します。
- Oracle Clusterwareは、このリソース・グループに依存する他のリソース・グループに対して即時チェック・アクションを開始します。
- リソースが正常に再起動された場合、リソース・グループはONLINE状態に移行し、Oracle Clusterwareはリソース・グループ内およびリソース・グループ間のpullup依存性評価を実行します。
- Oracle Clusterwareがリソースのすべてのローカル再起動試行数に達した場合、Oracle Clusterwareはリソース・グループ全体を停止します。また、Oracle Clusterwareは、リソース・グループに対する停止依存性を持つ他のリソース・グループも即時に停止します。Oracle Clusterwareは、リソース・グループのローカル再起動を試行します(そのように構成されている場合)。すべての再起動試行数に達した場合、リソース・グループはクラスタ内の別のサーバーにフェイルオーバーします。

クリティカルでないリソースの障害とリカバリ

- リソース・グループ内のクリティカルでないリソースに障害が発生した場合、Oracle Clusterwareは、RESTART_ATTEMPTSおよびUPTIME_THRESHOLDリソース属性の値に従って、障害の発生したクリティカル・リソースのローカル再起動を試行します。クリティカルでないリソースに障害が発生した場合、リソース・グループの状態に対する影響はありません。
- Oracle Clusterwareは、障害が発生したリソースに対する停止依存性を持つ同グループ内の他のリソースに対して即時チェック・アクションを開始します。
- リソースが正常に再起動された場合、Oracle Clusterwareはpullup依存性評価および対応する起動アクションを実行します。
- Oracle Clusterwareがリソースのすべてのローカル再起動試行数に達した場合、リソース・グループの状態への影響はありません。障害の原因を修正した後、クリティカルでないリソースを明示的に起動する必要があります。

関連トピック

- [リソース・グループ・タイプ](#)
- [リソース・グループの使用](#)

リソース・グループ・タイプ

Oracle Clusterwareにおいて、リソース・タイプはリソースのクラス用のテンプレートです。

リソース・グループ・タイプは、すべてのリソース・グループに共通に適用できる属性セットを提供します。リソース・グループの作成時に、リソース・グループ・タイプを指定する必要があります。Oracle Clusterwareには、2つのベース・リソース・グループ・タイプ (`local_resourcegroup` および `cluster_resourcegroup`) が用意されています。ベース・リソース・タイプの属性はリソースと似ており、その一部は構成可能です。

ローカル・リソース・グループ・タイプ

ローカル・リソースのみを含むリソース・グループを作成する場合は、`local_resourcegroup` タイプを使用します。このタイプのリソース・グループのインスタンスは、クラスタ内の各ノードで実行できます。ローカル・リソース・グループ・タイプの属性は次のとおりです。

- NAME
- DESCRIPTION
- ACL
- AUTO_START
- CRITICAL_RESOURCES
- DEBUG
- ENABLED
- INTERNAL_STATE
- RESOURCE_LIST
- RESTART_ATTEMPTS
- SERVER_CATEGORY
- START_DEPENDENCIES
- STOP_DEPENDENCIES
- STATE
- STATE_DETAILS
- UPTIME_THRESHOLD

クラスタ・リソース・グループ・タイプ

タイプ `cluster_resourcegroup` のリソース・グループには、クラスタ内の静的または動的なサーバー・セット上で実行される1つ以上のインスタンスを含めることができます。このようなリソース・グループは、グループの配置ポリシーに従って、クラスタ内の別のサーバーにフェイルオーバーできます。クラスタ・リソース・グループ・タイプの属性は次のとおりです。

- NAME
- DESCRIPTION
- ACL
- ACTIVE_PLACEMENT
- AUTO_START
- CARDINALITY
- CRITICAL_RESOURCES

- DEBUG
- ENABLED
- FAILURE_INTERVAL
- FAILURE_THRESHOLD
- HOSTING_MEMBERS
- INTERNAL_STATE
- PLACEMENT
- RESOURCE_LIST
- RESTART_ATTEMPTS
- SERVER_CATEGORY
- SERVER_POOLS
- START_DEPENDENCIES
- STOP_DEPENDENCIES
- STATE
- STATE_DETAILS
- UPTIME_THRESHOLD

関連トピック

- [リソース・グループの使用](#)
- [Oracle Clusterwareのリソース・リファレンス](#)

リソース・グループの使用

CRSCTLを使用して、リソース・グループおよびリソース・グループ・タイプを作成し、リソースをリソース・グループに追加します。

リソース・グループを使用するには、まず、組込みリソース・グループ・タイプまたは作成したリソース・グループ・タイプに基づいて、リソース・グループを作成する必要があります。リソース・グループを作成したら、リソースをそれに追加できます。

1. リソース・グループを作成するには、次のコマンドを使用します。

```
$ crsctl add resourcegroup group_name -type group_type
```

前述のコマンドでは空のリソース・グループが作成され、これにリソースを追加できます。リソース・グループの名前およびリソース・グループ・タイプを指定する必要があります。カスタム・リソース・グループ・タイプに基づくリソース・グループにすることを選択した場合は、次のステップの説明に従って、まずリソース・グループ・タイプを作成する必要があります。

2. カスタム・リソース・グループ・タイプに基づくリソース・グループを作成する場合は、次のようにして、リソース・グループ・タイプを作成する必要があります。

```
$ crsctl add resourcegroupstype group_type_name -basetype base_group_type {-file file_path | -attr "attribute_name=attribute_value"}
```

前述のコマンドではリソース・グループ・タイプが作成され、このリソース・グループ・タイプに基づいて作成したすべてのリソース・グループに対する単一の属性セットが提供されます。既存のリソース・グループ・タイプをベース・リソース・グループ・タイプとして指定し、行区切りの属性リスト/属性値ペアを含むファイルのパスを指定するか、コマンドラインでカンマ区切りの属性リスト/属性値ペアを指定する必要があります。

3. リソース・グループを作成した後は、次のようにして、リソース・グループへのリソースの追加を開始できます。

```
$ crsctl add resource resource_name -group group_name
```

リソースの追加先のリソース・グループが存在している必要があり、かつ、追加するリソースがオフライン状態になっている

必要があります。リソースがメンバーになれるリソース・グループは、1つのみです。複数のアプリケーションによって共有されているリソース(ファイル・システムなど)がある場合は、これらのリソースをそれぞれ個別のリソース・グループに配置することをお勧めします。

Oracle Clusterwareのリソース・タイプ

通常、すべてのリソースは一意ですが、共通属性を持つリソースもあります。Oracle Clusterwareでは、これらの類似リソースの編成にリソース・タイプが使用されます。リソース・タイプのメリットは次のとおりです。

- 必要なリソース属性のみの管理
- リソース・タイプに基づくすべてのリソースの管理

Oracle Clusterwareに登録するすべてのリソースには、特定のリソース・タイプが必要です。Oracle Clusterware制御(CRSCTL)ユーティリティを使用して、Oracle Clusterwareに含まれているリソース・タイプに加えて、カスタム・リソース・タイプを定義できます。含まれているリソース・タイプは次のとおりです。

- ローカル・リソース: ローカル・リソースのインスタンス(タイプ名は`local_resource`)は、クラスタの各サーバーで実行されるか(デフォルト)、または特定のサーバー・カテゴリに属しているサーバーで実行されるように制限できます。サーバーをクラスタに追加すると、Oracle Clusterwareではローカル・リソースが自動的に拡張され、インスタンスは新しいサーバーに関連付けられます。サーバーをクラスタから削除すると、Oracle Clusterwareでは、削除するサーバーで実行されていたローカル・リソースのインスタンスが自動的に削除されます。ローカル・リソースのインスタンスは、対応するサーバーに固定されており、あるサーバーから別のサーバーにフェイルオーバーされることはありません。
- クラスタ・リソース: タイプ名が`cluster_resource`のクラスタ対応リソース・タイプで、クラスタ環境を認識し、[カーディナリティ](#)およびサーバー間のスイッチオーバーとフェイルオーバーの対象となります。
- 汎用的なアプリケーション: このリソース・タイプ(タイプ名は`generic_application`)を使用すると、スクリプトを追加しなくても汎用的なアプリケーションを保護できます。アプリケーションの高可用性は、リソースを`generic_application`リソース・タイプで定義し、リソースの重要な属性に価値を提供することで実現されます。`generic_application`リソース・タイプは、`cluster_resource`リソース・タイプから取得されるため、`generic_application`リソース・タイプのすべてのリソースは、クラスタ対応のリソースです。属性は次のとおりです。
 - START_PROGRAM: アプリケーションを起動する実行可能ファイルの完全なパスで、適切なすべての引数が含まれています。実行可能ファイルは、Oracle Grid Infrastructureを構成してアプリケーションを実行する各サーバーに存在する必要があります。この属性は必須です。たとえば:

```
/opt/my_app -start
```

また、実行可能ファイルは、アプリケーションが正常に起動してオンラインになっていることを示すために、アプリケーションが起動したらゼロ(0)の終了ステータス値を返すことを確認する必要があります。実行可能ファイルがアプリケーションの起動に失敗した場合、実行可能ファイルは、ゼロ以外のステータス・コードで終了します。

- STOP_PROGRAM: アプリケーションを停止する実行可能ファイルの完全なパスで、適切なすべての引数が含まれています。実行可能ファイルは、Oracle Grid Infrastructureを構成してアプリケーションを実行する各サーバーに存在する必要があります。属性値を指定しない場合、Oracle Clusterwareは、オペレーティング・システムのkillに相当するコマンドを使用します。たとえば:

```
/opt/my_app -stop
```

また、実行可能ファイルは、アプリケーションが正常に停止したことを示すために、アプリケーションが停止したらゼロ(0)の終了ステータス値を返すことを確認する必要があります。実行可能ファイルがアプリケーションの停止

に失敗した場合、実行可能ファイルは、ゼロ以外のステータス・コードで終了し、Oracle Clusterwareはリソースのクリーンアップを開始します。

- **CLEAN_PROGRAM:** アプリケーションをクリーンアップする実行可能ファイルの完全なパスで、適切なすべての引数が含まれています。実行可能ファイルは、Oracle Grid Infrastructureを構成してアプリケーションを実行する各サーバーに存在する必要があります。この属性に値を指定しない場合、Oracle Clusterwareは、オペレーティング・システムの`kill -9`に相当するコマンドを使用します。たとえば:

```
/opt/my_app -clean
```

ノート:



STOP_PROGRAM および **CLEAN_PROGRAM** との違いは、**CLEAN_PROGRAM** は、アプリケーションを不正に停止する強制停止で、アプリケーションをいつでも停止できる必要があります、停止できない場合はアプリケーションが管理不能になります。

- **PID_FILES:** アプリケーションによって記述される、ファイルへの完全なパスのカンマ区切りのリストで、監視するためのプロセスID(PID)を含みます。プロセスに1回でも障害が起きると、完全なリソース障害として取り扱われます。たとえば:

```
/opt/app.pid
```

ノート:



PID_FILES 属性で指定するファイルは、**START** アクションが完了し、ファイルにリストされているPIDの監視が開始された直後に読み込まれます。

- **EXECUTABLE_NAMES:** 実行可能ファイルの名前のカンマ区切りのリストで、これは、アプリケーションが起動し、その後実行可能ファイルの状態が監視されると作成されます。実行可能ファイルに1回でも障害が起きると、完全なリソース障害として取り扱われます。たとえば:

```
my_app
```

ノート:



実行可能ファイルの完全な名前のみを指定する必要があります。この属性は実行可能ファイルまたはワイルド・カードのパスを受け入れません。**START** アクションが完了した後すぐに、実行可能ファイル名に一致するPIDがキャッシュされます。

- **CHECK_PROGRAMS:** 実行可能ファイルへの完全なパスのリストで、アプリケーションの状態を判断します。いずれのアプリケーションでも動作していない状態がレポートされると、リソース全体の障害として取り扱われます。たとえば:

```
/opt/my_app -check
```

- **ENVIRONMENT_FILE:** アプリケーションの起動時にソースへの環境変数を含んでいるファイルへの完全な

パスです。ファイルは、1行に1つのname=valueのペアを含んでいるテキスト・ファイルである必要があります。
たとえば:

```
/opt/my_app.env
```

- **ENVIRONMENT_VARS**: アプリケーションの起動時の環境に含まれるname=valueのペアのカンマ区切りのリストです。たとえば:

```
USE_FILES=No, AUTO_START=Yes
```

- **SEND_OUTPUT_ALWAYS**: この属性は、STDOUTに送られるアプリケーション出力を送り、表示する役割を果たします。値が0の場合、アクションが失敗しないかぎり、いずれのアプリケーション出力も表示されません。アクションが失敗すると、エージェントによって保存されたアプリケーション出力が表示されます。値が0より大きい場合、すべてのアプリケーション出力を表示します。デフォルトの値は0。たとえば:

```
SEND_OUTPUT_ALWAYS=1
```

ノート:



STOP_PROGRAM、**CHECK_PROGRAMS** および **CLEAN_PROGRAM** 属性を指定しない場合、**PID_FILES** または **EXECUTABLE_NAMES** のいずれかを指定する必要があり、これを行わない場合は、Oracle Clusterware でこのタイプのリソースを登録することはできません。

すべての属性を指定する場合、次のルールが適用されます。

- リソースを停止する場合、**STOP_PROGRAM**を指定すると、Oracle Clusterwareは**STOP_PROGRAM**をコールします。それ以外の場合では、Oracle Clusterwareは、**PID_FILES**または**EXECUTABLE_NAMES**属性のいずれかから取得されたPIDでオペレーティング・システムの`kill -9`に相当するコマンドを使用します。
- アプリケーションの現在の状態を確立する必要がある場合、**CHECK_PROGRAMS**を指定すると、Oracle Clusterwareは**CHECK_PROGRAMS**をコールします。それ以外の場合では、Oracle Clusterwareは、**PID_FILES**または**EXECUTABLE_NAMES**属性のいずれかから取得されたPIDでオペレーティング・システムの`ps -p`に相当するコマンドを使用します。
- リソースをクリーンアップする場合、**CLEAN_PROGRAM**を指定すると、Oracle Clusterwareは**CLEAN_PROGRAM**をコールします。それ以外の場合では、Oracle Clusterwareは、**PID_FILES**または**EXECUTABLE_NAMES**属性のいずれかから取得されたPIDでオペレーティング・システムの`kill -9`に相当するコマンドを使用します。

関連トピック

- [crsctl add resource](#)
- [リソースの状態](#)
- [generic_applicationリソース・タイプを使用するリソースの作成](#)

Oracle Clusterwareのエージェント

Oracle Clusterwareでは、リソース固有のすべてのコマンドはエージェントと呼ばれるエンティティを介して実行されます。

アプリケーションがOracle Clusterwareにリソースとして登録されると、Oracle Clusterwareでそれらのアプリケーションが管

理されます。Oracle Clusterwareでは、特定のリソースを起動、停止および監視できるアプリケーション固有のプリミティブにアクセスできます。

ノート:



セキュリティを強化し、管理業務をさらに分離するために、Oracle Clusterware エージェントは SYSRAC 管理権限で実行されるようになり、SYSDBA 管理権限は不要になりました。SYSRAC 管理権限は、SRVCTL などの Oracle RAC ユーティリティにかわって Oracle Clusterware エージェントによってデータベースに接続するためのデフォルト・モードであり、これにより Oracle RAC データベース・クラスタの日常の管理にデータベースへの SYSDBA 接続は不要になりました。

エージェントは、リソースを管理するためのエージェント・フレームワークおよびユーザー・コードを含むプロセスです。エージェント・フレームワークは、アプリケーション固有のコードを組み込んで、カスタマイズされたアプリケーションを管理できるライブラリです。アプリケーションの起動、停止、状態のチェックなど、実際のアプリケーション管理機能のすべてをエージェントにプログラムします。これらの機能はエントリ・ポイントと呼ばれます。

Oracle Clusterwareにかわって、エージェント・フレームワークによりこれらのエントリ・ポイント機能が起動されます。エージェントの開発者は、これらのエントリ・ポイントを使用して、リソースの起動、停止および監視方法に関する、機能を特定のリソースに必要な機能を組み込むことができます。エージェントで複数のリソースを管理できます。

エージェントの開発者は、コードへのコールバックとして次のエントリ・ポイントを設定できます。

- **ABORT:** エージェント・フレームワークでは、他のエントリ・ポイントが応答を停止した場合にABORTエントリ・ポイントがコールされ、進行中のアクションを停止します。エージェントの開発者によって停止機能が指定されていない場合、エージェント・フレームワークによってエージェント・プログラムが終了します。
- **ACTION:** ACTIONエントリ・ポイントは、カスタム・アクションが`crsctl request action`[コマンド](#)の`clscrs_request_action` APIを使用して起動されたときに起動されます。
- **CHECK:** CHECK (モニター)エントリ・ポイントは、リソースの状態をモニターします。エージェント・フレームワークでは、このエントリ・ポイントが定期的にコールされます。エージェント・フレームワークでは、このアクションで状態変更が検出されると、特定のリソースの状態における変更についてOracle Clusterwareに通知が行われます。
- **CLEAN:** CLEANエントリ・ポイントは、リソースのクリーンアップが必要な場合に機能します。ユーザーがリソースを強制終了する必要がある場合に起動される操作で、正常な場合の操作ではありません。このコマンドによってリソース固有の環境がクリーンアップされ、リソースを再起動できます。
- **DELETE:** DELETEエントリ・ポイントは、リソースが未登録のときにリソースを実行できるすべてのノードで起動されます。
- **MODIFY:** MODIFYエントリ・ポイントは、リソース・プロファイルが変更されているときにリソースを実行できるすべてのノードで起動されます。
- **START:** STARTエントリ・ポイントは、リソースをオンラインにします。エージェント・フレームワークでは、Oracle Clusterwareからstartコマンドを受信するたびにこのエントリ・ポイントがコールされます。
- **STOP:** STOPエントリ・ポイントは、リソースを正常に停止します。エージェント・フレームワークでは、Oracle Clusterwareからstopコマンドを受信するたびにこのエントリ・ポイントがコールされます。

START、STOP、CHECKおよびCLEANエントリ・ポイントは必須であり、エージェントの開発者はエージェントの作成時にこれらのエントリ・ポイントを指定する必要があります。エージェントの開発者には、C、C++、スクリプトの使用を含め、これらのエントリ・ポイントを実装するいくつかのオプションがあります。CまたはC++とスクリプト・タイプのエントリ・ポイントの両方を使用するエージェントを開発することもできます。エージェント・フレームワークの初期化時に、必須エントリ・ポイントのいずれかが指定されてい

ない場合、エージェント・フレームワークではACTION_SCRIPTリソース属性が指すスクリプトが起動されます。

エージェント・フレームワークでは、各アプリケーションに対して常に1つのエントリ・ポイントのみが起動されます。そのエントリ・ポイントが応答を停止した場合、エージェント・フレームワークはABORTエントリ・ポイントをコールして、現在の操作を終了します。エージェント・フレームワークによってCHECKエントリ・ポイントが定期的に起動され、リソースの状態が確認されます。このエントリ・ポイントによって、リソースの状態として次のいずれかが戻されます。

- CLSAGFW_ONLINE: リソースが正常に起動されて現在動作中の状態である場合には、CHECKエントリ・ポイントによってONLINEが戻されます。リソースがこの状態である場合、エージェント・フレームワークで引き続き監視されます。この状態では、scriptagentの数値は0です。

- CLSAGFW_UNPLANNED_OFFLINEおよびCLSAGFW_PLANNED_OFFLINE: OFFLINE状態は、リソースが現在実行されていないことを示します。これら2つの状態では、scriptagentの数値はそれぞれ1および2です。

plannedおよびunplannedの2つのカテゴリによってリソースのOFFLINE状態が示されます。

リソースの状態がOracle Clusterwareを介してOFFLINEに遷移すると、CHECKエントリ・ポイントから戻される値にかかわらず、このリソースはオフライン(TARGET=OFFLINE)になるように意図されているとみなされます。ただし、Oracle以外のインタフェースを使用してリソースが停止されたなど、Oracle Clusterwareに関係なくリソースの状態が変更されたことをエージェントで検出した場合、この意図はエージェントからクラスタ・レディ・サービス・デーモン(CRSD)に引き継がれる必要があります。この意図が次を決定する要因となります。

- リソースのTARGETリソース属性の値をそのままにしておくか、変更するか。PLANNED_OFFLINEは、リソースがそれまで実行されていた場合にのみTARGETリソース属性をOFFLINEに変更する必要があることを示します。リソースが実行されていなかった場合(STATE=OFFLINE、TARGET=OFFLINE)、リソースを起動するリクエストが着信すると、TARGETリソース属性の値はONLINEに変更されます。その後、エージェントにこの起動リクエストが送られ、エージェントはリソースの状態PLANNED_OFFLINEをOracle Clusterwareにレポートし、TARGETリソース属性の値はONLINEのままになります。TARGET属性はUNPLANNED_OFFLINEによって変更されません。
- リソースの状態をUNPLANNED_OFFLINEのままにしておくか、ローカルで再起動することによってリソースのリカバリを試行するか、クラスタ内の別のサーバーにフェイルオーバーするか。PLANNED_OFFLINE状態では、リソースはCRSDによってそのままにされますが、UNPLANNED_OFFLINE状態では、リソースのリカバリが求められます。
- CLSAGFW_UNKNOWN: リソースの現在の状態が判別できない場合は、CHECKエントリ・ポイントによってUNKNOWNが戻されます。この状態に対して、Oracle Clusterwareではフェイルオーバーまたはリソースの再起動は試行されません。リソースの以前の状態がONLINEまたはPARTIALのいずれかであった場合、エージェント・フレームワークでは引き続きリソースが監視されます。この状態では、scriptagentの数値は3です。
- CLSAGFW_PARTIAL: リソースが部分的にオンラインで、一部のサービスが使用可能であることが認識された場合は、CHECKエントリ・ポイントによってPARTIALが戻されます。Oracle Clusterwareでは、この状態は部分的にオンラインとみなされ、フェイルオーバーまたはリソースの再起動は試行されません。エージェント・フレームワークでは、この状態のリソースが引き続き監視されます。この状態では、scriptagentの数値は4です。
- CLSAGFW_FAILED: リソースが動作中の状態ではなく、一部のコンポーネントが失敗し、リソースを再起動するためにクリーンアップが必要であることが検出された場合は、CHECKエントリ・ポイントによってFAILEDが戻されます。この状態に対して、Oracle ClusterwareでCLEANアクションがコールされ、リソースがクリーンアップされます。CLEANアクションが終了したら、リソースの状態はOFFLINEになります。リソースのポリシーによっては、次に、Oracle Clusterwareでフェイルオーバーまたはリソースの再起動が試行される場合があります。エージェント・フレームワークで、

障害が発生したリソースが監視されることはありません。この状態では、scriptagentの数値は5です。

エージェント・フレームワークでは、[表8-2](#)に示す状態のリソースが、CHECK_INTERVALリソース属性またはOFFLINE_CHECK_INTERVALリソース属性で指定される一定の間隔で暗黙的に監視されます。

表8-2 エージェント・フレームワークの監視特性

状態	条件	頻度
ONLINE	常時	¥ CHECK_INTERVAL
PARTIAL	常時	CHECK_INTERVAL
OFFLINE	OFFLINE_CHECK_INTERVAL リソース属性の値が 0 より大きい場合のみ。	OFFLINE_CHECK_INTERVAL
UNKNOWN	監視されるのは、リソースが前述のいずれかの条件のために以前に監視されていた場合のみ。	ONLINE のときにのみ状態が UNKNOWN になる場合は、CHECK_INTERVAL の値が使用されています。それ以外の場合は、監視されていません。

エージェントが起動されるたびに、エージェントが監視するすべてのリソースの状態はUNKNOWNに設定されます。エージェント・フレームワークでは、Oracle Clusterwareからの最初のプローブ・リクエストを受信すると、すべてのリソースに対してCHECKエントリ・ポイントが実行され、現在の状態が確認されます。

リソースのCHECKアクションが正常に完了すると、リソースの状態は前述の状態のいずれかに遷移します。次に、エージェント・フレームワークによって、Oracle Clusterwareから発行されるコマンドに基づいてリソースが起動されます。エージェント・フレームワークでは、すべてのアクションが完了するとCHECKアクションが起動され、リソースの現在の状態が確認されます。エージェント・フレームワークでは、リソースが[表8-2](#)に示す監視対象の状態のいずれかである場合、CHECKエントリ・ポイントが定期的に実行され、リソースの状態に変更がないかどうかを確認されます。

デフォルトでは、エージェント・フレームワークでオフラインのリソースは監視されません。ただし、OFFLINE_CHECK_INTERVAL属性の値が0より大きい場合、エージェント・フレームワークでオフラインのリソースが監視されます。

関連トピック

- [ACTION_SCRIPT](#)
- [CHECK_INTERVAL](#)
- [OFFLINE_CHECK_INTERVAL](#)

Oracle Clusterwareの組み込みエージェント

Oracle Clusterwareでは、リソースの管理にエージェント・プログラム(エージェント)を使用しており、アプリケーションを保護する次の組み込みエージェントが含まれています。

- appagent: このエージェントは(Windowsではappagent.exe)、generic_applicationリソース・タイプのリソースおよびapplicationリソース・タイプの以前のバージョンのOracle Clusterwareのリソースを自動的に保護します。



ノート:

非推奨である application リソース・タイプは、以前の Oracle Clusterware 11g リリース 2 (11.2) リソースのサポートのみを目的として提供しているため使用しないことをお勧めします。

- **scriptagent**: このエージェント(Windowsではscriptagent.exe)は、シェル・スクリプトまたはバッチ・スクリプトを使用してアプリケーションを保護する場合に使用します。cluster_resourceとlocal_resourceの両方のリソース・タイプは、このエージェントを使用するように構成されており、また、これらのタイプのリソースは、このエージェントを自動的に利用します。

また、任意の方法でリソースを管理するために独自のエージェントを作成することもできます。

関連トピック

- [Oracle Clusterwareのリソース・タイプ](#)
- [エージェントの作成](#)

アクション・スクリプト

アクション・スクリプトは、リソースを開始、停止、確認またはクリーンアップする1つ以上のアクションを定義します。

エージェント・フレームワークは、C/C++アクションなしに、これらのアクションを起動します。アクション・スクリプトを使用すると、C/C++エントリ・ポイントおよびスクリプト・エントリ・ポイントを含むエージェントを作成できます。すべてのアクションがアクション・スクリプトに定義されている場合、スクリプト・エージェントを使用して、アクション・スクリプトに定義されているアクションを起動できます。

アクション・スクリプトに定義されているアクションを起動する前に、エージェント・フレームワークは、必要なすべての属性をリソース・プロファイルから環境にエクスポートします。アクション・スクリプトは、stdout/stderrにメッセージを記録でき、エージェント・フレームワークは、エージェント・ログにこれらのメッセージを出力します。ただし、アクション・スクリプトは、stdout/stderrに出力されるメッセージに次のいずれかのタグを接頭辞として追加することで、特別なタグを使用して、進捗、警告またはエラーのメッセージをcrs*クライアント・ツールに送信できます。

```
CRS_WARNING:
CRS_ERROR:
CRS_PROGRESS:
```

エージェント・フレームワークは、最終メッセージをcrs*クライアントに送信するときに、接頭辞として追加されたタグを削除します。

リソース属性は、接頭辞_CRS_を持つ環境変数としてアクション・スクリプト内からアクセス可能です。たとえば、START_TIMEOUT属性は、_CRS_START_TIMEOUTという環境変数になります。

関連トピック

- [crsctl start resource](#)
- [crsctl stop resource](#)

エージェントの作成

特定のアプリケーションの[エージェント](#)作成では、次のステップが実行されます。

1. スクリプト、CまたはC++のいずれかで[エージェント・フレームワーク](#)のエントリ・ポイントを実装します。
2. エージェントの実行可能ファイルを作成します(CエージェントおよびC++エージェントの場合)。

3. エントリ・ポイントで必要なすべてのパラメータを収集し、新しいリソース・タイプを定義します。AGENT_FILENAME属性を、新しく作成した実行可能ファイルの絶対パスに設定します。

CおよびC++エージェントの作成およびデプロイ

Oracle Clusterwareには、アプリケーションに高可用性エージェントを実装するエージェント・フレームワークを使用したCおよびC++エージェントの例が含まれています。

付録Fは、demoagent1.cppというエージェントの例を説明しています。このエージェントは、ディスク上のファイルを表すリソースを管理し、次のタスクを実行します：

- 起動時：ファイルを作成します
- 停止時：ファイルを正常に削除します
- チェック時：ファイルが存在するかどうかをチェックします
- クリーンアップ時：ファイルを強制的に削除します

Oracle Clusterwareに対するこの特定のリソースを記述するには、このリソース・クラスのすべての特徴的な属性を含むリソース・タイプを最初に作成する必要があります。この場合、記述する必要のある属性は、管理するファイルの名前のみです。次のステップは、リソースおよびそのエージェントを設定して、リソースの機能をテストする方法を示しています。

1. 提供されるdemoagent1.cppソース・ファイルおよびmakefileを使用してC++エージェントをコンパイルします。ローカルのコンパイラ、リンカーのパスおよびインストール場所に基づいたmakefileを変更します。出力は、demoagent1という実行可能ファイルです。この例は、実行可能ファイルが、クラスタの各ノードの/path/to/というディレクトリにあることを想定しています。
2. CRSCTLを使用して、次のように新しいリソース・タイプを追加します。

```
$ crsctl add type hotfile_type -basetype cluster_resource -attr  
"ATTRIBUTE=PATH_NAME, TYPE=string, DEFAULT_VALUE=default.txt,  
ATTRIBUTE=AGENT_FILENAME, TYPE=string, DEFAULT_VALUE=/path/to/demoagent1"
```

前述のコマンドの例では、PATH_NAMEがこのタイプの各ノードのディレクトリ・パスです。PATH_NAMEの値をディスク上の適切なディレクトリ場所に変更します。

AGENT_FILENAME属性は、このリソース・タイプのリソース管理コマンドを実装するエージェント・バイナリの場所を指定します。このステップにより、Oracle Clusterwareに新しいリソース・タイプが追加されます。

3. 上記ステップで定義されているタイプに基づいて、次のように新しいリソースを作成します。

```
$ crsctl add res file1 -type hotfile_type -attr "PATH_NAME=/var/log/file1.txt"  
$ crsctl add res file2 -type hotfile_type -attr "PATH_NAME=/var/log/file2.txt"
```

前述のコマンドにより、Oracle Clusterwareが管理および監視するfile1およびfile2というリソースが追加されます。

4. CRSCTLを使用して、次のようにリソースを起動および停止します。

```
$ crsctl start res file1  
$ crsctl start res file2  
$ crsctl relocate res file1  
$ crsctl stop res file2
```

リソースが起動および停止されると、Oracle Clusterwareはディスク・ファイルを作成および削除します。

関連トピック

- [Oracle Clusterware C Application Program Interfaceエージェント・フレームワーク](#)

Oracle Clusterwareへのリソースの登録

crsctl add resourceコマンドを使用して、Oracle Clusterwareにリソースを登録します。

アプリケーションをリソースとして登録するには:

```
$ crsctl add resource resource_name -type [-group group_name] resource_type  
[-file file_path] | [-attr "attribute_name='attribute_value', attribute_name='  
attribute_value', ..."]
```

リソースの作成対象のアプリケーションに基づいて、そのリソースの名前を選択します。たとえば、Apache Webサーバーのリソースを作成する場合は、このリソースの名前をmyApacheとします。-typeオプションの後に、既存のリソース・タイプの名前を指定します。オプションで、リソースを既存のリソース・グループに追加できます。

リソース属性は、-fileオプションで指定されたテキスト・ファイル、または-attrオプションの後の二重引用符(")で囲まれたリソース属性/値ペアのカンマ区切りのリストで指定できます。空白文字またはカンマで区切られた属性名およびカッコで囲まれた値は一重引用符(')で囲む必要があります。

属性ファイルの例を次に示します。

```
PLACEMENT=favored  
HOSTING_MEMBERS=node1 node2 node3  
RESTART_ATTEMPTS@CARDINALITYID(1)=0  
RESTART_ATTEMPTS@CARDINALITYID(2)=0  
FAILURE_THRESHOLD@CARDINALITYID(1)=2  
FAILURE_THRESHOLD@CARDINALITYID(2)=4  
FAILURE_INTERVAL@CARDINALITYID(1)=300  
FAILURE_INTERVAL@CARDINALITYID(2)=500  
CHECK_INTERVAL=2  
CARDINALITY=2
```

-attrオプションの使用例を次に示します。

```
$ crsctl add resource resource_name -type resource_type [-attr "PLACEMENT=  
favored', HOSTING_MEMBERS='node1 node2 node3', ..."]
```

関連トピック

- [ユーザー定義のリソースの追加](#)
- [crsctl add resource](#)
- [リソース属性](#)
- [リソース・グループ](#)

Oracle Clusterwareを使用した高可用性の実現の概要

Oracle Clusterwareでは、可用性が向上するような構成方法に基づいてリソースおよびリソース・グループが管理されます。

リソースおよびリソース・グループは、Oracle Clusterwareで次の操作が実行されるように構成できます。

- クラスタまたはサーバーの起動時のリソースおよびリソース・グループの起動
- 障害が発生した場合のリソースおよびリソース・グループの再起動
- 各サーバーへのリソースおよびリソース・グループの再配置(他のサーバーを使用できる場合)

Oracle Clusterwareを使用してアプリケーションを管理するには:

1. generic_applicationリソース・タイプを使用するか、スクリプト・エージェントにカスタム・スクリプトを記述するか、新しいエージェントを作成します。
2. アプリケーションをOracle Clusterwareにリソースとして登録します。

単一のアプリケーションで複数のリソースを登録する必要がある場合は、Oracle Clusterwareによって単一のリソースのように管理されるリソース・グループを作成できます。リソース・グループ内のリソース間の関連する依存性を定義することが必要となる場合があります。
3. リソースまたはリソース・グループに適切な権限を割り当てます。
4. リソースおよびリソース・グループを起動または停止します。

リソースで障害が発生すると、Oracle Clusterwareは、アプリケーションまたはプロセスをリソースとして登録したときに指定した属性値に基づいて、リソースの再起動を試行します。障害の発生したリソースがリソース・グループのクリティカルでないリソース・メンバーである場合、そのリソース・グループはONLINE状態を維持します。クラスタ内のサーバーで障害が発生した場合、障害が発生したサーバーで実行されるように指定されていたプロセスが別のサーバーで再起動されるようにリソースおよびリソース・グループを構成できます。Oracle Clusterwareでは、様々なリソース属性に基づいて様々な構成可能な例がサポートされています。

リソースをOracle Clusterwareに登録した場合、またはOracle Clusterwareでリソース・グループを作成した場合、アプリケーション関連の情報およびリソース関連の情報がOracle Cluster Registry (OCR)に格納されます。この情報には、次のものが含まれます。

- アクション・スクリプトまたはアプリケーション固有のエージェントへのパス：これは、Oracle Clusterwareによってアプリケーションで実行されるstartアクション、stopアクション、checkアクションおよびcleanアクションを定義するスクリプトまたはアプリケーション固有のエージェントへの絶対パスです。

関連項目:

これらのアクションの詳細は、「[Oracle Clusterwareのエージェント](#)」を参照してください。

- 権限: Oracle Clusterwareには、高可用性操作のために、アプリケーションのすべてのコンポーネントの制御に必要な権限(他のユーザーIDでプロセスを起動する権限を含む)があります。Oracle Clusterwareは、権限を持つユーザーとして実行し、正しい起動および停止プロセスでアプリケーションを制御する必要があります。
- リソース依存性: 操作の順序を示したり、クラスタ内のサーバーのリソース配置に影響するリソースおよびリソース・グループ間の関係を作成できます。たとえば、他のリソースが実行されている場合、Oracle Clusterwareは別のリソースに対するhard起動依存性が定義されているリソースのみを起動できます。Oracle Clusterwareでは、停止対象のリソースに依存する他のリソースが実行されている場合、そのリソースは停止されません。ただし、`crsctl stop resource -f`コマンドを使用して、停止対象のリソースに依存するすべてのリソースを最初に停止し、そのリソースを強制終了することができます。

関連トピック

- [リソース・グループ](#)
- [リソース属性](#)
- [リソースの状態](#)
- [リソース依存性](#)
- [リソースの配置](#)
- [Oracle Clusterwareのエージェント](#)

リソース属性

リソース属性は、Oracle Clusterwareで特定のリソース・タイプのリソースを管理する方法を定義します。各リソース・タイプには一意の属性セットがあります。一部のリソース属性はリソースの登録時に指定されますが、その他のリソース属性はOracle Clusterwareによって内部的に管理されます。

ノート:



新しいリソース属性を定義できる場合でも、使用できる文字は US-7 ASCII のみです。

関連トピック

- [Oracle Clusterwareのリソース・リファレンス](#)

リソースの状態

クラスタ内のすべてのリソースは、いつでも特定の状態になっています。その状態は、特定のアクションまたはイベントによって変更される可能性があります。

[表8-3](#)に、予想される[リソースの状態](#)を示します。

表8-3 予想されるリソースの状態

状態	説明
ONLINE	リソースは実行されています。
OFFLINE	リソースは実行されていません。
UNKNOWN	リソースの停止の試行に失敗しました。Oracle Clusterware では、この状態のリソースはアクティブに監視されません。アプリケーション固有のアクション(プロセスの停止など)を実行してリソースをオフラインにし、 <code>crsctl stop resource</code> コマンドを実行してリソースの状態を OFFLINE に再設定する必要があります。
INTERMEDIATE	次の 2 つのイベントのいずれかにより、リソースが INTERMEDIATE 状態になる場合があります。 1. Oracle Clusterware でリソースの状態を判別できませんが、リソースはオンラインになろうとしていたか、または最後に正確に認識されたときのリソースの状態はオンラインでした。通常、check アクションを阻止する条件が適用されなくなるため、リソースは時間の経過とともにこの状態から遷移します。 2. リソースが部分的にオンラインになっています。たとえば、Oracle Database VIP リソースのホーム・サーバーがクラスタから削除された場合、Oracle Database VIP リソースは別のサーバーにフェイルオーバーされます。ただし、この VIP がホーム・サーバー以外のサーバーに存在する間、アプリケーションはこの VIP を使用してデータベースにアクセスすることはできません。同様に、

状態	説明
	Oracle Database インスタンスが起動されていても、オープンされていない場合も、リソースは部分的にオンラインになり、このリソースは実行されていますが、サービスを提供することはできません。 Oracle Clusterware では、INTERMEDIATE 状態のリソースはアクティブに監視され、通常、ユーザーの介入は必要ありません。前述の理由 1 のためリソースが INTERMEDIATE 状態になっている場合、Oracle Clusterware は、リソースの状態が確立されるとすぐにリソースを INTERMEDIATE 状態から遷移させます。 前述の理由 2 のためリソースが INTERMEDIATE 状態になっている場合、リソースは、部分的にオンラインであるかぎり、この状態のままになります。たとえば、VIP のホーム・サーバーは、VIP がスイッチオーバーできるようにクラスタに再度参加する必要があります。データベース管理者は、データベース・インスタンスをオープンするコマンドを発行する必要があります。 ただし、いずれの場合も、Oracle Clusterware では、リソースは適切になるとすぐに INTERMEDIATE 状態から自動的に遷移します。STATE_DETAILS リソース属性を使用して、リソースが INTERMEDIATE 状態である理由を説明し、リソースをこの状態から遷移させるための解決策を提供します。

リソース依存性

リソースは他のリソースに依存するように構成でき、これによって、依存先リソースの特定の条件が満たされる場合にのみ、依存リソースを起動または停止できます。たとえば、Oracle Clusterwareでリソースを起動しようとする場合、初期リソースが依存するリソースは実行中で、同じ場所に存在する必要があります。Oracle Clusterwareでリソースをオンラインにすることができない場合は、初期(依存)リソースをオンラインにすることもできません。Oracle Clusterwareによってリソースが停止された場合またはリソースで障害が発生した場合は、依存リソースも停止されます。

一部のリソースでは、他のリソースより起動に必要な時間が長くなります。一部のリソースはサーバーが起動されるたびに起動する必要がありますが、その他のリソースは手動のstartアクションが必要です。これらのリソース固有の動作の例および他の多くの例は、各リソースを予測される動作および他のリソースとの関連から説明する必要があることを示しています。

リソースがOracleリソースに依存するように構成できます。ただし、リソースを作成する際、リソース名に接頭辞oraを使用しないでください。この接頭辞は、Oracleでのみ使用するために予約されています。

以前のバージョンのOracle Clusterwareには、REQUIRED_RESOURCESリソース属性およびOPTIONAL_RESOURCESリソース属性という2つの依存性仕様のみが含まれていました。REQUIRED_RESOURCESリソース属性は、リソースの起動依存性および停止依存性の両方に適用されます。

ノート:



REQUIRED_RESOURCES および OPTIONAL_RESOURCES リソース属性は application タイプのリソースの場合のみ引き続き使用可能です。これらを使用してリソース依存性を定義することは、Oracle Clusterware

12c 以降のリリースでは非推奨です。

リソース依存性は、起動および停止カテゴリに分類されます。この分割によって、リソースとリソース・タイプ間の起動依存性および停止依存性が向上され、拡張されます。

この項には次のトピックが含まれます：

- [起動依存性](#)
- [停止依存性](#)

起動依存性

Oracle Clusterwareでは、リソースの[起動試行の評価](#)が開始されると、そのリソースのプロファイルに含まれる起動依存性が考慮されます。START_DEPENDENCIESリソース属性を使用して、リソースの起動依存性を指定します。各依存性で修飾子を使用して、依存性をさらに詳細に構成できます。

この項では、次の起動依存性について説明します。

- [attraction](#)
- [dispersion](#)
- [exclusion](#)
- [hard](#)
- [pullup](#)
- [weak](#)

関連トピック

- [START_DEPENDENCIES](#)

attraction

リソースBに対するattraction依存性がリソースAに定義されている場合、Oracle ClusterwareはリソースBをホスティングしているサーバーにリソースAを配置します。この場合のリソースAのような依存リソースは、依存リソースでattraction依存性が定義されている依存先リソースが実行されているサーバーで実行される傾向があります。Oracle Clusterwareは、依存リソースが誘引されるリソースのサーバーに依存リソースを配置します。

次の制約を使用して、attraction起動依存性を構成できます。

- START_DEPENDENCIES=attraction(intermediate:resourceB)
intermediate修飾子を使用すると、INTERMEDIATE状態になっているリソースにリソースが誘引されるかどうかを指定できます。
- START_DEPENDENCIES=attraction(type:resourceB.type)
type修飾子を使用すると、特定のリソース・タイプに対して依存性を有効にするかどうかを指定できます。依存リソースは特定のタイプのリソースを最も多くホスティングしているサーバーに誘引されます。



ノート:

以前のバージョンの Oracle Clusterware では、現在非推奨の OPTIONAL_RESOURCES 属性を使用して

attraction 依存性が表現されています。

dispersion

リソースにdispersion起動依存性を指定すると、Oracle Clusterwareは、このリソースでdispersionが定義されているリソースが最も少ないサーバーでこのリソースを起動します。dispersionを使用するリソースでも、リソースを分散する十分なサーバーがない場合、同じサーバーで実行されることがあります。

次の修飾子を使用して、dispersion起動依存性を構成できます。

- `START_DEPENDENCIES=dispersion(intermediate:resourceB)`
intermediate修飾子を使用すると、リソースBがONLINE状態またはINTERMEDIATE状態のいずれであっても、Oracle ClusterwareによってリソースAが分散されるように指定できます。
- `START_DEPENDENCIES=dispersion:active(resourceB)`
通常、dispersionはリソースの起動時にのみ適用されます。リソースの起動時に十分なサーバーがないために、相互に分散するリソースが同じサーバーで起動される場合、サーバーがクラスタに追加されてもリソースが実行されていれば、Oracle Clusterwareはそのリソースを変更しません。active修飾子を指定する場合、Oracle Clusterwareは、新しいサーバーがクラスタに追加されると、リソースに対してdispersionを再適用します。
- `START_DEPENDENCIES=dispersion(pool:resourceB)`
Oracle Clusterwareがリソースを別のサーバーではなく別のサーバー・プールに分散させるように指定するにはpool修飾子を使用します。

exclusion

exclusion起動依存性には、起動中にリソース間の排他的関係を定義する句が含まれています。exclusion起動依存性が定義されているリソースは、同一のノード上で実行できません。たとえば、リソースBに対するexclusion起動依存性がリソースAに定義されている場合、CRSDポリシーは、リソースAを起動する必要があるサーバー上でリソースBが既に実行されていると、次のオプションを提供します。

- リソースBが既に実行されている場合、リソースAの起動を拒否します。
- リソースBに割り込んでリソースAを起動します。割り込み操作には2種類あります。
 - リソースBが停止され、可能であれば、別のノードで再起動されます。続いてリソースAが起動されます。
 - リソースAが最初に起動されます。続いてリソースBが停止され、可能であれば、別のノードで再起動されます。

次の修飾子を使用して、exclusion起動依存性を構成できます。

- `START_DEPENDENCIES=exclusion([[preempt_pre: | preempt_post:]] target_resource_name | type:target_resource_type*)`
指定されているすべての修飾子は、リソースまたはリソース・タイプに基づきます。Oracle Clusterwareは、リソース依存性ツリーに基づいて、1つのexclusion依存性のみを許可します。preempt修飾子がない場合、CRSDは、そのターゲット・リソースがオフラインのときにのみ起動しようとします。
 - `preempt_pre`: この割り込み修飾子を選択すると、CRSDは、指定したターゲット・リソースまたは特定のリソース・タイプが定義するリソースを起動してから、ソースのリソースを起動します。停止されたリソースの再起動が可能な場合、CRSDは、割り込みリソースを起動するのと同時にこれを実行できます。
 - `preempt_post`: この割り込み修飾子を選択すると、CRSDはソースのリソースを起動した後に停止し、可能であれば、指定のターゲットのリソースまたは特定のリソース・タイプが定義するリソースを再配置します。

CRSDがターゲットのリソースを停止できない場合、またはソースのリソースを起動できない場合、すべての操作が失敗します。次にOracle Clusterwareは、可能であれば、問題のあるリソースを元の状態に戻そうとします。

hard

依存リソースを起動する前に別のリソースが実行されている必要がある場合は、そのリソースにhard起動依存性を定義します。たとえば、リソースBに対するhard起動依存性がリソースAに定義されている場合は、リソースAを起動する前にリソースBが実行されている必要があります。同様に、両方のリソース(AとB)が最初にオフラインになっている場合、リソースAの依存性を満たすためにリソースBが最初に起動されます。

ノート:



hard 起動依存性を持つリソースには pullup 起動依存性も定義することをお勧めします。

次の制約を使用して、hard起動依存性を構成できます。

- `START_DEPENDENCIES=hard(global:resourceB)`

デフォルトでは、リソースAおよびBを同じサーバーに配置する必要があります。global修飾子を使用してリソースを同じ場所に配置する必要がないように指定します。たとえば、リソースBに対するhard(global:resourceB)起動依存性がリソースAに定義されている場合は、リソースBがクラスタ内のノードで実行されていると、リソースAを起動できます。

- `START_DEPENDENCIES=hard(intermediate:resourceB)`

intermediate修飾子を使用すると、依存先リソースの状態がONLINEまたはINTERMEDIATEのいずれかの場合に依存リソースを起動できるように指定できます。

- `START_DEPENDENCIES=hard(type:resourceB.type)`

type修飾子を使用すると、hard起動依存性を特定のリソースに対して有効にするか、リソース・タイプに対して有効にするかを指定できます。たとえば、resourceB.typeタイプに対するhard起動依存性がリソースAに定義されている場合は、resourceB.typeタイプのリソースが実行されているとリソースAを起動できます。

- `START_DEPENDENCIES=hard(uniform:resourceB)`

uniform修飾子を使用して、リソースBのすべてのインスタンスの起動を試行しますが、最低1つのインスタンスが起動されて依存性を満たす必要があります。

- `START_DEPENDENCIES=hard(resourceB, intermediate:resourceC, intermediate:global:type:resourceC.type)`

修飾子を組み合わせて、START_DEPENDENCIESリソース属性に複数のリソースを指定できます。

ノート:



修飾子句は、カンマで区切ります。type 修飾子句は必ずリストの最後の修飾子句である必要があり、type 修飾子を必ずタイプの直前に指定する必要があります。

pullup

リソースBが起動するたびにリソースAを自動的に起動するには、pullup起動依存性を使用します。この依存性は、リソースAが実行されていない場合にのみリソースAに影響します。他の依存性の場合と同様に、pullupによって、依存リソースがすべ

てのサーバーで起動することがあります。リソースAがリソースBに依存し、リソースBが失敗した後リカバリした場合に、リソースAが再起動されるように、hard停止依存性がある場合は必ずpullup依存性を使用します。

ノート:



hard 起動依存性を持つリソースには pullup 起動依存性も定義することをお勧めします。

次の制約を使用して、pullup起動依存性を構成できます。

- `START_DEPENDENCIES=pullup(intermediate:resourceB)`
intermediate修飾子を使用すると、リソースBがONLINE状態またはINTERMEDIATE状態のいずれかの場合にリソースAを起動できるように指定できます。
複数のリソースに対するpullup依存性がリソースAに定義されている場合、リソースAが依存するすべてのリソースが起動されたときにのみリソースAが起動されます。
- `START_DEPENDENCIES=pullup:always(resourceB)`
always修飾子を使用して、リソースAが依存するリソースのTARGET属性の値がONLINE またはOFFLINE のいずれであっても、Oracle ClusterwareでリソースAを起動するかどうかを指定できます。デフォルトでは、always修飾子を使用せず、依存先リソースのTARGET属性の値がONLINEの場合にのみpullupによってリソースが起動されます。
- `START_DEPENDENCIES=pullup(type:resourceB.type)`
type修飾子を使用すると、特定のリソース・タイプに対して依存性を有効にするように指定できます。

weak

リソースBに対するweak起動依存性がリソースAに定義されており、リソースBが実行されていない場合は、リソースAの起動を試行すると、リソースBの起動が試行されます。ただし、リソースBの起動の試行結果は、リソースAの起動結果に影響しません。

次の制約を使用して、weak起動依存性を構成できます。

- `START_DEPENDENCIES=weak(global:resourceB)`
デフォルトでは、リソースAおよびBは同じ場所に配置する必要があります。global修飾子を使用してリソースを同じ場所に配置する必要がないように指定します。たとえば、リソースBに対するweak(global:resourceB)起動依存性がリソースAに定義されている場合は、リソースBがクラスタ内のノードで実行されていると、リソースAを起動できます。
- `START_DEPENDENCIES=weak(concurrent:resourceB)`
concurrent修飾子を使用すると、リソースAおよびリソースBが同時に起動できるように指定できます。
- `START_DEPENDENCIES=weak(type:resourceB.type)`
type修飾子を使用すると、resourceB.typeなど、特定のリソース・タイプのリソースに対して依存性を有効にすることを指定できます。
- `START_DEPENDENCIES=weak(uniform:resourceB)`
uniform修飾子を使用して、リソースBのすべてのインスタンスの起動を試行します。

停止依存性

Oracle Clusterwareでは、リソースが停止される(リソースの状態がONLINEから他の状態に変更される)たびに、リソース間

の停止依存性が考慮されます。

hard

リソースBに対するhard停止依存性がリソースAに定義されている場合は、リソースBの実行が停止されるときにリソースAも停止される必要があります。2つのリソースは、構成方法に応じて、起動または別のサーバーへの再配置を試行する場合があります。hard停止依存性を持つリソースにはhard起動依存性も定義することをお勧めします。

次の修飾子を使用して、hard停止依存性を構成できます。

- `STOP_DEPENDENCIES=hard(intermediate:resourceB)`
`intermediate`修飾子を使用すると、リソースAをオンラインのままにするために、リソースBをONLINEまたはINTERMEDIATE状態のいずれかにする必要があるかどうかを指定できます。
- `STOP_DEPENDENCIES=hard(global:resourceB)`
`global`修飾子を使用すると、リソースAをオンラインのままにするために、リソースBが同じサーバーまたはクラスタ内の任意のサーバーに存在する必要があるかどうかを指定できます。この制約が指定されていない場合、リソースAおよびBは同じサーバーで実行されている必要があります。この条件が満たされなくなると、リソースAはOracle Clusterwareによって停止されます。
- `STOP_DEPENDENCIES=hard(shutdown:resourceB)`
`crsctl stop crs`コマンドまたは`crsctl stop cluster`コマンドのどちらかを使用してOracle Clusterwareスタックを停止する場合にのみ`shutdown`修飾子を使用してリソースを停止します。

関連トピック

- [STOP_DEPENDENCIES](#)

リソース状態のリカバリに対するリソース依存性の影響

リソースの実行を続行する計画である場合に、リソースが実行されている状態から実行されていない状態に遷移する場合、この遷移はリソース障害と呼ばれます。

この時点で、Oracle Clusterwareでは、リソース状態のリカバリ手順(リソースの高可用性ポリシーおよびその時点のエンティティの状態に応じて、リソースのローカルでの再起動、別のサーバーへの再配置または依存リソースの停止が試行される)が適用されます。

複数のリソースが相互に依存している場合、そのいずれかの障害が原因で他のリソースにも障害が発生することがあります。ほとんどの場合、障害が検出される順序を制御または予測することは困難です。たとえば、リソースAがリソースBに依存する場合でも、Oracle Clusterwareでは、リソースBの障害がリソースAの障害よりも後で検出されることがあります。

このように、障害の順序が予測可能でない場合、Oracle Clusterwareでは依存リソースの再起動がパラレルで試行される可能性があり、これによって、依存先リソースが不適切な順序で再起動されるため、最終的に一部のリソースの再起動に失敗します。

この場合、hard停止依存性およびpullup依存性のいずれかまたはその両方が使用されていると、Oracle Clusterwareでは依存リソースのローカルでの再起動が再試行されます。たとえば、リソースAにリソースBに対するhard停止依存性またはpullup依存性のいずれかあるいはその両方が定義されている場合、リソースBで障害が発生したためにリソースAでも障害が発生すると、Oracle Clusterwareではその両方のリソースの再起動が同時に試行される可能性があります。リソースAの再起動の試行に失敗した場合、Oracle Clusterwareでは、リソースBが正常に再起動されるとすぐにリソースAの再起動が試行されます。

リソースの配置

起動試行の評価の一部としてOracle Clusterwareで最初に決定される必要があるのは、リソースを起動(または配置)する場所です。

コール元がターゲットのサーバーを名前指定する場合、このような決定は簡単です。ただし、ターゲットのサーバーが指定されない場合、Oracle Clusterwareでは、リソースの構成およびクラスタの現在の状態で配置に最適なサーバーの検索が試行されます。

Oracle Clusterwareでは、最初にリソースの配置ポリシーが考慮され、ポリシーに適合しないサーバーは除外されます。

Oracle Clusterwareでは、リソースのPLACEMENTリソース属性の値に応じた特定の順序で残りのサーバーがソートされます。

これが考慮された結果、Oracle Clusterwareによってリソースを起動できる候補サーバーのリストが最大2つ作成されます。一方のリストには優先サーバーが含まれ、もう一方のリストには配置可能なサーバーが含まれます。リソースのPLACEMENTリソース属性の値がbalancedまたはrestrictedに設定されている場合、優先サーバーのリストは空になります。リソースの配置ポリシーによって、リソースを実行するサーバーが決定します。Oracle Clusterwareでは、優先リストにサーバーが示される場合、配置可能なサーバーよりも優先サーバーが考慮されます。

次に、リソースの依存性がある場合は、Oracle Clusterwareで考慮されてリソースの配置場所が判断されます。

[attraction](#)起動依存性および[dispersion](#)起動依存性は、一部の依存性修飾子と同様にリソースの配置決定に影響を及ぼします。Oracle Clusterwareによってこれらの配置ヒントが適用され、前述の2つのリストでサーバーがさらに順序付けられます。Oracle Clusterwareでは、リソースの配置ポリシーの影響と依存性の影響とを区別するために、サーバーの各リストが別々に処理されることに注意してください。

最後に、Oracle Clusterwareでは、優先サーバーがリストされている場合はそのリストから最初のサーバーが選択されます。Oracle Clusterwareでは、優先サーバーのリストにサーバーが示されていない場合、配置可能なサーバーがリストされていれば、その配置可能なサーバーのリストから最初のサーバーが選択されます。どちらのリストにもサーバーがない場合、Oracle Clusterwareではリソース配置エラーが生成されます。

ノート:



Oracle Clusterware で起動を試行しているリソースに関連する配置ポリシーまたはリソースの依存性のいずれも配置の決定に影響を及ぼしません。

関連トピック

- [アプリケーションの配置ポリシー](#)

リソースとしてのアプリケーションの登録

この項では、アプリケーションをOracle Clusterwareのリソースとして登録する手順の例を示します。

この手順では、(例として) Apache WebサーバーをOracle Clusterwareにリソースとして追加する方法を示します。この項の例では、Oracle ClusterwareおよびOracle Clusterwareで管理されるアプリケーションを所有するユーザーまたはグループに対する完全な管理権限がOracle Clusterware管理者に付与されていることを前提とします。登録処理が完了すると、オペレーティング・システム・ユーザーのかわりにOracle Clusterwareでアプリケーションを起動できます。

Oracle Clusterwareでは、登録されたリソースの所有者とユーザーは区別されます。リソースの所有者は、エージェントが実行されるオペレーティング・システム・ユーザーです。ユーザーおよび所有者の権限は、リソースのACLリソース属性によって定義さ

れます。リソースを変更できるのは、rootのみです。

ノート:

- 接頭辞に `crs_` を持つ Oracle Clusterware コマンドは、このリリースでサポート対象外となり、使用できなくなりました。これらのコマンドは、CRSCTL コマンドによって置換されます。CRSCTL コマンドのリストおよび対応する `crs_` コマンドは、「Oracle Clusterware 制御(CRSCTL)ユーティリティ・リファレンス」を参照してください。
- My Oracle Support で指示された場合を除き、接頭辞 `ora` を持つ名前のリソースで(Oracle リソースであるため)CRSCTL コマンドを使用しないでください。

Oracle リソースを構成するには、サーバー制御ユーティリティ(SRVCTL)を使用します。SRVCTL によって、すべての構成可能なオプションが提供されます。

関連トピック

- [Oracle Clusterware制御\(CRSCTL\)ユーティリティ・リファレンス](#)
- [ロール別管理](#)

Oracle Clusterwareで管理されるアプリケーションVIPの作成

アプリケーションVIPは、Oracle Clusterwareによって管理されるクラスタ・リソースです(Oracle ClusterwareではアプリケーションVIP用の標準VIPエージェントが提供されます)。

アプリケーションのクライアントがネットワークを介してアプリケーションにアクセスし、アプリケーションの配置ポリシーによって別のノードへのフェイルオーバーが可能である場合は、アプリケーションが依存する仮想インターネット・プロトコル・アドレス(VIP)を登録する必要があります。システムで、クラスタにデプロイされたすべてのVIPの動作の一貫性が保持されるように、新しいアプリケーションVIPはこのVIPタイプに基づく必要があります。

Oracle Clusterwareが管理するその他のリソースを追加するときと同じ方法でVIPを追加することができますが、Oracleでは、`Grid_home/bin/appvipcfg` コマンドライン・ユーティリティを使用して、`ora.net1.network` がデフォルトで作成されるデフォルトのネットワークにアプリケーションVIPを作成または削除することをお勧めします。

アプリケーションVIPを作成するには、次の構文を使用します。

```
appvipcfg create -network=network_nummber -ip=ip_address -vipname=vip_name  
-user=user_name [-group=group_name] [-failback=0 | 1]
```

ノート:

VIP 名は、リソースをオンラインにしたまま、リソースを再起動せずに変更できます。

デフォルトのネットワークにVIPを作成する場合、`-network=1`を設定します。

デフォルト以外のネットワークにアプリケーションVIPを作成するには、場合によって、最初に `srvctl add network` コマンドを使用してネットワークを作成する必要があります。その後、`-network=non-default_network_number` を設定してアプリケーションVIPを作成できます。

Oracle Flex Clusterでは、次のように`srvctl add network`コマンドを使用して、アプリケーションをハブではないノード上で実行できるように、アプリケーションVIPのハブではないノード・ネットワーク・リソースを追加することもできます。

```
srvctl add network -netnum=network_number -subnet subnet/netmask[/if1[|if2|...]]
```

アプリケーションVIPを削除するには、次の構文を使用します。

```
appvipcfg delete -vipname=vip_name
```

前述の構文の例で、`network_number`はネットワークの番号、`ip_address`はIPアドレス、`vip_name`はVIPの名前、`user_name`はOracle Databaseをインストールしたユーザーの名前、`group_name`はグループの名前です。`-failback`オプションのデフォルト値は0です。オプションを1に設定すると、VIP(および、VIPに依存するすべてのリソース)は、元のノードが再び使用可能になると、元のノードにフェイルバックします。

ノート:



`-ip=ip_address` パラメータは必須ですが、グリッド・プラグ・アンド・プレイおよび DHCP と GNS が構成されている場合、このパラメータでは常に DHCP サーバーからの IP アドレスが取得され、コマンドで指定された IP アドレスは無視されます。DHCP を使用する場合、`-vipname=vip_name` パラメータの値も無視されます。

たとえば、`root`として次のコマンドを実行します。

```
# Grid_home/bin/appvipcfg create -network=1 -ip=148.87.58.196 -vipname=appsVIP -user=root
```

スクリプトには、アプリケーションVIPリソースを所有するユーザーのほか、ネットワーク番号、IPアドレス、VIPリソースの名前のみがが必要です。VIP関連の操作には`root`権限が必要であるため、通常、VIPリソースは`root`が所有します。

アプリケーションVIPを削除するには、`delete`オプションで同じスクリプトを使用します。このオプションはVIPの名前をパラメータとして受け入れます。たとえば:

```
# Grid_home/bin/appvipcfg delete -vipname=appsVIP
```

この構成スクリプトを使用してアプリケーションVIPを作成したら、次のコマンドを使用してVIPプロファイルを表示できます。

```
$ Grid_home/bin/crsctl status res appsVIP -p
```

`Grid_home/bin/crsctl modify res`コマンドを使用して次のパラメータの検証と、必要に応じて変更を行います。

`appvipcfg`スクリプトは、`-network=1`に設定されていても`-network`オプションを指定する必要があります。

Oracle Databaseインストール所有者として、VIPリソースを起動します。

```
$ crsctl start resource appsVIP
```

関連トピック

- [Oracle Real Application Clusters管理およびデプロイメント・ガイド](#)
- [Oracle Flex Cluster](#)
- [Oracle Clusterwareのリソース・リファレンス](#)

Oracle Enterprise ManagerによるアプリケーションVIPの追加

Oracle Enterprise Managerを使用してアプリケーションVIPを追加するには、次の手順を実行します。

1. Oracle Enterprise Manager Cloud Controlにログインします。
2. 変更するクラスタ・ターゲットを選択します。
3. クラスタ・ターゲットのメニューから、「管理」→「リソース」→「管理」を選択します。
4. クラスタ管理者のユーザー名およびパスワードを入力して、「リソースの管理」ページを表示します。
5. 「アプリケーションVIPの追加」をクリックします。
6. 「名前」フィールドにVIPの名前を入力します。
7. 「ネットワーク番号」フィールドにネットワークの番号を入力します。
8. 「インターネット・プロトコル・アドレス」フィールドにVIPのIPアドレスを入力します。
9. 「プライマリ・ユーザー」フィールドにrootと入力します。Oracle Enterprise Managerは、ログイン時に使用するユーザー名をデフォルト値とします。
10. VIPをすぐに起動する場合は、「作成後にリソースを開始」を選択します。
11. 「続行」をクリックして「確認: VIPリソースの追加」ページを表示します。
12. クラスタの資格証明としてrootおよびルート・パスワードを入力します。
13. 「続行」をクリックしてアプリケーションVIPを作成します。

ユーザー定義のリソースの追加

Oracle Clusterwareにはいつでもリソースを追加できます。

ただし、別のリソースに依存しているリソースを追加する場合は、最初にそのリソースが依存しているリソースを追加します。

この項の例では、クラスタへのリソースの追加を簡単にするために、アクション・スクリプトmyApache.scrが各ノードの/opt/cluster/scriptsディレクトリに存在していると想定しています。また、アプリケーションをホストするために複数のプールが作成されていると想定します。このサーバー・プールは汎用のサブプールではありませんが、かわりに、最上位のサーバー・プールでアプリケーションをホスティングするために使用されます。

ノート:



スクリプトのメンテナンスを軽減するために、Oracle Automatic Storage Management Cluster File System (Oracle ACFS)などの共有ストレージを使用してアクション・スクリプトを格納することをお勧めします。

この項には次のトピックが含まれます:

- [デプロイメント・スキームの決定](#)
- [特定のサーバー・プールへのリソースの追加](#)
- [サーバー固有のデプロイメントを使用したリソースの追加](#)
- [generic_applicationリソース・タイプを使用するリソースの作成](#)

関連トピック

- [サード・パーティ製アプリケーションのアクション・スクリプトの例](#)

デプロイメント・スキームの決定

アプリケーションの管理者またはポリシー管理を使用するかどうかを決定する必要があります。管理者管理は、クラスタ構成が変更される可能性が低い、小規模な2ノード構成に使用します。ポリシー管理は、クラスタが3つ以上のノードで構成されている場合に、より動的な構成に使用します。たとえば、リソースがノード1およびノード2のみで実行されている場合、これらのノードにの

に必要なファイルが存在するため、管理者管理がより適切である可能性があります。

Oracle Clusterwareでは、匿名サーバーで構成され、必要なプール・サイズに厳密に基づく、アクセス制御されたサーバー・プールでのアプリケーションのデプロイメントがサポートされています。この場合、管理者定義のクラスタ・ポリシーを使用して、必要なサイズおよび重要度のレベルで、サーバー割当てを制御する必要があります。または、厳密なサーバー割当てまたは優先サーバー割当てを使用することもでき、この場合、リソースは具体的な名前が指定されたサーバーで実行されます。これは、以前のリリースのOracle Clusterwareで使用可能な既存のモデル(現在の管理者管理)を表します。

概念上、両方のデプロイメント・スキームで開発およびデプロイされたアプリケーションをホスティングするクラスタは、サーバーの、論理的に分割された2つのグループと考えることができます。一方のサーバー・グループはサーバー・プールに使用され、ロールの分離およびサーバーの容量制御が可能になります。もう一方のサーバー・グループでは、クラスタ内の名前付きのサーバーに基づく固定割当てが想定されます。

いずれかのデプロイメント・スキームを使用してアプリケーションを管理するには、クラスタにリソースを追加する前にサーバー・プールを作成する必要があります。Genericという名前の組込みサーバー・プールは、管理者ベース管理のアプリケーションによって使用されるサーバーを常に所有します。汎用サーバー・プールは論理的な分割で、異なる管理スキームを使用してクラスタの2つの部分を分割するために使用できます。

サード・パーティの開発者がこのモデルを使用してアプリケーションをデプロイするには、サーバー・プールを使用する必要があります。名前付きのサーバーに基づく、既存のアプリケーション開発およびデプロイメント・モデルを利用するには、汎用サブプール(汎用プールが親であるサーバー・プールで、サーバー・プール属性のPARENT_POOLSで定義される)を使用する必要があります。親として汎用を使用するサブプールを作成し、サーバーをサブプール定義に名前で列挙することによって、名前付きのサーバーが、汎用内に存在すること、および名前付きのサーバー・モデルを使用するアプリケーション専用で使用されることがアプリケーションで保証されます。

特定のサーバー・プールへのリソースの追加

crsctl add resourceコマンドを使用して、リソースをサーバー・プールに追加します。

ポリシー・ベースのデプロイメント・スキームを使用して、特定のサーバー・プールにApache Webサーバーをリソースとして追加するには、Apacheサーバーを実行するユーザーとして次のコマンドを実行します(Apache Serverの場合、これは通常rootユーザーです)。

```
$ crsctl add resource myApache -type cluster_resource -attr
"ACTION_SCRIPT=/opt/cluster/scripts/myapache.scr,
  PLACEMENT=restricted,
  SERVER_POOLS=server_pool_list,
  CHECK_INTERVAL=30,
  RESTART_ATTEMPTS=2,
  START_DEPENDENCIES=hard(appsvip),
  STOP_DEPENDENCIES=hard(appsvip)"
```

前述の例では、クラスタに追加されるリソースの名前はmyApacheです。

ノート:



- エラーにならないように、カンマ区切りまたはスペース区切りの属性値は単一引用符(' ')で囲む必要があります。単独の属性値を単一引用符で囲んだ場合、引用符は無視されてエラーは発生しません。
- リソース名の先頭をピリオドまたは ora という文字列にすることはできません。

リソースは次のように構成されます。

- このリソースはcluster_resourceタイプです。
- ACTION_SCRIPT=/opt/cluster/scripts/myapache.scr: 必要なアクション・スクリプトへのパスです。
- PLACEMENT=restricted
- SERVER_POOLS=server_pool_list: このリソースは、空白区切りリストで指定されているサーバー・プールでのみ実行できます。
- CHECK_INTERVAL=30: Oracle Clusterwareによって、リソースのステータスの確認のために30秒ごとにリソースがチェックされます。
- RESTART_ATTEMPTS=2: Oracle Clusterwareによって、リソースが別のノードにフェイルオーバーされる前に、リソースの再起動が2回試行されます。
- START_DEPENDENCIES=hard(appsvip): このリソースには、appsvipリソースに対するhard起動依存性が定義されています。myApacheを起動するには、appsvipリソースがオンライン状態である必要があります。
- STOP_DEPENDENCIES=hard(appsvip): このリソースには、appsvipリソースに対するhard停止依存性が定義されています。appsvipリソースがオフラインになると、myApacheリソースは停止します。

関連トピック

- [サード・パーティ製アプリケーションのアクション・スクリプトの例](#)
- [アプリケーションの配置ポリシー](#)

サーバー固有のデプロイメントを使用したリソースの追加

名前付きのサーバー・デプロイメントを使用するリソースとしてApache Webサーバーを追加する場合、定義上、汎用サーバー・プールのサブプールであるサーバー・プールにリソースが追加されていると想定します。crsctl add serverpoolコマンドを使用して、汎用サブプールであるサーバー・プールを作成します。これらのサーバー・プールでは、汎用サーバー・プールはPARENT_POOLSサーバー・プール属性で親として定義されます。また、SERVER_NAMESパラメータにはサーバー名のリストが含まれ、それぞれのプールに割り当てる必要があるサーバーが指定されます。たとえば:

```
$ crsctl add serverpool myApache_sp -attr  
"PARENT_POOLS=Generic, SERVER_NAMES=host36 host37"
```

サブプールを作成した後、Apache Webサーバー・リソースを次のように追加します。

```
$ crsctl add resource myApache -type cluster_resource -attr  
"ACTION_SCRIPT=/opt/cluster/scripts/myapache.scr,  
  PLACEMENT='restricted',  
  SERVER_POOLS=myApache_sp,  
  CHECK_INTERVAL='30',  
  RESTART_ATTEMPTS='2',  
  START_DEPENDENCIES='hard(appsvip)',  
  STOP_DEPENDENCIES='hard(appsvip)'"
```

ノート:



リソース名の先頭をピリオドまたは ora という文字列にすることはできません。

また、サーバー固有のデプロイメントを使用してリソースを追加する場合、SERVER_POOLSリソース・パラメータにリストされたサーバー・プールは、汎用サーバー・プールの下位プールである必要があることに注意してください。

関連トピック

- [ポリシー・ベースのクラスタおよび容量の管理](#)

generic_applicationリソース・タイプを使用するリソースの作成

高可用性が必要で、アクション・スクリプトの作成が不要なタイプのアプリケーションのモデル化する場合は、`crsctl add resource`コマンドでgeneric_applicationリソース・タイプを使用してリソースを作成します。

この項には、generic_applicationリソース・タイプを使用するリソースをLinux/UNIXプラットフォームで作成する2つの例が示されています。

次のコマンドの例では、高可用性のSambaサーバー・リソースが作成されます。

```
$ crsctl add resource samba1 -type generic_application -attr
"START_PROGRAM='/etc/init.d/smb start',
STOP_PROGRAM='/etc/init.d/smb stop',
CLEAN_PROGRAM='/etc/init.d/smb stop',
PID_FILES='/var/run/smbd.pid,/var/run/nmbd.pid'"
```

前述の例で、リソースを定義する属性は次のように構成されます。

- `START_PROGRAM='/etc/init.d/smb start'`: この属性には、Sambaサーバーを起動するスクリプトへの完全なパスおよび引数が含まれています
- `STOP_PROGRAM='/etc/init.d/smb stop'`: この属性には、Sambaサーバーを停止するスクリプトへの完全なパスおよび引数が含まれています
- `CLEAN_PROGRAM='/etc/init.d/smb stop'`: この属性には、Sambaサーバーの起動または停止に障害が起きた場合に、サーバーを強制的に終了し、クリーンアップするスクリプトの完全なパスおよび引数が含まれています
- `PID_FILES='/var/run/smbd.pid,/var/run/nmbd.pid'`: この属性には、プロセスID (PID)をリストしているテキスト・ファイルのパスが含まれており、このリストを監視して、Sambaサーバーおよびそのコンポーネントすべてが実行されているようにする必要があります

ノート:

- この Samba サーバー構成にスクリプト・ベースの監視が必要な場合、次のように、PID_FILES の代わりに、CHECK_PROGRAMS 属性を使用できます。

```
CHECK_PROGRAMS='/etc/init.d/smb status'
```

- Samba サーバー・リソースの HOSTING_MEMBERS、SERVER_POOLS、PLACEMENT および CARDINALITY 属性を構成して、標準的な Oracle Clusterware の配置およびカーディナリティ・プロパティを指定できます。

2番目のコマンドの例では、高可用性のデータベース・ファイル・サーバー(DBFS)リソースが作成されます。DBFSは、Oracle Databaseに格納されたデータにアクセスできるFilesystem in Userspace (FUSE)ファイル・システムを提供します。

generic_applicationリソース・タイプを使用してDBFSファイル・システムに相当するリソースを定義できます。このDBFSリソースを使用して、DBFSファイル・システム・マウント・ポイントを起動、停止、監視およびフェイルオーバーすることができます。このリソースを作成するコマンドの構文は、次のとおりです。

```
$ crsctl add resource dbfs1 -type generic_application -attr
```

```
"START_PROGRAM='/app/oracle/12.2/bin/dbfs_client -o wallet
/@inst1 /scratch/mjk/data/dbfs_mount',
STOP_PROGRAM='/bin/fusermount -u /scratch/mjk/data/dbfs_mount',
CHECK_PROGRAMS='ls /scratch/mjk/data/dbfs_mount/dbfsdata1',
ENVIRONMENT_VARS='ORACLE_HOME=/app/oracle/12.2,
LD_LIBRARY_PATH=/app/oracle/12.2/lib:/app/oracle/12.2/rdbms/lib,
TNS_ADMIN=/app/oracle/12.2/network/admin',
CLEAN_PROGRAM='/bin/fusermount -u -z /scratch/mjk/data/dbfs_mount',
START_DEPENDENCIES='hard(ora.inst1_srv.svc)',
STOP_DEPENDENCIES='hard(ora.inst1_srv.svc)'"
```

上記の例には、必須のSTART_PROGRAM、STOP_PROGRAM、CHECK_PROGRAMSおよびCLEAN_PROGRAM属性の他に、次のような属性が含まれています。

- ENVIRONMENT_VARS属性は、プログラムを起動したり停止するときに渡されるカスタム環境変数を指定します。
- START_DEPENDENCIESおよびSTOP_DEPENDENCIES依存属性は、DBFSファイル・システムのデータベース・ストアの基になっているデータベース・サービスに起動依存性および停止依存性を作成します。

DBFSファイル・システムのアプリケーション要件に基づいた上位レベルのアプリケーション・リソースのDBFSリソースに依存性を作成できます。

ノート:



- 前述の構文に示されている ORACLE_HOME ディレクトリは例です。
- DBFS ファイル・システム・リソースの HOSTING_MEMBERS、SERVER_POOLS、PLACEMENT および CARDINALITY 属性を構成して、Oracle Clusterware 標準の配置およびカーディナリティ・プロパティを指定できます。

関連トピック

- [Oracle Clusterwareのリソース・タイプ](#)
- [Oracle Database SecureFiles and Large Objects開発者ガイド](#)

Oracle Enterprise Managerを使用したリソースの追加

Enterprise Managerを使用してリソースを追加します。

Oracle Enterprise Managerを使用してOracle Clusterwareにリソースを追加するには:

1. Oracle Enterprise Manager Cloud Controlにログインします。
2. 変更するクラスタ・ターゲットを選択します。
3. クラスタ・ターゲットのメニューから、「管理」→「リソース」→「管理」を選択します。
4. クラスタ管理者のユーザー名およびパスワードを入力して、「リソースの追加」ページを表示します。
5. 「名前」フィールドにリソースの名前を入力します。



ノート:

リソース名の先頭をピリオドまたは ora という文字列にすることはできません。

6. 「リソース・タイプ」ドロップダウン・リストからcluster_resourceまたはlocal_resourceのいずれかを選択します。

7. 必要に応じて、「説明」フィールドにリソースの説明を入力します。
8. リソースをすぐに起動する場合は、「作成後にリソースを開始」を選択します。
9. 「配置」セクションのオプションのパラメータは、Oracle Clusterwareによってリソースが配置されるクラスタ内の場所を定義します。

この項の属性は、[「Oracle Clusterwareのリソース・リファレンス」](#)で説明されている属性に対応しています。

10. 「アクション・プログラム」セクションの「アクション・プログラム」ドロップダウン・リストから、リソースを管理するためにOracle Clusterwareでアクション・スクリプトをコールするか、エージェント・ファイルをコールするか、またはその両方をコールするかを選択します。

また、ドロップダウン・リストでの選択に応じて、スクリプト、ファイルまたはその両方へのパスを指定する必要があります。

「アクション・スクリプト」を選択した場合、「新規アクション・スクリプトの作成」をクリックし、Oracle Enterprise Managerアクション・スクリプト・テンプレートを使用してリソース用のアクション・スクリプトを作成することができます(作成済でない場合)。

11. これ以上のリソースの構成を行うには、「属性」をクリックします。このページでは、開始属性、停止属性、ステータス属性、オフライン監視および定義した属性を構成できます。
12. 「拡張設定」をクリックし、より詳細なリソース属性の構成を有効にします。
13. 「依存性」をクリックして、リソース間の起動および停止の依存性を構成します。
14. リソースの構成が終了したら、「発行」をクリックします。

関連トピック

- [アプリケーションの配置ポリシー](#)
- [リソース依存性](#)

リソース権限の変更

Oracle Clusterwareでは、リソースを追加したユーザーの権限に基づいてリソースが管理されます。最初にリソースを追加したユーザーがリソースを所有し、リソースはリソースの所有者として実行されます。特定のリソースはrootとして管理する必要があります。root以外のユーザーが、rootとして実行する必要があるリソースを追加する場合は、次のように、rootがリソースを管理するように権限をrootとして変更する必要があります。

1. rootとして次のコマンドを実行して、指定したリソースの権限をrootに変更します。

```
# crsctl setperm resource resource_name -o root
```

2. Oracle ClusterwareをインストールしたユーザーとしてOracle Databaseインストール所有者(次の例ではoracle)を有効にして、次のスクリプトを実行します。

```
$ crsctl setperm resource resource_name -u user:oracle:r-x
```

3. リソースを起動します。

```
$ crsctl start resource resource_name
```

アプリケーションの配置ポリシー

リソースは、配置ポリシー、リソース起動依存性およびサーバーでのアクション・スクリプトの可用性に従って、任意のサーバーで実行できます。

PLACEMENTリソース属性を使用して、リソースを起動するサーバーをOracle Clusterwareで選択する方法、およびサーバーの障害発生後にリソースを再配置する場所を指定します。HOSTING_MEMBERSおよびSERVER_POOLS属性を使用してリソースのホスティングに適切なサーバーを指定し、PLACEMENT属性を使用してリソースの配置をさらに調整します。

PLACEMENTリソース属性の値を使用して、クラスタへのリソース追加時またはサーバーの障害発生時にOracle Clusterwareでリソースを配置する方法を指定します。HOSTING_MEMBERSまたはSERVER_POOLS属性のいずれかとともに使用すると、Oracle Clusterwareでクラスタにリソースを配置する方法を構成できます。PLACEMENT属性の値は次のとおりです。

- **balanced:** Oracle Clusterwareによって、配置に任意のオンライン・サーバー・プールが使用されます。負荷が低いサーバーは負荷が大きいサーバーより優先されます。サーバーの負荷を測定する場合、Oracle Clusterwareでは、そのサーバーでONLINE状態になっているリソースのLOADリソース属性が使用されます。現行のサーバー負荷を測定する場合、Oracle Clusterwareでは、LOADの合計値が使用されます。
- **favored:** 値がHOSTING_MEMBERS、SERVER_POOLSまたはSERVER_CATEGORYリソース属性のいずれかに割り当てられている場合、この値がプリファレンスを示します。HOSTING_MEMBERSが移入されSERVER_POOLSまたはSERVER_CATEGORYのいずれかが設定されると、HOSTING_MEMBERSによって配置のプリファレンスが示され、SERVER_POOLSまたはSERVER_CATEGORYによって制限が示されます。たとえばora.cluster.vipリソースにPLACEMENTの値をfavoredに設定するポリシーがあり、SERVER_CATEGORYはHubに設定され、HOSTING_MEMBERSはserver_name1に設定されます。この場合、Oracle Clusterwareはora.cluster.vipの配置をハブ・カテゴリのサーバーに限定し、server_name1と呼ばれるサーバーを優先します。
- **restricted:** Oracle Clusterwareでは、SEVER_POOLSリソース属性に示されているサーバー・プールに属しているサーバー、SERVER_CATEGORYリソース属性で構成されているような特定のカテゴリのサーバーまたはHOSTING_MEMBERSリソース属性に示されているサーバーのみがリソースの配置の対象として考慮されます。値を指定できるのは、これらのリソース属性のいずれか1つのみであり、そうしないとエラーが発生します。

関連トピック

- [Oracle Clusterwareのリソース・リファレンス](#)
- [HOSTING_MEMBERS](#)
- [SERVER_CATEGORY](#)
- [SERVER_POOLS](#)

アプリケーションおよびアプリケーション・リソースの登録解除

リソースを登録解除するには、`crsctl delete resource`コマンドを使用します。アプリケーションまたはリソースがONLINEであるか、または別のリソースで必要とされている場合は、`-force`オプションを使用しないかぎり、そのアプリケーションまたはリソースを登録解除することはできません。次の例では、Apache Webサーバー・アプリケーションを登録解除します。

```
$ crsctl delete resource myApache
```

Oracle Clusterwareでリソースが管理されない場合は、クリーンアップ・ステップとして`crsctl delete resource`コマンドを実行します。不要なリソースは登録解除することをお勧めします。

リソースの管理

この項には次のトピックが含まれます：

- [アプリケーション・リソースの登録](#)
- [アプリケーション・リソースの起動](#)
- [アプリケーションおよびアプリケーション・リソースの再配置](#)
- [アプリケーションおよびアプリケーション・リソースの停止](#)
- [クラスタウェア・アプリケーションおよびアプリケーション・リソースのステータス情報の表示](#)

アプリケーション・リソースの登録

Oracle Clusterwareで管理する各アプリケーションは、リソースとしてOCRに登録されて格納されます。

crsctl add resourceコマンドを使用して、アプリケーションをOCRに登録します。たとえば、前述の例のApache Webサーバー・アプリケーションを登録するには次のコマンドを入力します。

```
$ crsctl add resource myApache -type cluster_resource
-group group_name -attr "ACTION_SCRIPT=/opt/cluster/scripts/myapache.scr,
PLACEMENT=restricted,
SERVER_POOLS=server_pool_list,CHECK_INTERVAL=30,RESTART_ATTEMPTS=2,
START_DEPENDENCIES=hard(appsvip),STOP_DEPENDENCIES=hard(appsvip)"
```

前述の例では、-groupパラメータを指定して、リソースをリソース・グループに割り当てることができます。

リソースを変更した場合は、crsctl modify resourceコマンドを実行してOCRを更新します。

関連トピック

- [crsctl add resource](#)
- [crsctl modify resource](#)

アプリケーション・リソースの起動

crsctl start resourceコマンドを使用してリソースを起動します。

Oracle Clusterwareの外部でリソースを手動で起動または停止すると、リソースのステータスを無効にできます。また、Oracle Clusterwareによって、手動で停止操作を実行したリソースの再起動が試行される場合もあります。

Oracle Clusterwareに登録されているアプリケーション・リソースを起動するには、crsctl start resourceコマンドを使用します。たとえば：

```
$ crsctl start resource myApache
```

このコマンドは、アクション・プログラムがコールされるたびにそのアクション・プログラムから成功または失敗の通知を受け取るために待機します。Oracle Clusterwareでは、障害しきい値を超えたために停止したアプリケーション・リソースを起動できます。起動する前に、crsctl add resourceを使用してリソースを登録する必要があります。

crsctl start resourceコマンドをリソースで実行すると、リソースのTARGET値がONLINEに設定されます。Oracle Clusterwareは、startアクションを指定してアクション・プログラムを実行し、TARGETと一致するように状態の変更を試行します。

クラスタ・サーバーでリソースを起動中にそのサーバーで障害が発生した場合、crsctl status resourceコマンドを使用して、クラスタ上のリソースの状態をチェックします。

関連トピック

- [Oracle Clusterware制御\(CRSCTL\)ユーティリティ・リファレンス](#)

アプリケーションおよびアプリケーション・リソースの再配置

`crsctl relocate resource`コマンドを使用して、アプリケーションおよびアプリケーション・リソースを再配置します。

たとえば、rac2というサーバーにApache Webサーバー・アプリケーションを再配置するには、次のコマンドを実行します。

```
# crsctl relocate resource myApache -n rac2
```

アクション・プログラムがコールされるたびに、`crsctl relocate resource`コマンドがSCRIPT_TIMEOUTリソース属性の値に指定された期間待機して、アクション・プログラムから成功または失敗の通知を受け取ります。再配置の試行は、次の場合に失敗します。

- アプリケーションに、初期サーバー上で実行される必須リソースが存在する。
- 指定したリソースが必要なアプリケーションが、初期サーバー上で実行される。

アプリケーションおよびその必須リソースを再配置するには、`crsctl relocate resource`コマンドで-fオプションを使用します。Oracle Clusterwareは、状態に関係なく、アプリケーションに必要なすべてのリソースを再配置または起動します。

また、`crsctl relocate resourcegroup`コマンドを使用してリソース・グループを再配置することもできます。このコマンドでは、最初にリソース・グループ内のリソースを停止してから、宛先サーバーでリソース・グループを再配置します。

オンライン再配置

一部のリソースまたはリソース・グループは、再配置操作中にも高可用性を保つ必要があります。RELOCATE_KINDリソース属性(リソース・グループでも使用可能)をonline (RELOCATE_KIND=online)に設定できます。これにより、`crsctl relocate resource`または`crsctl relocate resourcegroup`コマンドを実行すると、宛先サーバーで新しいリソース・インスタンス(またはリソース・グループに属するリソースの複数のインスタンス)が起動されてから、元のサーバーでインスタンスが停止されます。

ノート:



オンライン再配置を使用する前に、リソースがオンライン再配置中に起動された追加のリソース・インスタンスを管理できることを確認してください。

アプリケーションおよびアプリケーション・リソースの停止

`crsctl stop resource`コマンドを使用してアプリケーション・リソースを停止します。

コマンドによってリソースのTARGET値はOFFLINEに設定されます。Oracle Clusterwareは、常にリソースの状態をターゲットの状態に一致させようとするため、Oracle Clusterwareのサブシステムによって、アプリケーションが停止されます。次の例では、Apache Webサーバーを停止します。

```
# crsctl stop resource myApache
```

このリソースに対するhard停止依存性が別のリソースに定義されている場合、強制(-f)オプションを使用しないかぎり、このリソースを停止することはできません。他のリソースが依存しているリソース上で`crsctl stop resource resource_name -f`コマンドを使用し、それらの依存リソースが実行されている場合は、停止するリソースに依存するリソースがOracle Clusterwareによって停止されます。

クラスタウェア・アプリケーションおよびアプリケーション・リソースのステータス情報の表示

`crsctl status resource`コマンドを使用して、クラスタ・サーバー上のアプリケーションおよびリソースのステータス情報を表示します。

次の例では、Apache Webサーバー・アプリケーションのステータス情報を表示します。

```
# crsctl status resource myApache
NAME=myApache
TYPE=cluster_resource
TARGET=ONLINE
STATE=ONLINE on server010
```

このコマンドは、他にも次の情報を返します。

- リソースが再起動された回数
- 障害間隔内にリソースで障害が発生した回数
- リソースの再起動可能回数または障害発生許容回数の最大数
- リソースのターゲット状態および通常のステータスの情報

特定のリソースの詳細を表示するには、`crsctl status resource resource_name`コマンドで `-f` オプションを使用します。

次のコマンドを入力して、すべてのアプリケーションおよびリソースの情報を表形式で表示します。

```
# crsctl status resource -t
```

関連トピック

- [Oracle Clusterware制御\(CRSCTL\)ユーティリティ・リファレンス](#)

Oracle Clusterwareによるリソースの自動再起動の管理

いくつかのリソース属性を設定することによって、Oracle Clusterwareが自動的にリソースを再起動しないようにできます。また、Oracle Clusterwareによるリソースの再起動カウンタの管理を制御することもできます。さらに、Oracle Clusterwareがリソースで実行するstart、stopおよびcheckアクションのタイムアウト値をカスタマイズすることもできます。

この項には次のトピックが含まれます：

- [Oracle Clusterwareリソースの自動再起動の回避](#)
- [Oracle Clusterwareリソースの再起動試行カウンタの自動管理](#)

Oracle Clusterwareリソースの自動再起動の回避

自動再起動を管理するには、`AUTO_START`リソース属性を使用して、サーバーの再起動時にOracle Clusterwareによって自動的にリソースが起動されるかどうかを指定します。

サーバーを再起動すると、Oracle Clusterwareによって、サーバーの起動直後に実行されるリソースの起動が試行されます。ただし、リソースが依存するシステム・コンポーネント(ボリューム・マネージャ、ファイル・システムなど)が実行されていない場合は、リソースの起動が失敗する可能性があります。これは特に、リソースが依存するシステム・コンポーネントをOracle Clusterwareが管理しない場合に該当します。

ノート：



リソースの `AUTO_START` リソース属性の値にかかわらず、起動対象のリソースに対する `hard` 起動依存性または `weak` 起動依存性が別のリソースに定義されている場合、または別のリソースに対する `pullup` 起動依存性が起

動対象のリソースに定義されている場合は、そのリソースを起動できます。

関連トピック

- [起動依存性](#)
- [Oracle Clusterwareのリソース・リファレンス](#)

Oracle Clusterwareリソースの再起動試行カウンタの自動管理

リソースで障害が発生した場合、Oracle Clusterwareでは、RESTART_ATTEMPTSリソース属性に指定された回数だけリソースの再起動が試行されます。この属性は、障害が発生したリソースの再起動を試行する回数を指定するものではなく(常に1回の試行)、ローカルでリソース障害が何回発生した場合にOracle Clusterwareによってフェイルオーバーが試行されるかを指定するものです。CRSDプロセスは、内部カウンタを保持して、リソースがOracle Clusterwareによって再起動される回数を追跡します。Oracle Clusterwareがリソースの再起動を試行した回数はRESTART_COUNTリソース属性に反映されます。Oracle Clusterwareでは、リソースの安定性に基づいて再起動試行カウンタが自動的に管理されます。

UPTIME_THRESHOLDリソース属性は、リソースがオンラインである必要がある時間を決定し、その時間が経過すると、RESTART_COUNT属性は0にリセットされます。また、RESTART_COUNTリソース属性は、リソースがユーザーによって再配置または再起動された場合、あるいはリソースが別のサーバーにフェイルオーバーした場合にも、0にリセットされます。

A クラスタ検証ユーティリティのリファレンス

クラスタ検証ユーティリティ(CVU)について、それを使用してインストール、パッチ適用またはその他のシステム変更を準備する方法について学習します。

クラスタを変更する場合、クラスタ検証ユーティリティ(CVU)を使用してシステム・チェックを実行することをお勧めします。CVUを使用すると、必要なシステム構成やインストール前のステップを確実に行えるようになり、Oracle Grid InfrastructureやOracle Real Application Clusters (Oracle RAC)のインストール、更新またはパッチ操作を正常に完了できます。

Oracle Universal InstallerはCVUに完全に統合されており、CVUの多くの前提条件チェックは自動化されます。Oracle Universal Installerを実行すると、すべての前提条件チェックとこれに関連付けられたfixupスクリプトが実行されます。

関連項目:

- CVUを管理するサーバー制御ユーティリティ(SRVCTL)の使用の詳細は、[『Oracle Real Application Clusters 管理およびデプロイメント・ガイド』](#)を参照してください。
- CVUを手動でインストールする方法については、[『Oracle Grid Infrastructureインストール・ガイド』](#)および[『Oracle Real Application Clustersインストール・ガイド』](#)を参照してください。

ノート:



次の URL の Oracle Technology Network を確認して、CVU の最新バージョンをダウンロードしてください。

<http://www.oracle.com/technetwork/index.html>

この付録では、CVUについて説明します。内容は次のとおりです。

- [クラスタ検証ユーティリティについて](#)
 - [CVUの概要](#)
 - [CVUの操作上のノート](#)
 - [特別なCVU項目](#)
- [クラスタ検証ユーティリティのコマンド・リファレンス](#)
- [CVUのトラブルシューティングおよび診断出力](#)

クラスタ検証ユーティリティについて

この項では、CVUの使用に関連する項目について説明します。

- [CVUの概要](#)
- [CVUの操作上のノート](#)
- [特別なCVU項目](#)

CVUの概要

CVUは、操作のフェーズまたはステージで主なクラスタ・コンポーネントを検証できます。

コンポーネントは、ディスクの空き領域などの基本的なコンポーネントの場合と、Oracle Clusterwareの整合性チェックなどの複雑なコンポーネントの場合があります。たとえば、CVUは、Oracle Clusterwareレイヤー全体の複数のOracle Clusterwareサブコンポーネントを検証できます。また、CVUは、ディスク領域、メモリー、プロセスおよびその他の重要なクラスタ・コンポーネントをチェックできます。ステージにはデータベースのインストールなどがあり、CVUはこれらのステージで、システムがOracle Real Application Clusters(Oracle RAC)のインストールのための基準を満たしているかどうかを検証できます。その他のステージには、完全に稼働しているクラスタ設定を使用したハードウェアの初期設定やシステム要件の設定などが含まれます。

[表A-1](#)に、CVUで実行できる検証を示します。

表A-1 CVUの様々な検証の実行

実行する検証	使用するCVUのコマンド
システム要件の検証	cluvfy comp sys
Oracle Cluster File System の検証	cluvfy stage cfs
記憶域の検証	● cluvfy comp freespace ● cluvfy comp space ● cluvfy comp ssa ● cluvfy stage acfscfg
ネットワークの検証	cluvfy stage hwas
接続性の検証	● cluvfy comp nodecon ● cluvfy comp nodereach
クラスタ時刻同期化サービスの検証	cluvfy comp clocksync
ユーザーおよび権限の検証	cluvfy comp admprv
ノードの比較および検証	cluvfy comp peer
インストールの検証	● cluvfy stage appcluster

実行する検証	使用するCVUのコマンド
	● cluvfy stage hwos ● cluvfy stage dbcfg ● cluvfy stage -pre dbinst ● cluvfy stage crsinst ● cluvfy stage hacfg ● cluvfy stage nodeadd
削除の検証	cluvfy stage nodedel
Oracle Clusterware および Oracle ASM コンポーネントの検証	● cluvfy comp acfs ● cluvfy comp asm ● cluvfy comp clumgr ● cluvfy comp crs ● cluvfy comp dhcp ● cluvfy comp dns ● cluvfy comp gns ● cluvfy comp gnpn ● cluvfy comp ha ● cluvfy comp nodeapp ● cluvfy comp ocr ● cluvfy comp ohasd ● cluvfy comp olr ● cluvfy comp scan ● cluvfy comp software

- [cluvfy comp vdisk](#)

CVUの操作上のノート

この項には次のトピックが含まれます:

- [CVUのインストール要件](#)
- [CVUの使用方法](#)
- [CVUの構成ファイル](#)
- [権限およびセキュリティ](#)
- [CVUヘルプの使用](#)
- [非推奨およびサポート対象外のCLUVFYコマンド](#)

CVUのインストール要件

CVUのインストール要件を次に示します。

- CVUを実行するノード上には、CVUソフトウェア用に30MB以上の空き領域が必要です。
- 各ノード上に、空き領域が25MB以上の作業ディレクトリが必要です。作業ディレクトリのデフォルトの場所は、LinuxおよびUNIXシステムの場合は/tmpで、Windowsシステムの場合はTEMP環境変数に指定されている値の場所です。CV_DESTLOC環境変数を設定すると、別の場所を指定できます。

CVUを使用する際、ユーティリティは、必要な情報をCVUの作業ディレクトリにコピーしようとします。各ノードでの作業ディレクトリの存在が確認されます。見つからない場合は、作成が試行されます。クラスタ内のすべてのノード上にCVUの作業ディレクトリが存在するか、またはCVUを実行しているユーザーに対して各ノード上にディレクトリを作成するための適切な権限が付与されているかを確認します。

CVUの使用方法

CVUには、Oracleソフトウェアをインストールする前に使用するruncuvfy.sh (Windowsではruncuvfy.bat)と、cluvfy (Windowsではcluvfy.bat)の2つのスクリプトが含まれており、これらはGrid_homeディレクトリとGrid_home/binにそれぞれ格納されています。runcuvfy.shスクリプトには一時変数の定義が含まれており、これによって、Oracle Grid InfrastructureまたはOracle Databaseをインストールする前の実行が可能になります。Oracle Grid Infrastructureのインストール後には、cluvfyコマンドを使用して前提条件をチェックし、その他のシステム準備状態チェックを実行します。

ノート:



Oracle Universal Installer は、Oracle ソフトウェアのインストール時に、cluvfy を実行してすべての前提条件をチェックします。

Oracleソフトウェアをインストールする前に、Gridホームを配置するディレクトリからruncuvfy.shを次のように実行してください。

```
cd /u01/app/12.2.0/grid
./runcluvfy.sh options
```

前述の例で、options変数は選択したCVUコマンド・オプションを表します。たとえば:

```
$ ./runcluvfy.sh comp nodereach -n node1,node2 -verbose
```

CVUコマンドには、次のレポート・オプションが含まれる場合があります。

- **-json**: レポート出力をJSON形式で表示します。-jsonオプションを指定しない場合、詳細レポートは標準出力にテキストとして表示されます。
- **-html**: CVUレポート出力をHTML形式で表示します。-htmlオプションを指定しない場合、詳細レポートは標準出力にテキストとして表示されます。
- **-verbose**: すべてのCVUチェック(合格と失敗の両方)に関する説明を、出力の詳細サマリー・セクションに表示します。
- **-file file_location**: 指定した形式に応じて、.htm、.jsonまたは.txtのファイル拡張子を付けて出力を特定の場所に保存します。

CVUコマンドを入力すると、テストのサマリーが表示されます。インストール前には、CVUコマンドに**-verbose**引数を使用して詳細な出力を取得することをお勧めします。**-verbose**引数を指定すると、個々のチェックの詳細な出力が生成されます。適用可能な場合は、各ノードの結果が表レイアウトで表示されます。

c luvfyコマンドを使用して、CVUコマンドライン・ツールを実行します。c luvfyを使用しても、クラスタ環境またはインストールされたソフトウェアに悪影響を及ぼすことはありません。c luvfyコマンドは、Oracle Clusterwareのインストール前を含む任意のタイミングで実行できます。実際には、CVUは、ハードウェアおよびオペレーティング・システムが稼働するとすぐに使用できるように設計されています。ノード上のOracle Clusterwareを必要とするコマンドを実行したときにそのノード上にOracle Clusterwareがまだインストールされていない場合には、CVUはエラーをレポートします。

CVUコマンドとともに使用するノード・リストは、ドメインを含まないホスト名のカンマ区切りのリストである必要があります。CVUがノード・リストを処理する際にはドメインは無視されます。ドメイン情報を削除した後、CVUのコマンド・エントリに重複するノード・エントリがある場合には、CVUは重複するノード・エントリを削除します。

ネットワーク接続の検証では、CVUコマンドラインでインタフェースを指定しない場合、CVUによって、すべての使用可能なネットワーク・インタフェースが検出されます。記憶域のアクセス性の検証では、コマンドラインで特定の記憶域のIDを指定しない場合、CVUはサポートされるすべての記憶域タイプの共有記憶域を検出します。使用可能な場合、Oracle ClusterwareのホームもCVUにより検出されます。

関連項目:

使用方法のセキュリティ情報は、[権限およびセキュリティ](#)を参照してください。

CVU出力

CVU出力は、次のような5つの異なるセクションで構成されます。

- **ヘッダー**: ステージまたはコンポーネントが実行するチェックに関する情報が含まれる単一行。
- **構成**: CVUはその操作がすべてのノードで実行できるかどうかを評価します。指定したノードが停止しているか、必要な前提条件を満たしていない(たとえば、SSH設定がない)場合、これらのノードに関するエラー・メッセージがここに表示されます。このセクションは、出力に常に含まれているとはかぎりません。実行されるチェックが操作に関与するすべて

のノードで実行可能な場合、このセクションは出力に表示されません。

- 進捗メッセージ: CVUでは、様々なチェックを繰り返す間に、このセクションに進捗メッセージを表示します。進捗メッセージはCVUがハングしているかどうかの判別に役立ちます。
- 詳細サマリー: デフォルトで、CVUでは失敗したタスクまたはサブタスクのみをこのセクションに表示します。-verbose オプションを選択した場合、CVUではすべてのタスクおよびサブタスクに関する詳細情報を表示します。
- 実行サマリー: 最後に、CVUは、次の例のような、チェック・プロセス全体の簡潔なサマリーを表示します。

```
$ cluvfy stage -pre crsinst -n sales65
CVU operation performed: stage -pre crsinst
Date: Oct 13, 2016 9:43:39 PM
CVU home: /ade/scott_abc4/oracle/
User: scott
$ cluvfy comp baseline -collect all -n sales65
CVU operation performed: baseline
Date: Oct 13, 2016 9:48:19 PM
Operating system: Linux2.6.39-400.211.1.el6uek.x86_64
```

CVUの構成ファイル

CVUの構成ファイルを使用して、CVUの実行のための特定の入力を定義できます。構成ファイルのパスは、Grid_home/cv/admin/cvu_config(Windowsプラットフォームの場合は Staging_area¥clusterware¥stage¥cvu¥cv¥admin)です。このファイルをテキスト・エディタを使用して変更できます。CVUへの入力は、キー・エントリの形式で定義されています。CVUの構成ファイルを変更する場合、次の規則に従う必要があります。

- キー・エントリの構文はname=valueです。
- 各キー・エントリおよびキーに割り当てられた値は、1つのみのプロパティを定義します。
- シャープ記号(#)で始まる行はコメント行であり、無視されます。
- name=valueという構文に従っていない行は無視されます。

CVUによってサポートされるキーのリストを次に示します。

- CV_NODE_ALL: Oracle Clusterwareがインストールされていない場合に選択する必要があるノードのリストを指定します。デフォルトでは、このエントリはコメント・アウトされています。
- CV_ORACLE_RELEASE: 検証を実行する必要がある特定のOracleリリース(10.1、10.2、11.1、11.2、12.1または12.2)を指定します。これを設定すると、-r releaseオプションが適用される場合であっても、このオプションを使用する必要がなくなります。デフォルト値は12.2です。
- CV_RAW_CHECK_ENABLED: TRUEに設定すると、LinuxおよびUNIXシステム上の共有ディスクのアクセス性チェックが有効になります。この共有ディスクのアクセス性チェックには、すべてのノード上にcvuqdisk RPMパッケージ・マネージャ(rpm)がインストールされていることが必要です。デフォルトでは、このキーはTRUEに設定されており、共有ディスクのチェックが有効になっています。
- CV_ASSUME_DISTID: このプロパティは、CVUが特定のプラットフォームまたは配布を検出またはサポートできない場合に使用されます。CVUが無効になる可能性があるため、このプロパティを変更することはお勧めしません。
- CV_XCHK_FOR_SSH_ENABLED: TRUEに設定すると、sshを使用してユーザー等価関係を検証するためのX-Windowsチェックが有効になります。デフォルトでは、このエントリはコメント・アウトされており、X-Windowsチェックは無効になっています。

- ORACLE_SRVM_REMOTECPY: 設定すると、scpまたはrcpコマンドの位置を指定してCVUのデフォルト値が上書きされます。デフォルトでは、このエントリはコメント・アウトされており、CVUは/usr/bin/scpおよび/usr/sbin/rcpを使用します。
- ORACLE_SRVM_REMOTESHELL: 設定すると、sshコマンドの位置を指定してCVUのデフォルト値が上書きされます。デフォルトでは、このエントリはコメント・アウトされており、ツールは/usr/sbin/sshを使用します。
- CV_ASSUME_CL_VERSION: デフォルトで、コマンドライン解析機能は、コマンドライン構文の使用方法的表示および構文の検証にcrs activeversionを使用します。このプロパティを使用して、コマンドライン構文の表示および検証用にcrs activeversion以外のバージョンを渡します。デフォルトでは、このエントリはコメント・アウトされています。

CVUは構成ファイルに定義されたキー・エントリを検出できない場合、CVUはキーの名前に一致する環境変数を検索します。環境変数が設定されている場合、CVUはその値を使用し、環境変数が設定されていない場合、CVUはこのエンティティのデフォルト値を使用します。

権限およびセキュリティ

rootユーザーのユーザー等価関係がないため、リモート・ノード操作を実行するために、次を除いてほとんどのCVUコマンドをrootとして実行できません。

- [cluvfy comp dhcp](#)
- [cluvfy comp dns](#)

ただし、権限委任を使用すると、-methodパラメータを指定でき、2つの方法(sudoまたはroot)から1つを選択できます。これによって、チェックが有効化され、リモート・ノードで実行するためにroot権限が必要な修正スクリプトを実行できるようになります。パスワードの入力を求められます。パスワードはCVUコマンドの実行中に動的に使用され、ストレージ・デバイスには格納されません。

-methodパラメータの指定は、修正スクリプトのコンテキストに適しています。権限委任を選択すると、一度にすべての修正スクリプトをローカル・ノードから実行できます。権限委任を選択しない場合は、各関連ノードにrootとしてログインしてから修正スクリプトを実行する必要があります。

CVUヘルプの使用

cluvfyコマンドには、入力されたコマンドライン引数に基づいて使用方法を表示する状況依存ヘルプがあります。たとえば、cluvfyと入力すると、ステージおよびコンポーネントの構文について説明する、一般的な使用方法を示す高水準のテキストがCVUにより表示されます。状況依存ヘルプのコマンドのリストを次に示します。

- cluvfy -help: CVUは、CVUコマンドの詳細情報を表示します。
- cluvfy -version: CVUは、Oracle Clusterwareのバージョンを表示します。
- cluvfy comp -list: CVUは、チェック可能なコンポーネントのリストと、ユーティリティが各コンポーネントをチェックする方法についての簡単な説明を表示します。
- cluvfy comp -help: CVUは、それぞれの有効なコンポーネントをチェックするための詳細な構文を表示します。
- cluvfy stage -list: CVUは、有効なステージのリストを表示します。
- cluvfy stage -help: CVUは、それぞれの有効なステージをチェックするための詳細な構文を表示します。

また、任意のCVUコマンドに-helpオプションを使用することもできます。たとえば、cluvfy stage -pre nodeadd -helpを実行すると、この特定のコマンドに関する詳細情報が戻されます。

無効なCVUコマンドを入力すると、CVUによって、そのコマンドの正しい使用方法が表示されます。たとえば、`cluvfy stage -pre dbinst`と入力すると、dbinstステージの事前チェック・コマンドの正しい構文がCVUにより表示されます。CVUコマンドの詳細を表示するには、`cluvfy -help`と入力します。

非推奨およびサポート対象外のCLUVFYコマンド

次の表には、非推奨およびサポート対象外のCLUVFYコマンドが含まれています。

表A-2 非推奨およびサポート対象外のCluvfyコマンド

コマンド	非推奨	サポート対象外
<code>cluvfy comp cfs</code>	Oracle Database 12c リリース 1 (12.1)	いいえ

特別なCVU項目

この項には次のトピックが含まれます：

- [修正スクリプトの生成](#)
- [CVUを使用した、インストールの前提条件を満たしているかどうかの確認](#)
- [Oracle Database 10gリリース1または2でのCVUの使用](#)
- [開始基準および終了基準](#)
- [冗長モードおよびUNKNOWN出力](#)
- [CVUのノード・リストのショートカット](#)

修正スクリプトの生成

特定のCVUコマンドで `-fixup` フラグを指定すると、インストール前に[fixupスクリプト](#)を生成できます。また、Oracle Universal Installerでは、インストール中にfixupスクリプトを生成することもできます。その後、インストーラによって、個々のターミナル・セッションでrootとしてスクリプトを実行するように要求されます。fixupスクリプトをコマンドラインから生成した場合は、生成したスクリプトをrootとして実行できます。スクリプトを実行すると、カーネルの値が最低限必要な値に引き上げられ、必要に応じて、他のオペレーティング・システムの構成が実行されます。

または、特定のCVUコマンドには `-method` パラメータを指定できます。これによって、権限委任が有効化され、ローカル・ノードからrootとして修正スクリプトをリモート・ノードで実行できるようになります。

関連項目：

[権限およびセキュリティ](#)

CVUを使用した、インストールの前提条件を満たしているかどうかの確認

CVUを使用して、インストールに関するシステムの前提条件が満たされているかどうかを確認できます。

既存のOracleソフトウェア・インストールがあるシステムにOracle Databaseソフトウェアをインストールする場合に、このオプションを使用します。このオプションを使用する際には、次の点に注意してください。

- インストールの実行に使用する予定のユーザー・アカウントでCVUを実行する必要があります。CVUはrootでは実行できず、また、インストールを実行するユーザー以外のユーザーでCVUを実行すると、インストールまたはその他の構成

チェックに対するユーザーおよびグループの構成の正確さが保証されません。

- クラスタ全体のステータスのチェックを完了するには、その前にすべてのクラスタ・ノードでSSHを構成する必要があります。インストーラを使用してSSHの構成を実行するか、またはユーザー自身がクラスタ内のすべてのノード間でSSHの構成を実行できます。CVUを使用して、SSH接続を構成する修正スクリプトを生成することもできます。
- CVUでは完了しておく必要があるインストール前のステップを検出することができますが、インストール前のタスクは実行できません。

完了しているインストール前のステップ、および実行する必要があるインストール前のステップを確認するには、次の構文を使用します。必要に応じて、`-fixup`を指定してコマンドを実行し、カーネルの構成タスクを実行するfixupスクリプトを生成します。

```
$ ./runcluvfy.sh stage -pre crsinst -fixup -n node_list
```

前述の構文例の`node_list`変数には、クラスタ内のノードの名前をカンマで区切って指定します。Windowsの場合、カンマで区切られたノードのリストは二重引用符(" ")で囲む必要があります。

たとえば、マウントポイントが`/mnt/dvdrom/`で、ノードが`node1`、`node2`および`node3`のクラスタの場合は、次のようにコマンドを入力します。

```
$ cd /mnt/dvdrom/  
$ ./runcluvfy.sh stage -pre crsinst -fixup -n node1,node2,node3
```

CVUのレポートを確認して、必要に応じて追加のステップを実行します。

関連項目:

製品のインストールの詳細は、ご使用のプラットフォーム固有のインストレーション・ガイドを参照してください。

Oracle Database 10gリリース1または2でのCVUの使用

Oracle Database 12cメディアのCVUを使用して、Oracle Database 10gリリース1 (10.1)以上のインストールのシステム要件を確認します。CVUを使用してOracle Clusterwareのインストールをチェックするには、標準のCVUシステム・チェック・コマンドに、コマンド`-r release_code`フラグを追加します。

たとえば、メディアのマウントポイントが`/mnt/dvdrom/`で、クラスタ・ノードが`node1`、`node2`および`node3`であるシステムで、Oracle Clusterwareバージョン10.2のインストール前に検証チェックを実行するには、次のようにコマンドを入力します。

```
$ cd /mnt/dvdrom  
$ ./runcluvfy.sh stage -pre crsinst -n node1,node2,node3 -r 10.2
```

ノート:



チェックするリリース・バージョンを指定しなかった場合、CVU は、12c リリース 1 (12.1)の要件についてチェックします。

開始基準および終了基準

ステージを検証する際に、CVUは、開始基準および終了基準を使用します。各ステージに開始基準が設定されており、そのステージを開始する前に実行される特定の検証タスクのセットが定義されています。

このチェックを行うと、そのステージのOracle Clusterwareの前提条件を満たさないかぎり、Oracle Clusterwareのインス

ツールなどのステージを開始できません。

ステージの終了基準は、ステージの完了後に実行する必要がある検証タスクの別のセットを定義します。事後チェックによって、ステージのアクティビティが確実に完了します。事後チェックでは、後続のステージに伝播する前にステージ固有の問題が識別されます。

冗長モードおよびUNKNOWN出力

デフォルトでは、CVUレポートは非冗長モードでテストのサマリーのみがレポートされるため、`-verbose`引数を使用して詳細な出力を取得できます。

`-verbose`引数を指定すると、個別のチェックの詳細な出力が生成され、適用可能な場合には各ノードの結果が表レイアウトで表示されます。

`cluvfy`コマンドが特定のノードについてUNKNOWNと応答した場合、これは、チェックの合格または失敗をCVUが判別できないためです。この原因は、そのノードに対する到達可能性がないか、またはユーザー等価関係が存在しないことです。また、CVUによるチェックの実行時にそのノード上で発生していたシステムの問題が原因の場合もあります。

UNKNOWNという結果について考えられる原因のリストを次に示します。

- ノードが停止している。
- `Grid_home/bin`または`Oracle home`ディレクトリにCVUが必要とする実行可能ファイルが存在しない
- CVUを実行したユーザー・アカウントに対して、ノード上の一般的なオペレーティング・システム実行可能ファイルを実行する権限が付与されていない
- オペレーティング・システム・パッチまたは必要なパッケージがノードに適用されていない
- ノードの最大プロセス数または最大オープン・ファイル数を超過しているか、共有メモリー、セマフォなど、IPCセグメントに問題が発生している。

CVUのノード・リストのショートカット

CVUノード・リストを取得する場合は、`cluvfy`コマンドで利用できるオプションがいくつかあります。

CVUにクラスタのすべてのノードのリストを提供するには、`-n all`と入力します。CVUは、次の順序でノード・リストの取得を試行します。

1. ベンダーのクラスタウェアが使用可能な場合、CVUは、`lsnodes`ユーティリティを使用して、構成されているすべてのノードをベンダーのクラスタウェアから選択します。



ノート:

Oracle Database 19c 以降では、Oracle Clusterware とベンダー・クラスタウェアの統合は非推奨であり、将来のリリースでサポートされなくなる可能性があります。このため、次回のソフトウェアまたはハードウェアのアップグレードでは、ベンダー・クラスタ・ソリューションを使用しないように移行することをお勧めします。

2. Oracle Clusterwareがインストールされている場合、CVUは、`olsnodes`ユーティリティを使用して、構成されているすべてのノードをOracle Clusterwareから選択します。
3. ベンダーのクラスタウェアおよびOracle Clusterwareのいずれもインストールされていない場合、CVUは、構成ファイルで`CV_NODE_ALL`キーの値を検索します。
4. ベンダーのクラスタウェアおよびOracle Clusterwareのいずれもインストールされておらず、構成ファイルに

CV_NODE_ALLというキーが存在しない場合、CVUは、環境変数CV_NODE_ALLの値を検索します。この変数が設定されていない場合、CVUはエラーをレポートします。

環境変数を設定してその変数をCVUコマンドに使用すると、部分的なノード・リストを提供できます。たとえば、LinuxまたはUNIXのシステム上で次のように入力できます。

```
setenv MYNODES node1,node3,node5
cluvfy comp nodecon -n $MYNODES [-verbose]
```

クラスタ検証ユーティリティのコマンド・リファレンス

この項では、CVUコマンドについて説明します。

cluvfy comp acfs

クラスタ内のノード上のOracle Automatic Storage Management Cluster File System (Oracle ACFS)の整合性をチェックします。

構文

```
cluvfy comp acfs [-n node_list | -allnodes] [-f file_system] [-save [-savedir
directory_path]] [-verbose]
```

パラメータ

表A-3 cluvfy comp acfsコマンドのパラメータ

パラメータ	説明
-n node_list -allnodes	検証を実行するノードのドメイン修飾なしの名前のカンマ区切りリスト、またはクラスタ内のすべてのノードです。 このオプションを指定しない場合、CVU は、ローカル・ノードのみをチェックします。
-f file_system	チェックするファイル・システムの名前です。
-save [-savedir directory_path]	検証レポート(cvuheckdkreport_timestamp.txt および cvuheckreport_timestamp.htm)を保存するには、-save フラグまたは -save -savedir フラグを使用します (timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。
-verbose	CVU は詳細な出力を表示します。

cluvfy comp admprv

ノード・リストで指定したすべてのノード上、またはクラスタ内のすべてのノード上で、-oパラメータで指定された操作に必要な管理権限をチェックします。

構文

LinuxおよびUNIXプラットフォームの場合:

```
cluvfy comp admprv [-n node_list | -allnodes] {-o user_equiv [-sshonly] [-fixup] | -o crs_inst [-asmgrp asmadmin_group] [-asmdbagrp asmdba_group] [-asmopergrp asmoper_group] [-orainv orainventory_group] [-fixup] [-fixupnoexec] [-method sudo -user user_name [-location directory_path] | -method root] | -o db_inst [-osdba osdba_group] [-osoper osoper_group] [-fixup] [-fixupnoexec] [-method sudo -user user_name [-location dir_path] | -method root] | -o db_config -d oracle_home [-fixup] [-fixupnoexec] [-method sudo -user user_name [-location directory_path] | -method root]] [-verbose] [-json] [-html] [-save [-savedir directory_path]]
```

Windowsプラットフォームの場合:

```
cluvfy comp admprv [-n node_list | -allnodes] {-o user_equiv [-fixup] | -o crs_inst [-orainv orainventory_group] [-fixup] [-fixupnoexec] | -o db_inst [-fixup] [-fixupnoexec] | -o db_config -d oracle_home [-fixup] [-fixupnoexec]} [-verbose] [-json] [-html] [-savedir directory_path]]
```

パラメータ

表A-4 cluvfy comp admprvコマンドのパラメータ

パラメータ	説明
-n node_list -allnodes	検証を実行するノードのドメイン修飾なしの名前のカンマ区切りリスト、またはクラスタ内のすべてのノードです。 このオプションを指定しない場合、CVU は、ローカル・ノードのみをチェックします。
-o user_equiv [-sshonly]	ノード間のユーザー等価関係をチェックします。Linux および UNIX プラットフォームでは、-sshonly パラメータを追加することで、必要に応じて、ssh を使用してユーザー等価関係を検証できます。
-o crs_inst [option]	Oracle Clusterware をインストールするための管理権限をチェックします。必要に応じて、次の情報を指定できます。 -asmgrp: OSASM グループの名前を指定します。デフォルトは asmadmin です。 -asmdbagrp: ASMDBA グループの名前を指定します。デフォルトは asmdba です。

パラメータ	説明
	• -orainv: Oracle Inventory グループの名前を指定します。デフォルト値は oinstall です。
-o db_inst option]	Oracle RAC データベースをインストールするための管理権限をチェックします。必要に応じて、次の情報を指定できます。 • -osdba: OSDBA グループの名前です。デフォルトは dba です。 • -osoper: OSOPER グループの名前。
-o db_config -d oracle_home	Oracle RAC データベースを作成または構成するための管理権限をチェックします。Oracle RAC データベースの Oracle ホームの場所を指定します。
-fixup	検証が失敗した場合に、可能であれば CVU で fixup の指示を生成することを指定します。
-fixupnoexec	検証が失敗した場合に、CVU で修正データを生成し、生成された修正を手動で実行するための指示を表示するように指定します。
-method sudo -user user_name [-location dir_path] -method root	root ユーザー・アクセスの権限委任方法が sudo か root かを指定します。sudo を指定すると、root 権限ですべてのノードにアクセスするためにユーザー名を指定する必要があります。必要に応じて、sudo 実行可能ファイルのファイル・システムのフルパスも指定します。
-save [-savedir directory_path]	検証レポート(cvucheckreport_timestamp.txt および cvucheckreport_timestamp.htm)を保存するには、-save フラグまたは -save -savedir フラグを使用します (timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。
-verbose	CVU は詳細な出力を表示します。

使用上のノート

- -oパラメータの後の操作は相互に排他的であり、一度に1つの操作しか指定できません。
- デフォルトでは、等価関係のチェックでは、Xフォワードを無効にしているかどうか、環境変数DISPLAYを適切に設定しているかどうかなど、X-Windows構成の検証は行われません。

ユーザー等価関係のチェックでX-Windowsについて検証するには、`cluvfy comp admprv -o user_equiv`コマンドを実行する前にCV_HOME/cv/admin/cvu_configディレクトリに存在する構成ファイルでCV_XCHK_FOR_SSH_ENABLEDキーをTRUEに設定します。

例

次のコマンドを実行すると、Oracle Clusterwareのインストールに必要な権限が、ノードracnode1およびracnode2で構成されているかどうかを検証できます。

```
$ cluvfy comp admprv -n racnode1,racnode2 -o crs_inst -verbose
```

次のコマンドを実行すると、C:¥app¥oracle¥product¥19.0.0¥dbhome_1 Oracleホーム・ディレクトリを使用しているOracle RACデータベースの作成または変更に必要な権限を検証し、その権限を構成するスクリプトを生成できます。

```
cluvfy comp admprv -n racnode1,racnode2 -o db_config -d
C:¥app¥oracle¥product¥19.0.0¥dbhome_1 -fixup -verbose
```

cluvfy comp asm

クラスタ内の特定のノードのOracle Automatic Storage Management (Oracle ASM)の整合性をチェックします。

このチェックでは、指定ノードのOracle ASMインスタンスが同じOracleホームから実行されていること、およびasmlibが存在する場合は、これが有効なバージョンであり、所有権を持つことが確認されます。

構文

```
cluvfy comp asm [-n node_list | -allnodes] [-verbose] [-json]
               [-html] [-save [-savedir directory_path]] [-verbose]
```

パラメータ

表A-5 cluvfy comp asmコマンドのパラメータ

パラメータ	説明
-n node_list -allnodes	検証を実行するノードのドメイン修飾なしの名前のカンマ区切りリスト、またはクラスタ内のすべてのノードです。 このオプションを指定しない場合、CVU は、ローカル・ノードのみをチェックします。
-json	-json フラグを使用すると、詳細レポートが JSON 形式で生成されます。 -json フラグを指定しない場合、CVU によって詳細レポートがテキスト・ファイルで生成されます。
-html	-html フラグを使用すると、詳細レポートが HTML 形式で生成されます。

パラメータ	説明
	-html フラグを指定し、CVU によって認識されるブラウザがシステムで使用可能な場合は、チェックの完了時に CVU によってそのブラウザが起動され、レポートがブラウザに表示されます。 -html フラグを指定しない場合、CVU によって詳細なレポートがテキスト・ファイルで生成されます。
-save [-savedir directory_path]	検証レポート(cvuheckcdkreport_timestamp.txt および cvuheckreport_timestamp.htm)を保存するには、-save フラグまたは -save -savedir フラグを使用します (timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。
-verbose	CVU は詳細な出力を表示します。

例

このコマンドを実行すると、次のような出力が生成されます。

```
$ cluvfy comp asm -n all
Verifying ASM Integrity
Task ASM Integrity check started...
Starting check to see if ASM is running on all cluster nodes...
ASM Running check passed. ASM is running on all specified nodes
Starting Disk Groups check to see if at least one Disk Group configured...
Disk Group Check passed. At least one Disk Group configured
Task ASM Integrity check passed...
Verification of ASM Integrity was successful.
```

cluvfy comp baseline

システムおよびクラスタ構成情報を取得して、ベースラインを作成します。

このベースラインをシステムの状態との比較に使用できます。ベースラインの収集は、Oracle Clusterwareのインストール後や Oracle Clusterwareのアップグレード前後といった戦略的なタイミングで、またはOracle ClusterwareリソースとしてCVUを定期的に行うのに合わせて自動的に実行できます。また、複数のベースラインを比較することもできます。

構文

```
cluvfy comp baseline -collect {all | cluster | database | asm} [-n node_list | -allnodes]
    [-d Oracle_home | -db db_unique_name] [-bestpractice | -mandatory] [-binlibfilesonly]
    [-reportname report_name] [-savedir save_dir]
    [-method sudo -user user_name [-location directory_path] | -method root]
    [-orachk_loc location] [-skip check_utilities]
cluvfy comp baseline -compare baseline1,baseline2,... [-cross_compare] [-deviations]
    [-savedir save_dir]
```

表A-6 cluvfy comp baselineコマンドのパラメータ

パラメータ	説明
-collect {all cluster database asm}	-collect パラメータは、ベースラインを作成して Grid_home/cv/report/xml ディレクトリに保存するように CVU に指定します。 Oracle Clusterware (cluster)、データベース (database)、Oracle ASM、または 3 つすべて(all)に関するベースラインを収集できます。
-n node_list -allnodes	テスト対象として、ノードのドメイン修飾なしの名前のカンマ区切りリスト、またはクラスタ内のすべてのノードを指定します。
-d Oracle_home	データベース・ベースラインを収集するとき、Oracle ホームを指定すると、CVU は、Oracle ホームで実行しているすべてのデータベースのベースラインを収集します。 特定のデータベースのベースラインを収集するには -db パラメータを使用します。
-db db_unique_name	情報を収集するデータベースの名前。 データベース・ベースラインを収集する場合、-db パラメータを指定すると、CVU は指定されたデータベースのベースラインのみを収集します。-db を指定していない場合、CVU はシステムに構成されているすべてのクラスタ・データベースを検出し、それぞれのベースラインを収集します。
-bestpractice -mandatory	ベスト・プラクティス推奨事項のみのベースラインを収集するには、-bestpractice を指定します。必須要件のみのベースラインを収集するには、-mandatory を指定します。
-binlibfilesonly	ソフトウェア・ホームの bin/、lib/、および jlib/サブディレクトリにあるファイルのみを収集するには、-binlibfilesonly を指定します。
-report report_name	レポートの名前を指定するには、このオプション・パラメータを使用します。
-savedir save_dir	CVU がレポートを保存する場所を指定するには、このオプション

パラメータ	説明
	ン・パラメータを使用します。-savedir オプションを指定しない場合、CVU は、レポートを Grid_home/cv/report ディレクトリに保存します。
-method sudo -user user_name [-location dir_path] -method root	root ユーザー・アクセスの権限委任方法が sudo か root かを指定します。sudo を指定すると、root 権限ですべてのノードにアクセスするためにユーザー名を指定する必要があります。必要に応じて、sudo 実行可能ファイルのファイル・システムのフルパスも指定します。
-compare baseline1,baseline2,...	ベースラインを比較するには、-compare を指定します。1 つのベースラインのみを指定した場合、CVU は、収集の結果を表示します。カンマ区切りリストで複数のベースラインを指定した場合、CVU は、HTML ドキュメントでベースラインの値を相互に比較します。
-cross_compare	-cross_compare を指定して、クラスタ全体で、またはクラスタ・ノードとデータベース全体でベースラインを比較します。
-deviations	必要に応じて、このパラメータを指定して、ベスト・プラクティスの推奨事項からの差異のみ、または必須要件からの差異のみ、あるいは両方の差異を表示できます(-bestpractice パラメータと-mandatory パラメータの有無によって変わります)。

使用上のノート

- -collect または -compare パラメータのいずれかを指定する必要があります。
- このコマンドの実行時に CVU が収集するアイテムは次のとおりです。
 - 物理メモリー
 - 使用可能なメモリー
 - スワップ領域
 - 空き領域
 - 必須パッケージ
 - 推奨するカーネル・パラメータ
 - /etc/inittab permissions
 - /var/tmp/.oracle の下のドメイン・ソケット
 - Oracle Clusterware ソフトウェア・ファイルの属性
 - ネットワーク MTU サイズ
 - OCR 権限、グループおよび所有者 (OCR が共有ファイル・システムに格納されている場合)
 - OCR ディスク・グループ (OCR が Oracle ASM に格納されている場合)

- システム要件プラグブル・タスク(ゼロ構成ネットワーク設定、/bootマウント、Huge Pagesの存在、8888ポートの可用性、イーサネット・ジャンボ・フレーム)
- Oracle Clusterware事後チェック・プラグブル・タスク(css miscount、reboottime、disktimeout)[Foot 1](#)
- データベースのベスト・プラクティス

例

次の例は、-collectと-compareの両方のコマンド・パラメータの使用方法を示しています。

```
$ cluvfy comp baseline -collect all -n allnodes -db orcl -bestpractice -report bl1
-savedir /tmp
$ cluvfy comp baseline -compare bl1,bl2
```

cluvfy comp clocksync

ノード・リストのすべてのノードでのクロック同期をチェックします。

CVUは、時刻同期サービス(Oracleクラスタ時刻同期化サービス(CTSS)またはネットワーク・タイム・プロトコル(NTP))が実行されていること、各ノードがクロック同期用に同じ参照サーバーを使用していること、および各ノードの時間オフセットが許容範囲内であることを検証します。

構文

```
cluvfy comp clocksync [-noctss] [-n node_list | -allnodes] [-save [-savedir
directory_path]] [-verbose]
```

パラメータ

表A-7 cluvfy comp clocksyncコマンドのパラメータ

パラメータ	説明
-noctss	このパラメータを指定した場合、CVU は、CTSS に対してチェックを実行しません。かわりに、CVU は NTP などのプラットフォーム固有の時刻同期サービスをチェックします。
-n node_list -allnodes	検証を実行するノードのドメイン修飾なしの名前のカンマ区切りリスト、またはクラスタ内のすべてのノードです。 このオプションを指定しない場合、CVU は、ローカル・ノードのみをチェックします。
-save [-savedir directory_path]	検証レポート(cvuheckdkreport_timestamp.txt および cvuheckreport_timestamp.htm)を保存するには、-save フラグまたは -save -savedir フラグを使用します (timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。

パラメータ	説明
	-save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。
-verbose	CVU は詳細な出力を表示します。

clufy comp clu

すべてのノードでクラスタの整合性をチェックします。

構文

```
clufy comp clu [-verbose] [-json] [-html] [-save [-savedir directory_path]]
```

パラメータ

表A-8 clufy comp cluコマンドのパラメータ

パラメータ	説明
-verbose	CVU は詳細な出力を表示します。
-json	-json フラグを使用すると、詳細レポートが JSON 形式で生成されます。 -json フラグを指定しない場合、CVU によって詳細レポートがテキスト・ファイルで生成されます。
-html	-html フラグを使用すると、詳細レポートが HTML 形式で生成されます。 -html フラグを指定し、CVU によって認識されるブラウザがシステムで使用可能な場合は、チェックの完了時に CVU によってそのブラウザが起動され、レポートがブラウザに表示されます。 -html フラグを指定しない場合、CVU によって詳細なレポートがテキスト・ファイルで生成されます。
-save [-savedir directory_path]	検証レポート(cvuchecdkreport_timestamp.txt および cvuchekreport_timestamp.htm)を保存するには、-save フラグまたは-save -savedir フラグを使用します(timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。



ノート:

コマンドは非推奨となり、今後のリリースでサポートされない可能性があります。

詳細な出力の表示を選択することもできます。

関連項目:

レポート・オプションに関する情報などのCVU使用上のノートについては、[CVUの使用方法](#)

cluvfy comp clumgr

ノード・リスト内のノードのcluster managerサブコンポーネントまたはOracle Cluster Synchronization Services (CSS)の整合性をチェックします。

構文

```
cluvfy comp clumgr [-n node_list | -allnodes] [-save [-savedir directory_path]] [-verbose]
```

パラメータ

表A-9 cluvfy comp clumgrコマンドのパラメータ

パラメータ	説明
-n node_list -allnodes	検証を実行するノードのドメイン修飾なしの名前のカンマ区切りリスト、またはクラスタ内のすべてのノードです。 このオプションを指定しない場合、CVU は、ローカル・ノードのみをチェックします。
-save [-savedir directory_path]	検証レポート(cvuheckdkreport_timestamp.txt および cvuheckreport_timestamp.htm)を保存するには、-save フラグまたは -save -savedir フラグを使用します (timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。
-verbose	CVU は詳細な出力を表示します。

cluvfy comp containerhost

コンテナ・ホストに対してシステム要件チェックを実行し、Oracle ClusterwareおよびDatabaseをホストするコンテナの正常な作成を妨げる可能性のある問題を報告します。

構文

```
cluvfy comp containerhost [-type {docker|k8|podman}] [-fixup] [-fixupnoexec] [-method
sudo -user user_name
[-location directory_path] | -method root] [-verbose] [-json] [-html] [-save [-
savedir directory_path]]
```

パラメータ

表A-10 cluvfy comp containerhostコマンドのパラメータ

パラメータ	説明
-type {docker k8 podman}	ホストで構成されているコンテナ・マネージャのタイプを指定します。
-fixup	検証が失敗した場合に、可能であれば CVU で fixup の指示を生成することを指定します。
-fixupnoexec	検証が失敗した場合に、CVU で修正データを生成し、生成された修正を手動で実行するための指示を表示するように指定します。
-method sudo -user user_name [-location dir_path] -method root	root ユーザー・アクセスの権限委任方法が sudo か root かを指定します。sudo を指定すると、root 権限ですべてのノードにアクセスするためにユーザー名を指定する必要があります。必要に応じて、sudo 実行可能ファイルのファイル・システムのフルパスも指定します。
-verbose	CVU は詳細な出力を表示します。
-json	-json フラグを使用すると、詳細レポートが JSON 形式で生成されます。 -json フラグを指定しない場合、CVU によって詳細レポートがテキスト・ファイルで生成されます。
-html	-html フラグを使用すると、詳細レポートが HTML 形式で生成されます。 -html フラグを指定し、CVU によって認識されるブラウザがシステムで使用可能な場合は、チェックの完了時に CVU によってそのブラウザが起動され、レポートがブラウザに表示されます。 -html フラグを指定しない場合、CVU によって詳細なレポート

パラメータ	説明
	がテキスト・ファイルで生成されます。
-save [-savedir directory_path]	検証レポート(cvuchecdkreport_timestamp.txt および cvuchekreport_timestamp.htm)を保存するには、-save フラグまたは -save -savedir フラグを使用します (timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。

cluvfy comp crs

指定したノードのクラスタ・レディ・サービス(CRS)デーモンの整合性をチェックします。

構文

```
cluvfy comp crs [-n node_list | -allnodes] [-json] [-html] [-save [-savedir [-savedir directory_path]]]
```

パラメータ

表A-11 cluvfy comp clumgrコマンドのパラメータ

パラメータ	説明
-n node_list -allnodes	検証を実行するノードのドメイン修飾なしの名前のカンマ区切りリスト、またはクラスタ内のすべてのノードです。 このオプションを指定しない場合、CVU は、ローカル・ノードのみをチェックします。
-json	-json フラグを使用すると、詳細レポートが JSON 形式で生成されます。 -json フラグを指定しない場合、CVU によって詳細レポートがテキスト・ファイルで生成されます。
-html	-html フラグを使用すると、詳細レポートが HTML 形式で生成されます。 -html フラグを指定し、CVU によって認識されるブラウザがシステムで使用可能な場合は、チェックの完了時に CVU によってそのブラウザが起動され、レポートがブラウザに表示されます。 -html フラグを指定しない場合、CVU によって詳細なレポート

パラメータ	説明
	がテキスト・ファイルで生成されます。
-save [-savedir directory_path]	検証レポート(cvuheckdcreport_timestamp.txt および cvuheckreport_timestamp.htm)を保存するには、-save フラグまたは -save -savedir フラグを使用します (timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。

関連項目:

レポート・オプションに関する情報などのCVU使用上のノートについては、[CVUの使用方法](#)

clufy comp dhcp

DHCPサーバーがネットワークに存在し、必要な数のIPアドレスを提供できることを検証します。

必要なIPアドレス数は次のように計算されます。

- 3つのSCAN VIP(クラスタのサイズにかかわらず必要)
- -nオプションで指定するノードごとに1つのノードVIP
- -vipresnameオプションで指定するアプリケーションVIPリソースごとに1つのアプリケーションVIP

この検証では、DHCPサーバーの応答時間も検証します。

構文

```
clufy comp dhcp -clustername cluster_name [-vipresname
application_vip_resource_name]
[-port dhcp_port] [-n node_list | -allnodes] [-method sudo -user user_name [-
location directory_path] | -method root]
[-networks network_list] [-save [-savedir directory_path]] [-verbose]
```

パラメータ

表A-12 clufy comp dhcpコマンドのパラメータ

パラメータ	説明
-clustername cluster_name	DHCP の整合性を確認するクラスタの名前を指定する必要があります。
-vipresname application_vip_resource_name	必要に応じて、アプリケーション VIP リソースの名前のカンマ区切

パラメータ	説明
	りリストを指定できます。
-port dhcp_port	必要に応じて、DHCP パッケージが送信されるポートを指定できます。デフォルトのポートは 67 です。
-n node_list -allnodes	オプションで、検証対象として、ノードのドメイン修飾なしの名前のカンマ区切りリスト、またはクラスタ内のすべてのノードを指定できます。 このパラメータを指定しない場合、CVU は、ローカル・ノードのみをチェックします。
-method sudo -user user_name [-location directory_path] -method root	必要に応じて、root ユーザー・アクセスの権限委任方法が sudo か root かを指定できます。sudo を指定すると、root 権限ですべてのノードにアクセスするためにユーザー名を指定する必要があります。必要に応じて、sudo 実行可能ファイルのファイル・システムのフルパスも指定します。
-networks network_list	必要に応じて、CVU でチェックするクラスタのネットワーク分類の、スラッシュ(/)で区切られたリストを指定でき、ここで各ネットワークは、 "if_name"[:subnet_id[:if_type[,if_type...]]]の形式です。 この形式では、if_name を二重引用符(" ")で囲む必要があります。また、正規表現(".*")を使用できます。たとえば、 "eth*"は eth1 や eth02 のようなインタフェースと一致します。subnet_id はネットワーク・インタフェースのサブネット番号です。if_type はインタフェース・タイプ ({CLUSTER_INTERCONNECT PUBLIC ASM})のカンマ区切りリストです。
-save [-savedir directory_path]	検証レポート(cvuchecdkreport_timestamp.txt および cvuchekreport_timestamp.htm)を保存するには、 -save フラグまたは -save -savedir フラグを使用します (timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、 CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレ

パラメータ	説明
	ポートが保存されるディレクトリを指定します。
-verbose	CVU は詳細な出力を表示します。

使用上のノート

- このコマンドは、rootで実行する必要があります。
- DHCP提供のIPアドレスを使用するように構成されたデフォルト・ネットワークOracle Clusterwareリソースがオンラインになっている場合は、このチェックを実行しないでください(VIPがリリースされ、クラスタがオンラインになってからDHCPによってIPが提供されているため、DHCPサーバーへの負荷を倍増させる必要がないためです)。
- このチェックはローカル・ノードで実行します。これは、ノード・リストに指定されたすべてのノードで実行する他のCVUコマンドとは異なります。この結果、-nオプションで指定したノード・リストにローカル・ノードが含まれなくても、エラー・メッセージがローカル・ノードに報告されます。
- このコマンドを実行する前に、ネットワーク・リソースがオフラインであることを確認します。必要に応じて、`srvctl stop nodeapps`コマンドを使用してネットワーク・リソースをオフラインにします。

関連項目:

`srvctl stop nodeapps`コマンドの詳細は、[『Oracle Real Application Clusters管理およびデプロイメント・ガイド』](#)を参照してください。

cluvfy comp dns

グリッド・ネーミング・サービス(GNS)サブドメインの委譲がドメイン名サービス(DNS)サーバーに適切に設定されていることを検証します。

構文

```
cluvfy comp dns -server -domain gns_sub_domain -vipaddress gns_vip_address [-port dns_port]
[-method sudo -user user_name [-location directory_path] | -method root] [-save [-savedir directory_path]] [-verbose]
cluvfy comp dns -client -domain gns_sub_domain -vip gns_vip [-port dns_port]
[-last] [-method sudo -user user_name [-location directory_path] | -method root] [-n node_list|-allnodes]
[-save [-savedir directory_path]] [-verbose]
```

パラメータ

表A-13 cluvfy comp dnsコマンドのパラメータ

パラメータ	説明
-server	-domain オプションに指定したドメインでリスニングする GNS サブドメインのテスト DNS サーバーを起動します。
-client	特定のアドレスで起動されるテスト DNS サーバーへの接続を検

パラメータ	説明
	証します。DNS サーバーの起動時に指定した情報と同じ内容を指定する必要があります。
-domain gns_sub_domain	GNS サブドメインの名前を指定します。
-vipaddress gns_vip_address	GNS 仮想 IP アドレスを{ip_name ip_address}/net_mask/interface_name の形式で指定します。ip_name(IP アドレスに解決される名前)または IP_address(IP アドレス)のいずれかを指定できます。名前またはアドレスの後に、net_mask(IP アドレスのサブネット・マスク)、および interface_name(IP アドレスを開始するインターフェース)が続きます。
-vip gns_vip	GNS の仮想 IP アドレスを指定し、それは IP アドレスに解決される名前またはドット区切りの 10 進数数字の IP アドレスのいずれかです。
-port dns_port	DNS サーバーがリスニングするポートを指定します。デフォルトのポートは 53 です。
-last	必要に応じて、このパラメータを使用して、すべての検証が完了すると、テスト DNS サーバーに終了リクエストを送信できます。
-method sudo -user user_name [-location directory_path] -method root	必要に応じて、root ユーザー・アクセスの権限委任方法が sudo か root かを指定できます。sudo を指定すると、root 権限ですべてのノードにアクセスするためにユーザー名を指定する必要があります。必要に応じて、sudo 実行可能ファイルのファイル・システムのフルパスも指定します。
-n node_list -allnodes	検証を実行するノードのドメイン修飾なしの名前のカンマ区切りリスト、またはクラスタ内のすべてのノードです。 このオプションを指定しない場合、CVU は、ローカル・ノードのみをチェックします。
-save [-savedir directory_path]	検証レポート(cvuchecdkreport_timestamp.txt および cvucheckreport_timestamp.htm)を保存するには、-save フラグまたは -save -savedir フラグを使用します (timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、

パラメータ	説明
	CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。
-verbose	CVU は詳細な出力を表示します。

使用上のノート

- このコマンドは、rootで実行する必要があります。
- クラスタの1つのノードでcluvfy comp dns -serverを実行します。
- 最後のノードで、-lastオプションを指定してcluvfy comp dns -serverインスタンスを終了します。
- クラスタのDNSサーバー設定を確認するには、指定したノード上、またはクラスタのすべてのノード上で、cluvfy comp dns -clientを実行します。
- GNSリソースがオンラインのときは、このコマンドを実行しないでください。
- このコマンドはWindowsではサポートされません。

cluvfy comp freespace

Oracle Clusterwareホームの記憶域で使用可能な空き領域をチェックし、領域の合計の5%以上が使用可能であることを確認します。

たとえば、記憶域の合計が10GBの場合、このチェックで500MB以上が空いていることを確認します。

構文

```
cluvfy comp freespace [-n node_list | -allnodes]
```

-nオプションを含める場合は、このコマンドの実行元として、ノードの名前のカンマ区切りリスト、またはクラスタ内のすべてのノードを指定します。

cluvfy comp gns

クラスタ上のグリッド・ネーミング・サービス(GNS)の整合性を検証します。

構文

```
cluvfy comp gns -precrsinst {-vip gns_vip [-domain gns_domain] | -clientdata  
file_name}  
[-networks network_list] [-n node_list | -allnodes] [-verbose]  
cluvfy comp gns -postcrsinst [-save [-savedir directory_path]] [-verbose]
```

パラメータ

表A-14 cluvfy comp gnsコマンドのパラメータ

パラメータ	説明

パラメータ	説明
-precrsinst	このパラメータを使用して、Oracle Clusterware のインストール前に、GNS ドメイン名と VIP アドレスをチェックします。
-vip gns_vip	GNS 仮想 IP アドレスを指定します。-vip と -domain を一緒に指定すると、このクラスタが GNS サーバー(ローカル GNS)になれるかどうかを CVU が検証します。
-domain gns_domain	必要に応じて、GNS サブドメイン名を指定できます。
-clientdata file_name	GNS 資格証明を含むファイルの名前を指定します。CVU は、このクラスタが指定されたクライアント・データを使用して、別の GNS サーバー・クラスタ(共有 GNS)のクライアント GNS クラスタになれることを検証します。
-networks network_list	CVU でチェックする、クラスタのネットワーク分類を GNS のパブリック・ネットワークを含めて、スラッシュ(/)で区切ってリストとして指定します。各ネットワークの形式は、 "if_name"[:subnet_id[:if_type[,if_type...]]]です。 この形式では、if_name を二重引用符(" ")で囲む必要があります。また、正規表現(". *")を使用できます。たとえば、 "eth*"は eth1 や eth02 のようなインタフェースと一致します。subnet_id はネットワーク・インタフェースのサブネット番号です。if_type はインタフェース・タイプ ({CLUSTER_INTERCONNECT PUBLIC ASM})のカンマ区切りリストです。
-n node_list -allnodes	検証を実行するノードのドメイン修飾なしの名前のカンマ区切りリスト、またはクラスタ内のすべてのノードです。 このオプションを指定しない場合、CVU は、ローカル・ノードのみをチェックします。
-postcrsinst	このパラメータを使用して、クラスタのすべてのノードの GNS の整合性を確認します。 このパラメータを使用する場合は、-verbose 以外の他のパラメータは使用できません。
-save [-savedir directory_path]	検証レポート(cvuchecdkreport_timestamp.txt およ

パラメータ	説明
	び cvucheckreport_timestamp.htm)を保存するには、 -save フラグまたは -save -savedir フラグを使用します (timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、 CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。
-verbose	CVU は詳細な出力を表示します。

cluvfy comp gnpn

クラスタ内のノードのリストでグリッド・プラグ・アンド・プレイの整合性をチェックします。

構文

```
cluvfy comp gnpn [-n node_list | -allnodes] [-save [-savedir directory_path]] [-verbose]
```

パラメータ

表A-15 cluvfy comp gnpnコマンドのパラメータ

パラメータ	説明
-n node_list -allnodes	検証を実行するノードのドメイン修飾なしの名前のカンマ区切りリスト、またはクラスタ内のすべてのノードです。 このオプションを指定しない場合、CVU は、ローカル・ノードのみをチェックします。
-save [-savedir directory_path]	検証レポート(cvuheckdreport_timestamp.txt および cvucheckreport_timestamp.htm)を保存するには、 -save フラグまたは -save -savedir フラグを使用します (timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、 CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。
-verbose	CVU は詳細な出力を表示します。

cluvfy comp ha

ローカル・ノードのOracle Restartの整合性をチェックします。

構文

```
cluvfy comp ha [-save [-savedir directory_path]] [-verbose]
```

検証レポート(cvuchecdkreport_timestamp.txtおよびcvuchekreport_timestamp.htm)を保存するには、-saveフラグまたは-save -savedirフラグを使用します(timestampは検証レポートの日時)。-saveフラグを単独で使用する場合、CVUは、CVU_home/cv/reportディレクトリ(CVU_homeはCVUバイナリの場所)にレポートを保存します。-save -savedirフラグを使用する場合、CVUによってレポートが保存されるディレクトリを指定します。-verboseオプションを指定すると、CVUは詳細な出力を表示します。

cluvfy comp healthcheck

Oracle ClusterwareおよびOracle Databaseインストールをチェックして、必須要件およびベスト・プラクティス・ガイドラインへの適合性を調べ、正しく動作していることを確認します。

構文

```
cluvfy comp healthcheck [-collect {cluster|database|asm}] [-db db_unique_name]
                        [-bestpractice|-mandatory] [-deviations] [-json] [-html] [-save [-savedir
directory_path]]
                        [-orachk_loc location] [-skip check_utilities]
```

パラメータ

表A-16 cluvfy comp healthcheckコマンドのパラメータ

パラメータ	説明
-collect {cluster database asm}	Oracle Clusterware (cluster)、Oracle Database (database)、または Oracle ASM のチェックの実行を指定するには、-collect を使用します。healthcheck オプションで -collect フラグを使用しない場合、CVU では 3 つすべてのチェックが実行されます。
-db db_unique_name	-db フラグの後に入力する特定のデータベースでのチェックを指定するには、-db を使用します。-db フラグを使用し、一意の名前を指定しない場合は、CVU によってエラーが報告されます。 CVU では、JDBC を使用して dbsnmp ユーザーとしてデータベースに接続し、様々なデータベース・パラメータが検証されます。
[-bestpractice -mandatory] [-deviations]	ベスト・プラクティスのチェックを指定するためには -bestpractice フラグを使用し、必須のチェックを指定するためには -mandatory フラグを使用します。ベスト・プラクティ

パラメータ	説明
	スの推奨事項または必須要件からの差異のみを確認することを指定するには、-deviations フラグを追加します。-bestpractice または-mandatory のいずれかのフラグを指定できますが、両方のフラグを指定することはできません。-bestpractice または-mandatory のいずれも指定しない場合は、CVU によってベスト・プラクティスと必須要件の両方が表示されます。
-html	-html フラグを使用すると、詳細レポートが HTML 形式で生成されます。 -html フラグを指定し、CVU によって認識されるブラウザがシステムで使用可能な場合は、チェックの完了時に CVU によってそのブラウザが起動され、レポートがブラウザに表示されます。 -html フラグを指定しない場合、CVU によって詳細なレポートがテキスト・ファイルで生成されます。
-save [-savedir directory_path]	検証レポート(cvuchecdkreport_timestamp.txt および cvuchekreport_timestamp.htm)を保存するには、-save フラグまたは-save -savedir フラグを使用します(timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。
-orachk_loc location	ORAchk バイナリの場所を指定します。

cluvfy comp nodeapp

指定したすべてのノードのノード・アプリケーション、つまりVIP、NETWORKおよびONSの存在をチェックします。

構文

```
cluvfy comp nodeapp [-n node_list | -allnodes] [-save [-savedir directory_path]] [-verbose]
```

パラメータ

表A-17 cluvfy comp nodeappコマンドのパラメータ

パラメータ	説明
-n node_list -allnodes	検証を実行するノードのドメイン修飾なしの名前のカンマ区切りリスト、またはクラスタ内のすべてのノードです。 このオプションを指定しない場合、CVU は、ローカル・ノードのみをチェックします。
-save [-savedir directory_path]	検証レポート(cvuheckcdkreport_timestamp.txt および cvucheckreport_timestamp.htm)を保存するには、-save フラグまたは -save -savedir フラグを使用します (timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。
-verbose	CVU は詳細な出力を表示します。

cluvfy comp nodecon

ノード・リストで指定されたノード間の接続性をチェックします。インタフェース・リストを指定した場合、CVUは、指定したインタフェースのみを使用して接続性をチェックします。

構文

```
cluvfy comp nodecon [-n node_list | -allnodes] [-networks network_list] [-save [-savedir directory_path]] [-verbose]
```

パラメータ

表A-18 cluvfy comp nodeconコマンドのパラメータ

パラメータ	説明
-n node_list -allnodes	検証を実行するノードのドメイン修飾なしの名前のカンマ区切りリスト、またはクラスタ内のすべてのノードを指定します。 このオプションを指定しない場合、CVU は、ローカル・ノードのみをチェックします。
-networks network_list	検証を実行するネットワークのスラッシュ(/)区切りリストを指定します。
-save [-savedir directory_path]	検証レポート(cvuheckcdkreport_timestamp.txt および cvucheckreport_timestamp.htm)を保存するには、

パラメータ	説明
	-save フラグまたは -save -savedir フラグを使用します (timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。
-verbose	CVU は詳細な出力を表示します。

使用上のノート

- このコマンドを冗長モードで実行して、インタフェース、IPアドレスおよびサブネット間のマッピングを識別できます。
- このコマンドは、Solarisプラットフォームでは、非推奨とマークされているIPアドレスのテストを省略します。
- CVUを使用して次の操作を実行するには、-networksパラメータを指定せず、-nにallを設定してnodeconコマンドを使用します。
 - クラスタ・ノード上で使用可能なすべてのネットワーク・インタフェースの検出
 - インタフェースの対応するIPアドレスおよびサブネットの確認
 - VIPとしての使用に適したインタフェースのリストおよびプライベート・インターコネクトのインタフェースのリストの取得
 - これらのインタフェースを介したすべてのノード間の接続性の検証

例

例A-1 特定のネットワーク・インタフェースを介したノード間の接続性の検証

インタフェースeth0を介したノードnode1とnode3の間の接続性を検証するには、次のコマンドを実行します。

```
cluvfy comp nodecon -n node1/node3 -networks eth0 -verbose
```

例A-2 使用可能なすべてのネットワーク・インタフェースの検出およびそれらのネットワーク・インタフェースを介したクラスタ内のノード間の接続性の検証

クラスタ・ノード上で使用可能なすべてのネットワーク・インタフェースを検出するには、次のコマンドを使用します。これで、CVUによって、インタフェースの対応するIPアドレスおよびサブネットが確認されます。次に、この情報を使用して、VIPとしての使用に適したインタフェースのリストおよびプライベート・インターコネクトのインタフェースのリストがCVUにより取得されます。最後に、これらのインタフェースを介したクラスタ内のすべてのノード間の接続性がCVUにより検証されます。

```
cluvfy comp nodecon -n all -verbose
```

cluvfy comp nodereach

ソース・ノードが、クラスタ内の他の特定のノードまたはすべてのノードと通信できるかどうかを確認します。

構文


```
cluvfy comp nodereach {-n node_list | -allnodes} [-srcnode node] [-save [-savedir
directory_path]] [-verbose]
```

パラメータ

表A-19 cluvfy comp nodereachコマンドのパラメータ

パラメータ	説明
-n node_list -allnodes	検証を実行するノードのドメイン修飾なしの名前のカンマ区切りリスト、またはクラスタ内のすべてのノードを指定します。
-srcnode node	必要に応じて、CVU による到達可能性テストの実行元となるソース・ノードの名前を指定します。ソース・ノードを指定しなかった場合は、コマンドの実行元のノードがソース・ノードとして使用されます。
-save [-savedir directory_path]	検証レポート(cvuheckdkreport_timestamp.txt および cvuheckreport_timestamp.htm)を保存するには、-save フラグまたは -save -savedir フラグを使用します (timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。
-verbose	CVU は詳細な出力を表示します。

例

ローカル・ノードからnode3にネットワーク経由で到達可能かどうかを検証するには、次のコマンドを使用します。

```
cluvfy comp nodereach -n node3
```

このコマンドを実行すると、次のような出力が生成されます。

```
Verifying node reachability
Checking node reachability...
Node reachability check passed from node "node1"
Verification of node reachability was successful.
```

cluvfy comp ocr

指定されたすべてのノードで、Oracle Cluster Registry (OCR)の整合性をチェックします。

構文

```
cluvfy comp ocr [-n node_list | -allnodes] [-method sudo -user user_name [-location
directory_path] | -method root]
[-verbose] [-json] [-html] [-save [-savedir directory_path]]
```

表A-20 cluvfy comp ocrコマンドのパラメータ

パラメータ	説明
<code>-n node_list -allnodes</code>	検証を実行するノードのドメイン修飾なしの名前のカンマ区切りリスト、またはクラスタ内のすべてのノードです。 このオプションを指定しない場合、CVU は、ローカル・ノードのみをチェックします。
<code>-method {sudo -user user_name [-location directory_path] root}</code>	root ユーザー・アクセスの権限委任方法が sudo か root かを指定します。sudo を指定すると、root 権限ですべてのノードにアクセスするためにユーザー名を指定する必要があります。必要に応じて、sudo 実行可能ファイルのファイル・システムのフルパスも指定します。
<code>-save [-savedir directory_path]</code>	検証レポート(cvuchecdkreport_timestamp.txt および cvuchekreport_timestamp.htm)を保存するには、 -save フラグまたは -save -savedir フラグを使用します (timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。
<code>-verbose</code>	CVU は詳細な出力を表示します。

使用上のノート

このコマンドは、OCRの内容の整合性を検証しません。OCRの内容を検証するには、OCRCHECKユーティリティを使用する必要があります。

例

ローカル・ノードでのOCRの整合性を検証するには、次のコマンドを実行します。

```
cluvfy comp ocr
```

このコマンドによって、次のような出力が返されます。

```
Verifying OCR integrity
Checking OCR integrity...
Checking the absence of a non-clustered configurationl...
All nodes free of non-clustered, local-only configurations
ASM Running check passed. ASM is running on all specified nodes
Checking OCR config file "/etc/oracle/ocr.loc"...
OCR config file "/etc/oracle/ocr.loc" check successful
Disk group for ocr location "+DATA" available on all the nodes
```

NOTE:

This check does not verify the integrity of the OCR contents. Execute 'ocrcheck' as a privileged user to verify the contents of OCR.

OCR integrity check passed

Verification of OCR integrity was successful.

cluvfy comp ohasd

Oracle高可用性サービス・デーモンの整合性をチェックします。

構文

```
cluvfy comp ohasd [-n node_list | -allnodes] [-save [-savedir directory_path]] [-verbose]
```

パラメータ

表A-21 cluvfy comp ocrコマンドのパラメータ

パラメータ	説明
-n node_list -allnodes	検証を実行するノードのドメイン修飾なしの名前のカンマ区切りリスト、またはクラスタ内のすべてのノードです。 このオプションを指定しない場合、CVU は、ローカル・ノードのみをチェックします。
-save [-savedir directory_path]	検証レポート(cvuchecdkreport_timestamp.txt および cvuchekreport_timestamp.htm)を保存するには、-save フラグまたは -save -savedir フラグを使用します (timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。
-verbose	CVU は詳細な出力を表示します。

例

Oracle高可用性サービス・デーモンがクラスタのすべてのノードで正常に動作しているかどうかを検証するには、次のコマンドを使用します。

```
cluvfy comp ohasd -n allnodes -verbose
```

このコマンドによって、次のような出力が返されます。

```
Verifying OHASD integrity
Checking OHASD integrity...
ohasd is running on node "node1"
ohasd is running on node "node2"
ohasd is running on node "node3"
```

```
ohasd is running on node "node4"
OHASD integrity check passed
Verification of OHASD integrity was successful.
```

cluvfy comp olr

ローカル・ノードでOracle Local Registry (OLR)の整合性をチェックします。

構文

```
cluvfy comp olr [-n node_list | -allnodes] [-save [-savedir directory_path]] [-verbose]
```

パラメータ

表A-22 cluvfy comp ocrコマンドのパラメータ

パラメータ	説明
-n node_list -allnodes	検証を実行するノードのドメイン修飾なしの名前のカンマ区切りリスト、またはクラスタ内のすべてのノードです。 このオプションを指定しない場合、CVU は、ローカル・ノードのみをチェックします。
-save [-savedir directory_path]	検証レポート(cvuchecdkreport_timestamp.txt および cvuchekreport_timestamp.htm)を保存するには、-save フラグまたは -save -savedir フラグを使用します (timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。
-verbose	CVU は詳細な出力を表示します。

使用上のノート

- このコマンドは、OLRの内容の整合性を検証しません。OLRの内容を検証するには、ocrcheck -localコマンドを使用する必要があります。

例

現行のノードでOLRの整合性を検証するには、次のコマンドを実行します。

```
cluvfy comp olr -verbose
```

このコマンドによって、次のような出力が返されます。

```
Verifying OLR integrity
Checking OLR integrity...
Checking OLR config file...
```

```

OLR config file check successful
Checking OLR file attributes...
OLR file check successful
WARNING:
This check does not verify the integrity of the OLR contents. Execute 'ocrcheck -
local' as a privileged user to verify the contents of OLR.
OLR integrity check passed
Verification of OLR integrity was successful.

```

cluvfy comp peer

指定されたノードについて、参照ノードに対する互換性とプロパティをチェックします。

デフォルトのユーザー・グループ名以外の互換性、および異なるリリースのOracleソフトウェアの互換性をチェックできます。このコマンドは、メモリとスワップ領域などの物理的な属性、ユーザーとグループの値、カーネル設定およびインストール済オペレーティング・システム・パッケージを比較します。

構文

```

cluvfy comp peer {-n node_list | -allnodes} [-refnode node]
    [-r {10.1 | 10.2 | 11.1 | 11.2 | 12.1 | 12.2 | 18 | 19}] [-orainv
orainventory_group]
    [-osdba osdba_group] [-osoper osoper_group] [-verbose] [-json] [-html] [-save [-
savedir directory_path]]

```

パラメータ

表A-23 cluvfy comp peerコマンドのパラメータ

パラメータ	説明
-n node_list -allnodes	検証対象として、ノードのドメイン修飾なしの名前のカンマ区切りリスト、またはクラスタ内のすべてのノードを指定します。 このオプションを指定しない場合、CVU は、ローカル・ノードのみをチェックします。
-refnode node	必要に応じて、他のノードとの互換性チェックのための参照としてCVU で使用するノードを指定します。このオプションを指定しない場合、CVU は、ノード・リスト内のすべてのノードについての値をレポートします。
-r {10.1 10.2 11.1 11.2 12.1 12.2 18 19}	必要に応じて、Oracle Clusterware または Oracle RAC のインストール要件に合わせて、CVU がチェックするソフトウェア・リリースを指定します。このオプションを指定しない場合、CVU は、Oracle Clusterware 19c または Oracle Database 19c であると想定します。
-orainv orainventory_group	必要に応じて、Oracle Inventory グループの名前を指定できます。このオプションを指定しない場合、CVU は、oinstall

パラメータ	説明
	をインベントリ・グループとして使用します。 ノート: このパラメータは、Windows システムでは使用できません。
-osdba osdba_group	必要に応じて、OSDBA グループの名前を指定できます。このオプションを指定しない場合、CVU は、dba を OSDBA グループとして使用します。 ノート: このパラメータは、Windows システムでは使用できません。
-json	-json フラグを使用すると、詳細レポートが JSON 形式で生成されます。 -json フラグを指定しない場合、CVU によって詳細レポートがテキスト・ファイルで生成されます。
-html	-html フラグを使用すると、詳細レポートが HTML 形式で生成されます。 -html フラグを指定し、CVU によって認識されるブラウザがシステムで使用可能な場合は、チェックの完了時に CVU によってそのブラウザが起動され、レポートがブラウザに表示されます。 -html フラグを指定しない場合、CVU によって詳細なレポートがテキスト・ファイルで生成されます。
-save [-savedir directory_path]	検証レポート(cvuchecdkreport_timestamp.txt および cvuchekreport_timestamp.htm)を保存するには、-save フラグまたは-save -savedir フラグを使用します(timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。
-verbose	CVU は詳細な出力を表示します。

使用上のノート

-refnodeオプションによるピアの比較では、他のノードと参照ノードのシステム・プロパティが比較されます。他のノードの値が参照ノードの値より大きくない場合、CVUはその比較に参照ノードからの偏差としてフラグを付けます。グループまたはユーザーが参照ノードまたは別のノードのいずれにも存在しない場合、CVUは一致と報告して、参照ノードからの偏差はないことを示します。

同様に、参照ノードよりも総メモリー量が少ないノードの比較結果は、CVUにより不一致と報告されます。

例

次のコマンドを実行すると、様々なノードに対して事前選択された複数のプロパティの値が表示されます。

```
cluvfy comp peer -n node1,node2,node4,node7 -verbose
```

cluvfy comp scan

単一クライアント・アクセス名(SCAN)の構成をチェックします。

構文

```
cluvfy comp scan [-save [-savedir directory_path]] [-verbose]
```

使用上のノート

- 検証レポート(cvuchecdkreport_timestamp.txtおよびcvucheckreport_timestamp.htm)を保存するには、-saveフラグまたは-save -savedirフラグを使用します(timestampは検証レポートの日時)。-saveフラグを単独で使用する場合、CVUは、CVU_home/cv/reportディレクトリ(CVU_homeはCVUバイナリの場所)にレポートを保存します。-save -savedirフラグを使用する場合、CVUによってレポートが保存されるディレクトリを指定します。
- オプションで-verboseを含めれば、詳細な出力を表示できます。

例

SCANおよびSCANリスナーが構成されていて、クラスタのすべてのノードで動作することを検証するには、次のコマンドを使用します。

```
$ cluvfy comp scan
```

このコマンドによって、次のような出力が返されます。

```
Verifying scan
Checking Single Client Access Name (SCAN)...
Checking TCP connectivity to SCAN Listeners...
TCP connectivity to SCAN Listeners exists on all cluster nodes
Checking name resolution setup for "node1.example.com"...
Verification of SCAN VIP and Listener setup passed
Verification of scan was successful.
```

cluvfy comp software

Oracleソフトウェアとともにインストールされたファイルと属性をチェックします。

構文

```
cluvfy comp software [-n node_list | -allnodes] [-d oracle_home]
[-allfiles] [-checksum] [-json] [-html] [-save [-savedir directory_path]] [-verbose]
```

パラメータ

表A-24 cluvfy comp softwareコマンドのパラメータ

パラメータ	説明

パラメータ	説明
-n node_list -allnodes	オプションで、検証対象として、ノードのドメイン修飾なしの名前のカンマ区切りリスト、またはクラスタ内のすべてのノードを指定できます。このオプションを指定しない場合、CVU は、ローカル・ノードのみをチェックします。
-d oracle_home	必要に応じて、Oracle Database ソフトウェアをインストールするディレクトリを指定できます。このオプションを指定しなかった場合、CVU は Grid ホームにインストールされたファイルを検証します。
-allfiles	このパラメータを指定すると、CVU は、指定された Oracle ホームのすべてのファイルの属性をチェックします。このパラメータを指定しない場合、CVU は、指定された Oracle ホームの lib、jlib および bin ファイルの属性をチェックします。
-json	-json フラグを使用すると、詳細レポートが JSON 形式で生成されます。 -json フラグを指定しない場合、CVU によって詳細レポートがテキスト・ファイルで生成されます。
-html	-html フラグを使用すると、詳細レポートが HTML 形式で生成されます。 -html フラグを指定し、CVU によって認識されるブラウザがシステムで使用可能な場合は、チェックの完了時に CVU によってそのブラウザが起動され、レポートがブラウザに表示されます。 -html フラグを指定しない場合、CVU によって詳細なレポートがテキスト・ファイルで生成されます。
-save [-savedir directory_path]	検証レポート(cvuchecdkreport_timestamp.txt および cvucheckreport_timestamp.htm)を保存するには、-save フラグまたは -save -savedir フラグを使用します (timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。
-verbose	CVU は詳細な出力を表示します。

使用上のノート

ここで、コマンドに関する追加情報を追加します。

例

Oracle Clusterware 19cのインストール済ファイルが正しく構成されていることを検証するには、次のようなコマンドを使用します。

```
$ cluvfy comp software -n all -verbose
```

このコマンドによって、次のような出力が返されます。

```
Verifying software
Check: Software
  1021 files verified
Software check passed
Verification of software was successful.
```

cluvfy comp space

指定されたすべてのノードについて、-lパラメータで指定した場所の空きディスク容量をチェックします。

構文

```
cluvfy comp space [-n node_list | -allnodes] -l storage_location -z disk_space {B | K | M | G}
[-save [-savedir directory_path]] [-verbose]
```

パラメータ

表A-25 cluvfy comp spaceコマンドのパラメータ

パラメータ	説明
-n node_list	検証を実行するノードのドメイン修飾なしの名前のカンマ区切りリスト、またはクラスタ内のすべてのノードを指定します。このパラメータを指定しない場合、CVU は、ローカル・ノードのみをチェックします。
-l storage_location	チェックする記憶域の場所へのディレクトリ・パスを指定します。
-z disk_space {B K M G}	必要なディスク領域を指定します。単位はバイト(B)、キロバイト(K)、メガバイト(M)またはギガバイト(G)です。数値とバイト単位を示す文字の間に空白を入れないでください(「2G」などとする)。整数のみを使用してください。
-save [-savedir directory_path]	検証レポート(cvuchecdkreport_timestamp.txt および cvuchekreport_timestamp.htm)を保存するには、-save フラグまたは -save -savedir フラグを使用します(timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、

パラメータ	説明
	CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。
-verbose	CVU は詳細な出力を表示します。

使用上のノート

spaceコンポーネントは、ブロック・デバイスおよびRAWデバイスをサポートしていません。

関連項目:

動作保証されている記憶域オプションの最新情報については、My Oracle Supportの動作保証についてのサイトを参照してください。

<https://support.oracle.com>

例

次のコマンドを実行すると、各ノードの/home/dbadmin/productsディレクトリに5GBの空き領域があるかどうかを検証できます。

```
$ cluvfy comp space -n all -l /home/dbadmin/products -z 5G -verbose
```

cluvfy comp ssa

指定された記憶域の場所の共有を検出してチェックするには、cluvfy comp ssaコンポーネント検証コマンドを使用します。CVUは、ノード・リスト内のノードの共有をチェックします。

構文

```
cluvfy comp ssa [-n node_list | -allnodes] [-s storage_path_list]
  [-t {gi_software | rac_software | data | ocr_vdisk}] [-asm] [-asmdev
asm_device_list]
  [-r {10.1 | 10.2 | 11.1 | 11.2 | 12.1 | 12.2 | 18.1 | 19.1}] [-save [-savedir
directory_path]] [-verbose]
```

パラメータ

表A-26 cluvfy comp ssaコマンドのパラメータ

パラメータ	説明
-n node_list -allnodes	検証を行う非ドメイン修飾ノード名またはすべてのノードのカンマ区切りリストです。 これらのオプションを指定しない場合、CVU は、ローカル・ノードの

パラメータ	説明
	みをチェックします。
-s storage_path_list	記憶域パスのカンマ区切りリストです(たとえば、 /dev/sda, /dev/sdb)。 -s オプションを指定しない場合、CVU は、サポートされている記憶域タイプを検出し、各記憶域タイプの共有をチェックします。
-t {gi_software rac_software data ocr_vdisk}	ストレージ・デバイスに格納される Oracle ファイルのタイプ (Oracle Grid Infrastructure バイナリ、Oracle RAC、または Oracle Database バイナリ)。 -t を指定しない場合、CVU は、data ファイル・タイプを検出またはチェックします。
-asm	このパラメータを指定して、Oracle ASM での使用に適した記憶域をすべて検出します。
-asmdev asm_device_list	記憶域の場所の共有をチェックする Oracle ASM デバイスのカンマ区切りリスト。リストにシェル・メタ文字が含まれる場合は、リストを二重引用符(" ")で囲みます。
-r {10.1 10.2 11.1 11.2 12.1 12.2 18 19 20}	必要に応じて、検証を実行する製品のリリース番号を指定できます。-r を指定しないと、CVU は 20 の検証を実行します。
-save [-savedir directory_path]	検証レポート(cvuchecdkreport_timestamp.txt および cvuchekreport_timestamp.htm)を保存するには、-save フラグまたは -save -savedir フラグを使用します (timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。
-verbose	CVU は詳細な出力を表示します。

使用上のノート

- 現行のリリースのc luvfyには、共有記憶域アクセス性のチェックに関して、Linuxに対する次のような制限があります。
- 現在、NAS記憶域およびOCFS2(バージョン1.2.1以上)がサポートされています。

関連項目:

NASマウント・オプションの詳細は、[『Oracle Grid Infrastructureインストール・ガイド』](#)を参照してください。

- NASでの共有をチェックするとき、`cluvfy`コマンドを実行するには、指定されたパスに対する書き込み権限が必要です。`cluvfy`のユーザーに書き込み権限が付与されていない場合、`cluvfy`はそのパスがnot sharedであるとレポートします。
- CVUを使用してLinuxシステム上のSCSIディスクの検出および共有記憶域のアクセス性のチェックを実行するには、CVUQDISKパッケージが必要です。CVUの使用を試行したときにOracle RAC環境のいずれかのノード上にCVUQDISKパッケージがインストールされていない場合、CVUでエラーが発生します。CVUQDISKパッケージのインストール方法の詳細は、「[Red Hat Linuxでの共有ディスクの検出](#)」を参照してください。

例

システム上で使用可能なすべての共有記憶域システムを検出するには、次のコマンドを実行します。

```
$ cluvfy comp ssa -allnodes -verbose
```

指定したOracle ASM検出文字列に基づいて、Oracle ASMでの使用に適した記憶域をすべて検出するには、次のコマンドを実行します。

```
$ cluvfy comp ssa -n node1,node2 -asm -asmdev "/dev/xda*"
```

次のようなコマンドを実行すると、クラスターノード全体のデータ・ファイルを格納する特定の記憶域の場所(Oracle ASMディスク・グループOCR13など)のアクセス性を検証できます。

```
$ cluvfy comp ssa -allnodes -s OCR13
```

このコマンドを実行すると、次のような出力が生成されます。

```
Verifying shared storage accessibility
Checking shared storage accessibility...
"OCR13" is shared
Shared storage check was successful on nodes "node1,node2,node3,node4"
Verification of shared storage accessibility was successful.
```

cluvfy comp sys

指定されたすべてのノードで、指定された製品の最小システム要件が満たされていることをチェックします。

構文

```
cluvfy comp sys [-n node_list | -allnodes] -p {crs | ha | database} [-r {10.1 | 10.2 | 11.1 | 11.2 | 12.1 | 12.2 | 18 | 19}]
  [-osdba osdba_group] [-osoper osoper_group] [-orainv orainventory_group] [-fixup]
  [-fixupnoexec]
  [-method sudo -user user_name [-location directory_path] | -method root] [-verbose]
  [-json] [-html] [-save [-savedir directory_path]]
```

パラメータ

表A-27 cluvfy comp sysコマンドのパラメータ

パラメータ	説明

パラメータ	説明
<code>-n node_list -flex -hub hub_list</code>	検証を行う非ドメイン修飾ノード名のカンマ区切りリストを指定します。このオプションを指定しない場合、CVU は、ローカル・ノードのみをチェックします。 なお、検証を実行するハブ・ノードのリストを指定することもできます。
<code>-p {crs ha database}</code>	CVU が Oracle Clusterware、Oracle Restart(HA)または Oracle RAC に関するシステム要件をチェックするかどうかを指定します。 ノート: Oracle では、Oracle Database 10g で Oracle Restart はサポートされていません。-p ha オプションで -r 10.1 10.2 を使用すると、CVU でエラーが戻されます。
<code>-r {10.1 10.2 11.1 11.2 12.1 12.2 18 19}</code>	Oracle Clusterware または Oracle RAC のインストール要件に合わせて、CVU がチェックする Oracle Database のリリースを指定します。このオプションを指定しない場合、CVU は、Oracle Database 19c であると想定します。
<code>-osdba osdba_group</code>	OSDBA グループの名前です。このオプションを指定しない場合、CVU は、dba を OSDBA グループとして使用します。
<code>-orainv orainventory_group</code>	Oracle Inventory グループの名前です。このオプションを指定しない場合、CVU は、oinstall をインベントリ・グループとして使用します。
<code>-fixup</code>	検証が失敗した場合に、可能であれば CVU で修正操作を実行するように指定します。
<code>-fixupnoexec</code>	検証が失敗した場合に、CVU で修正データを生成し、生成された修正を手動で実行するための指示を表示するように指定します。
<code>-method sudo -user user_name [-location directory_path] -method root</code>	root ユーザー・アクセスの権限委任方法が sudo か root かを指定します。sudo を指定すると、root 権限ですべてのノードにアクセスするためにユーザー名を指定する必要があります。必要に応じて、sudo 実行可能ファイルのファイル・システムのフルパスも指定します。

パラメータ	説明
-verbose	CVU は詳細な出力を表示します。
-json	-json フラグを使用すると、詳細レポートが JSON 形式で生成されます。 -json フラグを指定しない場合、CVU によって詳細レポートがテキスト・ファイルで生成されます。
-html	-html フラグを使用すると、詳細レポートが HTML 形式で生成されます。 -html フラグを指定し、CVU によって認識されるブラウザがシステムで使用可能な場合は、チェックの完了時に CVU によってそのブラウザが起動され、レポートがブラウザに表示されます。 -html フラグを指定しない場合、CVU によって詳細なレポートがテキスト・ファイルで生成されます。
-save [-savedir directory_path]	検証レポート(cvucheckreport_timestamp.txt および cvucheckreport_timestamp.htm)を保存するには、-save フラグまたは -save -savedir フラグを使用します(timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。

例

クラスタ・ノードnode1、node2およびnode3上にOracle Clusterware 19cをインストールするためのシステム要件を検証するには、次のコマンドを実行します。

```
cluvfy comp sys -n node1,node2,node3 -p crs -verbose
```

cluvfy comp vdisk

指定されたすべてのノードで、投票ファイルの構成とudev設定をチェックします。

関連項目:

udevの設定の詳細は、[『Oracle Grid Infrastructureインストレーションおよびアップグレード・ガイドfor Linux』](#)を参照してください

構文

```
cluvfy comp vdisk [-n node_list | -allnodes] [-save [-savedir directory_path]] [-verbose]
```


表A-28 cluvfy comp vdiskコマンドのパラメータ

パラメータ	説明
<code>-n node_list -allnodes</code>	検証を実行するノードのドメイン修飾なしの名前のカンマ区切りリスト、またはクラスタ内のすべてのノードです。 このオプションを指定しない場合、CVU は、ローカル・ノードのみをチェックします。
<code>-save [-savedir directory_path]</code>	検証レポート(cvuheckdkreport_timestamp.txt および cvucheckreport_timestamp.htm)を保存するには、 -save フラグまたは -save -savedir フラグを使用します (timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、 CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。
<code>-verbose</code>	CVU は詳細な出力を表示します。

cluvfy comp vital

Oracle Grid Infrastructureの重要なヘルスチェックを実行し、安定性に影響を与える可能性のある問題を報告します。

構文

```
cluvfy comp vital [-verbose|-progress] [-json] [-html] [-save [-savedir directory_path]]
```

パラメータ

表A-29 cluvfy comp vitalコマンドのパラメータ

パラメータ	説明
<code>[-verbose -progress]</code>	詳細な出力を出力するには -verbose フラグを使用し、コマンドの現在の進行状況を出力するには -progress フラグを使用します。
<code>-json</code>	-json フラグを使用すると、詳細レポートが JSON 形式で生成されます。 -json フラグを指定しない場合、CVU によって詳細レポートがテキスト・ファイルで生成されます。

パラメータ	説明
-html	-html フラグを使用すると、詳細レポートが HTML 形式で生成されます。 -html フラグを指定し、CVU によって認識されるブラウザがシステムで使用可能な場合は、チェックの完了時に CVU によってそのブラウザが起動され、レポートがブラウザに表示されます。 -html フラグを指定しない場合、CVU によって詳細なレポートがテキスト・ファイルで生成されます。
-save [-savedir directory_path]	検証レポート(cvuheckdreport_timestamp.txt および cvucheckreport_timestamp.htm)を保存するには、-save フラグまたは -save -savedir フラグを使用します (timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。

cluvfy stage acfscfg

Oracle Automatic Storage Management Cluster File System (Oracle ACFS)を構成する前に、クラスタ・ノードが正しく設定されているかを検証するには、cluvfy stage -pre acfscfg コマンドを使用します。Oracle ACFSを構成した後に既存のクラスタをチェックするには、cluvfy stage -post acfscfg コマンドを使用します。

構文

```
cluvfy stage -pre acfscfg {-n node_list | -allnodes} [-asmdev asm_device_list] [-afdconfig]
[-save [-savedir directory_path]] [-verbose]
cluvfy stage -post acfscfg {-n node_list | -allnodes} [-save [-savedir directory_path]] [-verbose]
```

パラメータ

表A-30 cluvfy stage acfscfg コマンドのパラメータ

パラメータ	説明
-n node_list -allnodes	Oracle ACFS の構成前と構成後の両方について、検証対象として、ノードのドメイン修飾なしの名前のカンマ区切りリスト、またはクラスタ内のすべてのノードを指定します。
-asmdev asm_device_list	Oracle ASM で使用する予定のデバイスのリストです。このオプションを指定しない場合、CVU は、内部のオペレーティング・システムに依存した値(たとえば Linux システムの場合は

パラメータ	説明
	/dev/raw/*)を使用します。
-afdconfig	Oracle ASM フィルタ・ドライバ構成の前提条件を検証します。
-save [-savedir directory_path]	検証レポート(cvuchecdkreport_timestamp.txt およ び cvucheckreport_timestamp.htm)を保存するには、 -save フラグまたは -save -savedir フラグを使用します (timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、 CVU_home/cv/report ディレクトリ(CVU_home は CVU バ イナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレ ポートが保存されるディレクトリを指定します。
-verbose	CVU は詳細な出力を表示します。

cluvfy stage appcluster

指定されたノードで、Oracle Clusterwareアプリケーション・クラスターインストールについて適切なステージング前チェックおよび
ステージング後チェックを実行します。

構文

```
cluvfy stage -pre appcluster {-n node_list | -allnodes} [-verbose]
cluvfy stage -post appcluster {-n node_list | -allnodes} [-method sudo -user
user_name
[-location dir_path] | -method root] [-verbose]
```

パラメータ

表A-31 cluvfy stage appclusterコマンドのパラメータ

パラメータ	説明
-n node_list -allnodes	検証を実行する非ドメイン修飾ノード名のカンマ区切りリス トを指定します。クラスター内のすべてのノードで検証を実行す るには、-allnodes を指定します。
-method sudo -user user_name[-location dir_path] -method root	root ユーザー・アクセスの権限委任方法が sudo か root かを選択します。 sudo 方法を選択した場合は、root 権限ですべてのノ ードにアクセスするためにユーザー名を指定する必要がありま す。必要に応じて、sudo 実行可能ファイルのファイル・シス

パラメータ	説明
	テムのフルパスも指定します。

cluvfy stage cfs

OCFS2を構成する前にクラスタ・ノードが正しく設定されているかどうかを検証するには、`cluvfy stage -pre cfs`ステージ検証コマンドを使用します。OCFS2を構成した後に指定ノードで適切なチェックを実行するには、`cluvfy stage -post cfs`ステージ検証コマンドを使用します。

関連項目:

サポートされた共有記憶域タイプのリストについては、ご使用のプラットフォームの『[Oracle Grid Infrastructureインストールおよびアップグレード・ガイド](#)』を参照してください

構文

```
cluvfy stage -pre cfs {-n node_list | -allnodes} -s storageID_list [-save [-savedir
directory_path]] [-verbose]
cluvfy stage -post cfs {-n node_list | -allnodes} -f file_system [-save [-savedir
directory_path]] [-verbose]
```

パラメータ

表A-32 cluvfy stage cfsコマンドのパラメータ

パラメータ	説明
<code>-n node_list -allnodes</code>	OCFS2 の構成前と構成後の両方について、検証を行う非ドメイン修飾ノード名のカンマ区切りリストを指定します。クラスタ内のすべてのノードで検証を実行するには、 <code>-allnodes</code> を指定します。
<code>-s storageID_list</code>	OCFS2 の構成前にチェックする記憶域の場所のカンマ区切りリストを指定します。
<code>-f file_system</code>	OCFS2 の構成後にチェックするファイル・システムを指定します。
<code>-save [-savedir directory_path]</code>	検証レポート(<code>cvuheckdkreport_timestamp.txt</code> および <code>cvuheckdkreport_timestamp.htm</code>)を保存するには、 <code>-save</code> フラグまたは <code>-save -savedir</code> フラグを使用します (timestamp は検証レポートの日時)。 <code>-save</code> フラグを単独で使用する場合、CVU は、 <code>CVU_home/cv/report</code> ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 <code>-save -savedir</code> フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。

パラメータ	説明
-verbose	CVU は詳細な出力を表示します。

例

OCFS2の設定前に共有デバイスが正しく構成されているかどうかをチェックするには、次のようなコマンドを使用します。
/dev/sdd5をシステムの共有デバイスの名前に置き換えます。

```
$ cluvfy stage -pre cfs -n node1,node2,node3,node4 -s /dev/sdd5
```

cluvfy stage crsinst

Oracle Clusterwareをインストールまたはアップグレードする前に、cluvfy stage -pre crsinstコマンドと-file、-n、-flexまたは-upgradeパラメータを使用して、指定したノードをチェックします。Oracle Clusterwareをインストールした後、指定ノードをチェックするには、cluvfy stage -post crsinstコマンドを使用します。

構文

```
cluvfy stage -pre crsinst -responseFile file_name [-fixup] [-fixupnoexec]
  [-method sudo -user user_name [-location directory_path] | -method root] [-r
{10.1|10.2|11.1|11.2|12.1|12.2|18|19}]
  [-verbose] [-json] [-html] [-save [-savedir directory_path]] [-orachk_loc location]
[-skip check_utilities]
cluvfy stage -pre crsinst -file config_file [-fixup] [-fixupnoexec]
  [-method sudo -user user_name [-location directory_path] | -method root] [-verbose]
[-json] [-html] [-save [-savedir directory_path]]
cluvfy stage -pre crsinst {-n node_list | -allnodes}
  [-r {10.1 | 10.2 | 11.1 | 11.2 | 12.1 | 12.2 | 18 | 19}] [-c ocr_location_list] [-q
voting_disk_list]
  [-osdba osdba_group] [-osoper osoper_group] [-orainv orainventory_group] [-asm [-
presence {local | flex}
  | -asmcredentials client_data_file] [-asmgrp asmadmin_group] [-asmdev
asm_device_list] [-afdconfig]]
  [-crshome Grid_home] [-fixup] [-fixupnoexec] [-method sudo -user user_name [-
location directory_path] | -method root]
  [-networks network_list] [-dhcp -clustername cluster_name [-dhcpport dhcp_port]] [-
dns -vipaddress gns_vip_address
  [-domain gns_sub_domain]] [-software_only] [-verbose] [-orachk_loc location] [-skip
check_utilities]
  [-json] [-html] [-save [-savedir directory_path]] [-verbose]
cluvfy stage -pre crsinst -upgrade [-rolling] [-src_crshome src_crshome] -
dest_crshome dest_crshome
  -dest_version dest_version [-fixup] [-fixupnoexec] [-method sudo -user user_name
  [-location directory_path] | -method root] [-verbose] [-json] [-html] [-save [-
savedir directory_path]] [-orachk_loc location] [-skip check_utilities]
cluvfy stage -post crsinst {-n node_list | -allnodes}
  [-method sudo -user user_name [-location directory_path] | -method root] [-verbose]
[-verbose] [-json] [-html] [-save [-savedir directory_path]]
```

パラメータ

表A-33 cluvfy stage [-pre | -post] crsinstコマンドのパラメータ

パラメータ	説明
-responseFile	Oracle Universal Installer によって作成され、Oracle インストール変数を含むレスポンス・ファイルを指定します。

パラメータ	説明
-file config_file	Oracle のインストール変数を含む root スクリプト構成ファイルを指定します。
-n node_list -allnodes	検証を行う非ドメイン修飾ノード名のカンマ区切りリストを指定します。クラスタ内のすべてのノードで検証を実行するには、-allnodes を指定します。
-r {10.1 10.2 11.1 11.2 12.1 12.2 18 19}	Oracle Clusterware のインストール要件に合わせて、CVU がチェックする Oracle Clusterware のリリースを指定します。このオプションを指定しない場合、CVU は、Oracle Clusterware 19c であると想定します。
-c ocr_location_list	CVU がすべてのノードの可用性を検証する、OCR の場所またはファイルのディレクトリ・パスのカンマ区切りリストを指定します。このオプションを指定しない場合、OCR の場所はチェックされません。
-q voting_disk_list	CVU がすべてのノードの可用性を検証する、投票ファイルのディレクトリ・パスのカンマ区切りリストを指定します。このオプションを指定しなかった場合、CVU は投票ファイルの場所をチェックしません。
-osdba osdba_group	OSDBA グループの名前を指定します。このオプションを指定しない場合、CVU は、dba を OSDBA グループとして使用します。
-osoper osoper_group	OSOPER グループの名前を指定します。
-orainv orainventory_group	Oracle Inventory グループの名前を指定します。このオプションを指定しない場合、CVU は、oinstall をインベントリ・グループとして使用します。
-asm [-presence {local flex} -asmcredentials client_data_file	このパラメータは、Oracle ASM が Oracle Clusterware ファイルの格納に使用されることを示します。 この Oracle Clusterware インストールでの Oracle ASM の存在 (LOCAL または FLEX) を指定します。Oracle ASM クライアントの場合は、必要に応じて、Oracle ASM クライアント資格証明ファイルのパスを指定できます。

パラメータ	説明
-asmgrp asmadmin_group	OSASM グループの名前を指定します。このパラメータを指定しなかった場合、CVU は Oracle Inventory グループと同じグループを使用します。
-asmdev asm_device_list	CVU がすべてのノードの可用性を検証する、Oracle ASM で使用する予定のデバイスのリストを指定します。 このパラメータを指定しなかった場合、CVU は、内部のオペレーティング・システムに依存した値を使用します。
-afdconfig	Oracle ASM フィルタ・ドライバ構成の前提条件を検証します。
-crshome Grid_home	Oracle Grid Infrastructure または Oracle Clusterware ホーム・ディレクトリの場所を指定します。このパラメータを指定した場合、提供されたファイル・システムの場所で、Oracle Clusterware インストール用に十分な空き領域があるかどうかをチェックされます。
-networks network_list	"if_name" [:subnet_id [:public :cluster_interconnect]]の形式のネットワークのスラッシュ("/")区切りリストのネットワーク・パラメータを検証します。 ● ネットワーク・インタフェース名(if_name)を指定する場合、eth*のようにワイルドカード文字としてアスタリスク(*)を使用し、インタフェースを照合することができます。 ● subnet_id 変数のネットワーク・インタフェースのサブネット番号を指定し、ネットワーク・インタフェースのタイプを選択します。
-dhcp -clustername cluster_name [-dhcpport dhcp_port]	クラスタの名前を指定します。必要に応じて、DHCP パケットが送信されるポートを指定できます。このポートのデフォルト値は 67 です。
-dns -vipaddress gns_vip_address	GNS 仮想 IP アドレスを{ip_name ip_address}/net_mask/interface_name の形式で指定します。ip_name(IP アドレスに解決される名前)または IP_address(IP アドレス)のいずれかを指定できます。名前またはアドレスの後に、net_mask(IP アドレスのサブネット・マス

パラメータ	説明
	ク)、および interface_name(IP アドレスを開始するインタフェース)が続きます。
-domain gns_sub_domain	GNS サブドメインの名前を指定します。
-upgrade	アップグレードの前提条件を確認するにはこのパラメータを指定します。
-rolling	ローリング・アップグレードを実行するには、このパラメータを指定します。
-src_crshome src_crshome	アップグレード元の Grid ホームの場所を指定します。
-dest_crshome dest_crshome	アップグレード先の Grid ホームの場所を指定します。
-dest_version dest_version	アップグレード先のバージョンをパッチセットを含めて指定します(たとえば、11.2.0.1.0 または 11.2.0.2.0)。
-fixup	検証が失敗した場合に、可能であれば CVU で修正操作を実行するように指定します。
-fixupnoexec	検証が失敗した場合に、CVU で修正データを生成し、生成された修正を手動で実行するための指示を表示するように指定します。
-method sudo -user user_name [-location dir_path] -method root	root ユーザー・アクセスの権限委任方法が sudo か root かを指定します。sudo を指定すると、root 権限ですべてのノードにアクセスするためにユーザー名を指定する必要があります。必要に応じて、sudo 実行可能ファイルのファイル・システムのフルパスも指定します。
-verbose	CVU は詳細な出力を表示します。
-json	-json フラグを使用すると、詳細レポートが JSON 形式で生成されます。 -json フラグを指定しない場合、CVU によって詳細レポートがテキスト・ファイルで生成されます。
-html	-html フラグを使用すると、詳細レポートが HTML 形式で生

パラメータ	説明
	成されます。 -html フラグを指定し、CVU によって認識されるブラウザがシステムで使用可能な場合は、チェックの完了時に CVU によってそのブラウザが起動され、レポートがブラウザに表示されます。 -html フラグを指定しない場合、CVU によって詳細なレポートがテキスト・ファイルで生成されます。
-save [-savedir directory_path]	検証レポート(cvuchecdkreport_timestamp.txt および cvuchekreport_timestamp.htm)を保存するには、-save フラグまたは -save -savedir フラグを使用します(timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。

使用上のノート

- 新規インストールのためにチェックを実行するには、-fileまたは-nパラメータを指定します。別のバージョンへのアップグレードのチェックを実行するには、-upgradeを使用します。
- -cおよび-qオプションと-nパラメータを指定すると、OCRおよび投票ファイルに対する追加のチェックがCVUにより実行されます。

cluvfy stage dbcfg

Oracle RACデータベースを構成する前に指定したノードをチェックして、システムがデータベースの作成またはデータベースの構成の変更を行うためのすべての基準を満たしているかどうかを検証します。

構文

LinuxおよびUNIXプラットフォームの場合:

```
cluvfy stage -pre dbcfg {-n node_list | -allnodes} -d Oracle_home [-fixup] [-fixupnoexec]
[-method sudo -user user_name [-location directory_path] | root] [-verbose] [-orachk_loc location]
[-skip check_utilities] [-json] [-html] [-save [-savedir directory_path]]
```

Windowsプラットフォームの場合:

```
cluvfy stage -pre dbcfg {-n node_list | -allnodes} -d oracle_home [-fixup] [-fixupnoexec]
[-verbose] [-json] [-html] [-save [-savedir directory_path]]
```

パラメータ

表A-34 cluvfy stage -pre dbcfgコマンドのパラメータ

パラメータ	説明
-n node_list -allnodes	検証を行う非ドメイン修飾ノード名のカンマ区切りリストです。クラスタ内のすべてのノードで検証を実行するには、-allnodes を指定します。
-d Oracle_home	チェックされているデータベースの Oracle ホーム・ディレクトリの場所です。
-fixup	検証が失敗した場合に、可能であれば CVU で修正操作を実行するように指定します。
-fixupnoexec	検証が失敗した場合に、CVU で修正データを生成し、生成された修正を手動で実行するための指示を表示するように指定します。
-method sudo -user user_name [-location directory_path] root	root ユーザー・アクセスの権限委任方法が sudo か root かを指定します。sudo を指定すると、root 権限ですべてのノードにアクセスするためにユーザー名を指定する必要があります。必要に応じて、sudo 実行可能ファイルのファイル・システムのフルパスも指定します。
-save [-savedir directory_path]	検証レポート(cvuchecdkreport_timestamp.txt および cvuchekreport_timestamp.htm)を保存するには、-save フラグまたは -save -savedir フラグを使用します (timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。
-verbose	CVU は詳細な出力を表示します。
-json	-json フラグを使用すると、詳細レポートが JSON 形式で生成されます。 -json フラグを指定しない場合、CVU によって詳細レポートがテキスト・ファイルで生成されます。
-html	-html フラグを使用すると、詳細レポートが HTML 形式で生成されます。 -html フラグを指定し、CVU によって認識されるブラウザがシス

パラメータ	説明
	テムで使用可能な場合は、チェックの完了時に CVU によってそのブラウザが起動され、レポートがブラウザに表示されます。 -html フラグを指定しない場合、CVU によって詳細なレポートがテキスト・ファイルで生成されます。
-save [-savedir directory_path]	検証レポート(cvuchecdkreport_timestamp.txt および cvuchekreport_timestamp.htm)を保存するには、-save フラグまたは-save -savedir フラグを使用します (timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。

cluvfy stage -pre dbinst

Oracle RACデータベースのインストールまたは作成の前に指定したノードをチェックして、システムがOracle RACデータベースのインストールまたは作成のすべての基準を満たしているかどうかを検証します。

構文

LinuxおよびUNIXプラットフォームの場合:

```
cluvfy stage -pre dbinst {-n node_list | -allnodes} [-r {10.1 | 10.2 | 11.1 | 11.2 | 12.1 | 12.2 | 18 | 19}]
  [-osdba osdba_group] [-osoper osoper_group] [-osbackup osbackup_group] [-osdg
osdg_group]
  [-oskm oskm_group] [-d oracle_home] [-fixup] [-fixupnoexec] [-method sudo -user
user_name]
  [-location directory_path] | -method root] [-verbose] [-orachk_loc location] [-skip
check_utilities]
  [-json] [-html][-save [-savedir directory_path]]
cluvfy stage pre dbinst -upgrade -src_dbhome src_dbhome [-dbname dbname-list]
  -dest_dbhome dest_dbhome -dest_version dest_version [-pddb dbname-list]
  [-fixup] [-fixupnoexec] [-method sudo -user user_name [-location directory_path | -
method root] [-verbose]
  [-orachk_loc location] [-skip check_utilities] [-json] [-html] [-save [-savedir
directory_path]]
```

Windowsプラットフォームの場合:

```
cluvfy stage -pre dbinst {-n node_list | -allnodes} [-r {10.1 | 10.2 | 11.1 | 11.2 | 12.1 | 12.2 | 18 | 19}]
  [-d Oracle_home] [-fixup] [-fixupnoexec] [-serviceuser user_name [-servicepasswd]]
  [-verbose] [-json] [-html] [-save [-savedir directory_path]]
cluvfy stage -pre dbinst -upgrade -src_dbhome src_dbhome [-dbname dbname-list]
  -dest_dbhome dest_dbhome -dest_version dest_version [-fixup] [-fixupnoexec]
  [-verbose] [-json] [-html] [-save [-savedir directory_path]]
```

パラメータ

表A-35 cluvfy stage -pre dbinstコマンドのパラメータ

パラメータ	説明
-n node_list -allnodes	検証を行う非ドメイン修飾ノード名のカンマ区切りリストを指定します。クラスタ内のすべてのノードで検証を実行するには、-allnodes を指定します。
-r {10.1 10.2 11.1 11.2 12.1 12.2 18 19}	Oracle RAC のインストール要件に合わせて、CVU がチェックする Oracle Database のリリースを指定します。このオプションを指定しない場合、CVU は、Oracle Database 19c であると想定します。
-osdba osdba_group	OSDBA グループの名前です。このオプションを指定しない場合、CVU は、dba を OSDBA グループとして使用します。
-osoper osoper_group	OSOPER グループの名前を指定します。
-osbackup osbackup_group	OSBACKUP グループの名前を指定します。
-osdg osdg_group	OSDG グループの名前を指定します。
-oskm oskm_group	OSKM グループの名前を指定します。
-d oracle_home	Oracle RAC のインストールおよび Oracle RAC データベースの作成を行う Oracle ホーム・ディレクトリの場所です。このパラメータを指定した場合、指定した場所にデータベース・インストール用に十分な空きディスク領域があるかどうかチェックされます。
-fixup	検証が失敗した場合に、可能であれば CVU で修正操作を実行するように指定します。
-fixupnoexec	検証が失敗した場合に、CVU で修正データを生成し、生成された修正を手動で実行するための指示を表示するように指定します。
-upgrade	アップグレードの前提条件を確認するにはこのパラメータを指定します。
-src_dbhome src_dbhome	アップグレード元のデータベース・ホームの場所を指定します。
-dbname dbname-list	アップグレードするデータベースの一意名のカンマ区切りリストで

パラメータ	説明
	指定します。
-dest_dbhome dest_dbhome	アップグレード先のデータベース・ホームの場所を指定します。
-dest_version dest_version	アップグレード先のバージョンをパッチセットを含めて指定します (たとえば、11.2.0.1.0 または 11.2.0.2.0)。
-pdba	アップグレードするプラグブル・データベース(PDB)の一意名のカンマ区切りリストを指定します。
-method sudo -user user_name [-location directory_path] -method root	root ユーザー・アクセスの権限委任方法が sudo か root かを指定します。sudo を指定すると、root 権限ですべてのノードにアクセスするためにユーザー名を指定する必要があります。必要に応じて、sudo 実行可能ファイルのファイル・システムのフルパスも指定します。
-serviceuser user_name [-servicepasswd]	このオプションを指定した場合、CVU は次の内容をチェックします。 ● 指定したユーザーがドメイン・ユーザーかどうか。指定したユーザーがドメイン・ユーザーでない場合、CVU からエラーが戻され、ユーザーの検証についてそれ以降のチェックは実行されません。 ノート: Oracle ホーム・ユーザーのユーザー名を指定する必要があります。 ● 指定したユーザーがクラスタ内のすべてのノードで管理者かどうか。ユーザーがクラスタ内のいずれかのノードで管理者でない場合、チェックにパスします。それ以外の場合、チェックに失敗します。 ● -servicepasswd オプションを指定しなかった場合、CVU は、そのユーザーのパスワードが OCR のウォレットに格納されているかどうかをチェックします。指定したユーザーのパスワードが存在しない場合、CVU は動作を続行します。 ● -servicepasswd オプションを指定した場合、パスワードがウォレットが存在する場合でも、指定したユーザーのパスワードが CVU から求められます。

パラメータ	説明
	ノート: -serviceuser パラメータと-servicepwd パラメータは Windows のみに適用されます。
-save [-savedir directory_path]	検証レポート(cvuchecdkreport_timestamp.txt および cvucheckreport_timestamp.htm)を保存するには、-save フラグまたは-save -savedir フラグを使用します(timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。
-verbose	CVU は詳細な出力を表示します。

cluvfy stage hacfg

Oracle Restartの構成の前と後にローカル・ノードをチェックします。

構文

```
cluvfy stage -pre hacfg [-osdba osdba_group] [-osoper osoper_group] [-orainv
orainventory_group]
  [-fixup] [-fixupnoexec] [-method sudo -user user_name [-location directory_path] |
-method root]
  [-verbose] [-json] [-html] [-save [-savedir directory_path]]
cluvfy stage -post hacfg [-verbose] [-json] [-html] [-save [-savedir directory_path]]
```

パラメータ

表A-36 cluvfy stage hacfgコマンドのパラメータ

パラメータ	説明
-osdba osdba_group	OSDBA グループの名前を指定します。このオプションを指定しない場合、CVU は、dba を OSDBA グループとして使用します。
-osoper osoper_group	OSOPER グループの名前を指定します。
-orainv orainventory_group	Oracle Inventory グループの名前を指定します。このオプションを指定しない場合、CVU は、oinstall をインベントリ・グループとして使用します。
-fixup	検証が失敗した場合に、可能であれば CVU で修正操作を実

パラメータ	説明
	行するように指定します。
-fixupnoexec	検証が失敗した場合に、CVU で修正データを生成し、生成された修正を手動で実行するための指示を表示するように指定します。
-method sudo -user user_name [-location dir_path] -method root	root ユーザー・アクセスの権限委任方法が sudo か root かを指定します。sudo を指定すると、root 権限ですべてのノードにアクセスするためにユーザー名を指定する必要があります。必要に応じて、sudo 実行可能ファイルのファイル・システムのフルパスも指定します。
-save [-savedir directory_path]	検証レポート(cvuchecdkreport_timestamp.txt および cvuchekreport_timestamp.htm)を保存するには、-save フラグまたは-save -savedir フラグを使用します(timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。
-verbose	CVU は詳細な出力を表示します。
-json	-json フラグを使用すると、詳細レポートが JSON 形式で生成されます。 -json フラグを指定しない場合、CVU によって詳細レポートがテキスト・ファイルで生成されます。
-html	-html フラグを使用すると、詳細レポートが HTML 形式で生成されます。 -html フラグを指定し、CVU によって認識されるブラウザがシステムで使用可能な場合は、チェックの完了時に CVU によってそのブラウザが起動され、レポートがブラウザに表示されます。 -html フラグを指定しない場合、CVU によって詳細なレポートがテキスト・ファイルで生成されます。
-save [-savedir directory_path]	検証レポート(cvuchecdkreport_timestamp.txt および cvuchekreport_timestamp.htm)を保存するには、-save フラグまたは-save -savedir フラグを使用しま

パラメータ	説明
	す(timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。

cluvfy stage hwos

クラスタ内の指定されたノードでネットワークおよび記憶域をチェックします。また、このコマンドは、サポートされる記憶域タイプをチェックし、共有に関して各記憶域タイプをチェックします。

構文

```
cluvfy stage -post hwos {-n node_list | -allnodes} [-s storageID_list] [-save [-savedir directory_path]] [-verbose]
```

パラメータ

表A-37 cluvfy stage hwosコマンドのパラメータ

パラメータ	説明
-n node_list	検証を行う非ドメイン修飾ノード名のカンマ区切りリストです。クラスタ内のすべてのノードで検証を実行するには、-allnodes を指定します。
-s storageID_list	サポートされている記憶域タイプの共有用の記憶域の場所のカンマ区切りリストをチェックします。 -s パラメータを指定しなかった場合、CVU は、サポートされている記憶域タイプを検出し、各記憶域タイプの共有をチェックします。
-save [-savedir directory_path]	検証レポート(cvuchecdkreport_timestamp.txt および cvuchekreport_timestamp.htm)を保存するには、-save フラグまたは -save -savedir フラグを使用します (timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。
-verbose	CVU は詳細な出力を表示します。

cluvfy stage nodeadd

指定したノードを既存のクラスタに追加する前に、そのノードが正しく構成されているかどうかを検証する場合、および指定したノードを追加する前にクラスタの整合性を検証する場合は、`cluvfy stage -pre nodeadd`コマンドを使用します。指定したノードがネットワーク、共有記憶域およびクラスタウェアのレベルでクラスタに正常に追加されていることを検証するには、`cluvfy stage -post nodeadd`コマンドを使用します。

`cluvfy stage -pre nodeadd`コマンドは、追加するノードのシステム構成(オペレーティング・システムのバージョン、ソフトウェア・パッチ、パッケージ、カーネル・パラメータなど)が既存のクラスタ・ノードと互換性があること、およびクラスタウェアが既存のノードで正常に動作していることを検証します。既存のクラスタの任意のノードでこのコマンドを実行します。

構文

```
cluvfy stage -pre nodeadd -n node_list [-vip vip_list] [-fixup] [-fixupnoexec]
  [-method sudo -user user_name [-location directory_path] | -method root]
  [-verbose] [-json] [-html] [-save [-savedir directory_path]]
cluvfy stage -post nodeadd -n node_list [-verbose] [-json] [-html] [-save [-savedir
directory_path]]
```

パラメータ

表A-38 cluvfy stage nodeaddコマンドのパラメータ

パラメータ	説明
<code>-n node_list</code>	検証を行う非ドメイン修飾ノード名のカンマ区切りリストを指定します。これらは、クラスタに追加するノード、または追加されたノードです。
<code>-vip vip_list</code>	新規ノードで使用される仮想 IP アドレスのカンマ区切りリストです。
<code>-flex [-hub hub_list [-vip vip_list]]</code>	Oracle Flex Cluster にノードを追加する場合、 <code>-flex</code> を指定します。必要に応じて、クラスタにハブ・ノードとして追加する、ドメインで修飾されていないノード名のカンマ区切りリストを指定できます。 さらに、指定したハブ・ノードのリストに適用される仮想 IP アドレスのカンマ区切りリストも指定できます。
<code>-fixup</code>	このオプションのパラメータを指定すると、検証が失敗した場合に、可能であれば CVU で修正操作が実行されます。
<code>-fixupnoexec</code>	このオプションのパラメータを指定すると、検証が失敗した場合に、CVU で修正データが生成され、生成された修正を手動で実行するための指示が表示されます。
<code>-method sudo -user user_name [-</code>	<code>root</code> ユーザー・アクセスの権限委任方法が <code>sudo</code> か <code>root</code> かを

パラメータ	説明
location directory_path] -method root	指定します。sudo を指定すると、root 権限ですべてのノードにアクセスするためにユーザー名を指定する必要があります。必要に応じて、sudo 実行可能ファイルのファイル・システムのフルパスも指定します。
-verbose	CVU は詳細な出力を表示します。
-json	-json フラグを使用すると、詳細レポートが JSON 形式で生成されます。 -json フラグを指定しない場合、CVU によって詳細レポートがテキスト・ファイルで生成されます。
-html	-html フラグを使用すると、詳細レポートが HTML 形式で生成されます。 -html フラグを指定し、CVU によって認識されるブラウザがシステムで使用可能な場合は、チェックの完了時に CVU によってそのブラウザが起動され、レポートがブラウザに表示されます。 -html フラグを指定しない場合、CVU によって詳細なレポートがテキスト・ファイルで生成されます。
-save [-savedir directory_path]	検証レポート(cvucheckreport_timestamp.txt および cvucheckreport_timestamp.htm)を保存するには、-save フラグまたは -save -savedir フラグを使用します (timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。

cluvfy stage nodedel

特定のノードがクラスタから正常に削除されたことを確認します。通常、このコマンドは、ノード固有のインタフェース構成の詳細が削除されたこと、ノードがクラスタ構成の一部ではなくなったこと、および適切な Oracle ASM クリーンアップが実行されたことを検証します。

構文

```
cluvfy stage -post nodedel -n node_list [-save [-savedir directory_path]] [-verbose]
```

パラメータ

表A-39 cluvfy stage nodedelコマンドのパラメータ

パラメータ	説明
-n node_list -allnodes	OCFS2 の構成前と構成後の両方について、検証を行う非ドメイン修飾ノード名のカンマ区切りリストを指定します。クラスタ内のすべてのノードで検証を実行するには、-allnodes を指定します。
-save [-savedir directory_path]	検証レポート(cvuheckdkreport_timestamp.txt および cvucheckreport_timestamp.htm)を保存するには、-save フラグまたは -save -savedir フラグを使用します (timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。
-verbose	CVU は詳細な出力を表示します。

関連項目:

[クラスタ・ノードの追加および削除](#)

使用上のノート

- cluvfy stage -post nodedelの検証に失敗した場合、ノードの削除手順を繰り返します。

cluvfy stage osupg

ノードリスト内のすべてのノードでOracle Grid Infrastructure前提条件チェックを実行し、オペレーティング・システム・アップグレードの準備状況を確認します。

構文



ノート:

このコマンドは Linux プラットフォームでのみサポートされます。

```
cluvfy stage -pre osupg {-n node_list | -allnodes} [-checkGIdriverSupport
kernel_information]
[-verbose] [-json] [-html] [-save [-savedir directory_path]]
```

パラメータ

表A-40 cluvfy stage osupgコマンドのパラメータ

パラメータ	説明

パラメータ	説明
-n node_list -allnodes	検証を実行するノードのドメイン修飾なしの名前のカンマ区切りリスト、またはクラスタ内のすべてのノードです。 このオプションを指定しない場合、CVU は、ローカル・ノードのみをチェックします。
-checkGIdriverSupport kernel_information	オプションで、オペレーティング・システムのカーネル・バージョンまたはカーネル RPM ファイルへの絶対パスを指定して、Oracle Grid Infrastructure ドライバのサポートを確認します。
-verbose	CVU は詳細な出力を表示します。
-json	-json フラグを使用すると、詳細レポートが JSON 形式で生成されます。 -json フラグを指定しない場合、CVU によって詳細レポートがテキスト・ファイルで生成されます。
-html	-html フラグを使用すると、詳細レポートが HTML 形式で生成されます。 -html フラグを指定し、CVU によって認識されるブラウザがシステムで使用可能な場合は、チェックの完了時に CVU によってそのブラウザが起動され、レポートがブラウザに表示されます。 -html フラグを指定しない場合、CVU によって詳細なレポートがテキスト・ファイルで生成されます。
-save [-savedir directory_path]	検証レポート (cvuchecdkreport_timestamp.txt および cvuchekreport_timestamp.htm)を保存するには、-save フラグまたは-save -savedir フラグを使用します(timestamp は検証レポートの日時)。 -save フラグを単独で使用する場合、CVU は、CVU_home/cv/report ディレクトリ(CVU_home は CVU バイナリの場所)にレポートを保存します。 -save -savedir フラグを使用する場合、CVU によってレポートが保存されるディレクトリを指定します。

cluvfy stage patch

すべてのノードでパッチ適用の適切なステージング前チェックを実行します。パッチ適用のステージング後チェックを実行します。

構文

```
cluvfy stage -pre patch [-db -dbhome oracle_home [-dbname dbname-list]] [-non_rolling]
    [-verbose] [-orachk_loc location] [-skip check_utilities]
cluvfy stage -post patch [-db -dbhome oracle_home] [-verbose]
```

パラメータ

表A-41 cluvfy stage patchコマンドのパラメータ

パラメータ	説明
-db -dbhome oracle_home	パッチを適用するデータベース・ホームの場所を指定します。
-dbname dbname-list	パッチを適用するデータベースの一意名のカンマ区切りリストを指定します。
-non_rolling	非ローリング・モードでパッチを適用するには、-non_rolling フラグを使用します。
-verbose	CVU は詳細な出力を表示します。

使用上のノート

デフォルトでは、ステー징前チェックはローリング・モードでパッチ適用のために実行されます。

CVUのトラブルシューティングおよび診断出力

この項では、次のCVUのトラブルシューティングに関する項目について説明します。

- [トレースの有効化](#)
- [クラスタ検証ユーティリティの既知の問題](#)

トレースの有効化

トレースを無効にしないかぎり、CVUによってトレース・ファイルが生成されます。トレースを無効化するには、SRVM_TRACE環境変数をfalseまたはFALSEに設定します。たとえば、tcshでsetenv SRVM_TRACE FALSEなどのエントリを使用するとトレースが無効になります。

デフォルトではCVUトレース・ファイルはORACLE_BASE/crsdata/host_name/cvuディレクトリに作成されます。ログ・ファイルは自動的にローテーションされ、最新のログ・ファイルの名前はcvutrace.log.0になります。必要に応じて不要なログ・ファイルを削除またはアーカイブしてディスク領域を解放する必要があります。

Oracle Clusterwareによって、CVUの定期的な実行時に生成されたログ・ファイルがORACLE_BASE/crsdata/host_name/cvu/cvutrcディレクトリに格納されます。

トレース・ファイルのデフォルト以外の場所を使用するには、CV_TRACELOC環境変数に適切なトレース・ディレクトリの絶対パスを設定します。

クラスタ検証ユーティリティの既知の問題

この項では、次に示すクラスタ検証ユーティリティ(CVU)の既知の制限について説明します。

- [クラスタ検証ユーティリティによってサポートされるデータベースのバージョン](#)
- [Linuxの共有記憶域アクセス性\(ssa\)チェックのレポートの制限](#)
- [Red Hat Linuxでの共有ディスクの検出](#)

クラスタ検証ユーティリティによってサポートされるデータベースのバージョン

現行のリリースのCVUでは、Oracle Database 10g以上、Oracle RACおよびOracle Clusterwareのみがサポートされており、CVUには下位互換性がありません。CVUは、Oracle Database 10gより前のOracle Database製品をチェックまたは検証することはできません。

Linuxの共有記憶域アクセス性(ssa)チェックのレポートの制限

現行のリリースのc_luvfyには、共有記憶域アクセス性のチェックに関して、Linuxに対する次のような制限があります。

- OCFS2(バージョン1.2.1以上)がサポートされています。
- NASの共有性チェックの場合、c_luvfyコマンドを実行するには、指定されたパスに対する書込み権限が必要です。
c_luvfyコマンドを実行するユーザーに書込み権限が付与されていない場合、c_luvfyはそのパスがnot sharedであるとレポートします。

Red Hat Linuxでの共有ディスクの検出

CVUを使用してRed Hat Linux 5.0(以上)、Oracle Linux 5.0 (以上)およびSUSE Linux Enterprise Server上のSCSIディスクの検出および共有記憶域のアクセス性のチェックを実行する場合、CVUQDISKパッケージが必要です。CVUの使用を試行したときにOracle RAC環境のいずれかのノード上にCVUQDISKパッケージがインストールされていない場合、CVUでエラーが発生します。

CVUQDISKパッケージをインストールするには、次の手順を実行します。

1. rootユーザーとしてログインします。
2. パッケージcvuqdisk-1.0.9-1.rpm (またはこれ以上のバージョン)をローカル・ディレクトリにコピーします。このrpmは、Oracle Clusterwareのインストール・メディアの最上位ディレクトリのrpmサブディレクトリ内にあります。たとえば、cvuqdisk-1.0.9-1.rpmは、/mountpoint/clusterware/rpm/というディレクトリにあり、この場合のmountpointはディレクトリが配置されているディスクのマウント・ポイントです。

```
# cp /mount_point/clusterware/rpm/cvuqdisk-1.0.9-1.rpm /u01/oradba
```

3. CVUQDISK_GRP環境変数を、CVUQDISKパッケージ・バイナリを所有する必要があるオペレーティング・システム・グループに設定します。CVUQDISK_GRPが設定されていない場合、デフォルトで、oinstallグループが所有者グループになります。

```
# set CVUQDISK_GRP=oinstall
```

4. コマンドrpm -q cvuqdiskを実行して、以前のバージョンのCVUQDISKパッケージがインストールされているかどうかを判別します。以前のバージョンのCVUQDISKパッケージが検出された場合、コマンドrpm -e cvuqdisk previous_versionを実行してこのパッケージを削除します(この場合のprevious_versionは、以前のバージョンのCVUQDISKの識別子です)。

```
# rpm -q cvuqdisk
cvuqdisk-1.0.2-1
# rpm -e cvuqdisk-1.0.2-1
```

5. コマンド `rpm -iv cvuqdisk-1.0.9-1.rpm`を実行して、最新のCVUQDISKパッケージをインストールします。

```
# cd /u01/oradba
# rpm -iv cvuqdisk-1.0.9-1.rpm
```

脚注の凡例

脚注1: UNIXプラットフォームでのみ適用可能

B Oracle Clusterwareのリソース・リファレンス

この付録は、Oracle Clusterwareのリソースに対するリファレンスです。この付録には、リソース属性の説明と使用例、およびリソース属性のアクション・スクリプトの詳細な説明および例が示されています。この付録の内容は次のとおりです。

- [リソース属性](#)
- [サード・パーティ製アプリケーションのアクション・スクリプトの例](#)

リソース属性

この項では、アプリケーションをOracle Clusterwareのリソースとして登録する際に使用する属性について説明します。この属性は、次に示すように、`crsctl add resource`コマンドで使用します。

```
$ crsctl add resource resource_name -type resource_type -group group_name  
{[-attr "attribute_name='attribute_value', attribute_name='attribute_value'  
, ..."] | [-file file_name]}
```

属性/値ペアは、`-attr`フラグの後にカンマ区切りのリストとして指定し、各属性の値は一重引用符(' ')で囲みます。一部のリソース属性は、構成できない読み取り専用の属性です。

または、属性/値ペアを含むテキスト・ファイルを作成できます。たとえば：

```
PLACEMENT=favored  
HOSTING_MEMBERS=node1 node2 node3  
RESTART_ATTEMPTS@CARDINALITYID(1)=0  
RESTART_ATTEMPTS@CARDINALITYID(2)=0  
FAILURE_THRESHOLD@CARDINALITYID(1)=2  
FAILURE_THRESHOLD@CARDINALITYID(2)=4  
FAILURE_INTERVAL@CARDINALITYID(1)=300  
FAILURE_INTERVAL@CARDINALITYID(2)=500  
CHECK_INTERVAL=2  
CARDINALITY=2
```

ノート：



このような属性の長さは、最大で 254 文字に制限されています。

この項には次のトピックが含まれます：

- [構成可能なリソース属性](#)
- [読み取り専用のリソース属性](#)

構成可能なリソース属性

この項では、アプリケーションをOracle Clusterwareのリソースとして登録する際に構成可能な次のリソース属性について説明します。

ノート：



すべての属性の値は小文字で指定する必要があります。属性名はすべて大文字で指定する必要があります。

- [ACL](#)
- [ACTION_SCRIPT](#)
- [ACTION_TIMEOUT](#)
- [ACTIONS](#)
- [ACTIVE_PLACEMENT](#)
- [AGENT_FILENAME](#)
- [ALERT_TEMPLATE](#)
- [AUTO_START](#)
- [CARDINALITY](#)
- [CARDINALITY_ID](#)
- [CHECK_INTERVAL](#)
- [CHECK_TIMEOUT](#)
- [CLEAN_TIMEOUT](#)
- [CRITICAL_RESOURCES](#)
- [DELETE_TIMEOUT](#)
- [DESCRIPTION](#)
- [ENABLED](#)
- [FAILURE_INTERVAL](#)
- [FAILURE_THRESHOLD](#)
- [HOSTING_MEMBERS](#)
- [INSTANCE_FAILOVER](#)
- [INTERMEDIATE_TIMEOUT](#)
- [LOAD](#)
- [MODIFY_TIMEOUT](#)
- [NAME](#)
- [OFFLINE_CHECK_INTERVAL](#)
- [ONLINE_RELOCATION_TIMEOUT](#)
- [PLACEMENT](#)
- [RELOCATE_BY_DEPENDENCY](#)
- [RELOCATE_KIND](#)
- [RESTART_ATTEMPTS](#)

- [SCRIPT_TIMEOUT](#)
- [SERVER_CATEGORY](#)
- [SERVER_POOLS](#)
- [START_CONCURRENCY](#)
- [START_DEPENDENCIES](#)
- [START_TIMEOUT](#)
- [STOP_CONCURRENCY](#)
- [STOP_DEPENDENCIES](#)
- [STOP_TIMEOUT](#)
- [UPTIME_THRESHOLD](#)
- [USER_WORKLOAD](#)
- [USE_STICKINESS](#)

ACL

ACLは、リソースの所有者、および様々なオペレーティング・システムのユーザーとグループに付与するアクセス権限を定義します。リソースの所有者には、所有者になるオペレーティング・システム・ユーザーとその権限を定義します。

これは、リソースの作成時に構成するオプションの属性です。この属性を構成しない場合、リソースを作成するプロセスのIDに基づいて値が決定されます。リソースの既存の権限に基づいて変更が許可される場合には、属性の値を変更できます。

ノート:



owner、pggrp、user および group を含むすべてのオペレーティング・システムのユーザー名およびユーザー・グループは、クラスタ内のすべてのサーバー上に登録される必要があります。

文字列は次のように指定します。

- owner: リソースを所有するオペレーティング・システム・ユーザーと、アクション・スクリプトまたはアプリケーション固有のエージェントを実行するユーザーの後に、所有者の権限が続きます。
- pggrp: リソース所有者のプライマリ・グループであるオペレーティング・システム・グループです。続いてプライマリ・グループのメンバーの権限を指定します。
- other: 所有者でもプライマリ・グループのメンバーでもないオペレーティング・システム・ユーザー
- r: リソース、その状態および構成を表示する権限のみを付与する読取りオプション
- w: リソースの属性を変更する権限およびリソースを削除する権限を付与する書込みオプション
- x: リソースを開始、停止および再配置する権限を付与する実行オプション

デフォルトでは、リソースを作成するクライアントのIDはownerです。また、デフォルトでは、root、およびownerに指定されたユーザーは完全な権限を持ちます。ACL属性に次の行を追加することで、必要なオペレーティング・システム・ユーザーおよびオペレーティング・システム・グループに権限を付与できます。

```
user:user_name:rwX
```

```
group:group_name:rwx
```

使用例

```
ACL=owner:user_1:rwx,pgrp:osdba:rwx,other::r-
```

前述の例では、リソースの所有者はuser_1で、プライマリ・グループはosdbaです。他のユーザーはリソースを表示することしかできませんが、ユーザーuser_1は、osdbaグループと同様すべての権限を持ちます。

ACTION_SCRIPT

アクション・スクリプトのパスおよびファイル名を含む絶対ファイル名。AGENT_FILENAME属性に指定されたエージェントが、ACTION_SCRIPT属性に指定されたスクリプトをコールします。

使用例

```
ACTION_SCRIPT=fully_qualified_path_to_action_script
```

ACTION_TIMEOUT

リソース上でOracle Clusterwareによって実行可能な、サポート対象のすべてのアクションに関する単一のタイムアウト値(秒単位)。

使用例

```
ACTION_TIMEOUT=30
```

ACTIONS

ACTIONS属性によって、リソース上でOracle Clusterwareが実行可能なアクションおよびこのアクションに対応する権限をリストする、名前の表が宣言されます。ACTIONS属性には、アクションの仕様に関するスペース区切りリストが含まれ、各仕様は次の形式となっています。

- actionNameは、アクション名です(最大長は32バイトで、US7ASCIIの英数字に対応し、大文字小文字を区別します)。
- userNameは、当該アクションを実行可能な、オペレーティング・システムのユーザー名です。
- groupNameは、当該アクションを実行可能な、オペレーティング・システムのグループ名です。

```
actionName [,user:userName | group:groupName][ ...]
```

userNameまたはgroupNameを指定しない場合、Oracle Clusterwareでは、誰でも当該アクションにアクセス可能であると想定します。

使用例

次の例によって、複数のアクションが利用可能となります。

```
ACTIONS='action1 action2,user:user2 action3,group:group1'
```

ACTIVE_PLACEMENT

1に設定すると、Oracle Clusterwareは、この属性を使用してクラスタ・サーバーの追加または再起動時にリソースの配置を再評価します。PLACEMENT=favoredのリソースの場合、Oracle Clusterwareは、favoredのサーバーがクラスタに追加されているときにリソースがfavored以外のサーバーで実行されていると、実行中のリソースを再配置することがあります。

```
ACTIVE_PLACEMENT=1
```

AGENT_FILENAME

リソース・タイプがそのリソースを管理するために使用するエージェント・プログラムの完全修飾されたファイル名。すべてのリソース・タイプに、そのリソースを管理するためのエージェント・プログラムが必要です。この属性に値を指定するか、またはベースになるリソース・タイプから値を継承することで、リソース・タイプはエージェント・プログラムを使用できます。Oracle Clusterware 12c以降のリリースには、applicationおよびscriptagentの2つのスクリプト・エージェントが含まれています。Oracle Clusterwareは、非推奨のapplicationリソース・タイプのリソースに対してapplicationスクリプト・エージェントを使用します。この属性のデフォルト値はscriptagentです。



ノート:

一度リソースが作成されると、この属性を変更できません。

使用例

```
AGENT_FILENAME=%Grid_home%/bin/application
```

ALERT_TEMPLATE

リソース状態のアラート・メッセージに含めようとする、追加のリソース属性を指定するために使用します。リソース属性に関するスペース区切りのリストとして、属性を指定できます。このような属性は、アラート・メッセージに表示されるために、リソース・タイプからアクセス可能である必要があります。

使用例

```
ALERT_TEMPLATE="DESCRIPTION HOSTING_MEMBERS"
```

AUTO_START

クラスタ・サーバーの再起動後、Oracle Clusterwareによってリソースを自動的に起動するかどうかを示します。

AUTO_STARTの有効な値は次のとおりです。

- **always**: サーバーが停止したときのリソースの状態に関係なく、サーバーの再起動時にリソースを再起動します。
- **restore**: サーバーが停止したときと同じ状態にリソースをリストアします。サーバーが停止する前にTARGETの値がONLINEだった場合、Oracle Clusterwareはリソースの再起動を試行します。
- **never**: サーバーが停止したときのリソースの状態に関係なく、Oracle Clusterwareはリソースを再起動しません。

CARDINALITY

リソースまたはリソース・グループを同時に実行できるサーバーの数。これが、リソース・カーディナリティの上限になります。

使用例

```
CARDINALITY=1
```

リソースまたはリソース・グループが実行されるように構成されているサーバー・プールに割り当てられるサーバー数とともに、カーディナリティが常に増減するような値を使用することもできます。値は、次のとおりです。


```
CARDINALITY=%CRS_SERVER_POOL_SIZE%  
CARDINALITY=%CRS_HUB_SIZE%
```

リソースでPLACEMENT=restrictedが指定され、SERVER_POOLS属性を使用する場合にのみ、この値を使用できます。

CARDINALITY_ID

INSTANCE_COUNTリソース・インスタンスごとに異なる整数。

このパラメータの値は1からCARDINALITY*2です。リソース・インスタンスの数は、オンライン再配置(RELOCATE_KIND=online)を使用する場合、またはリソース・インスタンス数を減らす場合にのみ、CARDINALITYの値よりも大きくすることができます。

CHECK_INTERVAL

checkアクションの実行を繰り返す時間間隔(秒単位)。間隔を短くするとより頻繁にチェックできますが、スクリプト・エージェントを使用するとリソースの消費も増大します。リソースの消費を低減するには、アプリケーション固有のエージェントを使用します。

使用例

```
CHECK_INTERVAL=60
```

CHECK_TIMEOUT

チェック・アクションを実行できる最大時間(秒単位)。アクションが指定された時間内に完了しなかった場合、Oracle Clusterwareはエラー・メッセージを戻します。この属性が指定されていない場合または0秒が指定されている場合、Oracle Clusterwareは[SCRIPT_TIMEOUT](#)属性の値を使用します。

使用例

```
CHECK_TIMEOUT=30
```

CLEAN_TIMEOUT

クリーンアップ・アクションを実行できる最大時間(秒単位)。アクションが指定された時間内に完了しなかった場合、Oracle Clusterwareはエラー・メッセージを戻します。この属性に値が指定されていなかったり、0秒が指定されている場合、Oracle Clusterwareは[STOP_TIMEOUT](#)属性の値を使用します。

使用例

```
CLEAN_TIMEOUT=30
```

CRITICAL_RESOURCES

リソース・グループでクリティカルとマークされたリソースのリストを含むリソース・グループ属性。

クリティカル・リソースの空白区切りリストを、リソース・グループのCRITICAL_RESOURCES属性で指定します。ローカルおよびクラスタの両方のリソース・グループ・タイプにこの属性を使用します。type: オプションにリソース・タイプ名を続けて追加することで、指定したタイプのすべてのメンバー・リソースがグループのクリティカル・リソースであることを示すことができます。

例B-1 使用例

```
CRITICAL_RESOURCES="r1 r2 r3"  
CRITICAL_RESOURCES="appvip type:ora.export.type"
```

DELETE_TIMEOUT

削除アクションを実行できる最大時間(秒単位)。アクションが指定された時間内に完了しなかった場合、Oracle Clusterwareはエラー・メッセージを戻します。この属性に値が指定されていなかったり、0秒が指定されている場合、Oracle Clusterwareは[SCRIPT_TIMEOUT](#)属性の値を使用します。

使用例

```
DELETE_TIMEOUT=30
```

DESCRIPTION

追加するリソースの説明を入力します。

使用例

```
DESCRIPTION=Apache Web server
```

ENABLED

Oracle Clusterwareは、この属性を使用してリソースの状態を管理します。Oracle Clusterwareは、直接的にも、別のリソースへの依存性による場合にも無効(ENABLED=0)になっているリソースの管理は試行しません。無効になっているリソースを起動することはできませんが、停止することはできます。Oracle Clusterwareは、無効なリソースをアクティブに監視せず、つまり、Oracle Clusterwareは無効なリソースの状態をチェックしません。

使用例

```
ENABLED=1
```

FAILURE_INTERVAL

リソースがFAILURE_THRESHOLD属性で指定した障害の回数を超えた場合にOracle Clusterwareがリソースを停止するまでの間隔(秒単位)。値が0(ゼロ)の場合、障害の追跡は無効になります。

使用例

```
FAILURE_INTERVAL=30
```

FAILURE_THRESHOLD

リソースに指定されたFAILURE_INTERVAL内で検出される障害の数で、この値を超えると、Oracle Clusterwareがリソースを使用不可としてマーク付けし、そのリソースの監視を停止します。指定した回数の障害がリソースで発生すると、Oracle Clusterwareはそのリソースを停止します。値が0(ゼロ)の場合、障害の追跡は無効になります。最大値は20です。

使用例

```
FAILURE_THRESHOLD=3
```

HOSTING_MEMBERS

リソースのホスティングが可能なクラスタ・サーバーの名前のリストで、スペース区切りで順序付けします。この属性は、管理者管理型を使用し、PLACEMENT属性の値がfavoredまたはrestrictedに設定されている場合にのみ指定する必要があります。アプリケーションをOracle Clusterwareリソースとして登録する際は、かわりにSERVER_POOLS属性を使用します。

ノート:



application タイプのリソースの場合、Oracle Clusterware は、汎用サーバー・プールの HOSTING_MEMBERS 属性に示されたサーバーを配置の対象とします。

関連項目:

- 汎用サーバー・プールの詳細は、「[デフォルト・サーバー・プール](#)」を参照してください。
- この属性の詳細は、「[LOAD](#)」を参照してください。
- この属性の詳細は、「[PLACEMENT](#)」を参照してください。

候補ノード名のリストを取得するには、olsnodesコマンドを実行して、使用しているサーバー名のリストを表示します。

使用例

```
HOSTING_MEMBERS=server1 server2 server3
```

INSTANCE_FAILOVER

CLUSTER_RESOURCEタイプのリソースには、INSTANCE_FAILOVER属性を使用します。この属性を使用すると、障害が発生したサーバーからのリソース・インスタンスのフェイルオーバーを拒否できます。これによって、特定のサーバーにリソースを結合できます。

0を設定して、インスタンス・フェイルオーバーを無効にします。

使用例

```
INSTANCE_FAILOVER=1
```

INTERMEDIATE_TIMEOUT

リソースに障害が発生したと宣言される前に、リソースがINTERMEDIATE状態のままである最大時間を(秒単位)表示します。INTERMEDIATE_TIMEOUTの値が有効となるには、0よりも大きいものである必要があります。

使用例

```
INTERMEDIATE_TIMEOUT=60
```

LOAD

Oracle Clusterwareは、PLACEMENT属性の値とともにこの属性の値を解析します。PLACEMENTの値がbalancedの場合は、LOADの値によってリソースの最適な配置場所が決定されます。これは、リソースの1つのインスタンスが消費するサーバー容量を、他のリソースとの相対値として数値的に示す負以外の数値です。Oracle Clusterwareは、実行中のリソースによる負荷の合計が最小のサーバーにリソースの配置を試行します。

使用例

```
LOAD=1
```

MODIFY_TIMEOUT

修正アクションを実行できる最大時間(秒単位)。アクションが指定された時間内に完了しなかった場合、Oracle

Clusterwareはエラー・メッセージを戻します。この属性に値が指定されていなかったり、0秒が指定されている場合、Oracle Clusterwareは[SCRIPT_TIMEOUT](#)属性の値を使用します。

使用例

```
MODIFY_TIMEOUT=30
```

NAME

リソースの名前を示す英数字の文字列で、大/小文字が区別されます。ネーミング規則として、名前の先頭を英数字の接頭辞(myApacheなど)にし、末尾をリソースについて説明する識別子にすることをお勧めします。リソース名には、プラットフォームでサポートされているすべての文字(感嘆符(!)およびチルダ(~)を除く)を使用できます。リソース名の先頭をピリオド(.)またはoraという文字列にすることはできません。

使用例

```
NAME=myApache
```

OFFLINE_CHECK_INTERVAL

リソースのオフライン監視を制御します。値は、リソースの状態がOFFLINEの場合に、Oracle Clusterwareがそのリソースを監視する時間間隔(秒単位)を表します。値が0の場合、監視は無効になります。

使用例

```
OFFLINE_CHECK_INTERVAL=30
```

ONLINE_RELOCATION_TIMEOUT

再配置されたインスタンスが起動されサービスが移行されてから、前のサービス・インスタンスが停止されるまでの時間(分)。デフォルト値は、30分です。

関連項目:

- オンラインでのデータベースの再配置については、[Oracle Real Application Clusters管理およびデプロイメント・ガイド](#)
- srvctl relocate databaseについては、[Oracle Real Application Clusters管理およびデプロイメント・ガイド](#)

PLACEMENT

Oracle Clusterwareがリソースを起動するクラスタ・サーバーを選択する方法を指定します。有効な値は、balanced、favoredまたはrestrictedです。

PLACEMENT属性をfavoredまたはrestrictedに設定した場合は、SERVER_POOLSおよびHOSTING_MEMBERS属性にも値を割り当てる必要があります。PLACEMENT属性の値をbalancedに設定した場合は、HOSTING_MEMBERS属性を指定する必要はありません。

関連項目:

- PLACEMENT属性の詳細は、「[アプリケーションの配置ポリシー](#)」を参照してください。

- この属性の詳細は、「[HOSTING_MEMBERS](#)」を参照してください。
- この属性の詳細は、「[SERVER_POOLS](#)」を参照してください。

使用例

```
PLACEMENT=favored
```

RELOCATE_KIND

リソースが再配置される方法を制御する場合に使用します。オフライン再配置の処理中に、ターゲット・リソース・インスタンスがソース・サーバーで停止され、その後、宛先サーバーで起動されます。オンライン再配置の操作では、リソース・インスタンスをソース・サーバーで停止する前に、宛先サーバーで最初に起動します。start-onlyは、宛先サーバーのリソース・インスタンスを起動しますが、ソース・サーバーのリソース・インスタンスは停止しません。かわりに、再配置が完了すると、エージェントがリソース・インスタンスを停止します。

このパラメータの値として、offline/0、online/1またはstart-only/2を選択できます。

使用例

```
RELOCATE_KIND=1  
RELOCATE_KIND=start-only
```

RELOCATE_BY_DEPENDENCY

再配置が要求されたリソースへの依存関係のため、リソースの再配置が要求された場合に、この再配置が可能かどうかを宣言するために使用します。0の場合、再配置が要求されたリソースへの依存関係のため、このリソースの再配置は許可されません。有効値は1または0です。

使用例

```
RELOCATE_BY_DEPENDENCY=1
```

RESTART_ATTEMPTS

Oracle Clusterwareが、リソースの再配置を試行する前に、リソースの現行のサーバー上でリソースの再起動を試行する回数。1の値を指定した場合、Oracle Clusterwareはサーバー上でリソースの再起動を1回のみ試行します。2回失敗すると、Oracle Clusterwareの再配置が試行されます。0の値を指定した場合、再起動は試行されませんが、Oracle Clusterwareは常時、別のサーバーへのリソースのフェイルオーバーを試行します。

使用例

```
RESTART_ATTEMPTS=2
```

SCRIPT_TIMEOUT

アクションを実行する最大時間(秒単位)。アクション・スクリプトが指定された時間内に完了しなかった場合、Oracle Clusterwareはエラー・メッセージを戻します。このタイムアウトは、すべてのアクション(start、stop、checkおよびclean)に適用されます。

使用例

```
SCRIPT_TIMEOUT=45
```

SERVER_CATEGORY

ローカル・リソースの場合、local_resourceタイプの定義はカテゴリ認識に拡張されます。すなわち、ローカル・リソースを制限して特定のサーバー・カテゴリに属させることができます。クラスタ・リソースの場合、SERVER_CATEGORY属性の値は、[PLACEMENT](#)属性の値で常に機能します。[PLACEMENT](#)が制限され、SERVER_CATEGORYが使用されている場合、[SERVER_POOLS](#)を*に設定します。[PLACEMENT](#)をrestrictedに設定すると、次の属性の1つも設定されるとOracle Clusterwareでは予測されます。

- [HOSTING_MEMBERS](#)
- [SERVER_POOLS](#)
- [SERVER_CATEGORY](#)

たとえばresource1と呼ばれるリソースはrestrictedの対象の[PLACEMENT](#)の値を設定するポリシーを所有でき、SERVER_CATEGORYがHubCategoryに設定されます。このような場合、Oracle Clusterwareによって、HubCategoryに属するサーバー上でのみresource1が実行されます。

[PLACEMENT](#)がfavoredに設定され、かつ[HOSTING_MEMBERS](#)、[SERVER_POOLS](#)または[SERVER_CATEGORY](#)の1つのみが設定されている場合、この値がプリファレンスを示します。[HOSTING_MEMBERS](#)が移入され[SERVER_POOLS](#)または[SERVER_CATEGORY](#)の1つが設定されると、[HOSTING_MEMBERS](#)によって配置のプリファレンスが示され、[SERVER_POOLS](#)またはSERVER_CATEGORYによって制限が示されます。たとえばora.cluster.vipリソースに[PLACEMENT](#)の値をfavoredに設定するポリシーが設定可能で、[SERVER_CATEGORY](#)はHubCategoryに設定され、[HOSTING_MEMBERS](#)はserver_name1に設定されます。このような場合にOracle Clusterwareは、ora.cluster.vipの配置をHubCategoryにおけるサーバーに限定し、server_name1と呼ばれるサーバーを優先します。

使用例

```
SERVER_CATEGORY=my_category
```

SERVER_POOLS

SERVER_POOLSリソース属性の説明です。

特定のリソースが属することができる、サーバー・プールのスペース区切りリスト。リソースがクラスタ内の任意のサーバー上で動作できる場合は、デフォルト値の*を使用し、リソースがcluster_resourceタイプ以外の場合は、SERVER_POOLS属性のデフォルト値は空白です。この属性の値に*を指定できるのはクラスタ管理者のみです。

- 次に示すように、SERVER_POOLS属性とともに[PLACEMENT](#)属性を使用します。[PLACEMENT](#)属性の値をrestrictedまたはfavoredに設定した場合は、リソースのポリシー管理を使用する際にSERVER_POOLS属性にも値を指定する必要があります。
- [PLACEMENT](#)の値をbalancedに設定した場合、SERVER_POOLS=*でないかぎり、リソースは汎用プールおよび空きプールのみで実行されます。

この属性は、配置に関してリソースと1つ以上のサーバー・プールとの間でアフィニティを作成します。また、[PLACEMENT](#)属性の値に依存します。

使用例

```
SERVER_POOLS=pool1 pool2 pool3
```

関連項目:

- サーバー・プールの詳細は、「[ポリシー・ベースのクラスタおよび容量の管理](#)」を参照してください。
クラスタ管理の詳細は、「[ロール別管理](#)」を参照してください。
- この属性の詳細は、「[HOSTING_MEMBERS](#)」を参照してください。
- この属性の詳細は、「[PLACEMENT](#)」を参照してください。

START_CONCURRENCY

同時に実行可能なアクションの最大数を説明しています。0を指定すると、上限は設定されません。

使用例

```
START_CONCURRENCY=10
```

START_DEPENDENCIES

リソースを起動する際にOracle Clusterwareで考慮する一連の関係を指定します。特定のリソースが依存する可能性がある、いくつかのリソースおよびリソース・タイプに対して、依存性についてスペース区切りのリストを指定できます。

構文

```
START_DEPENDENCIES=dependency(resource_set) [dependency(resource_set)] [...]
```

前述の構文例で、変数は次のように定義されます。

- **dependency**: 指定可能な値は、**attraction**、**dispersion**、**exclusion**、**hard**、**pullup**および**weak**です。これらの依存性は、それぞれ1回のみ指定できますが、**pullup**は、複数回指定できます。
- **resource_set**: 構成するリソースが依存しているリソース・エンティティ(個々のリソースまたはリソース・タイプのいずれか)をカッコ()で囲み、**res1[, res2[, ...]]**の形式で指定したカンマ区切りリスト。

各リソース・エンティティは次のように定義されます。

```
[modifier1:[modifier2:]] {resource_name | type:resource_type}
```

前述の構文例で、**resource_name**は特定のリソースの名前、**type:resource_type**は特定のリソース・タイプの名前です。リソース・タイプの前に**type**に指定する必要があり、この**type**修飾子はリストで最後のリソース・エンティティである必要があります。

必要に応じて、修飾子を指定してリソース・エンティティの依存性をさらに詳細に構成できます。リソース・エンティティに接頭辞として次の修飾子を指定することで、各依存性を変更することができます。

- **attraction([intermediate:]{resource_name | type:resource_type})**: 指定した特定のリソースまたは特定のタイプの任意のリソースと同じサーバーでこのリソースを実行する場合、**attraction**起動依存性を使用します。

intermediateを使用すると、**INTERMEDIATE**状態になっている、このリソースが依存するリソース・エンティティにこのリソースが誘引されることを指定できます。指定しない場合、依存リソースを誘引するには、リソースの状態が**ONLINE**である必要があります。

リソースのリソース・タイプに**attraction**依存性を指定した場合、依存リソースは、指定したタイプの任意のリソースに誘引されます。

- **exclusion([[preempt_pre: | preempt_post:]] target_resource_name |**

`type:target_resource_type]`): `exclusion`起動依存性を使用して、この依存性のあるリソースが同じノードで動作することを回避します。

ソース・リソースを起動する前に、`preempt_pre`修飾子を使用して、指定されたターゲット・リソース、または特定のリソース・タイプによって定義されるリソースを停止するための`exclusion`依存性を構成します。

ソース・リソースの起動後に`preempt_post`修飾子を使用して、指定されたターゲット・リソース、または特定のリソース・タイプによって定義されるリソースを停止し、可能であれば再配置するための`exclusion`依存性を構成します。

- `dispersion[:active]([intermediate:][pool:]){resource_name | type:resource_type}`): 指定したリソースまたは特定のタイプのリソースとは異なるサーバーで実行するリソースに`dispersion`起動依存性を指定します。リソースは、サーバーの可用性に応じて、同じサーバーで実行されることがあります。

`active`修飾子を使用して`dispersion`依存性を構成すると、依存リソースが別のリソースと同じ場所に配置されていて、別のサーバーがオンラインになった場合、Oracle Clusterwareは別サーバーへの依存リソースの再配置を試行します。Oracle Clusterwareは、`active`修飾子が指定されないかぎり、新しく利用可能になったサーバーへのリソースの再配置を行いません。

`intermediate`修飾子を使用すると、リソースの状態が`ONLINE`または`INTERMEDIATE`のいずれかである場合にOracle Clusterwareで依存リソースを再配置できるように指定できます。指定しない場合、依存リソースを分散するには、リソースの状態が`ONLINE`である必要があります。

リソースを別のサーバーではなく、ターゲットとは別のサーバー・プールに配置する場合は、`pool`修飾子を使用します。

- `hard([intermediate:][global:][uniform:]){resource_name | type:resource_type}`): 特定のリソースまたは特定のタイプのリソースが起動するときのみあるリソースを起動させる場合、そのリソースに`hard`起動依存性を指定します。

`intermediate`修飾子を使用すると、依存先リソースの状態が`ONLINE`または`INTERMEDIATE`のいずれかの場合にOracle Clusterwareでこのリソースを起動できるように指定できます。指定しない場合、Oracle Clusterwareでこのリソースを起動するには、リソースの状態が`ONLINE`である必要があります。

`global`修飾子を使用すると、Oracle Clusterwareでこのリソースを起動する条件としてリソースが同じサーバー上に存在する必要があるように指定できます。指定しない場合、Oracle Clusterwareでこのリソースを起動するには、同じサーバー上にリソースが存在する必要があります。

`uniform`修飾子を使用して、リソースBのすべてのインスタンスの起動を試行しますが、最低1つのインスタンスが起動されて依存性を満たす必要があります。

リソースのリソース・タイプに`hard`依存性を指定した場合は、指定したタイプの任意のリソースが実行されていると、このリソースを起動できます。

ノート:



`hard` 起動依存性を持つリソースには `pullup` 起動依存性も定義することをお勧めします。

- `pullup[:always]([intermediate:][global:]){resource_name |`

`type:resource_type})): リソースにpullup起動依存性を指定すると、このリソースは、指定したリソースの起動により起動します。`

pullupにalways修飾子を使用すると、TARGET属性の値がONLINEまたはOFFLINEのいずれでも、その値に関係なくOracle Clusterwareはこのリソースを起動します。そうでない場合は、always修飾子を指定しないと、このリソース用のTARGET属性の値がONLINEのときにのみ、Oracle Clusterwareはこのリソースを起動します。

intermediate修飾子を使用すると、依存先リソースの状態がONLINEまたはINTERMEDIATEのいずれかの場合にOracle Clusterwareでこのリソースを起動できるように指定できます。指定しない場合、Oracle Clusterwareでこのリソースを起動するには、リソースの状態がONLINEである必要があります。

global修飾子を使用すると、Oracle Clusterwareでこのリソースを起動する条件として、このリソースの依存先リソースが同じサーバー上に存在する必要があるように指定できます。指定しない場合、Oracle Clusterwareでこのリソースを起動するには、このリソースの依存先リソースが同じサーバー上に存在する必要があります。

リソースのリソース・タイプにpullup依存性を指定した場合、指定したタイプの任意のリソースが起動すると、Oracle Clusterwareはこのリソースを起動できます。

ノート:



hard 起動依存性を持つリソースには pullup 起動依存性も定義することをお勧めします。

- `weak([concurrent:][global:][uniform:]{resource_name | type:resource_type}):` 指定したリソースが実行されているかどうかにかかわらず、このリソースを起動する場合、リソースにweak起動依存性を指定します。このリソースの起動を試行すると、このリソースが依存する任意のリソースが実行されていない場合は、それらのリソースの起動も試行されます。

concurrent修飾子を使用すると、依存先リソースの起動時にOracle Clusterwareで依存リソースを起動できるように指定できます。concurrentを指定しない場合、Oracle Clusterwareで依存リソースを起動するには、リソースの起動が完了している必要があります。

global修飾子を使用すると、Oracle Clusterwareで依存リソースを起動する条件としてリソースが同じサーバー上に存在する必要があることを指定できます。

uniform修飾子を使用すると、リソースを実行可能なすべての場所でそのリソースのすべてのインスタスを起動できます。修飾子を指定しない場合(デフォルトの場合)、リソースは、依存先リソースと同じサーバー上で起動します。

リソースのリソース・タイプにweak起動依存性を指定した場合は、指定したタイプの任意のリソースが実行されていると、このリソースを起動できます。

関連項目:

起動依存性の詳細は、「[起動依存性](#)」を参照してください。

START_TIMEOUT

startアクションを実行できる最大時間(秒単位)。アクションが指定された時間内に完了しなかった場合、Oracle Clusterwareはエラー・メッセージを戻します。この属性に値が指定されていないか、0秒が指定されている場合、Oracle Clusterwareは[SCRIPT_TIMEOUT](#)属性の値を使用します。

使用例

```
START_TIMEOUT=30
```

関連項目:

この属性の詳細は、「[SCRIPT_TIMEOUT](#)」を参照してください。

STOP_CONCURRENCY

同時に停止可能なアクションの最大数を説明しています。0を指定すると、上限は設定されません。

使用例

```
STOP_CONCURRENCY=10
```

STOP_DEPENDENCIES

リソースを停止する際にOracle Clusterwareで考慮する一連の関係を指定します。

構文

```
STOP_DEPENDENCIES=dependency(resource_set) [dependency(resource_set)] ...
```

前述の構文例で、変数は次のように定義されます。

- dependency: 指定可能な値はhardのみです。
- resource_set: 構成するリソースが依存しているリソース・エンティティ(個々のリソースまたはリソース・タイプのいずれか)をres1[, res2[, ...]]の形式で指定したカンマ区切りリスト。

各リソース・エンティティは次のように定義されます。

```
[modifier1:[modifier2:][modifier3:]] resource_name | type:resource_type
```

前述の構文例で、resource_nameは特定のリソースの名前、type:resource_typeは特定のリソース・タイプの名前です。リソース・タイプの前にはtype:と記述する必要があります。

必要に応じて、修飾子を指定してリソース・エンティティの依存性をさらに詳細に構成できます。リソース・エンティティに接頭辞として次の修飾子を指定することで、各依存性を変更することができます。

hard([intermediate:][global:][shutdown:]){resource_name | type:resource_type}: 指定したリソースまたは特定のリソース・タイプのリソースが停止した場合に停止させるリソースにhard停止依存性を指定します。

intermediateを使用すると、リソースの状態がONLINEまたはINTERMEDIATEのいずれかである場合に依存リソースの状態をONLINEのままにできるように指定できます。指定しない場合、リソースの状態がONLINEでないかぎり、Oracle Clusterwareは依存リソースを停止します。

globalを使用すると、クラスタ内の任意のノード上にあるリソースの状態がONLINEの場合に、依存リソースの状態

がONLINEのままになるように指定できます。指定しない場合、同じサーバー上にあるリソースがオフラインになると、Oracle Clusterwareは依存リソースを停止します。

shutdownを使用すると、Oracle Clusterwareスタックが停止する場合にこの依存性を適用します。これは、個々のリソースの計画済または計画外イベントには影響を及ぼすことなく、スタックを停止するときにリソースの停止順序に影響を与える便利な方法です。この依存性は、shutdown修飾子とともに使用されると、リソースが直接停止されてもスタックが停止している場合にのみ有効になりません。

関連項目:

停止依存性の詳細は、「[停止依存性](#)」を参照してください。

STOP_TIMEOUT

stopまたはcleanアクションを実行できる最大時間(秒単位)。アクションが指定された時間内に完了しなかった場合、Oracle Clusterwareはエラー・メッセージを戻します。この属性が指定されていない場合または0秒が指定されている場合、Oracle Clusterwareは[SCRIPT_TIMEOUT](#)属性の値を使用します。

使用例

```
STOP_TIMEOUT=30
```

関連項目:

- この属性の詳細は、「[SCRIPT_TIMEOUT](#)」を参照してください。
- この属性の詳細は、「[START_TIMEOUT](#)」を参照してください。

UPTIME_THRESHOLD

UPTIME_THRESHOLDの値は、Oracle Clusterwareがリソースが安定しているとみなすまでに、リソースが稼働中である必要がある時間の長さを表します。UPTIME_THRESHOLD属性の値を設定すると、リソースの安定性を示すことができます。

この属性の値は、数値の後に秒(s)、分(m)、時間(h)、日(d)または週(w)を示す文字を付けて入力します。たとえば、7hという値は、稼働時間のしきい値が7時間であることを示します。

Oracle Clusterwareは、UPTIME_THRESHOLDに指定した時間が経過すると、停止、起動、再配置または障害のようなリソース状態が変更されるイベントが次に発生したときに、RESTART_COUNTの値を0にリセットします。RESTART_COUNTの値がRESTART_ATTEMPTSに設定した値に達すると、Oracle Clusterwareはアラートを生成します。カウンタは、次回リソースの障害発生時や再起動時に、効果的にリセットされます。しきい値は、再起動がカウントされて破棄される時間の長さを示しています。しきい値に達してリソースが失敗すると、再起動されます。

ノート:



RESTART_COUNT の値が RESTART_ATTEMPTS に設定した値に達すると、Oracle Clusterware はクラスタウェア・アラート・ログ・ファイルにアラートを記録します。

関連項目:

- この属性の詳細は、「[RESTART_ATTEMPTS](#)」を参照してください。
- この属性の詳細は、「[RESTART_COUNT](#)」を参照してください。

USER_WORKLOAD

What-If分析に関して、リソースがワークロードの生成リソースであるかどうかを示すために使用します。使用可能な値はyesまたはnoです。

使用例

```
USER_WORKLOAD=yes
```

USE_STICKINESS

可能な場合はリソースが最後に実行された場所で実行され、そうでない場合に適用されるロード・バランシングを許可しないことを示すために使用します。1に設定すると、Oracle Clusterwareは、最後に動作したリソースの開始を試行します。

USE_STICKINESSの無効化によっても、ロード・バランシングを無効化できます。デフォルト値は0です。使用可能な値は、0および1です。

使用例

```
USE_STICKINESS=1
```

読取り専用のリソース属性

次の属性は、特定のリソースに対してcrsctl status resourceコマンドを実行すると参照できます。Oracle Clusterwareは、ユーザーがリソースを登録した際にこれらの属性を設定します。

- [ACTION_FAILURE_EVENT_TEMPLATE](#)
- [INSTANCE_COUNT](#)
- [INTERNAL_STATE](#)
- [LAST_SERVER](#)
- [LAST_STATE_CHANGE](#)
- [PROFILE_CHANGE_EVENT_TEMPLATE](#)
- [RESTART_COUNT](#)
- [STATE](#)
- [STATE_CHANGE_EVENT_TEMPLATE](#)
- [STATE_DETAILS](#)
- [TARGET](#)
- [TARGET_SERVER](#)
- [TYPE](#)

ACTION_FAILURE_EVENT_TEMPLATE

これは、ora.*リソース用に内部的に管理される属性の1つです。この属性を編集することはできません。

INSTANCE_COUNT

INSTANCE_COUNT属性は内部的に管理されている属性で、これにはリソースが現在所有しているインスタンス数が含まれています。

INTERNAL_STATE

現在ポリシー・エンジンがリソースで実行しているアクション(ある場合)について説明する、内部的に管理される読み取り専用の属性です。

指定可能な値とその説明は、次のとおりです。

- STARTING: ポリシー・エンジンは、現在リソースを起動しています。
- STOPPING: ポリシー・エンジンは、現在リソースを停止しています。
- CLEANING: ポリシー・エンジンは、現在リソースをクリーンアップしています。
- STABLE: ポリシー・エンジンは、現在リソースでアクションを実行していません

ただし、その他のいくつかのコマンドの一環として、そのリソースがまだロックされている可能性があります。

LAST_SERVER

cluster_resourceタイプのリソースの場合、内部的に管理される読み取り専用の属性の1つで、ソースのstartアクションが最後に成功したサーバーの名前が設定されています。

local_resourceタイプのリソースの場合、リソース・インスタンスが固定されるサーバーの名前です。

LAST_STATE_CHANGE

ポリシー・エンジンがリソースの現在の状態を登録した時点について説明する、内部的に管理される読み取り専用の属性です。これは、CRSDの再起動時に発生する、リソースの状態が変更された時点、またはポリシー・エンジンがその状態を検出した時点のタイムスタンプである場合があります。

PROFILE_CHANGE_EVENT_TEMPLATE

これは、ora.*リソース用に内部的に管理される属性の1つです。この属性を編集することはできません。

RESOURCE_LIST

リソース・グループに属しているリソースのリスト。リソースがグループに追加されると、Oracle Clusterwareはこの属性に値を移入して更新します。

この属性をリソース・グループとともに使用します。

RESTART_COUNT

Oracle Clusterwareデーモンが使用する内部的に管理される属性で、リソース再起動の試行回数を0(ゼロ)からRESTART_ATTEMPTS属性に指定した値までカウントします。この属性を編集することはできません。

STATE

Oracle Clusterwareからレポートされるとリソースの現在の状態が反映される、内部的に管理される属性です。リソースの状態は次のいずれかです。

- ONLINE: リソースはオンラインで、リソースの監視が有効です([「CHECK_INTERVAL」](#)を参照)。

- OFFLINE: リソースはオフラインで、オフラインのリソースの監視が構成されている場合は、そのみが有効です([「OFFLINE_CHECK_INTERVAL」](#)を参照)。
- INTERMEDIATE: リソースは部分的にオンラインであるか、前にオンラインであることが認識されているものの、それ以降の状態確認の試行が失敗しています。リソースの監視は有効です([「CHECK_INTERVAL」](#)を参照)。
- UNKNOWN: リソースは管理不可能で、現在の状態は不明であるため、手動操作を行ってその操作を再開する必要があります。この状態のリソースは監視されません。

STATE_CHANGE_EVENT_TEMPLATE

これは、ora.*リソース用に内部的に管理される属性の1つです。この属性を編集することはできません。

STATE_DETAILS

内部的に管理される読取り専用の属性の1つです。リソースの状態の詳細が設定されています。

リソースの4つの状態(ONLINE、OFFLINE、UNKNOWNおよびINTERMEDIATE)は、マウント、アンマウント、オープンなどの異なるリソース固有の値にマッピングされることがあります。リソース・エージェントの開発者は、STATE_DETAILS属性を使用して、リソースとリソースの状態とのマッピングに関する詳細な説明を提供できます。

詳細の提供は任意です。詳細が提供されない場合、Oracle Clusterwareは、予想される4つのリソースの状態のみを使用します。また、エージェントがこれらの詳細を提供できない場合(リソースの状態の値に対しても発生する可能性がある)、Oracle Clusterwareはこの属性の値を設定して、リソースが現在の状態になった理由を示す最小限の詳細情報を提供します。

TARGET

内部用の読取り専用属性の1つで、リソースの望ましい状態について説明します。ただし、`crsctl start resource_name`または`crsctl stop resource_name`コマンドを使用すると、この属性の値に影響が及ぶ場合があります。

TARGET_SERVER

これは、リソースが起動しているサーバー名を含んでいる、内部管理属性です。この値は、リソースの起動時に関連します。

TYPE

リソースの作成時に示されるリソースのタイプ。この属性は、リソースの作成時に指定する必要があり、リソースの作成後には変更できません。

関連項目:

リソース・タイプの詳細は、「[Oracleクラスタウェアのリソース・タイプ](#)」を参照してください。

非推奨のリソース属性

次のリソース属性は、Oracle Clusterware 12cでは非推奨です。

DEGREE

単一サーバー上で実行可能なクラスタ・リソースのインスタンスの数。

サード・パーティ製アプリケーションのアクション・スクリプトの例

この項では、スクリプト・エージェントを使用するサード・パーティ製アプリケーションの例を示します。

[例B-2](#)に、Apache Webサーバーをフェイルオーバーするアクション・スクリプトを示します。

例B-2 Apacheアクション・スクリプト

```
#!/bin/sh
HTTPDCONFLOCATION=/etc/httpd/conf/httpd.conf
WEBPAGECHECK=http://<MyVIP>:80/icons/apache_pb.gif
case $1 in
'start')
    /usr/sbin/apachectl -k start -f $HTTPDCONFLOCATION
    RET=$?
    ;;
sleep(10)
    ;;
'stop')
    /usr/sbin/apachectl -k stop
    RET=$?
    ;;
'clean')
    /usr/sbin/apachectl -k stop
    RET=$?
    ;;
'check')
    /usr/bin/wget -q --delete-after $WEBPAGECHECK
    RET=$?
    ;;
*)
    RET=0
    ;;
esac
# 0: success; 1 : error
if [ $RET -eq 0 ]; then
exit 0
else
exit 1
fi
```

[例B-3](#)に、xclockスクリプトを示します。このスクリプトは、すべてのLinuxおよびUNIXプラットフォームでデフォルト・バイナリとして利用できるxclockを使用した簡単なアクション・スクリプトです。

例B-3 xclockアクション・スクリプト

```
#!/bin/bash
# start/stop/check script for xclock example
# To test this change BIN_DIR to the directory where xclock is based
# and set the DISPLAY variable to a server within your network.
BIN_DIR=/usr/X11R6/bin
LOG_DIR=/tmp
BIN_NAME=xclock
DISPLAY=yourhost.domain.com:0.0
export DISPLAY
exit_code=0
if [ ! -d $BIN_DIR ]
then
    echo "start failed"
    exit 2
fi
PID1=`ps -ef | grep $BIN_NAME | grep -v grep | grep -v xclock_app | awk '{ print $2 }'`
case $1 in
'start')
```

```

        if [ "$PID1" != "" ]
        then
            status_p1="running"
        else
            if [ -x $BIN_DIR/$BIN_NAME ]
            then
                umask 002
                ${BIN_DIR}/${BIN_NAME} & 2>${LOG_DIR}/${BIN_NAME}.log
                status_p1="started"
            else
                echo `basename $0`: $BIN_NAME: Executable not found"
                exit_code=1
            fi
        fi
        echo "$BIN_NAME: $status_p1"
        exit $exit_code
    ;;
'stop')
    if [ "${PID1}" != "" ]
    then
        kill -9 ${PID1} && echo "$BIN_NAME daemon killed"
    else
        echo "$BIN_NAME: no running Process!"
    fi
    exit $exit_code
    ;;
'check')
    if [ "$PID1" != "" ]
    then
        echo "running"
        exit 0
    else
        echo "not running"
        exit 1
    fi
    ;;*)
    echo "Usage: "`basename $0`" {start|stop|check}"
    ;;
esac

```

[例B-4](#)に、ファイルを監視するエージェントのシェル・スクリプトの例を示します。エージェントは、起動するとファイル(属性を使用して指定されるファイル)を作成し、停止するとファイルを削除します。CHECKアクションは、ファイルが存在するかどうかを単にチェックするものです。接頭辞_CRS_を持つ変数は、環境のスクリプトに指定する属性値です。

例B-4 アクション・スクリプトの例

```

#!/bin/sh
TOUCH=/bin/touch
RM=/bin/rm
PATH_NAME=/tmp/$_CRS_NAME
#
# These messages go into the CRSD agent log file.
echo " ***** `date` ***** "
echo "Action script '$_CRS_ACTION_SCRIPT' for resource[$_CRS_NAME] called for action $1"
#
case "$1" in
'start')
    echo "START entry point has been called.."
    echo "Creating the file: $PATH_NAME"
    $TOUCH $PATH_NAME
    exit 0
    ;;
'stop')
    echo "STOP entry point has been called.."

```

```
    echo "Deleting the file: $PATH_NAME"
    $RM $PATH_NAME
    exit 0
;;
'check')
    echo "CHECK entry point has been called.."
    if [ -e $PATH_NAME ]; then
        echo "Check -- SUCCESS"
        exit 0
    else
        echo "Check -- FAILED"
        exit 1
    fi
;;
'clean')
    echo "CLEAN entry point has been called.."
    echo "Deleting the file: $PATH_NAME"
    $RM -f $PATH_NAME
    exit 0
;;
esac
```

C OLSNODESのコマンド・リファレンス

この付録では、olsnodesコマンドの構文およびコマンド・オプションについて説明します。

この付録の内容は次のとおりです。

- [OLSNODESの使用](#)
 - [概要](#)
 - [操作上のノート](#)
- [OLSNODESコマンドのサマリー](#)

OLSNODESの使用

この項では、OLSNODESコマンドの使用に関連する項目について説明します。

- [概要](#)
- [操作上のノート](#)

概要

olsnodesコマンドは、クラスタに参加しているすべてのノードのリストおよびその他の情報を表示します。

このコマンドを使用して、クラスタが操作可能な状態であること、およびすべてのノードがクラスタのメンバーとして登録されていることをチェックできます。このコマンドは、ノード番号を取得する方法も提供します。

操作上のノート

このコマンドは、`-n all`オプションが使用されている場合にノード名のリストを取得するためにクラスタ検証ユーティリティ (CLUVFY)で使用されます。

このコマンドは、`$ORA_CRS_HOME/bin`ディレクトリに格納されています。このコマンドは、CRSデーモンが起動されている場合にのみ使用できます。

権限およびセキュリティ

このコマンドは、rootユーザー、Oracle ClusterwareをインストールしたユーザーまたはOracle Databaseをインストールしたユーザーとして実行できます。

OLSNODESコマンドのサマリー

olsnodesコマンドでは、キーワードは使用しませんが、1つ以上のオプションを指定できます。使用可能なオプションは、[表C-1](#)を参照してください。

構文

```
olsnodes [[-n] [-s] [-t] [node_name | -l [-p]] | [-c]] [-a] [-g] [-v]
```

コマンド・パラメータを指定せずに、olsnodesコマンドを発行すると、クラスタのノードのリストが戻されます。

```
[root@node1]# olsnodes
```

```
node1
node2
node3
node4
```

表C-1 OLSNODESのコマンド・オプション

オプション	説明
-n	クラスタに参加しているすべてのノードを表示し、割り当てられているノード番号も含まれます。
-s	ノードのステータス(active または inactive)を表示します。
-t	ノード・タイプ(pinned または unpinned)を表示します。
node_name	特定のノードの情報を表示します。
-l [-p]	ローカル・ノードを表示し、ローカル・ノードへのプライベート・インターコネクトを含めます。-p オプションは、-l オプションとともに指定したときのみ有効です。
-c	クラスタの名前を表示します。
-a	クラスタ内のアクティブなノードのみを重複なしで表示します。
-g	クラスタ検証の詳細情報を記録します。
-v	クラスタ検証の情報を冗長モードで記録します。My Oracle Support からの指示があった場合にのみ、デバッグ・モードで使します。

例

例1: 現在、クラスタ内にあるすべてのノードのリスト

現在、クラスタのメンバーであるノードをリストするには、次のコマンドを使用します。

```
[root@node1]# olsnodes
node1
node2
node3
node4
```

例2: クラスタ・メンバーのノード名およびノード番号の表示

クラスタ内の各ノードのノード名およびノード番号を表示するには、次のコマンドを使用します。

```
[root@node1]# olsnodes -n
node1      1
node2      2
node3      3
node4      4
```

例3: クラスタ・メンバーのノード・ロールの表示

クラスタ内の各ノードのノード・ロールを表示するには、次のコマンドを使用します。

```
[root@node1]# olsnodes -a
node1      Hub
node2      Hub
node3
node4
```

D Oracle Interface Configurationツール(OIFCFG)のコマンド・リファレンス

Oracle Interface Configurationツール(OIFCFG)のコマンドライン・インタフェースにより、ネットワーク・インタフェースを定義および管理できます。Oracle Clusterware環境では、次の作業に対してOIFCFGコマンドを使用できます。

- ネットワーク・インタフェースのコンポーネントへの割当ておよび割当て解除
- 特定のネットワーク・インタフェースを使用するようにコンポーネントを設定
- コンポーネント構成情報の取得

この付録の内容は次のとおりです。

- [OIFCFGコマンドライン・インタフェースの起動](#)
- [OIFCFGの使用方法的概要](#)

OIFCFGコマンドライン・インタフェースの起動

OIFCFGを起動する前に、コマンド構文に `-global` オプションを含める場合は、少なくともローカル・ノード(できればすべてのノード)でOracle Clusterwareを起動してください。

ノート:



グローバル・ネットワーク・インタフェースを変更するには、Oracle Clusterware がすべてのクラスタ・ノードで実行されている必要があります。

Oracle Clusterwareソフトウェアをインストールしたユーザーとして、`Grid_home/bin/`ディレクトリからOIFCFGを実行します。たとえば:

```
$ ./oifcfg
```

OIFCFGのオンライン・ヘルプを表示するには、`oifcfg -help`コマンドを実行します。

```
$ ./oifcfg -help
```

OIFCFGの使用方法的概要

この項では、次の項目について説明します。

- [OIFCFGコマンドの書式](#)
- [OIFCFGコマンド](#)
- [OIFCFGコマンドのパラメータ](#)
- [OIFCFGの使用上のノート](#)
- [OIFCFGの例](#)

OIFCFGコマンドの書式

```
oifcfg iflist [-p] [-n]
oifcfg setif {-node nodename | -global} {if_name/subnet:if_type[,if_type]}[,...]
oifcfg getif [-node nodename | -global] [-if if_name[/subnet] [-type if_type]]
oifcfg delif {-node nodename | -global} [if_name[/subnet]] [-force] | -force}
oifcfg [-help]
```

OIFCFGコマンド

これらのコマンドを使用して、ネットワーク・インタフェースを定義および管理します。

[表D-1](#)に示されたOIFCFGコマンドのいずれかを入力できます。

表D-1 OIFCFGコマンド

コマンド	説明
oifcfg iflist [-p [-n]]	setif で構成できる使用可能なインタフェースを示します。iflist コマンドは、オペレーティング・システムに問い合わせ、このノードに存在するネットワーク・インタフェースを検出します。このコマンドを使用して、次の 2 つのオプションを指定できます。 ● -p: 経験則に基づいて想定されるインタフェース・タイプ(プライベート、パブリックまたは不明)が表示されます。 ● -n: ネットマスクが表示されます。 インタフェースに IPv6 ネットワークがある場合、OIFCFG は、次のように別個の行に IPv6 サブネットを出力します。 <pre>eth1 fec0::80 PUBLIC 64 eth1 10.229.184.0 PUBLIC 255.255.248.0</pre>
oifcfg setif	インタフェースのインタフェース・タイプ(パブリック、クラスタ・インターコネクトまたは Oracle ASM)を設定します。
oifcfg getif	setif コマンドによってインタフェース・タイプが定義されているインタフェースを、インタフェースのタイプとともに示します。 ノート: このコマンドで利用できる -node コマンド・パラメータは、oifcfg setif -node コマンドを使用してインタフェースの詳細を設定していない場合は結果を戻しません。
oifcfg delif	グローバル・インタフェースまたはノード固有のインタフェースとして格納されたネットワーク構成を削除します。クラスタのすべてのノードから格納されたネットワーク構成を削除するには、-force オプションを使用し、-node または -global オプションは指定しません。

OIFCFGコマンドのパラメータ

この項では、OIFCFGコマンドのパラメータを示します。実行するコマンドによっては、一部のパラメータがオプションとなることに注意してください。

- `-node node_name`: `olsnodes`コマンドの出力結果にリストされるOracle Clusterwareノードの名前。
`olsnodes`コマンドの詳細は、[\[OLSNODESのコマンド・リファレンス\]](#)を参照してください。
- `-global`: ネットワーク・インタフェースは、グローバル・インタフェース(`iflist`コマンドによってレポートされる)としても、ノード固有のインタフェースとしても格納できます。
 - Oracle Real Application Clusters(Oracle RAC)クラスタのすべてのノードが同じサブネットに接続された同じインタフェースを使用している場合、インタフェースはグローバル・インタフェースとして格納されます。グローバル・インタフェース(各パブリック・サブネットのすべてのノード、および各プライベート・サブネットのすべてのノードが、それぞれ同一ネットワーク・インタフェースを持つ構成)は、推奨構成であるだけでなく、デフォルトのインストール構成です。
 - インタフェースは、ノード固有の(ローカル)インタフェースとして格納できます。

ノート:



ノード間の異なるインタフェース名に対して、インタフェース名のワイルドカードを使用できます。異なるノード上の異なる具体的なインタフェース名を構成する場合には、`-node` は使用できません。

- `-if if_name`: システムでインタフェースが構成されときの名前。
- `subnet`: インタフェースのサブネット・アドレス。
- `-type if_type`: 1つ以上のカンマ区切りのインタフェースのタイプ: `public`、`cluster_interconnect`または`asm`。
- `-help`: OIFCFGコマンドのオンライン・ヘルプを表示します。

関連トピック

- [\[OLSNODESのコマンド・リファレンス\]](#)

OIFCFGの使用上のノート

- ネットワーク・インタフェースの指定には、次の形式を使用します。

```
if_name/subnet:if_type
```

この指定では、次の項目を使用して、ネットワーク・インタフェースを一意に識別します。

- インタフェース名

`oifcfg setif`コマンドを使用する場合、インタフェース名には、任意の文字列に一致するアスタリスク(*)などのワイルドカード文字を含めることができます。ただし、ワイルドカードを含むインタフェース名は二重引用符("")で囲む必要があります。

ノート:

Oracle Database 11g リリース 2 (11.2.0.2) データベースより前のクラスタではワイルドカードを使用しないことをお勧めします。Oracle では、ローカル・ノード上でワイルドカードを拡張することでインタフェース名を解決し、`oifcfg getif` コマンドを実行するたびに PRIF-0029 警告も発行するためです。

11g リリース 2 (11.2.0.2) より前の Oracle Databases でワイルドカードを使用する場合、これらのデータベースではかわりに `CLUSTER_INTERCONNECTS` パラメータを使用する必要があります。

- 関連付けられたサブネット
- インタフェース・タイプ

インタフェース・タイプは、ネットワークを構成する目的を示します。サポートされるインタフェース・タイプは次のとおりです。

- `public`: Oracle Net や仮想インターネット・プロトコル (VIP) ・アドレスなど、Oracle RAC インスタンスの外部にあるコンポーネントと通信するために使用できるインタフェース
- `cluster_interconnect`: インスタンス間の通信やキャッシュ・フュージョン^{脚注1}通信を提供するため、クラスタ・インターコネクトに使用されるプライベート・インタフェース
- `asm`: Oracle ASM との通信に使用できるインタフェース。

`cluster_interconnect` のインタフェース・タイプを設定すると、それは起動時にインスタンスに適用されます。変更内容はインスタンスを再起動するまで反映されません。

次の指定では、アドレス 204.152.65.0 にあるクラスタ・インターコネクトとして `qfe0` を識別します。

```
qfe0/204.152.65.0:cluster_interconnect
```

- Oracle Universal Installer は、OIFCFG を使用して、使用可能なインタフェースの識別および表示を行います。
- インタフェース名の変更による影響は、どの名前を変更するか、また IP アドレスも変更するかどうかによって異なります。インタフェース名のみを変更する場合、影響はわずかです。Oracle Cluster Registry (OCR) 内に格納されたパブリック・インタフェースの名前を変更する場合、各ノードの `nodeapps` を変更する必要があります。したがって、この変更を有効にするために、`nodeapps` を停止する必要があります。
- グローバル変更を実行する場合は、クラスタのすべてのメンバー上で Oracle Clusterware を再起動する必要があります。ローカル変更の場合は、ノードの再起動のみ実行する必要があります。データベースのインターコネクトの変更は、インスタンスの起動時に行われます。ただし、Oracle Clusterware のインターコネクトは、異なる場合があります。
- `interconnect` はインスタンスの起動時に選択されるため、OIFCFG コマンドの発行のみで、実行中のシステムにすぐに影響することはありません。そのかわり、このコマンドにより影響される可能性のあるコンポーネントの再起動後に、変更内容は有効になります。
- OIFCFG を使用してクラスタ全体のネットワーク分類を変更するには、そのグリッド・プラグ・アンド・プレイ・プロファイルが更新可能になるように、既知のすべてクラスタ・ノードが起動されている必要があります。

OIFCFGの例

次の例では、OIFCFGコマンドの一般的な使用方法をいくつか示します。

ネットワーク・インタフェースの名前のリスト

OIFCFGを使用すると、次の例に示すようにiflistキーワードを実行することにより、ローカル・ノード上で使用可能なすべてのインタフェースのインタフェース名とサブネットを表示できます。

```
oifcfg iflist
eth0      172.19.141.0
eth1      172.21.65.0
```

ネットワーク情報の取得

また、getifコマンドを使用すると、特定のOIFCFG情報を取得できます。

たとえば、Oracle Clusterwareのインストール後に、次のコマンドを入力することで、パブリックおよびクラスタ・インターコネクが適切な値に設定されているかどうかを確認できます。

```
$ oifcfg getif
```

このコマンドは、次のようなglobal publicおよびcluster_interconnectの値を戻します。

```
eth0 172.19.141.0 global public
eth1 172.21.65.0 global cluster_interconnect
```

グローバル・インタフェースの格納

インタフェースを格納するには、setifキーワードを使用します。たとえば、サブネットが172.19.141.0のインタフェースeth0をグローバル・インタフェース(クラスタおよびOracle Clusterware内のすべてのOracle RACインスタンスに対するインターコネクとして使用するインタフェース)として格納するには、次のコマンドを使用します。

```
oifcfg setif -global eth0/172.19.141.0:cluster_interconnect
```

ノート:



実行されていないノードでグリッド・プラグ・アンド・プレイ・プロファイルを更新できないため、setif コマンドを実行するときはすべてのノードが実行されていることを確認します。

格納されたインタフェースの削除

グローバルまたはノード固有のインタフェースとして格納された構成を削除するには、oifcfg delifコマンドを使用します。特定のノード固有のインタフェースまたはグローバル・インタフェースは、コマンドラインにインタフェース名を指定することで(オプションでサブネットを指定して)削除できます。

ノート:



コマンドラインで oifcfg delif のみを入力し、他の引数を指定しない場合、OIFCFG はクラスタ内のすべてのノードですべてのインタフェースを削除します。

たとえば、次のコマンドでは、サブネット172.21.65.0のグローバル・インタフェースeth1が削除されます。

```
oifcfg delif -global eth1/172.21.65.0
```

次の例では、コマンドによって、OIFCFGで割り当てられたすべてのグローバル・インタフェースが削除されます。

```
oifcfg delif -global
```

脚注の凡例

脚注1:

キャッシュ・フュージョンは、ブロックを保持しているインスタンスのメモリー・キャッシュから要求側インスタンスのメモリー・キャッシュにブロックを直接コピーするディスク不要のキャッシュ一貫性メカニズムです。

E Oracle Clusterware制御(CRSCTL)ユーティリティ・リファレンス

この付録には、Oracle Clusterware制御(CRSCTL)ユーティリティの参照情報が含まれています。

ノート:



My Oracle Support で指示された場合を除き、ora で始まる名前を持つ Oracle エンティティ(リソース、リソース・タイプ、サーバー・プールなど)で CRSCTL コマンドを使用しないでください。サーバー制御ユーティリティ (SRVCTL)は、Oracle エンティティに対して使用する適切なユーティリティです。

この付録の内容は次のとおりです。

- [CRSCTLの概要](#)
- [デュアル環境のCRSCTLコマンド](#)
- [Oracle RAC環境のCRSCTLコマンド](#)
- [Oracle Restart環境のCRSCTLコマンド](#)
- [トラブルシューティングおよび診断出力](#)

CRSCTLの概要

CRSCTLはユーザーとOracle Clusterwareのインタフェースであり、Oracle Clusterwareオブジェクト用のOracle Clusterware APIを解析およびコールします。

CRSCTLではクラスタ対応コマンドが提供されており、これらを使用して、チェック、起動、停止操作をクラスタで実行できます。これらのコマンドは、操作に応じてクラスタ内の任意のノードから、クラスタ内の別のノードで、またはクラスタ内のすべてのノードで実行できます。

CRSCTLコマンドを使用すると、次に示すいくつかの操作をOracle Clusterwareで実行できます。

- Oracle Clusterwareリソースの起動および停止
- Oracle Clusterwareデーモンの有効化および無効化
- クラスタの状態のチェック
- サード・パーティ製アプリケーションのリソースの管理
- Intelligent Platform Management Interface(IPMI)とOracle Clusterwareとの統合による障害分離サポートおよびクラスタの整合性の実現
- Oracle Clusterwareコンポーネントのデバッグ

この項には次のトピックが含まれます:

- [クラスタ用に変更された\(クラスタ対応\)コマンド](#)
- [CRSCTLの操作上のノート](#)

- [非推奨のサブプログラムまたはコマンド](#)

クラスタ用に変更された(クラスタ対応)コマンド

あるノードでクラスタ対応コマンドを実行してクラスタ内の別のノードで操作を実行できます。これらの操作はリモート操作と呼ばれます。これによって、たとえば、すべてのノードのOracle Clusterwareのステータスをチェックするために各ノードにログインする必要がなくなるため、管理が簡素化されます。

クラスタ対応コマンドはオペレーティング・システムにはまったく依存しませんが、OHASD(Oracle高可用性サービス・デーモン)に依存します。このデーモンが実行されているときに、リモート・ノードの起動、停止、ステータスのチェックなどのリモート操作を実行できます。

クラスタ対応コマンドには次のコマンドがあります。

- [crsctl check cluster](#)
- [crsctl start cluster](#)
- [crsctl stop cluster](#)

CRSCTLの操作上のノート

CRSCTLコマンドライン・インタフェースの使用に関する情報です。

使用方法

- CRSCTLユーティリティは、Grid_home/binディレクトリに格納されています。CRSCTLコマンドを実行するには、次の例に示すように、オペレーティング・システム・プロンプトでまず`crsctl`、次にコマンドおよび引数を入力します。

```
crsctl stop crs
```

- CRSCTLコマンドには3つのカテゴリがあります。
 - Oracle Real Application Clusters(Oracle RAC)環境またはOracle Restart環境のいずれかで使用するコマンド
 - Oracle RAC環境でのみ使用するコマンド
 - Oracle Restart環境でのみ使用するコマンド
- 多くのCRSCTLコマンドでは、`-f`パラメータを使用することによってチェックが無視されコマンドが強制実行されます。

たとえば、あるリソースに対する`crsctl stop resource`コマンドに強制パラメータを指定すると、そのリソースが実行中であり、それを依存先としている依存リソースも実行中である場合、強制パラメータはエラー・メッセージを省略し、かわりに、すべての依存リソースを停止または再配置してからコマンドに示すリソースを停止します。
- Oracle Clusterware 12cの管理に、12cリリース1 (12.1)より前のバージョンのCRSCTLを使用しないでください。

フィルタ

次のようにフィルタを使用して、CRSCTLコマンドが動作するOracle Clusterwareエンティティを絞り込むことができます。

- 簡易フィルタは、演算子を使用した属性/値ペアです。
- 演算子は、例に示すように空白で囲む必要があります。
- 簡易フィルタは、Boolean演算子を使用した式フィルタと呼ばれる式に組み込むことができます。

サポートされるフィルタの演算子は次のとおりです。

- =
- >
- <
- !=
- co: 次を含む
- st: 次で始まる
- en: 次で終わる
-

サポートされるブール演算子はANDおよびORです。

次にフィルタの例を示します。

- TYPE = type1
- ((TYPE = type1) AND (CHECK_INTERVAL > 50))
- (TYPE = type1) AND ((CHECK_INTERVAL > 30) OR (AUTO_START co never))
- NAME en network.res
- TYPE st ora.db

CRSCTLヘルプの使用

CRSCTLのヘルプ情報を出力するには、次のコマンドを使用します。

```
crsctl -help
```

startなどの特定のコマンドのヘルプを出力する場合は、次の例に示すように、コマンドを入力し、-helpを末尾に追加します。

```
crsctl start -help
```

-helpのかわりに省略形-hまたは-?を使用することもできます(このパラメータはLinux、UNIXおよびWindows環境で機能します)。

非推奨のサブプログラムまたはコマンド

[表E-1](#)は、非推奨のコマンドおよびそれらの代替(同じまたは類似した機能を実行するのに使用可能)を示しています。

表E-1 非推奨のCRSCTLコマンドおよびその代替

非推奨コマンド	代替コマンド
crs_stat	crsctl check cluster crsctl status resource
crs_register	crsctl add resource crsctl add type crsctl modify resource crsctl modify type
crs_unregister	crsctl stop resource crsctl delete resource
crs_start	crsctl start resource crsctl start crs crsctl start cluster

非推奨コマンド	代替コマンド
<code>crs_stop</code>	<code>crsctl stop resource</code> <code>crsctl stop crs crsctl stop cluster</code>
<code>crs_getperm</code>	<code>crsctl getperm resource</code> <code>crsctl getperm type</code>
<code>crs_profile</code>	<code>crsctl add resource</code> <code>crsctl add type crsctl status resource crsctl</code> <code>status type crsctl modify resource crsctl modify</code> <code>type</code>
<code>crs_relocate</code>	<code>crsctl relocate resource</code>
<code>crs_setperm</code>	<code>crsctl setperm resource</code> <code>crsctl setperm type</code>
<code>crsctl add crs administrator</code>	アクセス制御リスト(ACL)を使用して、サーバー・プールを追加できるユーザーを制御します。
<code>crsctl check crsd</code>	<code>crsctl check crs</code>
<code>crsctl check cssd</code>	<code>crsctl check css</code>
<code>crsctl check evmd</code>	<code>crsctl check evm</code>
<code>crsctl debug res log</code> <code>resource_name:level</code>	<code>crsctl set log</code>
<code>crsctl set css votedisk</code>	<code>crsctl add css votedisk</code> <code>crsctl delete css votedisk crsctl query css</code> <code>votedisk crsctl replace votedisk</code>
<code>crsctl start resources</code>	<code>crsctl start resource -all</code>
<code>crsctl stop resources</code>	<code>crsctl stop resource -all</code>
<code>crsctl set css logfilesize</code>	<code>crsctl get tracefileopts css</code> <code>crsctl set tracefileopts css</code>
<code>crsctl get css logfilesize</code>	<code>crsctl set tracefileopts css</code>
<code>crsctl set trace module_name</code> <code>"component:debugging_level</code> <code>[,component:debugging_level][</code> <code>,...]"</code>	

```
crsctl get trace
```

デュアル環境のCRSCTLコマンド

Oracle RACまたはOracle Restart環境で次のコマンドを使用できます。

この項に示すコマンドは、Oracle RAC環境およびOracle Restart環境の両方のOracle Clusterwareリソースを管理します。

crsctl check css

クラスタ同期サービス(CSS)のステータスをチェックします。

このコマンドを最も頻繁に使用するのは、Oracle Automatic Storage Management (Oracle ASM)がローカル・サーバーにインストールされている場合です。

構文

```
crsctl check css
```

例

このコマンドによって、次のような出力が返されます。

```
CRS-4529: Cluster Synchronization Services is online
```

crsctl check evm

イベント・マネージャのステータスをチェックします。

構文

```
crsctl check evm
```

例

このコマンドによって、次のような出力が返されます。

```
CRS-4533: Event Manager is online
```

crsctl get hostname

ローカル・サーバーのホスト名を取得します。

構文

```
crsctl get hostname
```

例

Oracle Clusterwareによってローカル・サーバーのホスト名が戻されます。

```
$ crsctl get hostname  
node2
```

crsctl add resource

Oracle Clusterwareによって管理されるリソースを登録します。

リソースには、アプリケーション・プロセス、データベース、サービス、リスナーなどがあります。

構文

```
crsctl add resource resource_name -type resource_type [-group group_name [-file
file_path |
    -attr "attribute_name=attribute_value[,...]"
    [-i] [-f]
```

パラメータ

表E-2 crsctl add resourceコマンドのパラメータ

パラメータ	説明
resource_name	短く、わかりやすいリソース名を指定します。
-type resource_type	前に -type フラグを付けて、追加するリソースの種類を指定します。
-group group_name	リソースを追加するリソース・グループを指定します。
-file file_path	リソースを定義する属性名/値ペアが行区切りで記載されているテキスト・ファイルのパス名(絶対または相対)。
-attr "attribute_name=attribute_value[,...]"	追加するリソースの属性を2つの異なる方法で指定できます。 ● -attr フラグの後に、1つ以上の属性名/値ペアをカンマで区切り、二重引用符("")で囲んで指定できます。たとえば: <pre>-attr "CHECK_INTERVAL=30, START_TIMEOUT=25"</pre> いくつかの属性は複数の値を持つことができます。そのような場合は、空白で値を区切り、値のリストを一重引用符で囲みます。たとえば: <pre>-attr "SERVER_POOL_NAMES= 'ora.pool1 ora.pool2', START_TIMEOUT=25"</pre> ● また、特定のカーディナリティ値および特定の程度値を付加してリソース・インスタンスの属性値を指定できます。この方法は特定の

パラメータ	説明
	サーバーに関連付けられるアプリケーションに役立ちます。-attr フラグの後の構文は次のとおりです。 <pre>attribute_name{@SERVERNAME(server_name) [@DEGREEID(did)] @CARDINALITYID(cid) [@DEGREEID(did)]}=attribute_value</pre> @SERVERNAME(server_name) 構文を指定すると、指定した属性の属性値は、指定したサーバーに存在するリソース・インスタンスに限定されます。 または、@CARDINALITYID(cid) 構文を指定すると、指定した属性の属性値は、特定のカーディナリティ ID(cid) を持つリソース・インスタンスに限定されます。 必要に応じて、@DEGREEID(did) 構文を SERVERNAME または CARDINALITYID 構文のいずれか、または両方と組み合わせて、特定の DEGREE が設定されているリソースに属性値を限定することができます。 例： <pre>CHECK_INTERVAL@SERVERNAME(node1)=45 STOP_TIMEOUT@CARDINALITYID(2)=65 STOP_TIMEOUT@SERVERNAME(node1)@DEGREEID(2)=65 STOP_TIMEOUT@CARDINALITYID(3)@DEGREEID(2)=65</pre>
-i	-i を指定すると、CRSCTL は、コマンドの処理で Oracle Clusterware がリソースまたはその依存リソースをロック解除するのを待機する必要がある場合、エラーを戻します。コマンドが互いに影響しないように、Oracle Clusterware によってリソースまたは他のオブジェクトがロックされることがあります。
-f	強制パラメータは次の場合に使用します。 ● まだ存在しない他のリソースに依存するリソースを追加する場合。強制パラメータによってチェックが無効になるため、コマンドの完了が妨げられません。 ● 他のリソースへの hard 依存性がリソースに設定されており、1 つ以上の依存性に対する実行権限がリ

パラメータ	説明
	ソース所有者にない状態でリソースを追加する場合。この場合には、強制パラメータを指定しないとエラーが表示されます。 ● 汎用サーバー・プールにサーバーを移動する必要がある場合があり、このために、application タイプのリソースを追加する場合。停止する必要のあるリソースがサーバーで現在ホスティングされている場合、強制パラメータが必要です。

使用上のノート

- resource_nameパラメータおよび-type resource_typeパラメータのどちらも必須です。関連付けられたリソース・タイプを[crsctl add type](#)を使用して作成できます。
- 任意のユーザーがリソースを作成できますが、SERVER_POOLS=*が設定されているlocal_resourceタイプのリソースまたはcluster_resourceタイプのリソースを作成できるのはクラスタウェア管理者のみです。

リソースの定義後、どのユーザーがリソースに関する特定の操作を実行できるかはACLによって制御されます。Oracle Clusterware管理者のリストは影響しなくなります。

Windowsでは、管理者グループのメンバーがすべてを完全に制御できます。

関連項目:

ACLの設定の詳細は、「[crsctl setperm resource](#)」を参照してください。

- 属性名/値ペアの属性値にカンマが含まれる場合、値は一重引用符(' ')で囲む必要があります。
- 属性ファイルの例を次に示します。

```
PLACEMENT=favored
HOSTING_MEMBERS=node1 node2 node3
RESTART_ATTEMPTS@CARDINALITYID(1)=0
RESTART_ATTEMPTS@CARDINALITYID(2)=0
FAILURE_THRESHOLD@CARDINALITYID(1)=2
FAILURE_THRESHOLD@CARDINALITYID(2)=4
FAILURE_INTERVAL@CARDINALITYID(1)=300
FAILURE_INTERVAL@CARDINALITYID(2)=500
CHECK_INTERVAL=2
CARDINALITY=2
```

- oraで始まる名前を持つリソースはOracleリソースであるため、これらのリソースにはこのコマンドを使用しないでください。

例

例E-1 例1

Oracle Clusterwareを使用してVIPをリソースとして登録するには、次のコマンドを実行します。

```
$ crsctl add resource app.appvip -type app.appvip.type -attr "RESTART_ATTEMPTS=2,
START_TIMEOUT=100,STOP_TIMEOUT=100,CHECK_INTERVAL=10,
USR_ORA_VIP=172.16.0.0,
START_DEPENDENCIES=hard(ora.net1.network)pullup(ora.net1.network),
```

```
STOP_DEPENDENCIES=hard(ora.net1.network)"
```

例E-2 例2

test_type1リソース・タイプに基づくリソースを登録するには、次のコマンドを実行します。

```
$ crsctl add resource r1 -type test_type1 -attr "PATH_NAME=/tmp/r1.txt"
$ crsctl add resource r1 -type test_type1 -attr "PATH_NAME=/tmp/r2.txt"
```

例E-3 例3

EXECUTABLE_NAMES属性を使用してgeneric_applicationリソース・タイプのSambaサーバー・リソースを登録するには、次のコマンドを実行します。

```
# crsctl add resource my_samba -type generic_application -attr
"EXECUTABLE_NAMES=smbd,START_PROGRAM='/etc/rc.d/init.d/smb start',
STOP_PROGRAM='/etc/rc.d/init.d/smb stop'"
```

例E-4 例4

EXECUTABLE_NAMES属性を使用してgeneric_applicationリソース・タイプのDNSサーバーを登録するには、次のコマンドを実行します。

```
# crsctl add resource my_dns -type generic_application -attr
"EXECUTABLE_NAMES=named,START_PROGRAM='/etc/rc.d/init.d/named start',
STOP_PROGRAM='/etc/rc.d/init.d/named stop'"
```

例E-5 例5

PID_FILES属性を使用してgeneric_applicationリソース・タイプのApache Webサーバーを登録するには、次のコマンドを実行します。

```
# crsctl add resource my_apache -type generic_application -attr
"START_PROGRAM='/usr/sbin/httpd -k start',STOP_PROGRAM='/usr/sbin/httpd -k stop',
PID_FILES=/etc/httpd/run/httpd.pid"
```

例E-6 例6

環境変数を使用してgeneric_applicationリソース・タイプのアプリケーションを登録するには、次のコマンドを実行します。

```
# crsctl add resource my_app -type generic_application -attr
"START_PROGRAM='/opt/my_app start', EXECUTABLE_NAMES=my_app,
ENVIRONMENT_VARS='USE_NETAPP=no,USE_BACKUP=yes,CLEAN_ON_KILL=yes'"
```

関連トピック

- [Oracle Clusterwareのリソース・リファレンス](#)

crsctl delete resource

Oracle Clusterware構成からリソースを削除します。

構文

```
crsctl delete resource {"resource_name [...]"} [-w "filter"] [-i] [-f]
```

パラメータ

表E-3 crsctl delete resourceコマンドのパラメータ

パラメータ	説明
resource_name	削除対象の 1 つ以上のリソース名を指定します。複数のリソースは空白で区切り、二重引用符("")で囲む必要があります。
-w "filter"	削除が評価されるリソースの数を制限するために Oracle Clusterware によって使用されるリソース・フィルタを指定します。フィルタは二重引用符("")で囲む必要があります。次にリソース・フィルタの例を示します。 ● "TYPE == cluster_resource": このフィルタは、cluster_resource タイプのリソースのみを再配置するように Oracle Clusterware を制限します。 ● "CHECK_INTERVAL > 10": このフィルタは、CHECK_INTERVAL リソース属性に 10 より大きい値を持つリソースを再配置するように Oracle Clusterware を制限します。 ● "(CHECK_INTERVAL > 10) AND (NAME co 2)": このフィルタは、CHECK_INTERVAL リソース属性に 10 より大きい値を持ち、かつ、リソースの名前に数字 2 が含まれるリソースを再配置するように Oracle Clusterware を制限します。 ノート: すべての演算子は空白で囲む必要があります。 関連項目: 演算子の詳細は、「フィルタ」を参照してください。
-i	-i を指定すると、CRSCTL は、コマンドの処理で Oracle Clusterware がリソースまたはその依存リソースをロック解除するのを待機する必要がある場合、エラーを戻します。コマンドが互いに影響しないように、Oracle Clusterware によってリソースまたは他のオブジェクトがロックされることがあります。
-f	このパラメータを使用して、オンラインのリソースを削除します。

使用上のノート

- resource_nameまたは-w "filter"パラメータは必須です。
- 指定したリソースを削除するには、読取りおよび書込み権限が必要です。
- oraで始まる名前を持つリソースはOracleリソースであるため、これらのリソースにはこのコマンドを使用しないでください。

例

Oracle Clusterwareからリソースを削除するには、次のコマンドを実行します。

```
# crsctl delete resource myResource
```

crsctl eval add resource

システムを変更せずに、リソースを追加した場合の影響を予測します。

このコマンドはアプリケーション管理者にとって役に立つ場合があります。

構文

```
crsctl eval add resource resource_name -type resource_type [-file file_path  
| -attr "attribute_name=attribute_value[,...]" ] [-group group_name] [-f]
```

パラメータ

表E-4 crsctl eval add resourceコマンドのパラメータ

パラメータ	説明
resource_name	短く、わかりやすいリソース名を指定します。
-type resource_type	-type フラグの後に追加するリソースのタイプを指定します。
-file file_path	リソースを定義する属性名/値ペアが行区切りで記載されているテキスト・ファイルのパス名(絶対または相対)。
-attr "attribute_name=attribute_value"	追加するリソースの属性を 2 つの異なる方法で指定できます。 ● -attr フラグの後に、1 つ以上の属性名/値ペアをカンマで区切り、二重引用符(" ")で囲んで指定できます。たとえば: -attr "CHECK_INTERVAL=30, START_TIMEOUT=25" いくつかの属性は複数の値を持つことができます。そのような場合は、空白で値を区切り、値のリストを一重引用符で囲みます。たとえば: -attr "SERVER_POOL_NAMES= 'ora.pool1 ora.pool2', START_TIMEOUT=25" ● 必要に応じて、特定のカーディナリティ値および特定の程度値を付加してリソース・インスタンスの属性値を指定できます。この方法は特定のサーバーに関連付けられるアプリケーションに役立ちます。-attr フラグの後の構文は次のとおりです。 attribute_name{@SERVERNAME(server_name) [@DEGREEID(did)] @CARDINALITYID(cid) [@DEGREEID(did)]}=attribute_value @SERVERNAME(server_name)構文を指定すると、指定

パラメータ	説明
	した属性の属性値は、指定したサーバーに存在するリソース・インスタンスに限定されます。 または、<code>@CARDINALITYID(cid)</code>構文を指定すると、指定した属性の属性値は、特定のカーディナリティ <code>ID(cid)</code>を持つリソース・インスタンスに限定されます。 必要に応じて、<code>@DEGREEID(did)</code>構文を <code>SERVERNAME</code> または <code>CARDINALITYID</code> 構文のいずれか、または両方と組み合わせて、特定の <code>DEGREE</code> が設定されているリソースに属性値を限定することができます。 例： <pre>CHECK_INTERVAL@SERVERNAME(node1)=45 STOP_TIMEOUT@CARDINALITYID(2)=65 STOP_TIMEOUT@SERVERNAME(node1)@DEGREEID(2)=65 STOP_TIMEOUT@CARDINALITYID(3)@DEGREEID(2)=65</pre>
<code>-group group_name</code>	リソースを追加するリソース・グループを指定します。
<code>-f</code>	強制パラメータは次の場合に使用します。 ● まだ存在しない他のリソースに依存するリソースを追加する場合。強制パラメータによってチェックが無効になるため、コマンドの完了が妨げられません。 ● 他のリソースへの <code>hard</code> 依存性がリソースに設定されており、1 つ以上の依存性に対する実行権限がリソース所有者にない状態でリソースを追加する場合。この場合には、強制パラメータを指定しないとエラーが表示されます。 ● 汎用サーバー・プールにサーバーを移動する必要がある場合があり、このために、<code>application</code> タイプのリソースを追加する場合。停止する必要があるリソースがサーバーで現在ホスティングされている場合、強制パラメータが必要です。

crsctl eval fail resource

リソース障害の結果を予測します。

構文

```
crsctl eval fail resource {resource_name | -w "filter"} [-n server]
[-env "attr=val[,...]" ]
```

表E-5 crsctl eval fail resourceコマンドのパラメータ

パラメータ	説明
resource_name	障害をシミュレートするリソースの名前を指定します。リソースの名前、または-w パラメータを使用してリソース・フィルタを指定する必要があります。
-w "filter"	または、評価されるリソースの数を制限するために Oracle Clusterware によって使用されるリソース・フィルタを指定することもできます。フィルタは二重引用符("")で囲む必要があります。次にリソース・フィルタの例を示します。 ● "TYPE == cluster_resource": このフィルタは、cluster_resource タイプのリソースのみを再配置するように Oracle Clusterware を制限します。 ● "CHECK_INTERVAL > 10": このフィルタは、CHECK_INTERVAL リソース属性に 10 より大きい値を持つリソースを再配置するように Oracle Clusterware を制限します。 ● "(CHECK_INTERVAL > 10) AND (NAME co 2)": このフィルタは、CHECK_INTERVAL リソース属性に 10 より大きい値を持ち、かつ、リソースの名前に数字 2 が含まれるリソースを再配置するように Oracle Clusterware を制限します。 ノート: すべての演算子は空白で囲む必要があります。 関連項目: 演算子の詳細は、「フィルタ」を参照してください。
-n server	障害をシミュレートするリソースがあるサーバーの名前を指定します。
-env "attr=val[,...]"	必要に応じて、-env コマンド・パラメータを使用して、1 つ以上のリソース・プロファイル属性値を上書きできます。複数の環境名/値ペアを指定する場合、各ペアをカンマで区切り、リスト全体を二重引用符("")で囲む必要があります。 たとえば、-env "RESTART_ATTEMPTS=0"を使用すると、リソースがローカルでの再起動の試行限度を超えローカルで再起動できない場合の動作を予測できます。

例

このコマンドによって、次のような出力が返されます。

```
$ crsctl eval fail res cs1
```

Stage Group 1:

Stage Number	Required	Action
1	Y	Resource 'cs1' (1/1) will be in state [ONLINE INTERMEDIATE] on server [mjkeenan_node_0]
	Y	Resource 'cs1' (2/1) will be in state [ONLINE INTERMEDIATE] on server [mjkeenan_node_1]

crsctl eval relocate resource

システムを変更せずに、リソースの再配置をシミュレートします。

構文

```
crsctl eval relocate resource {resource_name | -all} {-s source_server | -w "filter"} [-n destination_server] [-f]
```

パラメータ

表E-6 crsctl eval relocate resourceコマンドのパラメータ

パラメータ	説明
resource_name	再配置するリソースの名前を指定します。
-all	または、このパラメータを使用して、再配置対象のすべてのリソースを指定できます。
-s source_server	リソースの再配置元である特定のサーバーを指定します。
-w "filter"	-sを使用するかわりに、再配置されるリソースの数を制限するために CRSCTL によって使用されるリソース・フィルタを指定できます。フィルタは二重引用符("")で囲む必要があります。次にリソース・フィルタの例を示します。 ● "TYPE == cluster_resource": このフィルタは、cluster_resource タイプのリソースのみを再配置するように Oracle Clusterware を制限します。 ● "CHECK_INTERVAL > 10": このフィルタは、CHECK_INTERVAL リソース属性に 10 より大きい値

パラメータ	説明
	を持つリソースを再配置するように Oracle Clusterware を制限します。 ● "(CHECK_INTERVAL > 10) AND (NAME co 2)": このフィルタは、CHECK_INTERVAL リソース属性に 10 より大きい値を持ち、かつ、リソースの名前に数字 2 が含まれるリソースを再配置するように Oracle Clusterware を制限します。 関連項目: 「フィルタ」を参照してください。
-n destination_server	リソースの再配置先サーバーの名前を指定します。宛先サーバーを指定しない場合、CRSCTL によって、各リソースの属性プロファイルに従い、最適なサーバーにリソースが再配置されます。
-f	再配置対象のリソースに依存している他のリソースが実行中である場合に再配置対象のリソースを強制的に再配置するには、-f パラメータを指定します。このパラメータを使用すると、CRSCTL によって、依存リソースは停止または再配置されます。 ノート: 1 より大きいカーディナリティを持つリソースを再配置する場合、-k または -s のいずれかを使用して、再配置対象のリソース・インスタンスを絞り込む必要があります。

crsctl eval modify resource

システムを変更せずに、リソースを変更した場合の影響を予測します。

構文

```
crsctl eval modify resource resource_name -attr
"attribute_name=attribute_value[,...]"
[-group group_name] [-f]
```

パラメータ

表E-7 crsctl eval modify resourceコマンドのパラメータ

パラメータ	説明
resource_name	評価対象のリソースの名前を指定します。
-attr "attribute_name=attribute_value [,...]"	評価対象のリソースの属性を次の 2 つの異なる方法で指定できます。 ● -attr フラグの後に、変更対象の 1 つ以上の属性名/値ペアをカンマで区切り、二重引用符(" ")で囲んで指定できます。たとえ

パラメータ	説明
	ば: <pre>-attr "CHECK_INTERVAL=30, START_TIMEOUT=25"</pre> いくつかの属性は複数の値を持つことができます。そのような場合は、空白で値を区切り、値のリストを一重引用符で囲みます。たとえば: <pre>-attr "SERVER_POOL_NAMES= 'ora.pool1 ora.pool2', START_TIMEOUT=25"</pre> ● または、特定のカーディナリティ値および特定の程度値を付加して特定のサーバーに存在するリソースの属性値を指定できます。この方法は特定のサーバーに何らかの方法で関連付けられるアプリケーションに役立ちます。 -attr フラグの後の構文は次のとおりです。 <pre>attribute_name{@SERVERNAME(server_name) [@DEGREEID(did)] @CARDINALITYID(cid) [@DEGREEID(did)]}=attribute_value</pre> @SERVERNAME(server_name)構文を指定すると、指定した属性の属性値は、指定したサーバーに存在するリソースに限定されます。 または、@CARDINALITYID(cid)構文を指定すると、指定した属性の属性値は、特定のカーディナリティID(cid)を持つリソース・インスタンスに限定されます。 必要に応じて、@DEGREEID(did)構文を SERVERNAME または CARDINALITYID 構文のいずれか、または両方と組み合わせて、特定の DEGREE が設定されているリソースに属性値を限定することができます。 例: <pre>CHECK_INTERVAL@SERVERNAME(node1)=45 STOP_TIMEOUT@CARDINALITYID(2)=65 STOP_TIMEOUT@SERVERNAME(node1)@DEGREEID(2)=65 STOP_TIMEOUT@CARDINALITYID(3)@DEGREEID(2)=65</pre>
-group group_name	評価対象のリソースが属するリソース・グループを指定します。
-f	強制パラメータを指定してコマンドを実行した場合に発生する事象を評価するには、このパラメータを指定します。

crsctl eval start resource

システムを変更せずに、リソースを起動した場合の影響を予測します。

構文

```
crsctl eval start resource {resource_name [...] | -w "filter"}  
[-n server_name] [-f]
```

パラメータ

表E-8 crsctl eval start resourceコマンドのパラメータ

パラメータ	説明
resource_name [...]	空白区切りの 1 つ以上の起動対象のリソース名を指定します。
-w "filter"	Oracle Clusterware によってリソースの照合に使用されるリソース・フィルタを、二重引用符(" ")で囲んで指定します。たとえば、-w "TYPE=ora.database.type"や-w "NAME=cs1"とします。 関連項目: 「 フィルタ 」を参照してください。
-n server_name	起動するリソースが存在するサーバーの名前を指定します。サーバーを指定しない場合、Oracle Clusterware では各リソースの属性プロファイルに従い、最適なサーバー上のリソースが起動されます。
-f	起動するリソースが別のサーバーで実行中であり、そのサーバーに対する hard 起動依存性が定義されている場合、-f パラメータを使用してそのリソースを再配置します。この場合には、強制パラメータを指定しないと、start コマンドは失敗します。

crsctl eval stop resource

システムを変更せずに、リソースを停止した場合の影響を予測します。

構文

```
crsctl eval stop resource {resource_name [...] | -w "filter"} [-f]
```

パラメータ

表E-9 crsctl eval stop resourceコマンドのパラメータ

パラメータ	説明
resource_name [...]	空白区切りの 1 つ以上の停止対象のリソース名を指定します。

パラメータ	説明
-w "filter"	停止されるリソースの数を制限するために Oracle Clusterware によって使用されるリソース・フィルタを指定します。フィルタは二重引用符(" ")で囲む必要があります。次にリソース・フィルタの例を示します。 ● "TYPE == cluster_resource": このフィルタは、cluster_resource タイプのリソースのみを停止するように Oracle Clusterware を制限します。 ● "CHECK_INTERVAL > 10": このフィルタは、CHECK_INTERVAL リソース属性に 10 より大きい値を持つリソースを停止するように Oracle Clusterware を制限します。 ● "(CHECK_INTERVAL > 10) AND (NAME co 2)": このフィルタは、CHECK_INTERVAL リソース属性に 10 より大きい値を持ち、かつ、リソースの名前に数字 2 が含まれるリソースを停止するように Oracle Clusterware を制限します。 関連項目: 「フィルタ」を参照してください。

-f	停止対象のリソースに依存している他のリソースが実行中である場合に停止対象のリソースを強制的に停止するには、-f パラメータを指定します。このパラメータを使用すると、依存リソースは再配置または停止されます。

crsctl getperm resource

指定されたリソースのユーザー権限とグループ権限を表示します。

構文

```
crsctl getperm resource resource_name [{-u user_name | -g group_name}]
```

パラメータ

表E-10 crsctl getperm resourceコマンドのパラメータ

パラメータ	説明
resource_name	権限を取得するリソースの名前を指定します。
-u user_name	-u を指定すると、Oracle Clusterware では特定のユー

パラメータ	説明
	ザーの権限が取得されます。
-g group_name	-g を指定すると、Oracle Clusterware では特定のグループの権限が取得されます。

使用上のノート

- resource_nameパラメータは必須です。
- 指定したリソースの権限を取得するには、それらのリソースに対する読取り権限が必要です。
- oraで始まる名前を持つリソースはOracleリソースであるため、これらのリソースにはこのコマンドを使用しないでください。

関連項目:

リソースおよびリソース属性の詳細は、[「Oracle Clusterwareのリソース・リファレンス」](#)を参照してください。

例

crsctl getperm resourceコマンドでは、選択するコマンド・オプションに応じて次のような出力が戻されます。

```
$ crsctl getperm resource app.appvip
Name: app.appvip
owner:root:rwX,pgrp:oinstall:rwX,other::r--

$ crsctl getperm resource app.appvip -u oracle
Name: app.appvip
rwx

$ crsctl getperm resource app.appvip -g dba
Name: app.appvip
r--
```

crsctl modify resource

Oracle Clusterwareの特定のリソースの属性を変更します。

構文

```
crsctl modify resource resource_name -attr "attribute_name=attribute_value"
[-i] [-f] [-delete] [-group group_name] [-env "env1=val1,env2=val2,..."]
```

パラメータ

表E-11 crsctl modify resourceコマンドのパラメータ

パラメータ	説明
resource_name	変更するリソースの名前。
-attr "attribute_name=attribute_value"	変更するリソースの属性を 2 つの異なる方法で指定できます。

- `-attr` フラグの後に、変更対象の 1 つ以上の属性名/値ペアをカンマで区切り、二重引用符("`"`)で囲んで指定できます。たとえば:

```
-attr "CHECK_INTERVAL=30,
START_TIMEOUT=25"
```

いくつかの属性は複数の値を持つことができます。そのような場合は、空白で値を区切り、値のリストを一重引用符で囲みます。たとえば:

```
-attr "SERVER_POOL_NAMES=
'ora.pool1 ora.pool2', START_TIMEOUT=25"
```

- または、特定のカーディナリティ値および特定の程度値を付加して特定のサーバーに存在するリソースの属性値を指定できます。この方法は特定のサーバーに何らかの方法で関連付けられるアプリケーションに役立ちます。`-attr` フラグの後の構文は次のとおりです。

```
attribute_name{@SERVERNAME(server_name)
[@DEGREEID(did)] | @CARDINALITYID(cid)
[@DEGREEID(did)]}=attribute_value
```

`@SERVERNAME(server_name)` 構文を指定すると、指定した属性の属性値は、指定したサーバーに存在するリソースに限定されます。

または、`@CARDINALITYID(cid)` 構文を指定すると、指定した属性の属性値は、特定のカーディナリティ `ID(cid)` を持つリソース・インスタンスに限定されます。

必要に応じて、`@DEGREEID(did)` 構文を `SERVERNAME` または `CARDINALITYID` 構文のいずれか、または両方と組み合わせて、特定の `DEGREE` が設定されているリソースに属性値を限定することができます。

例:

```
CHECK_INTERVAL@SERVERNAME(node1)=45
STOP_TIMEOUT@CARDINALITYID(2)=65
STOP_TIMEOUT@SERVERNAME(node1)@DEGREEID(2)=65
STOP_TIMEOUT@CARDINALITYID(3)@DEGREEID(2)=65
```

`-i` を指定すると、`CRSCTL` は、コマンドの処理で Oracle Clusterware がリソースまたはその依存リソースをロック解除するのを待機する必要がある場合、エラーを戻します。コマンドが互いに影響しないように、Oracle Clusterware によってリソースまたは他のオブジェクトがロックされることがあ

パラメータ	説明
	ります。
-f	-f パラメータは次の場合に使用します。 ● 存在しないリソースへの hard 依存性がリソースに設定されている場合 ● 1 つ以上の hard 依存性に対する実行権限がリソース所有者にない場合 ● 変更によってサーバーが汎用プールに移動され、サーバーの移動を完了するためにリソースが停止または再配置される場合
-delete	-delete パラメータを使用すると、指定した属性が Oracle Clusterware によって削除されます。
-group group_name	リソースが属するリソース・グループを指定します。リソース・グループは存在する必要があり、リソースは他の明示的なリソース・グループのメンバーになることはできません。リソース・グループを指定しない場合、リソースは、CRSCTL によってそのリソースと同じ名前のリソース・グループに追加されます。
-env "env1=val1, env2=val2, ..."	必要に応じて、このコマンドの 1 つ以上のリソース・プロファイル属性値を上書きできます。複数の環境名/値ペアを指定する場合、各ペアをカンマで区切り、リスト全体を二重引用符("")で囲む必要があります。

使用上のノート

- resource_nameパラメータは必須です。
- 属性名/値ペアの属性値にカンマが含まれる場合、値は一重引用符(' ')で囲む必要があります。たとえば:

```
"START_DEPENDENCIES='hard(res1,res2,res3)'"
```
- 指定したリソースを変更するには、それらのリソースに対する読取りおよび書込み権限が必要です。
- oraで始まる名前を持つリソースはOracleリソースであるため、これらのリソースにはこのコマンドを使用しないでください。

例

appsvipリソースの属性を変更するには、次のコマンドを実行します。

```
$ crsctl modify resource appsvip -attr USR_ORA_VIP=10.1.220.17 -i
```

関連トピック

- [Oracle Clusterwareのリソース・リファレンス](#)

crsctl relocate resource

クラスタ内の別のサーバーにリソースを再配置します。

構文

```
crsctl relocate resource {resource_name [-k cid] | -all | -w "filter"} -s
source_server |
[-n destination_server] [-env "env1=val1,env2=val2,..."] [-i] [-f]
```

パラメータ

表E-12 crsctl relocate resourceコマンドのパラメータ

パラメータ	説明
resource_name	再配置するリソースの名前を指定します。
-k cid	必要に応じて、リソースのカーディナリティ ID を指定できます。このパラメータを指定すると、指定したカーディナリティのリソース・インスタンスが CRSCTL によって再配置されます。
-all	または、このパラメータを使用して、再配置対象のすべてのリソースを指定できます。
-w "filter"	-k または -all を使用するかわりに、再配置されるリソースの数を制限するために CRSCTL によって使用されるリソース・フィルタを指定できます。フィルタは二重引用符(" ")で囲む必要があります。次にリソース・フィルタの例を示します。 ● "TYPE == cluster_resource": このフィルタは、cluster_resource タイプのリソースのみを再配置するように Oracle Clusterware を制限します。 ● "CHECK_INTERVAL > 10": このフィルタは、CHECK_INTERVAL リソース属性に 10 より大きい値を持つリソースを再配置するように Oracle Clusterware を制限します。 ● "(CHECK_INTERVAL > 10) AND (NAME co 2)": このフィルタは、CHECK_INTERVAL リソース属性に 10 より大きい値を持ち、かつ、リソースの名前に数字 2 が含まれるリソースを再配置するように Oracle Clusterware を制限します。

関連項目: 「[フィルタ](#)」を参照してください。

パラメータ	説明
-s source_server	リソースの再配置元である特定のサーバーを指定します。
-n destination_server	リソースの再配置先サーバーの名前を指定します。宛先サーバーを指定しない場合、CRSCTL によって、各リソースの属性プロファイルに従い、最適なサーバーにリソースが再配置されます。
-env "env1=val1, env2=val2, ..."	必要に応じて、このコマンドの 1 つ以上のリソース・プロファイル属性値を上書きできます。複数の環境名/値ペアを指定する場合、各ペアをカンマで区切り、リスト全体を二重引用符("")で囲む必要があります。
-i	-i を指定すると、このコマンドの処理で Oracle Clusterware がリソースまたはその依存リソースをロック解除するのを待機する必要がある場合、コマンドはエラーを戻します。コマンドが互いに影響しないように、Oracle Clusterware によってリソースまたは他のオブジェクトがロックされることがあります。
-f	再配置対象のリソースに依存している他のリソースが実行中である場合に再配置対象のリソースを強制的に再配置するには、-f パラメータを指定します。このパラメータを使用すると、CRSCTL によって、依存リソースは停止または再配置されます。 ノート: 1 より大きいカーディナリティを持つリソースを再配置する場合、-k または -s のいずれかを使用して、再配置対象のリソース・インスタンスを絞り込む必要があります。

使用上のノート

- 再配置対象のリソースを指定するには、次の3つの方法のいずれかを使用する必要があります。
 - 再配置対象の1つの特定のリソースを指定します。
 - または、特定のソース・サーバーから再配置する1つの特定のリソースまたはすべてのリソースを指定できます。
 - 3つ目の方法として、CRSCTLによって再配置対象のリソースの照合に使用されるリソース・フィルタを指定できます。
- リソースの程度IDが1より大きい場合、このリソースのすべてのインスタンスがCRSCTLによって再配置されます。
- 指定したリソースを再配置するには、それらのリソースに対する読取りおよび実行権限が必要です。
- oraで始まる名前を持つリソースはOracleリソースであるため、これらのリソースにはこのコマンドを使用しないでください。

例

あるサーバーから別のサーバーに1つの特定のリソースを再配置するには、次のコマンドを実行します。

```
# crsctl relocate resource myResource1 -s node1 -n node3
```

crsctl restart resource

2つのコマンドを実行してリソースを停止および起動するかわりに、クラスタ内のアイドル・リソースを再起動します。

構文

```
crsctl restart resource {resource_name [...] | -w "filter"} [-k cid] [-d did]
[-env "env1=val1,env2=val2,..."] [-i] [-f]
```

パラメータ

表E-13 crsctl restart resourceコマンドのパラメータ

パラメータ	説明
resource_name [...]	再起動する 1 つ以上のリソースの名前を空白区切りで指定します。
-w "filter"	Oracle Clusterware によってリソースの照合に使用されるリソース・フィルタを、二重引用符("")で囲んで指定します。たとえば、-w "TYPE=ora.database.type"や-w "NAME=cs1"とします。 関連項目: 「フィルタ」を参照してください。
-k cid	リソースのカーディナリティ ID を指定します。このパラメータを指定すると、指定したカーディナリティのリソース・インスタンスが Oracle Clusterware によって再起動されます。
-d did	リソースの程度 ID を指定します。このパラメータに 1 より大きい程度 ID を指定すると、この基準を満たすすべてのリソース・インスタンスが Oracle Clusterware によって再起動されます。 ノート: -k パラメータを指定せずに -d パラメータを使用することはできません。
-env "env1=val1,env2=val2,..."	必要に応じて、-env コマンド・パラメータを使用して、1 つ以上のリソース・プロファイル属性値を上書きできます。複数の環境名/値ペアを指定する場合、各ペアをカンマで区切り、リスト全体を二重引用符("")で囲む必要があります。
-i	-i を指定すると、このコマンドの処理で Oracle Clusterware がリソースまたはその依存リソースをロック解除するのを待機する必要がある場合、コマンドはエラーを戻します。コマンドが互いに影響しないように、Oracle Clusterware によってリソースまたは他のオブジェクトがロックされることがあります。

パラメータ	説明
-f	再起動するリソースが別のサーバーで実行中で、そのサーバーに対する hard 起動依存性が定義されている場合、-f パラメータを使用してそのリソースを再配置します。この場合には、強制パラメータを指定しないと、start コマンドは失敗します。

使用上のノート

- 再起動対象のリソースを指定するには、次のいずれかの方法を使用する必要があります。
 - 再起動対象の1つ以上のリソースを名前で指定します
 - Oracle Clusterwareによって再起動対象のリソースの照合に使用されるリソース・フィルタを指定します
- 指定したリソースを再起動するには、それらのリソースに対する読取りおよび実行権限が必要です
- oraで始まる名前を持つリソースはOracleリソースであるため、これらのリソースの再起動にはこのコマンドを使用しないでください

crsctl setperm resource

特定のリソースの権限を設定します。

構文

```
crsctl setperm resource resource_name {-u acl_string | -x acl_string | -o user_name | -g group_name}
```

パラメータ

表E-14 crsctl setperm resourceコマンドのパラメータ

パラメータ	説明
resource_name	権限を設定するリソースの名前を指定します。
-u -x -o -g	リソースには次の権限のうち 1 つのみを設定できます。 ● -u acl_string: リソースのアクセス制御リスト(ACL)を更新できます。 ● -x acl_string: リソースの ACL を削除できます。 ● -o user_name: ユーザー名を入力することによってリソースの所有者を変更できます。 ● -g group_name: グループ名を入力することによってリソースのプライマリ・グループを変更できます。 ユーザー、グループ、他の ACL 文字列は次のように指定します。

パラメータ	説明
	<pre>user:user_name[:readPermwritePermexecPerm] group:group_name[:readPermwritePermexecPerm] other[:readPermwritePermexecPerm]</pre> ● user: ユーザーACL ● group: グループ ACL ● other: その他の ACL ● readPerm: リソースに対する読取り権限。文字 r はユーザー、グループ、その他の読取り権限を付与し、マイナス記号(-)は読取り権限を付与しません。 ● writePerm: リソースに対する書込み権限。文字 w はユーザー、グループ、その他の書込み権限を付与し、マイナス記号(-)は書込み権限を付与しません。 ● execPerm: リソースに対する実行権限。文字 x はユーザー、グループ、その他の実行権限を付与し、マイナス記号(-)は実行権限を付与しません。

関連項目:

リソースおよびリソース属性の詳細は、[「Oracle Clusterwareのリソース・リファレンス」](#)を参照してください。

使用上のノート

- oraで始まる名前を持つリソースはOracleリソースであるため、これらのリソースにはこのコマンドを使用しないでください。
- 権限を設定するには、指定したリソースに対する読取りおよび書込み権限が必要です。

例

リソースに対する読取り、書込み、および実行権限をユーザーScottに付与するには、次のコマンドを実行します。

```
$ crsctl setperm resource myResource -u user:scott:rwX
```

crsctl start resource

クラスタ内の特定のサーバー上のアイドル・リソースを起動します。

構文

```
crsctl start resource {resource_name [...] | -w "filter" | -all}
    [-n server_name | -s server_pool_names] [-k cid] [-d did]
    [-env "env1=val1,env2=val2,..."] [-begin] [-end] [-i] [-f] [-l]
```

パラメータ

表E-15 crsctl start resourceコマンドのパラメータ

パラメータ	説明

パラメータ	説明
resource_name [...]	起動する 1 つ以上のリソースの名前を空白区切りで指定します。
-w "filter"	Oracle Clusterware によってリソースの照合に使用されるリソース・フィルタを、二重引用符(" ")で囲んで指定します。たとえば、-w "TYPE=ora.database.type"や-w "NAME=cs1"とします。 関連項目: 「フィルタ」を参照してください。
-all	クラスタ内のすべてのリソースを起動するには、このパラメータを使用します。
-n server_name	起動するリソースが存在するサーバーの名前を指定します。サーバーを指定しない場合、Oracle Clusterware では各リソースの属性プロファイルに従い、最適なサーバー上のリソースが起動されます。
-s server_pool_names	起動するリソースが存在する 1 つのサーバー・プール名または空白区切りのサーバー・プールのリストを指定します。
-k cid	リソースのカーディナリティ ID を指定します。このパラメータを指定すると、指定したカーディナリティのリソース・インスタンスが Oracle Clusterware によって起動されます。
-d did	リソースの程度 ID を指定します。このパラメータに 1 より大きい程度 ID を指定すると、この基準を満たすすべてのリソース・インスタンスが Oracle Clusterware によって起動されます。 ノート: -k パラメータを指定せずに -d パラメータを使用することはできません。
-env "env1=val1, env2=val2, ..."	必要に応じて、-env コマンド・パラメータを使用して、1 つ以上のリソース・プロファイル属性値を上書きできます。複数の環境名/値ペアを指定する場合、各ペアをカンマで区切り、リスト全体を二重引用符(" ")で囲む必要があります。
-begin	このパラメータを指定して、透過的 HA アクションを開始できます。
-end	このパラメータを指定して、透過的 HA アクションを終了できます。
-i	-i を指定すると、このコマンドの処理で Oracle Clusterware がリ

パラメータ	説明
	ソースまたはその依存リソースをロック解除するのを待機する必要がある場合、コマンドはエラーを戻します。コマンドが互いに影響しないように、Oracle Clusterware によってリソースまたは他のオブジェクトがロックされることがあります。
-f	起動するリソースが別のサーバーで実行中であり、そのサーバーに対する hard 起動依存性が定義されている場合、-f パラメータを使用してそのリソースを再配置します。この場合には、強制パラメータを指定しないと、start コマンドは失敗します。
-l	start コマンドが失敗した場合は、-l パラメータを使用してリソースを元の状態のままにします。

使用上のノート

- 起動対象のリソースを指定するには、次の3つの方法のいずれかを使用する必要があります。
 - 起動対象の1つ以上のリソースを指定します。
 - Oracle Clusterwareによって起動対象のリソースの照合に使用されるリソース・フィルタを指定します。
 - -allパラメータを指定して、指定したサーバー上のすべてのリソースを起動できます。
- 指定したリソースを起動するには、それらのリソースに対する読取りおよび実行権限が必要です。
- oraで始まる名前を持つリソースはOracleリソースであるため、これらのリソースの起動にはこのコマンドを使用しないでください
- Oracleでは、Oracle Grid Infrastructure外部での管理対象アプリケーションの起動はサポートされていません

例

リソースを起動するには、次のコマンドを実行します。

```
# crsctl start resource myResource -n server1
```

crsctl status resource

任意の数の特定のリソースのステータスおよび構成情報を取得します。

構文

```
crsctl status resource resource_name [...] | -w "filter" [-p | -v] | [-f | -l | -g] [[-k cid | -n server_name] [-e [-p | -v]] [-d did]] | [-s -k cid [-d did]] [-p [-attr attribute_name[,...] [-noname]]]
```

リソースの状態を表形式で印刷するには、次のコマンドを実行します。

```
crsctl status resource resource_name [...] | -w "filter" -t
```

リソースの依存関係のリストを印刷するには、次のコマンドを実行します。

```
crsctl status resource [resource_name [...]] -dependency [-stop | -pullup]
```

表E-16 crsctl status resourceコマンドのパラメータ

パラメータ	説明
resource_name [...] -w "filter"	ステータスのチェック対象である 1 つ以上のリソースの名前を空白区切りで指定します。 必要に応じて、表示されるリソースの数を制限するために CRSCCTL によって使用されるリソース・フィルタを指定できます。フィルタは二重引用符("")で囲む必要があります。値にカッコまたは空白が含まれる場合、一重引用符(' ')で囲む必要があります。演算子は空白で囲む必要があります。次にリソース・フィルタの例を示します。 ● "TYPE == cluster_resource": このフィルタは、cluster_resource タイプのリソースのみを表示するように制限します。 ● "CHECK_INTERVAL > 10": このフィルタは、CHECK_INTERVAL リソース属性に 10 より大きい値を持つリソースを表示するように制限します。 ● "(CHECK_INTERVAL > 10) AND (NAME co 2)": このフィルタは、CHECK_INTERVAL リソース属性に 10 より大きい値を持ち、かつ、リソースの名前に数字 2 が含まれるリソースを表示するように制限します。 ● "START_DEPENDENCIES='hard(appsvip)'": このフィルタは appsvip リソースに対する hard 起動依存性を持つリソースを表示するように制限します。 関連項目: 「フィルタ」を参照してください。
[-p -v] [-f -l -g]	オプションで、次のパラメータを指定できます。 ● -p パラメータを指定してリソースの静的な構成を表示するか、または、-v パラメータを指定してリソースの実行時の構成を表示します。 ● リソースのすべての構成を表示する場合は -f パラメータを、リソースのすべてのカーディナル値および程度値を表示する場合は -l パラメータを、指定したリソースが登録されているかどうかをチェックする場合は -g パラメータを指定します。

パラメータ	説明
<pre>[[-k cid -n server_name] [-e [-p -v]] [-d did [-s -k cid [-d did]]]</pre>	次の 2 つのオプションのいずれかを指定できます。 ● 問合せ対象のリソースのカーディナリティ ID を指定する場合は -k cid を指定します。または、-n パラメータを指定して、リソースをチェックする特定のサーバーを指定できます。必要に応じて -n パラメータとともに -d パラメータを指定して、チェックするリソースの程度 ID を指定できます。1 より大きい程度 ID を指定すると、この基準を満たすサーバー上のすべてのリソース・インスタンスが Oracle Clusterware によってチェックされます。 リソース・インスタンスの特別な値を評価するには、-e パラメータを使用します。-e パラメータとともに -p または -v も指定する必要があります。 ● 再配置対象のサーバーのリストを取得する場合は、-k パラメータとともに -s パラメータを指定します。-d パラメータとともに程度 ID を指定することによって、出力をさらに制限できます。
-t	表形式で出力を表示する場合は -t パラメータを指定します。
-p [-attr attribute_name[,...]] [-noname]	特定の属性を出力するか、デフォルトであるリソース名の出力を省略する場合は、-p の後に、それぞれ、-attr (単一の属性名、またはカンマ区切りの属性名リストが続く) または -noname を指定します。
-dependency [-stop -pullup]	リソースの依存性を表示するには、-dependency パラメータを指定します。-stop または -pullup オプションのいずれかを指定しなかった場合、CRSCTL ではリソースの起動依存性が表示されます。 -dependency パラメータでは次のいずれかのオプションを使用します。 ● リソースの停止依存性を表示するには、-stop パラメータを指定します。 ● リソースのプルアップ依存性を表示するには、-pullup パラメータを指定します。

- リソースの空白区切りのリストまたはリソース・フィルタのいずれかが必要です。
- 指定したリソースのステータスを取得するには、それらのリソースに対する読取り権限が必要です。
- クラスタにデプロイされている任意のリソースのステータス情報を問い合わせるには、`crsctl status resource`を使用します。ただし、Oracle(ora.*)リソースのステータス情報を問い合わせる場合は、対応するSRVCTLコマンドを使用することをお勧めします。

例

`crsctl status resource`コマンドでは次のような出力が戻されます。

```
$ crsctl status resource ora.staii14.vip
NAME=ora.staii14.vip
TYPE=ora.cluster_vip_net1.type
TARGET=ONLINE
STATE=ONLINE on staii14
```

次の例では、`ora.newdb.db`というリソースの起動依存性を表示します。

```
$ crsctl status resource ora.newdb.db -dependency ora.newdb.db(ora.database.type)
ora.ACFS_DG1.dg(ora.diskgroup.type)[hard,pullup]
ora.asm(ora.asm.type)[hard,pullup]
ora.LISTENER.lsnr(ora.listener.type)[weak]
type:ora.cluster_vip_net1.type[hard:type,pullup:type]
ora.net1.network(ora.network.type)[hard,pullup]
ora.dbhome_dg.dbhome_dg_v.acfs(ora.acfs.type)[hard,pullup]
ora.asm(ora.asm.type)[pullup:always]
```

crsctl stop resource

実行中のリソースを停止します。

構文

```
crsctl stop resource {resource_name [...] | -w "filter" | -all} [-n server_name
| -s server_pool] [-k cid] [-d did] [-env "env1=val1,env2=val2,..."]
[-begin | -end] [-i] [-f] [-l]
```

パラメータ

表E-17 `crsctl stop resource`コマンドのパラメータ

パラメータ	説明
<code>resource_name [...]</code>	<code>-w</code> または <code>-all</code> を使用するかわりに、停止対象である 1 つ以上のリソース名を空白区切りで指定できます。
<code>-w "filter"</code>	リソース名を指定したり <code>-all</code> を使用するかわりに、停止するリソースの数を制限するために CRSCTL によって使用される、リソース・フィルタを指定できます。フィルタは二重引用符(" ")で囲む必要があります。次にリソース・フィルタの例を示します。 ● <code>"TYPE == cluster_resource"</code>: このフィルタは、<code>cluster_resource</code> タイプのリソースのみを停止する

パラメータ	説明
	ように Oracle Clusterware を制限します。 ● "CHECK_INTERVAL > 10": このフィルタは、CHECK_INTERVAL リソース属性に 10 より大きい値を持つリソースを停止するように Oracle Clusterware を制限します。 ● "(CHECK_INTERVAL > 10) AND (NAME co 2)": このフィルタは、CHECK_INTERVAL リソース属性に 10 より大きい値を持ち、かつ、リソースの名前に数字 2 が含まれるリソースを停止するように Oracle Clusterware を制限します。 関連項目: 「フィルタ」を参照してください。
-all	リソース名を指定したり-wを使用するかわりに、このパラメータを使用して、クラスタ内のすべてのリソースを停止できます。
-n server_name -s server_pool	必要に応じて、停止するリソース・インスタンスが存在するサーバーの名前を指定したり、停止するリソース・インスタンスが存在するサーバー・プールの名前を指定できます。サーバーまたはサーバー・プールを指定しない場合、CRSCTL によって、リソースのすべてのインスタンスが停止されます。
-k cid	必要に応じて、リソースのカーディナリティ ID を指定できます。このパラメータを指定すると、指定したカーディナリティのリソース・インスタンスが CRSCTL によって停止されます。
-d did	リソースの程度 ID を指定します。このパラメータに 1 より大きい程度 ID を指定すると、この基準を満たすすべてのリソース・インスタンスが CRSCTL によって停止されます。
-env "env1=val1,env2=val2,..."	必要に応じて、-env コマンド・パラメータを使用して、1 つ以上のリソース・プロファイル属性値を上書きできます。複数の環境名/値ペアを指定する場合、各ペアをカンマで区切り、リスト全体を二重引用符(" ")で囲む必要があります。
-begin -end	必要に応じて、これらのパラメータのいずれかを指定して、透過的 HA アクションの開始または終了をそれぞれ実行できます。
-i	-i を指定すると、このコマンドの処理で Oracle Clusterware が

パラメータ	説明
	リソースまたはその依存リソースをロック解除するのを待機する必要がある場合、コマンドはエラーを戻します。コマンドが互いに影響しないように、Oracle Clusterware によってリソースまたは他のオブジェクトがロックされることがあります。
-f	停止対象のリソースに依存している他のリソースが実行中である場合に停止対象のリソースを強制的に停止するには、-f パラメータを指定します。このパラメータを使用すると、依存リソースは再配置または停止されます。
-l	stop コマンドが失敗した場合は、-l パラメータを使用してリソースを元の状態のままにします。

使用上のノート

- 停止対象のリソースを指定するには、次の3つの方法のいずれかを使用する必要があります。
 - 停止する1つ以上のリソースを名前で指定できます。
 - CRSCTLによって停止対象のリソースの照合に使用されるリソース・フィルタを指定できます。
 - -allパラメータを指定して、特定のサーバー上のすべてのリソースを停止できます。
- 指定したリソースを停止するには、それらのリソースに対する読取りおよび実行権限が必要です。
- oraで始まる名前を持つリソースはOracleリソースであるため、これらのリソースにはこのコマンドを使用しないでください。
- Oracleでは、Oracle Grid Infrastructure外部での管理対象アプリケーションの停止はサポートされていません

例

リソースを停止するには、次のコマンドを実行します。

```
$ crsctl stop resource -n node1 -k 2
```

crsctl add resourcegroup

リソース・グループ・タイプに基づいてリソース・グループを作成します。

このコマンドを使用して、リソースを追加できる空のリソース・グループを作成します。

構文

```
crsctl add resourcegroup group_name -type group_type
[[ -file file_path ] | [ -attr "attribute_name=
attribute_value" ] ] [ -f ] [ -i ]
```

パラメータ

表E-18 crsctl add resourcegroupコマンドのパラメータ

パラメータ	説明

パラメータ	説明
group_name	リソース・グループの名前を指定します。
-type group_type	リソース・グループのグループ・タイプを指定します。
-file file_path -attr "attribute_name=attribute_value[,...]"	リソース・グループを定義する属性のファイルまたはリストを指定します。 リソース・グループを定義する属性名/値ペアが行区切りで記載されているテキスト・ファイルのパス名(絶対または相対)。 -attr フラグの後に、1 つ以上の属性名/値ペアをカンマで区切り、二重引用符("")で囲んで指定できます。
-f	このパラメータを使用して、オンラインのリソース・グループを追加します。
-i	-i を指定すると、CRSCTL は、コマンドの処理で Oracle Clusterware がリソースまたはその依存リソースをロック解除するのを待機する必要がある場合、エラーを戻します。コマンドが互いに影響しないように、Oracle Clusterware によってリソースまたは他のオブジェクトがロックされることがあります。

crsctl check resourcegroup

指定されたグループのすべてのメンバー・リソースに対するチェック操作を開始します。

構文

```
crsctl check resourcegroup {group_name [...] | -w "filter"}
[-n server_name] [-k cid]
```

パラメータ

表E-19 crsctl check resourcegroup コマンドのパラメータ

パラメータ	説明
group_name	チェックするリソース・グループを(複数のリソース・グループの場合は空白区切りのリストで)指定します。
-w "filter"	または、Oracle Clusterware によってリソース・グループの照合に使用されるリソース・グループ・フィルタを、二重引用符("")で囲んで指

パラメータ	説明
	定できます。たとえば、<code>-w "TYPE=ora.database.type"</code>や<code>-w "NAME=cs1"</code>とします。 関連項目: フィルタの使用の詳細は、「フィルタ」を参照してください。
<code>-n server_name</code>	必要に応じて、チェックするリソース・グループが存在するサーバーの名前を指定できます。
<code>-k cid</code>	必要に応じて、リソース・グループのカーディナリティ ID を指定できます。このパラメータを指定すると、指定したカーディナリティのリソース・インスタンスが Oracle Clusterware によってチェックされます。

crsctl delete resourcegroup

1つ以上のリソース・グループをOracle Clusterwareから削除します。

構文

```
crsctl delete resourcegroup {"group_name [...] " | -w "filter"} [-f] [-i]
```

パラメータ

表E-20 crsctl delete resourcegroupコマンドのパラメータ

パラメータ	説明
<code>group_name</code>	削除対象の 1 つ以上のグループ名を指定します。複数のグループは空白で区切り、二重引用符("")で囲む必要があります。
<code>-w "filter"</code>	削除が評価されるリソースの数を制限するために Oracle Clusterware によって使用されるリソース・フィルタを指定します。フィルタは二重引用符("")で囲む必要があります。次にリソース・フィルタの例を示します。 ● <code>"TYPE == cluster_resource"</code>: このフィルタは、<code>cluster_resource</code> タイプのリソースのみを再配置するように Oracle Clusterware を制限します。 ● <code>"CHECK_INTERVAL > 10"</code>: このフィルタは、<code>CHECK_INTERVAL</code> リソース属性に 10 より大きい値を持つリソースを再配置するように Oracle Clusterware を制限します。 ● <code>"(CHECK_INTERVAL > 10) AND (NAME co 2)"</code>: このフィルタは、<code>CHECK_INTERVAL</code> リソース属性

パラメータ	説明
	に 10 より大きい値を持ち、かつ、リソースの名前に数字 2 が含まれるリソースを再配置するように Oracle Clusterware を制限します。 ノート: すべての演算子は空白で囲む必要があります。 関連項目: 演算子の詳細は、「フィルタ」を参照してください。
-f	このパラメータを使用して、オンラインのリソース・グループを削除します。
-i	-i を指定すると、CRSCTL は、コマンドの処理で Oracle Clusterware がリソースまたはその依存リソースをロック解除するのを待機する必要がある場合、エラーを戻します。コマンドが互いに影響しないように、Oracle Clusterware によってリソースまたは他のオブジェクトがロックされることがあります。

使用上のノート

- グループ名(1つまたは複数)またはリソース・フィルタを指定する必要があります。
- リソース・グループは、他のリソース・グループがそのリソース・グループに依存していない場合にのみ削除できます。

crsctl eval add resourcegroup

システムを変更せずに、リソース・グループを追加した場合の影響を予測します。

構文

```
crsctl eval add resourcegroup group_name -type group_type [-file file_path
| -attr "attribute_name=attribute_value[,...]" [-f]
```

パラメータ

表E-21 crsctl eval add resourcegroupコマンドのパラメータ

パラメータ	説明
group_name	リソース・グループの名前を指定します。
-type group_type	リソース・グループのグループ・タイプを指定します。
-file file_path -attr "attribute_name=attribute_value[,...]"	リソース・グループを定義する属性のファイルまたはリストを指定します。 リソース・グループを定義する属性名/値ペアが行区切りで記載されているテキスト・ファイルのパス名(絶対または相

パラメータ	説明
	対)。 -attr フラグの後に、1 つ以上の属性名/値ペアをカンマで区切り、二重引用符(" ")で囲んで指定できます。
-f	このパラメータを使用して、オンラインのリソース・グループを追加します。

crsctl eval fail resourcegroup

リソース障害の結果を予測します。

構文

```
crsctl eval fail resourcegroup {group_name | -w filter} [-n server]
[-env "attr=val[,...]" ]
```

パラメータ

表E-22 crsctl eval fail resourcegroup コマンドのパラメータ

パラメータ	説明
group_name	評価するリソース・グループの名前を指定します。
-w filter	特定のリソース・グループを指定するかわりに、CRSCTL によってリソース・グループの照合に使用されるリソース・グループ・フィルタを、二重引用符(" ")で囲んで指定できます。たとえば、-w "TYPE=ora.database.type"や-w "NAME=cs1"とします。 関連項目: フィルタの詳細は、「フィルタ」を参照してください。
-n server	必要に応じて、評価するリソース・グループが存在するサーバーの名前を指定できます。
-env "attr=val[,...]"	必要に応じて、-env コマンド・パラメータを使用して、1 つ以上のリソース・グループ・プロファイル属性値を上書きできます。複数の環境名/値ペアを指定する場合、各ペアをカンマで区切り、リスト全体を二重引用符(" ")で囲む必要があります。

crsctl eval relocate resourcegroup

システムを変更せずに、リソース・グループを再配置した場合の影響を予測します。

構文

```
crsctl eval relocate resourcegroup {group_name | -all} -s server | -w filter}
{-n server} [-f]
```

パラメータ

表E-23 crsctl eval relocate resourcegroupコマンドのパラメータ

パラメータ	説明
group_name	評価対象のリソース・グループの名前を指定します。
-all	特定のリソース・グループを指定するかわりに、このパラメータを使用し てすべてのリソース・グループの再配置を評価できます。
-s server	再配置するリソース・グループが存在するソース・サーバーの名前を指 定します。このパラメータは必須です。
-w filter	特定のリソース・グループを指定するかわりに、CRSCTL によってリソー ス・グループの照合に使用されるリソース・グループ・フィルタを、二重引 用符(" ")で囲んで指定できます。たとえば、-w "TYPE=ora.database.type"や-w "NAME=cs1"とします。 -s を使用しない場合、このパラメータは必須です。 関連項目：フィルタの詳細は、 「フィルタ」 を参照してください。
-n server	リソース・グループの再配置先サーバーの名前を指定します。このパラ メータは必須です。
-f	このパラメータを使用して、強制オプションを指定した場合の操作を評 価します。

crsctl eval start resourcegroup

システムを変更せずに、リソース・グループを起動した場合の影響を予測します。

構文

```
crsctl eval start resourcegroup {group_name | -w filter} [-n server] [-f]
```

パラメータ

表E-24 crsctl eval start resourcegroupコマンドのパラメータ

パラメータ	説明
group_name	評価するリソース・グループを(複数のリソース・グループの場合は空白

パラメータ	説明
	区切りのリストで)指定します。このパラメータは必須です。
-w filter	または、CRSCTL によってリソース・グループの照合に使用されるリソース・グループ・フィルタを、二重引用符("")で囲んで指定できます。たとえば、-w "TYPE=ora.database.type"や-w "NAME=cs1"とします。特定のリソース・グループを指定しない場合、このパラメータは必須です。 関連項目: フィルタの詳細は、「フィルタ」を参照してください。
-n server	評価するリソース・グループが存在するサーバーの名前を指定します。
-f	起動対象のリソース・グループ内のいずれかのリソースでサーバーに対する hard 起動依存性が定義されている場合、-f パラメータを使用してそのサーバー上のリソース・グループを評価します。

crsctl eval stop resourcegroup

システムを変更せずに、リソース・グループを停止した場合の影響を予測します。

構文

```
crsctl eval stop resourcegroup {group_name | -w filter} [-f]
```

パラメータ

表E-25 crsctl eval stop resourcegroupコマンドのパラメータ

パラメータ	説明
group_name	停止するリソース・グループを(複数のリソース・グループの場合は空白区切りのリストで)指定します。
-w "filter"	リソース・グループを指定するかわりに、CRSCTL によってリソース・グループの照合に使用されるリソース・グループ・フィルタを、二重引用符("")で囲んで指定できます。たとえば、-w "TYPE=ora.database.type"や-w "NAME=cs1"とします。 関連項目: フィルタの使用の詳細は、「フィルタ」を参照してください。
-f	リソース・グループ内の停止対象のいずれかのリソースが別のサーバーで実行中で、そのサーバーに対する hard 停止依存性が定義されている場合、-f パラメータを使用してそのリソースを停止します。

crsctl export resourcegroup

リソース・グループおよびそのメンバー・リソースの詳細な情報および属性をスクリプト・ファイルにエクスポートします。

また、このコマンドは、リソース・グループ・タイプおよびメンバー・リソース・タイプの構成情報をエクスポートします。別のクラスタでスクリプト・ファイルを実行して、リソース・グループ構成をインポートできます。

構文

```
crsctl export resourcegroup [group_name [...]] -file file_name
```

パラメータ

表E-26 crsctl export resourceコマンドのパラメータ

パラメータ	説明
group_name	エクスポートするリソース・グループを(複数のリソース・グループの場合は空白区切りのリストで)指定します。
-file file_name	グループをインポートするために実行する出力スクリプト・ファイルの名前を指定します。

crsctl modify resourcegroup

リソース・グループを変更して、属性を追加するか、既存の属性の値を更新します。

構文

```
crsctl modify resourcegroup group_name -attr "attribute_name=attribute_value" [-i] [-f] [-delete]
```

パラメータ

表E-27 crsctl modify resourcegroupコマンドのパラメータ

パラメータ	説明
group_name	変更するリソース・グループの名前を指定します。
-attr "attribute_name=attribute_value"	変更するリソース・グループの属性を次の 2 つの異なる方法で指定できます。 ● -attr フラグの後に、変更対象の 1 つ以上の属性名/値ペアをカンマで区切り、二重引用符("")で囲んで指定できます。たとえば: -attr "CHECK_INTERVAL=30, START_TIMEOUT=25" いくつかの属性は複数の値を持つことができます。そのような場合

パラメータ	説明
	は、空白で値を区切り、値のリストを一重引用符で囲みます。たとえば: <pre>-attr "SERVER_POOL_NAMES='ora.pool1 ora.pool2',START_TIMEOUT=25"</pre> ● または、特定のカーディナリティ値および特定の程度値を付加して特定のサーバーに存在するリソースの属性値を指定できます。この方法は特定のサーバーに何らかの方法で関連付けられるアプリケーションに役立ちます。-attr フラグの後の構文は次のとおりです。 <pre>attribute_name{@SERVERNAME(server_name) [@DEGREEID(did)] @CARDINALITYID(cid) [@DEGREEID(did)]}=attribute_value</pre> @SERVERNAME(server_name)構文を指定すると、指定した属性の属性値は、指定したサーバーに存在するリソースに限定されます。 または、@CARDINALITYID(cid)構文を指定すると、指定した属性の属性値は、特定のカーディナリティ ID(cid)を持つリソース・インスタンスに限定されます。 必要に応じて、@DEGREEID(did)構文を SERVERNAME または CARDINALITYID 構文のいずれか、または両方と組み合わせて、特定の DEGREE が設定されているリソースに属性値を限定することができます。 例: <pre>CHECK_INTERVAL@SERVERNAME(node1)=45 STOP_TIMEOUT@CARDINALITYID(2)=65 STOP_TIMEOUT@SERVERNAME(node1)@DEGREEID(2)=65 STOP_TIMEOUT@CARDINALITYID(3)@DEGREEID(2)=65</pre>
-i	-i を指定すると、このコマンドの処理で Oracle Clusterware がリソース・グループまたはその依存グループをロック解除するのを待機する必要がある場合、CRSCTL はエラーを戻します。コマンドが互いに影響しないように、Oracle Clusterware によってリソース・グループまたは他のオブジェクトがロックされることがあります。
-f	-f パラメータは次の場合に使用します。 ● 存在しないリソースへの hard 依存性がリソース・グループに設定されている場合

パラメータ	説明
	● 1 つ以上の hard 依存性に対する実行権限がリソース・グループ所有者にない場合 ● 変更によってサーバーが汎用プールに移動され、サーバーの移動を完了するためにリソースが停止または再配置される場合
-delete	-delete パラメータを使用すると、指定した属性が Oracle Clusterware によって削除されます。

使用上のノート

リソース・グループの名前を変更し、既存の属性を追加または変更できます。

crsctl relocate resourcegroup

オンライン・リソース・グループをソース・サーバーからターゲット・サーバーに再配置します。

このコマンドは、ソース・サーバー上の指定したリソース・グループをまず停止し、次にターゲット・サーバー上でそれらのグループを起動します。

構文

```
crsctl relocate resourcegroup {group_name [...] [-k cid] | -all | -w "filter"}
    [-env "attr=val[,...]" ] [-f] [-i]
```

パラメータ

表E-28 crsctl relocate resourcegroupコマンドのパラメータ

パラメータ	説明
group_name	再配置するリソース・グループを(複数のリソース・グループの場合は空白区切りのリストで)指定します。
-k cid	必要に応じて、リソース・グループのカーディナリティ ID を指定できます。このパラメータを指定すると、指定したカーディナリティのリソース・インスタンスが CRSCTL によって再配置されます。
-all	ソース・サーバー上のすべてのリソース・グループを再配置するには、このパラメータを指定します。
-w "filter"	CRSCTL によってリソースの照合に使用されるリソース・グループ・フィルタを、二重引用符(" ")で囲んで指定します。たとえば、-w "TYPE=ora.database.type"や-w "NAME=cs1"とします。

パラメータ	説明
	関連項目: フィルタの詳細は、 「フィルタ」 を参照してください。
-s source_server_name	再配置するリソース・グループが存在するソース・サーバーの名前を指定します。このパラメータは必須です。
-n target_server_name	必要に応じて、リソース・グループの再配置先サーバーの名前を指定できます。ターゲット・サーバーを指定しない場合、CRSCTL によって、各リソース・グループの属性プロファイルに従い、最適なサーバーにリソース・グループが再配置されます。
-env "attr=val[,...]"	必要に応じて、-env コマンド・パラメータを使用して、1 つ以上のリソース・グループ・プロファイル属性値を上書きできます。複数の環境名/値ペアを指定する場合、各ペアをカンマで区切り、リスト全体を二重引用符(" ")で囲む必要があります。
-f	このオプションを指定すると、指定したリソース・グループの再配置が強制されます。
-i	-i を指定すると、このコマンドの処理で Oracle Clusterware がリソースまたはその依存リソースをロック解除するのを待機する必要がある場合、コマンドはエラーを戻します。コマンドが互いに影響しないように、Oracle Clusterware によってリソースまたは他のオブジェクトがロックされることがあります。

例

node1で実行中のすべてのリソース・グループをnode2に再配置するには、次のコマンドを実行します。

```
$ crsctl relocate resourcegroup resGrp1 -s node1 -n node2
```

crsctl restart resourcegroup

リソース・グループが現在存在するサーバー上で、それらのグループを再起動します。

構文

```
crsctl restart resourcegroup {group_name [...] | -w "filter"} [-k cid]
    [-env "env1=val1,env2=val2,..."] [-f] [-i]
```

パラメータ

表E-29 crsctl restart resourcegroupコマンドのパラメータ

パラメータ	説明
group_name	再起動するリソース・グループを(複数のリソース・グループの場合は空

パラメータ	説明
	白区切りのリストで)指定します。
-w "filter"	Oracle Clusterware によってリソース・グループの照合に使用されるリソース・グループ・フィルタを、二重引用符(" ")で囲んで指定します。たとえば、-w "TYPE=ora.database.type"や-w "NAME=cs1"とします。 関連項目: フィルタの詳細は、 「フィルタ」 を参照してください。
-k cid	リソース・グループのカーディナリティ ID を指定します。このパラメータを指定すると、指定したカーディナリティのリソース・インスタンスが Oracle Clusterware によって再起動されます。
-env "env1=val1,env2=val2,..."	必要に応じて、-env コマンド・パラメータを使用して、1 つ以上のリソース・グループ・プロファイル属性値を上書きできます。複数の環境名/値ペアを指定する場合、各ペアをカンマで区切り、リスト全体を二重引用符(" ")で囲む必要があります。
-f	再起動するリソース・グループが存在するサーバー上でリソースが実行中で、別のリソースに対する hard 起動依存性がそのサーバーに定義されている場合、-f パラメータを指定して、そのリソースを再起動します。この場合には、強制パラメータを指定しないと、start コマンドは失敗します。
-i	-i を指定すると、このコマンドの処理で Oracle Clusterware がリソースまたはその依存リソースをロック解除するのを待機する必要がある場合、コマンドはエラーを戻します。コマンドが互いに影響しないように、Oracle Clusterware によってリソースまたは他のオブジェクトがロックされることがあります。

例

次のコマンドは、リソース・グループresGrp1を再起動します。

```
$ crsctl restart resourcegroup resGrp1
```

crsctl start resourcegroup

特定または事前構成済のサーバーやサーバー・プール上の1つ以上のリソース・グループを起動します。

CRSCTLは、hardまたはweak起動依存性が定義されているリソース・グループ、およびそのグループのすべての有効なメンバー・リソースを起動します。このコマンドは、CRSCTLが起動した各リソースおよびリソース・グループの進捗メッセージをレポートします。

構文

```
crsctl start resourcegroup {group_name [...] | -w "filter" | -all}  
  [-n server_name | -s server_pool] [-k cid] [-env "attr=val[,...]" ]  
  [-f] [-i] [-l]
```

パラメータ

表E-30 crsctl start resourcegroupコマンドのパラメータ

パラメータ	説明
group_name	起動するリソース・グループを(複数のリソース・グループの場合は空白区切りのリストで)指定します。
-w "filter"	CRSCTL によってリソースの照合に使用されるリソース・グループ・フィルタを、二重引用符("")で囲んで指定します。たとえば、-w "TYPE=ora.database.type"や-w "NAME=cs1"とします。 関連項目: フィルタの詳細は、 「フィルタ」 を参照してください。
-all	クラスタ上のすべてのリソース・グループを起動するには、このパラメータを指定します。
-n server_name	起動するリソース・グループが存在するサーバーの名前を指定します。サーバーを指定しない場合、CRSCTL によって、各リソースの属性プロファイルに従い、最適なサーバー上のリソースが起動されます。
-s server_pool	起動するリソース・グループが存在する 1 つのサーバー・プール名または空白区切りのサーバー・プールのリストを指定します。
-k cid	リソース・グループのカーディナリティ ID を指定します。このパラメータを指定すると、指定したカーディナリティのリソース・インスタンスが CRSCTL によって起動されます。
-env "attr=val[,...]"	必要に応じて、-env コマンド・パラメータを使用して、1 つ以上のリソース・グループ・プロファイル属性値を上書きできます。複数の環境名/値ペアを指定する場合、各ペアをカンマで区切り、リスト全体を二重引用符("")で囲む必要があります。
-f	起動対象のリソース・グループ内のいずれかのリソースでサーバーに対する hard 起動依存性が定義されている場合、-f パラメータを使用してそのサーバー上のリソース・グループを起動します。この場合には、強制パラメータを指定しないと、start コマンドは失敗します。

パラメータ	説明
-i	-i を指定すると、このコマンドの処理で Oracle Clusterware がリソースまたはその依存リソースをロック解除するのを待機する必要がある場合、コマンドはエラーを戻します。コマンドが互いに影響しないように、Oracle Clusterware によってリソースまたは他のオブジェクトがロックされることがあります。
-l	start コマンドが失敗した場合は、-l パラメータを使用してリソース・グループを元の状態のままにします。

例

サーバー・プールのリソース・グループを起動するには、次のコマンドを実行します。

```
$ crsctl start resourcegroup rg1 -s pool1
```

crsctl status resourcegroup

リソース・グループの状態を表示します。

構文

```
crsctl status resourcegroup [group_name [...] | -w "filter"]
[-p [-attr "attribute_name=attribute_value" [-noname]] | -v [-attr
"attribute_name=attribute_value" [-noname]]] | [-f] [-l] [-g] | [[-k cid |
-n server_name] [-e [-p | -v]]] [-s -k cid] | -noauto]]
```

パラメータ

表E-31 crsctl status resourcegroupコマンドのパラメータ

パラメータ	説明
group_name	問い合わせるリソース・グループを(複数のリソース・グループの場合は空白区切りのリストで)指定します。リソース・グループ名またはフィルタを指定しない場合、CRSCTL によって、すべてのリソース・グループのステータスが表示されます。
-w "filter"	Oracle Clusterware によってリソース・グループの照合に使用されるリソース・グループ・フィルタを、二重引用符(" ")で囲んで指定します。たとえば、-w "TYPE=ora.database.type"や-w "NAME=cs1"とします。 関連項目: フィルタの詳細は、 フィルタ を参照してください。
-p -attr "attribute_name=attribute_value" [-noname]	
-v [-attr	必要に応じて、このパラメータを指定して、特定のリソース・グループ

パラメータ	説明
"attribute_name=attribute_value" [-noname]]	のすべてのメンバー・リソースのステータスを表示できます。
-f	必要に応じて、このパラメータを指定して、リソース・グループのすべての構成を表示できます。
-l	
-g	
-k cid	リソース・グループのカーディナリティ ID を指定します。このパラメータを指定すると、指定したカーディナリティのリソース・グループのステータスが CRSCTL によって表示されます。
-n server_name	
-e	必要に応じて、このパラメータを指定して、automatic リソース・グループを除外できます。デフォルトでは、すべてのリソース・グループのステータスが表示されます。
-p -v	
-s -k cid	
-noauto	

例

次のコマンドは、node1に存在するリソース・グループrg1のステータスを示します。

```
$ crsctl status resourcegroup rg1 -n node1
```

crsctl stop resourcegroup

オプションで指定した事前構成済のサーバーやサーバー・プール上の1つ以上のリソース・グループを停止します。

CRSCTLによって、リソース・グループのすべてのメンバー・リソースが停止されます。停止対象のリソース・グループに対する停止依存性が、別のリソース・グループで定義されている場合は、-fを使用する必要があります。

構文

```
crsctl stop resourcegroup {group_name [...] | -w "filter" | -all}
    [-n server_name | -s server_pool] [-k cid] [-env "attr=val[,...]" ]
    [-f] [-i] [-l]
```

パラメータ

表E-32 crsctl stop resourcegroupコマンドのパラメータ

パラメータ	説明
group_name	-w または -all を使用するかわりに、停止するリソース・グループを

パラメータ	説明
	(複数のリソース・グループの場合は空白区切りのリストで)指定できます。
-w "filter"	リソース・グループ名を指定したり-allを使用するかわりに、CRSCTLによってリソース・グループの照合に使用されるリソース・グループ・フィルタを、二重引用符(" ")で囲んで指定できます。たとえば、-w "TYPE=ora.database.type"や-w "NAME=cs1"とします。 関連項目: フィルタの使用の詳細は、 「フィルタ」 を参照してください。
-all	リソース名を指定したり-wを使用するかわりに、このパラメータを指定して、クラスタ上のすべてのリソース・グループを停止できます。
-n server_name	停止するリソース・グループが存在するサーバーの名前を指定します。サーバーを指定しない場合、CRSCTLによって、各リソースの属性プロファイルに従い、最適なサーバー上のリソースが停止されます。
-s server_pool	停止するリソース・グループが存在する1つのサーバー・プール名または空白区切りのサーバー・プールのリストを指定します。
-k cid	リソース・グループのカーディナリティ ID を指定します。このパラメータを指定すると、指定したカーディナリティのリソース・インスタンスが Oracle Clusterware によって停止されます。
-env "attr=val[,...]"	必要に応じて、-env コマンド・パラメータを使用して、1つ以上のリソース・グループ・プロファイル属性値を上書きできます。複数の環境名/値ペアを指定する場合、各ペアをカンマで区切り、リスト全体を二重引用符(" ")で囲む必要があります。
-f	リソース・グループ内の停止対象のいずれかのリソースが別のサーバーで実行中で、そのサーバーに対する hard 停止依存性が定義されている場合、-f パラメータを使用してそのリソースを停止します。この場合には、強制パラメータを指定しないと、stop コマンドは失敗します。
-i	-i を指定すると、このコマンドの処理で Oracle Clusterware がリソースまたはその依存リソースをロック解除するのを待機する必要がある場合、コマンドはエラーを戻します。コマンドが互いに影響しないように、Oracle Clusterware によってリソースまたは他のオブジェクト

パラメータ	説明
	トがロックされることがあります。
-l	stop コマンドが失敗した場合にリソース・グループが元の状態に保たれるようにするには、-l パラメータを使用します。

例

次のコマンドは、リソース・グループrg1を停止します。

```
$ crsctl stop resourcegroup rg1
```

crsctl add resourcegroup type

新しいリソース・グループ・タイプをOracle Clusterwareに登録します。

複数のリソースで構成されるアプリケーションでは、リソース・グループ・タイプを定義して、個々のIDを取得し、新しいグループレベルの属性も作成することをお勧めします。

構文

```
crsctl add resourcegroup type group_type_name -basetype base_group_type
{-file file_path | -attr "attribute_name=attribute_value"} [-i]
```

パラメータ

表E-33 crsctl add resourcegroup typeコマンドのパラメータ

パラメータ	説明
group_type_name	リソース・グループ・タイプの名前を指定します。
-basetype base_group_type	リソース・グループ・タイプのベース・グループ・タイプを指定します。
-file file_path -attr "attribute_name=attribute_value"	リソース・グループ・タイプを定義する属性のファイルまたはリストを指定する必要があります。 リソース・グループ・タイプを定義する属性名/値ペアが行区切りで記載されているテキスト・ファイルのパス名(絶対または相対)。 -attr フラグの後に、1 つ以上の属性名/値ペアをカンマで区切り、二重引用符("")で囲んで指定できます。
-i	-i を指定すると、CRSCTL は、コマンドの処理で Oracle Clusterware がリソースまたはその依存リソースをロック解除するのを待機する必要がある場合、エラーを戻します。コ

パラメータ	説明
	マンドが互いに影響しないように、Oracle Clusterware によってリソースまたは他のオブジェクトがロックされることがあります。

crsctl delete resourcegrouptype

1つ以上のリソース・グループ・タイプをOracle Clusterwareから削除します。

構文

```
crsctl delete resourcegrouptype {"group_type_name [...]"} | -w "filter" [-i]
```

パラメータ

表E-34 crsctl delete resourcegrouptypeコマンドのパラメータ

パラメータ	説明
group_type_name	削除対象の 1 つ以上のグループ・タイプ名を指定します。複数のグループ・タイプは空白で区切り、二重引用符("")で囲む必要があります。
-w "filter"	削除が評価されるリソースの数を制限するために Oracle Clusterware によって使用されるリソース・フィルタを指定します。フィルタは二重引用符("")で囲む必要があります。次にリソース・フィルタの例を示します。 ● "TYPE == cluster_resource": このフィルタは、cluster_resource タイプのリソースのみを再配置するように Oracle Clusterware を制限します。 ● "CHECK_INTERVAL > 10": このフィルタは、CHECK_INTERVAL リソース属性に 10 より大きい値を持つリソースを再配置するように Oracle Clusterware を制限します。 ● "(CHECK_INTERVAL > 10) AND (NAME co 2)": このフィルタは、CHECK_INTERVAL リソース属性に 10 より大きい値を持ち、かつ、リソースの名前に数字 2 が含まれるリソースを再配置するように Oracle Clusterware を制限します。

ノート: すべての演算子は空白で囲む必要があります。

関連項目: 演算子の詳細は、[「フィルタ」](#)を参照してください。

パラメータ	説明
-i	-i を指定すると、CRSCTL は、コマンドの処理で Oracle Clusterware がリソースまたはその依存リソースをロック解除するのを待機する必要がある場合、エラーを戻します。コマンドが互いに影響しないように、Oracle Clusterware によってリソースまたは他のオブジェクトがロックされることがあります。

crsctl modify resourcegrouptype

リソース・グループ・タイプを変更して、属性を追加するか、既存の属性の値を更新します。

構文

```
crsctl modify resourcegrouptype group_type_name -attr
"attribute_name=attribute_value"
```

使用上のノート

リソース・グループ・タイプの名前を変更し、既存の属性を追加または変更できます。

crsctl get tracefileopts

特定のモジュールのトレース設定を戻します。

構文

```
crsctl get tracefileopts module_name
```

使用上のノート

トレース情報を取得するモジュールを次の中から1つ指定します。

- mdns: マルチキャスト・ドメイン・ネーム・サーバー
- gpnp: グリッド・プラグ・アンド・プレイ・サービス
- css: クラスタ同期サービス
- crf: クラスタ状態モニター
- crs: クラスタ・レディ・サービス
- ctss: クラスタ時刻同期化サービス
- evm: Event Manager
- gipc: グリッド・プロセス間通信

crsctl set tracefileopts

特定のモジュールのトレース・パラメータを設定します。

構文

```
crsctl set tracefileopts module_name [-filesize file_size[K|k|M|m|G|g]]
[-numsegments number_of_segments]
```

パラメータ

表E-35 crsctl set tracefileoptsコマンドのパラメータ

パラメータ	説明
module_name	トレース情報を設定するための次のモジュールをどれか 1 つ指定します。 - mdns: マルチキャスト・ドメイン・ネーム・サーバー - gpnv: グリッド・プラグ・アンド・プレイ・サービス - css: クラスタ同期サービス - crf: クラスタ状態モニター - crs: クラスタ・レディ・サービス - ctss: クラスタ時刻同期化サービス - evm: Event Manager - gipc: グリッド・プロセス間通信
-filesize file_size	単一トレース・ファイルのサイズを KB 単位(K k)、MB 単位(M m)または GB 単位(G g)で指定します。
-numsegments number_of_segments	Oracle Clusterware でトレース・ファイルをローテーションするために、より古いファイルをいくつ保存するかを指定します。

crsctl add type

Oracle Clusterwareのリソース・タイプを作成します。

構文

```
crsctl add type type_name -basetype base_type_name {-attr "ATTRIBUTE=attribute_name
| -file file_path,TYPE={string | int}
[,DEFAULT_VALUE=default_value][,FLAGS=typeFlags"]} [-i]
```

パラメータ

表E-36 crsctl add typeコマンドのパラメータ

パラメータ	説明
type_name	リソース・タイプの名前を、xxx.yyy.type の形式で指定します。リソース・タイプの名前は一意である必要があり、リソース・タイプの登録後は変更できません。
-basetype base_type_name	ベースになる既存のタイプの名前。作成するリソース・タイプのベースになるリソース・タイプは、local_resource または cluster_resource のいずれかである必要があります。
-attr "ATTRIBUTE=attribute_name	-attr 引数を使用してリソース・タイプ属性を指定できます。各タイプ属性定義には最大 4 つのタイプ属性キーワードを含めることができ、それらはここに示す順序で表示される必要があります。

パラメータ	説明
	1 つ以上のリソース・タイプ属性の説明をカンマで区切り、二重引用符(" ")で囲んで入力します。属性のキーワードは次のとおりです。 1. ATTRIBUTE: 属性の名前を指定します。名前では大/小文字が区別され、空白は使用できません。 2. TYPE: 属性のタイプが integer か string かを指定します。 3. DEFAULT_VALUE: (オプション)属性が必須である場合、デフォルト値は不要です。必須でない属性の場合は、このリソース・タイプに基づいてリソースを作成する際に Oracle Clusterware によって使用されるデフォルト値を指定する必要があります。 4. FLAGS: (オプション)次のタイプから 1 つ以上のタイプを指定し、縦線()で区切ります。 CONFIG: このリソース・タイプのリソースを登録した後に、この属性を構成できます。 READONLY: このリソース・タイプのリソースを登録した後は、この属性を変更できません。 REQUIRED: このリソース・タイプに基づきリソースを作成するときには、この属性の名前および値を指定する必要があります。この属性は必須でないと指定した場合、Oracle Clusterware ではこの属性に指定されたデフォルト値が使用されます。 HOTMOD: このタイプのリソースの属性値を変更した場合、変更はただちに適用され、リソースを再起動する必要があります。 複数の -attr 引数を使用して、このリソース・タイプに複数の引数を定義することはできません。かわりに、-attr フラグの後の二重引用符の中に複数のタイプを指定できます。たとえば: <pre>"ATTRIBUTE=F00, TYPE=integer, DEFAULT_VALUE=0, FLAGS=REQUIRED HOTMOD, ATTRIBUTE=BAR, TYPE=string"</pre> 前述の例では、2 つのタイプ属性(F00 および BAR)を

パラメータ	説明
	定義しています。ATTRIBUTE キーワードを指定すると、前のタイプ属性(存在する場合)が終わり、新しいタイプ属性が始まります。
-file file_path	リソース・タイプを定義するリソース・タイプのキーワード/値ペアが行区切りで記載されているテキスト・ファイルのパス名(絶対または相対)。ファイルの内容の例は、次のとおりです。 <pre>ATTRIBUTE=FOO TYPE=integer DEFAULT_VALUE=0 FLAGS=REQUIRED ATTRIBUTE=BAR TYPE=string</pre> ノート：キーワードは、ATTRIBUTE、TYPE、DEFAULT_VALUE および FLAGS の順序である必要があります。ATTRIBUTE キーワードを指定すると、前のタイプ属性(存在する場合)が終わり、新しいタイプ属性が始まります。 前述の例では、2 つのタイプ属性(FOO および BAR)を定義しています。 ノート：すべての演算子は空白で囲む必要があります。 関連項目：演算子の詳細は、「フィルタ」を参照してください。
-i	-i を指定すると、Oracle Clusterware でリクエストをすぐに処理できない場合にコマンドが失敗します。

関連項目:

リソース・タイプの詳細は、「[Oracle クラスタウェアのリソース・タイプ](#)」を参照してください。

使用上のノート

- type_name パラメータおよび base_type_name パラメータのどちらも必須です。
- タイプ情報を含むファイルを指定するか、またはコマンドラインにタイプ情報を指定することができます。
- ora で始まる名前を持つリソース・タイプは Oracle リソース・タイプであるため、これらのリソース・タイプにはこのコマンドを使用しないでください。
- ベースとなるタイプに対する読取り権限が必要です。

例

demoActionScript のリソース・タイプを作成するには、次のコマンドを実行します。

```
# crsctl add type test_type1 -basetype cluster_resource -attr
"ATTRIBUTE=FOO,TYPE=integer,DEFAULT_VALUE=0"
```

crsctl delete type

Oracle Clusterwareの構成からリソース・タイプを削除します。

構文

```
crsctl delete type type_name [-i]
```

使用上のノート

- type_nameパラメータは必須です。複数のタイプを空白で区切って指定できます。
- -iを指定すると、Oracle Clusterwareでリクエストをすぐに処理できない場合にコマンドが失敗します。
- oraで始まる名前を持つリソース・タイプはOracleリソース・タイプであるため、これらのリソース・タイプにはこのコマンドを使用しないでください。

例

2つのリソース・タイプを削除するには、そのリソース・タイプに対する書込み権限を持つユーザーとして次のコマンドを実行します。

```
$ crsctl delete type test_type1 test_type2
```

crsctl getperm type

特定のリソース・タイプの権限を取得します。

構文

```
crsctl getperm type resource_type [-u user_name] | [-g group_name]
```

パラメータ

表E-37 crsctl getperm typeコマンドのパラメータ

パラメータ	説明
resource_type	権限を取得するリソース・タイプを指定します。
-u user_name	-u を指定すると、Oracle Clusterware では特定のユーザーの権限が取得されます。
-g group_name	-g を指定すると、Oracle Clusterware では特定のグループの権限が取得されます。

使用上のノート

- resource_typeパラメータは必須です。
- oraで始まる名前を持つリソース・タイプはOracleリソース・タイプであるため、これらのリソース・タイプにはこのコマンドを使用しないでください。

関連項目:

リソース・タイプの詳細は、「[Oracleクラスタウェアのリソース・タイプ](#)」を参照してください。

例

このコマンドによって、次のような出力が返されます。

```
$ crsctl getperm type app.appvip.type
Name: app.appvip.type
owner:root:rwx,pgrp:oinstall:rwx,other::r--
```

crsctl modify type

既存のリソース・タイプを変更します。

構文

```
crsctl modify type type_name -attr "ATTRIBUTE=attribute_name,TYPE={string | int}
[,DEFAULT_VALUE=default_value [,FLAGS=[READONLY][| REQUIRED]]" [-i] [-f]]
```

パラメータ

表E-38 crsctl modify typeコマンドのパラメータ

パラメータ	説明
type_name	変更するリソース・タイプの名前を指定します。複数のリソース・タイプの名前は変更できません。
-attr	次のリソース・タイプ・キーワードを変更できます。 ● TYPE ● DEFAULT_VALUE ● FLAGS ノート: ATTRIBUTE キーワードは、指定する必要がありますが、変更することはできません。 関連項目: これらのキーワードの詳細は、「crsctl add type」を参照してください
-i	-i パラメータを指定すると、Oracle Clusterware でリクエストをすぐに処理できない場合にコマンドが失敗します。

使用上のノート

- type_nameパラメータは必須です。
- oraで始まる名前を持つリソース・タイプはOracleリソース・タイプであるため、これらのリソース・タイプにはこのコマンドを使用しないでください。

関連項目:

リソース・タイプの詳細は、「[Oracle クラスウェアのリソース・タイプ](#)」を参照してください。

例

次の例では、2つのタイプ属性(F00およびBAR)を変更します。

```
$ crsctl modify type myType.type -attr "ATTRIBUTE=F00,DEFAULT_VALUE=0  
ATTRIBUTE=BAR,DEFAULT_VALUE=baz"
```

crsctl setperm type

特定のリソース・タイプのリソースの権限を設定します。

構文

```
crsctl setperm type resource_type_name {-u acl_string | -x acl_string |  
-o user_name | -g group_name}
```

パラメータ

表E-39 crsctl setperm typeコマンドのパラメータ

パラメータ	説明
resource_type_name	権限を設定するリソース・タイプの名前を指定します。
-u -x -o -g	リソース・タイプには次のパラメータのうち 1 つのみを指定できます。 ● -u acl_string: リソース・タイプのアクセス制御リスト(ACL)を更新できます。 ● -x acl_string: リソース・タイプの ACL を削除できます。 ● -o user_name: ユーザー名を入力することによってリソース・タイプの所有者を変更できます。 ● -g group_name: グループ名を入力することによってリソース・タイプのプライマリ・グループを変更できます。 ユーザー、グループ、他の ACL 文字列は次のように指定します。 <pre>user:user_name[:readPermwritePermexecPerm] group:group_name[:readPermwritePermexecPerm] other[:readPermwritePermexecPerm]</pre> ● user: ユーザー ACL ● group: グループ ACL ● other: その他の ACL ● readPerm: リソース・タイプに対する読取り権限。文

パラメータ	説明
	字 r はユーザー、グループ、その他の読取り権限を付与し、マイナス記号(-)は読取り権限を付与しません。 ● writePerm: リソース・タイプに対する書込み権限。文字 w はユーザー、グループ、その他の書込み権限を付与し、マイナス記号(-)は書込み権限を付与しません。 ● execPerm: リソース・タイプに対する実行権限。文字 x はユーザー、グループ、その他の実行権限を付与し、マイナス記号(-)は実行権限を付与しません。

使用上のノート

- resource_type_nameパラメータは必須です。
- 権限を設定するには、指定したリソースに対する読取りおよび書込み権限が必要です。
- oraで始まる名前を持つリソース・タイプはOracleリソース・タイプであるため、これらのリソース・タイプにはこのコマンドを使用しないでください。

例

リソース・タイプに対する読取り、書込み、および実行権限をユーザーScottに付与するには、次のコマンドを実行します。

```
$ crsctl setperm type resType -u user:scott:rwX
```

crsctl status type

1つ以上の特定のリソース・タイプの構成情報を取得します。

構文

```
crsctl status type [resource_type_name [...] | -w "filter"] [-g] [-p] [-f] [-r]
```

パラメータ

表E-40 crsctl status typeコマンドのパラメータ

パラメータ	説明
resource_type_name [...] -w "filter"	ステータスのチェック対象である 1 つ以上のリソース・タイプ名を空白区切りで指定します。 または、CRSCTL によってリソース・タイプの照合に使用されるリソース・タイプ・フィルタを、二重引用符("")で囲んで指定できます。たとえば、 -w "TYPE = ora.database.type"のようにします。 関連項目: 「フィルタ」を参照してください。
[-g] [-p] [-f] [-r]	CRSCTL で特定のリソース・タイプのステータスをチェックする場合のオ

パラメータ	説明
	ブションとして、次のパラメータを指定できます。 ● -g: 指定したリソース・タイプが登録されているかどうかをチェックする場合にこのパラメータを使用します。 ● -p: 指定したリソース・タイプの静的な構成を表示する場合にこのパラメータを使用します。 ● -f: リソース・タイプのすべての構成を表示する場合にこのパラメータを使用します。 ● -r: すべてのベース・タイプを出力する場合にこのパラメータを使用します。

使用上のノート

resource_type_nameパラメータまたはフィルタは必須です。

例

crsctl status typeコマンドでは次のような出力が戻されます。

```
$ crsctl status type ora.network.type
TYPE_NAME=ora.network.type
BASE_TYPE=ora.local_resource.type
```

Oracle RAC環境のCRSCTLコマンド

次のコマンドは、Oracle RAC環境でのみ使用できます。

この項に示すコマンドは、次のもので構成されるOracle RAC環境のOracle Clusterwareスタックを管理します。

- Oracle Clusterware、メンバー・ノードおよびサーバー・プール
- Oracle ASM(インストールされている場合)
- クラスタ同期サービス
- クラスタ時刻同期化サービス

crsctl request action

特定のリソースに対して特定のアクションを実行します。

構文

```
crsctl request action action_name {-r resource_name [...] | -w "filter"}
[-env "env1=val1,env2=val2,..."] [-i]
```

パラメータ

表E-41 crsctl request actionコマンドのパラメータ

パラメータ	説明
action_name	実行するアクションの名前を指定します。特定のリソースでサポートされているアクションは、そのリソースの ACTIONS リソース属性に示されます。
-r resource_name [...]	特定のリソースを指定します。複数のリソース名は空白で区切る必要があります。
-w "filter"	リソース名を指定するかわりに、Oracle Clusterware によってアクションが実行されるリソースの数を制限するために使用されるリソース・フィルタを指定できます。次にリソース・フィルタの例を示します。 ● TYPE == cluster_resource: このフィルタは、cluster_resource タイプのリソースのみでアクションを実行するように Oracle Clusterware を制限します。 ● CHECK_INTERVAL > 10: このフィルタは、CHECK_INTERVAL リソース属性に 10 より大きい値を持つリソースのみのアクションを実行するように、Oracle Clusterware を制限します。 ● (CHECK_INTERVAL > 10) AND (NAME co 2): このフィルタは、CHECK_INTERVAL リソース属性に 10 より大きい値を持ち、リソースの名前に数字 2 が含まれるリソースのみでアクションを実行するように、Oracle Clusterware を制限します。 関連項目: 「フィルタ」を参照してください。
-env "env1=val1, env2=val2, ..."	必要に応じて、-env コマンド・パラメータを使用して、1 つ以上のリソース・プロファイル属性値を上書きできます。複数の環境名/値ペアを指定する場合、各ペアをカンマで区切り、リスト全体を二重引用符(" ")で囲む必要があります。
-i	-i を指定すると、Oracle Clusterware でリクエストをすぐに処理できない場合にコマンドが失敗します。

使用上のノート

action_nameはサポートされたアクションで、エージェントおよびACTIONSリソース属性の両方で定義されます。ユーザーには、このアクションを実行する権限が付与されている必要があります。権限は、ACTIONSリソース属性で設定できます。

例

特定のリソースのアプリケーション固有のエージェント内で指定されたアクションを開始するには:

```
$ crsctl request action action1 -r res1
```

crsctl add category

サーバー・カテゴリを追加します。

構文

```
crsctl add category category_name [-attr "attr_name=attr_value  
[,attr_name=attr_value[,...]]" [-i]
```

パラメータ

表E-42 crsctl add categoryコマンドのパラメータ

パラメータ	説明
category_name	追加するサーバー・カテゴリの名前を指定します。
-attr "attr_name=attr_value[,...]"	前に -attr フラグを付けて、追加するカテゴリ属性の名前/値ペアを指定します。 ノート: attr_name および attr_value パラメータは、二重引用符(" ")で囲み、名前と値の複数のペアをカンマで区切る必要があります。たとえば: <pre>-attr "EXPRESSION='(CPU_COUNT > 2) AND (MEMORY_SIZE > 2048)'"</pre> 関連項目: ● このコマンドで利用できる属性のリストについては、表 3-4 を参照してください。 ● 演算子の詳細は、「フィルタ」を参照してください。
-i	-i を指定すると、Oracle Clusterware でリクエストをすぐに処理できない場合にコマンドが失敗します。

使用上のノート

属性名/値ペアの属性値にカンマが含まれる場合、値は一重引用符(' ')で囲む必要があります。たとえば:

```
"START_DEPENDENCIES='hard(res1,res2,res3)'"
```

例

サーバー・カテゴリを追加するには、次のコマンドを実行します。

```
$ crsctl add category cat1 -attr "EXPRESSION='(CPU_COUNT > 2) AND (MEMORY_SIZE > 2048)'"
```

crsctl delete category

サーバー・カテゴリを削除します。

構文

```
crsctl delete category category_name [category_name [...]] [-i]
```

使用上のノート

- 削除するサーバー・カテゴリの名前または空白区切りのサーバー・カテゴリのリストを指定します。
- -iを指定すると、Oracle Clusterwareでリクエストをすぐに処理できない場合にコマンドが失敗します。

例

サーバー・カテゴリを削除するには、次のコマンドを実行します。

```
$ crsctl delete category blue_server -i
```

crsctl modify category

既存のサーバー・カテゴリを変更します。

構文

```
crsctl modify category category_name [-attr "attr_name=attr_value  
[,attr_name=attr_value[,...]]"] [-i] [-f]
```

パラメータ

表E-43 crsctl modify categoryコマンドのパラメータ

パラメータ	説明
category_name	変更するサーバー・カテゴリの名前を指定します。
-attr "attr_name=attr_value[,...]"	前に -attr フラグを付けて、変更するカテゴリ属性の名前/値ペアを指定します。 ノート: attr_name および attr_value パラメータは、二重引用符(" ")で囲み、名前と値の複数のペアをカンマで区切る必要があります。たとえば: <pre>"ACL='owner:st-cdc¥cdctest:rwx,pgrp:---', ACTIVE_CSS_ROLE=hub"</pre> 関連項目: ● このコマンドで利用できる属性のリストについては、表 3-4 を参照してください。 ● 演算子の詳細は、「フィルタ」を参照してくだ

パラメータ	説明
	さい。
-i	-i を指定すると、Oracle Clusterware でリクエストをすぐに処理できない場合にコマンドが失敗します。
-f	操作を強制するには、このパラメータを使用します。

使用上のノート

属性名/値ペアの属性値にカンマが含まれる場合、値は一重引用符(' ')で囲む必要があります。たとえば:

```
"START_DEPENDENCIES=s'hard(res1,res2,res3)'"
```

例

サーバー・カテゴリを変更するには、次のコマンドを実行します。

```
$ crsctl modify category blue_server -attr "EXPRESSION=(LOCATION=hub)"
```

crsctl status category

サーバー・カテゴリに関する情報を取得します。

構文

```
crsctl status category {category_name [category_name [...]] | [-w "filter" |  
-server server_name]}
```

パラメータ

表E-44 crsctl status categoryコマンドのパラメータ

パラメータ	説明
category_name [...]	ステータスを取得するサーバー・カテゴリの名前またはサーバー・カテゴリの空白区切りリストを指定します。
-w "filter"	または、前に -w フラグを付けて、カテゴリ・フィルタを指定できます。 関連項目: 「 フィルタ 」を参照してください。
-server server_name	または、特定のサーバーを指定して、そのサーバーに一致するすべてのカテゴリを表示できます。

例

フィルタを使用してサーバー・カテゴリのステータスを取得するには、次のコマンドを実行します。

```
$ crsctl stat category -w "ACTIVE_CSS_ROLE = hub"
```

```

NAME=my_category_i
ACL=owner:scott:rw, pgrp:svrtech:rw, other::r--
ACTIVE_CSS_ROLE = hub
EXPRESSION=(CPU_COUNT > 3)

```

サーバーごとにサーバー・カテゴリのステータスを取得するには、次のコマンドを実行します。

```

$ crsctl stat category -server node1
NAME=my_category
ACL=owner:scott:rw, pgrp:svrtech:rw, other::r--
ACTIVE_CSS_ROLE = hub
EXPRESSION=

```

ASMを含むすべてのリソースを出力するには:

```

$ crsctl stat res -w "NAME co asm" -t
-----
Name      Target    State    Server                State details
-----
Cluster Resources
-----
ora.asm(ora.asmgroup)
1         ONLINE   ONLINE  stbm000042-vm1       Started, STABLE
2         ONLINE   ONLINE  stbm000042-vm2       Started, STABLE
3         ONLINE   ONLINE  stbm000042-vm3       Started, STABLE
ora.asmnet1.asmnetwork(ora.asmgroup)
1         ONLINE   ONLINE  stbm000042-vm1       STABLE
2         ONLINE   ONLINE  stbm000042-vm2       STABLE
3         ONLINE   ONLINE  stbm000042-vm3       STABLE
-----

```

dbを含むすべてのリソースを出力するには:

```

$ crsctl stat res -w "NAME co db" -t
-----
Name      Target    State    Server                State details
-----
Cluster Resources
-----
ora.cdb19.db
1         ONLINE   ONLINE  stbm000042-vm1
Open, HOME=/u01/app/oracle/product/19c/dbhome, STABLE
2         ONLINE   ONLINE  stbm000042-vm2
Open, HOME=/u01/app/oracle/product/19c/dbhome, STABLE
3         ONLINE   ONLINE  stbm000042-vm3
Open, HOME=/u01/app/oracle/product/19c/dbhome, STABLE
4         ONLINE   ONLINE  stbm000042-vm4
Open, HOME=/u01/app/oracle/product/19c/dbhome, STABLE
-----

```

dbを含み、その状態がONLINEであるすべてのリソースを出力するには:

```

crsctl stat res -w "(NAME en db) AND (STATE = ONLINE)" -t
-----
Name      Target    State    Server                State details
-----
Cluster Resources
-----
ora.cdb19.db
1         ONLINE   ONLINE  stbm000042-vm1
Open, HOME=/u01/app/oracle/product/19c/dbhome, STABLE
2         ONLINE   ONLINE  stbm000042-vm2
Open, HOME=/u01/app/oracle/product/19c/dbhome, STABLE
3         ONLINE   ONLINE  stbm000042-vm3
Open, HOME=/u01/app/oracle/product/19c/dbhome, STABLE
-----

```


crsctl check cluster

クラスタ内の任意のノードからOracle Clusterwareスタックのステータスをチェックします。

構文

```
crsctl check cluster [-all | [-n server_name [...]]]
```

使用上のノート

- -allパラメータを使用してクラスタ内のすべてのノードでOracle Clusterwareスタックのステータスをチェックするか、1つ以上のノードを空白区切りで指定できます。どちらのパラメータも指定しない場合、Oracle Clusterwareはローカル・サーバーのOracle Clusterwareスタックのステータスをチェックします。
- クラスタ内の任意のノードでこのクラスタ対応コマンドを使用できます。

例

crsctl check clusterコマンドでは次のような出力が戻されます。

```
$ crsctl check cluster -all
*****
node1:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
*****
node2:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
*****
```

crsctl start cluster

クラスタ内の任意のノードで実行されている場合に、Oracle Clusterwareスタックを起動します。

構文

```
crsctl start cluster [[-all] | [-n server [...]] | -hub | -w "filter"]
```

パラメータ

表E-45 crsctl start clusterコマンドのパラメータ

パラメータ	説明
-all	クラスタ内のすべてのサーバーで Oracle Clusterware スタックを起動するには、このパラメータを使用します。
-n server [...]	必要に応じて、クラスタ内の指定した 1 つ以上のサーバー(複数のサーバー名は空白で区切る)を指定するか、または-all も-n も指定しない場合はローカル・サーバーを指定できま

パラメータ	説明
	す。
- hub	必要に応じて、- hub を指定して、クラスタのハブ・ノードで Oracle Clusterware スタックを起動できます。
-w "filter"	必要に応じて、このパラメータを使用して、 "ACTIVE_CSS_ROLE = hub"などのフィルタに一致するサーバーのみを起動できます。 関連項目: フィルタの詳細は、 「フィルタ」 を参照してください。

crsctl stop cluster

クラスタ内の任意のノードでコマンドを実行するときに、クラスタ内のすべてのサーバー上または特定のサーバー上のOracle Clusterwareスタックを停止します。

構文

```
crsctl stop cluster [[-all] | [-n server [...]] | -hub | -w "filter"] [-f]
```

パラメータ

表E-46 crsctl stop clusterコマンドのパラメータ

パラメータ	説明
-all	クラスタ内のすべてのサーバーで Oracle Clusterware スタックを起動するには、このパラメータを使用します。
-n server [...]	必要に応じて、クラスタ内の指定した 1 つ以上のサーバー(複数のサーバー名は空白で区切る)を指定するか、または-all も-n も指定しない場合はローカル・サーバーを指定できます。
- hub	必要に応じて、- hub を指定して、クラスタの特定のハブ・ノードで Oracle Clusterware スタックを停止できます。
-w "filter"	必要に応じて、このパラメータを使用して、 "ACTIVE_CSS_ROLE = hub"などのフィルタに一致するサーバーのみを停止できます。 関連項目: フィルタの詳細は、 「フィルタ」 を参照してください。

使用上のノート

- このコマンドは、rootで実行する必要があります。
- -allを指定しない場合、または空白区切りの1つ以上のサーバー名を指定しない場合、Oracle Clusterwareによってローカル・サーバーのOracle Clusterwareスタックが停止されます。
- クラスタ内の任意のノードでこのクラスタ対応コマンドを使用できます。
- このコマンドでは、Oracle Clusterwareスタックの停止試行中、Oracle Clusterwareによって管理されるリソースの正常な停止が試行されます。

Oracle Clusterwareによって管理される任意のリソースが、`crsctl stop cluster`コマンドの実行後も実行されている場合、コマンドは失敗します。すべてのリソースを無条件で停止し、Oracle Clusterwareスタックを停止するには、`-f`パラメータを使用します。

- すべてのノードまたはリストされたノードでOracle Clusterwareを停止する場合は、`crsctl stop cluster`コマンドを使用しますが、これは、このコマンドを使用すると、Oracle Clusterwareスタックが特定のサーバーで停止される前に特定のリソースがクラスタ内の他のサーバーに再配置されないためです。1つ以上のノードでOracle高可用性サービスを停止する必要がある場合は、`crsctl stop cluster`コマンドが完了するまで待機してから、必要に応じて、特定のノードで[crsctl stop crs](#)コマンドを実行します。

例

特定のサーバー上のOracle Clusterwareスタックを停止するには、次のコマンドを実行します。

```
# crsctl stop cluster -n node1
```

crsctl get cluster class

クラスタ・クラス情報を取得します。

構文

```
crsctl get cluster class
```

使用上のノート

このコマンドにパラメータはありません。

crsctl get cluster configuration

クラスタ構成情報を表示します。

構文

```
crsctl get cluster configuration
```

使用上のノート

このコマンドにパラメータはありません。

crsctl set cluster disabledtlsciphersuite

ノード間通信の保護に使用される、無効化された暗号スイートのリストとの間で、暗号スイートを追加または削除します。

構文

```
crsctl set cluster disabledtlsciphersuite add | delete cipher_suite_name
```

使用上のノート

addまたはdelete動詞を使用して必要な操作を実行し、追加または削除する暗号スイートの名前を指定します。

crsctl get cluster extended

クラスタが拡張クラスタであるかどうかを判別します。

構文

```
crsctl get cluster extended
```

使用上のノート

このコマンドにパラメータはありません。

例

このコマンドによって、次のような出力が返されます。

```
$ crsctl get cluster extended
CRS-XXXX: The cluster is 'EXTENDED'
$ crsctl get cluster extended
CRS-XXXX: The cluster is 'NOT EXTENDED'
```

crsctl get cluster hubsize

Oracle Flex Cluster内のハブ・ノードの値を取得します。

構文

```
crsctl get cluster hubsize
```

例

このコマンドによって、次のような出力が返されます。

```
CRS-4950: Current hubsize parameter value is 32
```

crsctl set cluster hubsize

Oracle Flex Clusterのハブ・ノードの最大数を設定します。

構文

```
crsctl set cluster hubsize
```

例

次のコマンド例は、ハブ・ノード最大数を32に設定しています。

```
$ crsctl set cluster hubsize 32
```

crsctl get cluster mode

クラスタがOracle Flex Clusters用に構成されているかどうかを確認するか、現在のステータスを取得します。

構文

```
crsctl get cluster mode [config | status]
```

使用上のノート

- クラスタが構成されているモードを取得するには、configオプションを指定します。
- クラスタの現在のステータスを取得するには、statusオプションを指定します。

crsctl set cluster mode

標準クラスタをOracle Flex Clusterに変更します。

構文

```
crsctl set cluster mode flex
```

使用上のノート

このコマンドはflexパラメータのみを取り、これは必須です。

crsctl get cluster name

現在のクラスタ名を表示します。

構文

```
crsctl get cluster name
```

使用上のノート

このコマンドにパラメータはありません。

crsctl add cluster site

拡張クラスタにサイトを追加するには、crsctl add cluster siteコマンドを使用します。

構文

```
$ crsctl add cluster site site_name [-guid site_guid]
```

使用上のノート

- 追加するサイトの名前を指定する必要があります。必要に応じて、サイトのGUIDを指定できます。
- 拡張クラスタには最大5つのサイトを含めることができます。
- 拡張されていないクラスタでこのコマンドを実行することはできません。

例

次のコマンド例では、SiteAというサイトを拡張クラスタに追加します。

```
$ crsctl add cluster site SiteA
```

crsctl delete cluster site

拡張クラスタから特定のサイトを削除するには、crsctl delete cluster siteコマンドを使用します。

構文

```
$ crsctl delete cluster site site_name
```

使用上のノート

- 削除するサイトの名前を指定する必要があります。
- 拡張されていないクラスタでこのコマンドを実行することはできません。
- ノードやディスクを含むサイトは削除できません。サイトを削除する前に、ノードおよびディスクを削除する必要があります。

例

次のコマンドは、SiteAというサイトを拡張クラスタから削除します。

```
$ crsctl delete cluster site SiteA
```

crsctl modify cluster site

拡張クラスタ内のサイトの構成を変更します。

構文

```
crsctl modify cluster site site_name {-n node1, node2, ... | -s {rejuvenate | quarantine} | [-local]}
```

パラメータ

表E-47 crsctl modify cluster siteコマンドのパラメータ

パラメータ	説明
site_name	変更するサイトの名前を指定します。
-n node1, node2, ...	Oracle Cluster Registry (OCR)でグローバル変更を実行する場合、および変更内容をすべてのノード上の Oracle Local Registry (OLR)にもプッシュする場合は、このパラメータを指定します。
-s {rejuvenate quarantine}	サイトの状態を指定します。 ● rejuvenate: 以前隔離したサイトを有効にするには、このオプションを使用します。 記憶域の分割が検出された場合、サイトは通常、Oracle Clusterware および Oracle ASM により隔離されます。 ● quarantine: サイトを明示的に隔離する場合は、このオプションを使用します。Oracle Clusterware は、隔離されたサイトではデータベース・インスタンスまたは Oracle ASM を起動しません。記憶域の分割が発生した場合、Oracle Clusterware はその状況を検出し、サイトを隔離ステータスに設定します。

パラメータ	説明
-local	必要に応じて、このパラメータを指定して、ローカル・ノード上の Oracle Local Registry にのみ変更を加えることができます。OCR を使用できない場合や、グローバル変更が実行されて停止した場合のユーザーのために、オラクル社では、このことを実行できる特定のスクリプト用に、コマンドを提供しています。

例

次のコマンド例では、node11というノードをSiteAに追加します。

```
$ crsctl modify cluster site SiteA -n node1, node2,node11
```

crsctl query cluster site

拡張クラスタ内のサイト情報を表示します。

構文

```
$ crsctl query cluster site {site_name | -all}
```

使用上のノート

- 名前で特定のサイトを指定したり、-allを使用して拡張クラスタ内のすべてのサイトの情報を取得できます。
- このコマンドは、サイトの名前、サイトGUID、サイトの状態、およびサイトのコンポーネントを返します。
- このコマンドを実行するには、Oracle Clusterware技術スタックがアクティブで、起動している必要があります。
- 拡張クラスタには最大5つのサイトを含めることができます。

例

このコマンドによって、次のような出力が返されます。

```
$ crsctl query cluster site -all
Site 'SiteA' identified by GUID '7b7b3bef4c1f5ff9ff8765bceb45433a' in state 'ONLINE'
contains nodes 'node1,node2,node3', and disks 'disk1, disk2, disk3'.
Site 'SiteB' identified by GUID '23453bef4c1f5ff9ff8765bceb45433a' in state
'QUARANTINED' contains nodes 'node4, node5, node6', and disks 'disk4, disk5, disk6'.
Site 'SiteQ' identified by GUID '98763bef4c1f5ff9ff8765bceb45433a' in state 'ONLINE'
contains no nodes and disk 'disk7'.
Site 'SiteD' identified by GUID '4abcd453c6bc6fc1ffd3a58849d5ba41' in state 'ONLINE'
contains nodes 'cuj1234' and no disk.
```

crsctl get cluster tlsciphersuite

ノード間の通信を保護するためにトランスポート層セキュリティで使用する、使用可能な暗号スイートをリストします。

構文

```
crsctl get cluster tlsciphersuite [disabled|all]
```

使用上のノート

必要に応じて、disabledを指定して使用できない暗号スイートのみをリストするか、allを指定してすべての暗号スイートをリ

ストできます。

crsctl get cluster type

クラスタがOracle Flex Clusterであるかアプリケーション・クラスタであるかを判別するには、`crsctl get cluster type`コマンドを使用します。

構文

```
crsctl get cluster type
```

このコマンドは、flexまたはapplicationのいずれかのクラスタ・タイプを返します。

crsctl set cluster type

クラスタのタイプを設定します。

構文

```
crsctl Set cluster type flex | application
```

使用上のノート

設定するクラスタのタイプを選択するには、flexまたはapplicationを指定します。

crsctl get cpu equivalency

`crsctl get cpu equivalency`コマンドは、CPU_EQUIVALENCYサーバー構成属性の値を取得します。

構文

```
crsctl get cpu equivalency
```

例

このコマンドによって、次のような出力が返されます。

```
$ crsctl get cpu equivalency
CRS-4963: Current cpu equivalency parameter value is 1000
```

関連トピック

- [サーバー構成およびサーバー状態の属性](#)

crsctl set cpu equivalency

`crsctl set cpu equivalency`コマンドは、CPU_EQUIVALENCYサーバー構成属性の値を設定します。

構文

```
crsctl set cpu equivalency number
```

使用上のノート

このコマンドは、CPU_EQUIVALENCYサーバー構成属性を設定するための数値のみを取ります。

例

次の例では、CPU_EQUIVALENCYサーバー構成属性の値を2000に設定します。


```
$ crsctl set cpu equivalency 2000
CRS-4416: Server attribute 'CPU_EQUIVALENCY' successfully changed.
Restart Oracle High Availability Services for new value to take effect.
```

関連トピック

- [サーバー構成およびサーバー状態の属性](#)

crsctl check crs

ローカル・サーバー上のOracle High Availability ServicesとOracle Clusterwareスタックのステータスをチェックします。

構文

```
crsctl check crs
```

例

このコマンドによって、次のような出力が返されます。

```
CRS-4638: Oracle High Availablity Services is online
CRS-4537: Cluster Ready Services is onlin
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
```

crsctl config crs

Oracle高可用性サービスの自動起動構成を表示します。

構文

```
crsctl config crs
```

例

このコマンドによって、次のような出力が返されます。

```
CRS-4622: Oracle High Availability Services autostart is enabled.
```

crsctl disable crs

サーバー起動時のOracle高可用性サービスの自動起動を禁止します。

構文

```
crsctl disable crs
```

使用上のノート

- このコマンドはローカル・サーバーにのみ影響します。
- Oracle高可用性サービスの自動起動を無効にしている場合にOracle高可用性サービスを起動するには、[crsctl start crs](#)コマンドを使用する必要があります。

例

このコマンドによって、次のような出力が返されます。

```
CRS-4621: Oracle High Availability Services autostart is disabled.
```

crsctl enable crs

サーバー起動時のOracle高可用性サービスの自動起動を有効化します。

構文

```
crsctl enable crs
```

使用上のノート

このコマンドはローカル・サーバーにのみ影響します。

例

このコマンドによって、次のような出力が返されます。

```
CRS-4622: Oracle High Availability Services autostart is enabled.
```

crsctl start crs

ローカル・サーバーでOracle高可用性サービスを起動します。

構文

```
crsctl start crs [-excl [-nocrs] [-cssonly]] | [-wait | -waithas | -nowait] | [-noautostart]
```

パラメータ

表E-48 crsctl start crsコマンドのパラメータ

パラメータ	説明
-excl [-nocrs] [-cssonly]	次の 2 つのオプションを使用して Oracle Clusterware を排他モードで起動します。 ● CRSD を起動せずに排他モードで Oracle Clusterware を起動するには、-nocrs パラメータを指定します。 ● CSSD のみを起動するには -cssonly パラメータを指定します。
-wait -waithas -nowait	次のいずれかを選択します。 ● 起動が完了するまで待機し、すべての進捗状況およびステータス・メッセージを表示するには、-wait を指定します。 ● 起動が完了するまで待機し、OHASD の進捗状況およびステータス・メッセージを表示するには -waithas を指定します。

パラメータ	説明
	● OHASD の起動を待機しない場合は、-nowait を指定します。
-noautostart	OHASD のみを起動します。

使用上のノート

- このコマンドは、rootで実行する必要があります
- このコマンドによって起動されるのは、ローカル・サーバーのOracle高可用性サービスのみです。

例

ローカル・サーバーのOracle高可用性サービスを起動するには、次のコマンドを実行します。

```
# crsctl start crs
```

crsctl stop crs

ローカル・サーバーでOracle高可用性サービスを停止します。

構文

```
crsctl stop crs [-f]
```

使用上のノート

- このコマンドは、rootで実行する必要があります。
- このコマンドでは、ローカル・サーバーのOracle高可用性サービスの停止試行中、Oracle Clusterwareによって管理されるリソースの正常な停止が試行されます。

Oracle Clusterwareによって管理される任意のリソースが、crsctl stop crsコマンドの実行後も実行されている場合、コマンドは失敗します。すべてのリソースを無条件で停止し、ローカル・サーバーのOracle高可用性サービスを停止するには、-fパラメータを使用します。
- すべてのノードまたはリストされたノードでOracle Clusterwareを停止する場合は、[crsctl stop cluster](#)コマンドを使用しますが、これは、このコマンドを使用すると、Oracle Clusterwareスタックが特定のサーバーで停止される前に特定のリソースがクラスタ内の他のサーバーに再配置されないためです。1つ以上のノードでOracle高可用性サービスを停止する必要がある場合は、crsctl stop clusterコマンドが完了するまで待機してから、必要に応じて、特定のノードでcrsctl stop crsコマンドを実行します。
- Oracle Flex ASMが有効になっているOracle Clusterwareの標準クラスタ内で、Oracle ASMインスタンスがパラレルで実行されているすべてのノード上のOracle Clusterware技術スタックを停止しようとする場合は、その前に1つ以上のOracle ASMインスタンスを、Oracle ASMが実行されていない別のノードに再配置する必要があります。
- CRSCTLでは、高可用性およびフェイルオーバー時間を改善するためにこのコマンドを実行する場合に、shutdown abort停止オプションを使用します。
- crsctl stop crsを実行する前に、SRVCTLを使用してデータベースをシャットダウンした場合は、Oracle Clusterwareが起動した後、リソースを再起動する必要があります。

- Oracle Clusterware 11gリリース2(11.2.0.3)では、このコマンドをSolaris Sparc環境およびSolaris X64環境で実行すると、停止時およびその後の起動時にドライバはロードされたままになります。これは、Linux環境では発生しません。

例

ローカル・サーバーのOracle高可用性サービスを停止するには、次のコマンドを実行します。

```
# crsctl stop crs
```

crsctl query crs activeversion

クラスタ内で実行されているOracle Clusterwareソフトウェアのアクティブ・バージョンと構成済パッチ・レベルを表示します。

ただし、ローリング・アップグレード時は、クラスタ全体でアップグレードが終了するまでアクティブなバージョンは表示されず、その時点まで、クラスタはアップグレード前のバージョンで動作します。また、ローリング・パッチ適用時は、クラスタ全体でパッチ適用が終了するまでアクティブなパッチ・レベルは表示されず、その時点まで、クラスタはアップグレード前のパッチ・レベルで動作します。

構文

```
crsctl query crs activeversion [-f]
```

使用上のノート

また、-fパラメータを指定すると、このコマンドではクラスタ内の各構成済ノードのパッチ・レベルが出力されます。

例

このコマンドによって、次のような出力が返されます。

```
Oracle Clusterware active version on the cluster is [12.2.0.1.0]. The cluster
upgrade state is [NORMAL]. The cluster active patch level is [456789126].
```

crsctl add crs administrator

クラスタ管理者のリストにユーザーを追加します。

構文

```
crsctl add crs administrator -u user_name [-f]
```

パラメータ

表E-49 crsctl add crs administratorコマンドのパラメータ

パラメータ	説明
-u user_name	Oracle Clusterware 管理権限の付与対象のユーザー名。
-f	ユーザー名の妥当性チェックを無視するには、このパラメータを使用します。

使用上のノート

- このコマンドは、Oracle Clusterware 12c以降のリリースでは非推奨です。
- このコマンドは、rootまたはクラスタ管理者として、またはWindowsシステムの場合、管理者として実行する必要があります。

ります。

- デフォルトでは、root、Oracle Clusterwareをインストールしたユーザーおよび*ワイルドカードが、Oracle Clusterware管理権限を持つユーザーのリストのメンバーです。ワイルドカードを削除し、Oracle Clusterwareのロール別管理を有効にするには、[crsctl delete crs administrator](#)コマンドを実行します。

例

Oracle Clusterware管理者のリストにユーザーを追加するには、次のコマンドを実行します。

```
# crsctl add crs administrator -u scott
```

関連トピック

- [ロール別管理](#)

crsctl delete crs administrator

Oracle Clusterware管理者のリストからユーザーを削除します。

構文

```
crsctl delete crs administrator -u user_name [-f]
```

パラメータ

表E-50 crsctl delete crs administratorのパラメータ

パラメータ	説明
-u user_name	削除対象の Oracle Clusterware 管理権限を持つユーザーの名前。 デフォルトでは、Oracle Clusterware 管理権限を持つユーザーのリストは、Oracle Clusterware をインストールしたユーザー、root および*で構成されます。Oracle Clusterware をインストールしたユーザーおよび root は、このリストの永続的なメンバーです。*値によって Oracle Clusterware 管理権限がすべてのユーザーに付与されるため、ロール別管理を有効にするには、この値を削除する必要があります。
-f	ユーザー名の妥当性チェックを無視するには、このパラメータを使用します。

使用上のノート

- user_nameパラメータは必須です。
- このコマンドは、rootまたはクラスタ管理者として、またはWindowsシステムの場合、管理者として実行する必要があります。
- ロール別管理を有効にするには、二重引用符("")で囲まれた*値を削除する必要があります。

例

クラスタ管理者のリストからユーザーを削除するには、次のコマンドを実行します。

```
# crsctl delete crs administrator -u scott
```

関連項目:

[ロール別管理](#)

crsctl query crs administrator

Oracle Clusterware管理者権限を持つユーザーのリストを表示します。

構文

```
crsctl query crs administrator
```

使用上のノート

ここで、コマンドに関する追加情報を追加します。

例

このコマンドによって、次のような出力が返されます。

```
CRS Administrator List: scott
```

crsctl query crs autostart

Oracle Clusterwareの自動リソース起動基準の値を取得します。

構文

```
crsctl query crs autostart
```

例

このコマンドによって、次のような出力が返されます。

```
'Autostart delay':          60
'Autostart servercount':    2
```

crsctl set crs autostart

Oracle Clusterwareの自動リソース起動基準を設定します。

自動起動遅延および最小サーバー数の基準を設定すると、この2つの条件のいずれかが満たされるまでOracle Clusterwareリソースの自動起動は遅延します。

構文

```
crsctl set crs autostart [delay delay_time] [servercount count]
```

パラメータ

表E-51 crsctl set crs autostartコマンドのパラメータ

パラメータ	説明
delay delay_time	Oracle Clusterware の自動起動を遅らせる秒数を指定

パラメータ	説明
	します。
servercount count	Oracle Clusterware の自動起動に必要なサーバーの最小数を指定します。

例

クラスタ内の最初のサーバーがオンラインになった後、Oracle Clusterwareによるリソースの自動起動を60秒間遅らせるには、次のコマンドを実行します。

```
$ crsctl set crs autostart delay 60
```

Oracle Clusterwareが2つ以上のサーバーがオンラインになるのを待機してから、リソースの自動起動を開始するようにするには、次のコマンドを実行します。

```
$ crsctl set crs autostart servercount 2
```

前述の2つの条件のいずれかが満たされるまで(順不同) Oracle Clusterwareによるリソースの自動起動を遅らせるには、次のコマンドを実行します。

```
$ crsctl set crs autostart delay 60 servercount 2
```

crsctl query crs releasepatch

ノードへのパッチ適用時にGridホーム・パッチ・リポジトリ内で更新されたパッチ・レベルを表示します。

パッチ・レベルは、コマンドを実行するローカル・ノードのみに対応します。このコマンドは、Oracle Clusterwareスタックが実行されていないときに実行できます。

構文

```
crsctl query crs releasepatch
```

例

このコマンドは、パッチが適用されていないノードについて、次のような出力を戻します。

```
Oracle Clusterware release patch level is [3180840333] and the complete list of patches is [13559647] on the local node.
```

crsctl query crs releaseversion

ローカル・ノード上のバイナリに格納されているOracle Clusterwareソフトウェアのバージョンを表示します。

構文

```
crsctl query crs releaseversion
```

例

このコマンドによって、次のような出力が返されます。

```
Oracle High Availability Services release version on the local node is [12.2.0.1.0]
```

crsctl query crs site

特定のホストまたはディスクが属するサイトを判別するには、`crsctl query crs site`コマンドを使用します。

構文

```
crsctl query crs site {-n host_name | -d disk_name}
```

使用上のノート

- ホスト名またはディスク名を指定する必要があります。
- このコマンドを実行するには、Oracle Clusterware技術スタックがアクティブである必要があります。

例

このコマンドによって、次のような出力が返されます。

```
$ crsctl query crs site -n hostA  
Host 'hostA' is a part of site 'SiteA'
```

crsctl query crs softwarepatch

インストールされたOracle Clusterwareの構成済パッチ・レベルを表示します。

構文

```
crsctl query crs softwarepatch [host_name]
```

使用上のノート

ホスト名を指定して、特定のホストにインストールされているOracle Clusterwareのパッチ・レベルを表示することもできます。そうでない場合、CRSCTLではローカル・ホストにインストールされているOracle Clusterwareのパッチ・レベルが表示されます。

例

このコマンドによって、次のような出力が返されます。

```
Oracle Clusterware patch level on node [node1] is [456789126]
```

crsctl query crs softwareversion

特定のノードで正常に起動したソフトウェアの最新バージョンを表示します。

構文

```
crsctl query crs softwareversion [node_name]
```

使用上のノート

ノード名を指定しない場合、Oracle Clusterwareではローカル・サーバーで実行中のOracle Clusterwareのバージョンが表示されます。

例

このコマンドによって、次のような出力が返されます。

```
Oracle Clusterware version on node [node1] is [12.2.0.1.0]
```


crsctl get css

特定のクラスタ同期サービス・パラメータの値を取得します。

構文

```
crsctl get css parameter
```

使用上のノート

- クラスタ同期サービスのパラメータとそのデフォルト値は次のとおりです。

```
clusterguid  
diagwait  
disktimeout (200 (seconds))  
misscount (30 (seconds))  
reboottime (3 (seconds))  
priority (4 (UNIX), 3 (Windows))  
logfilesize (50 (MB))
```

- このコマンドはデフォルト値を表示しません。
- このコマンドはローカル・サーバーにのみ影響します。

例

このコマンドによって、次のような出力が返されます。

```
$ crsctl get css disktimeout  
CRS-4678: Successful get disktimeout 200 for Cluster Synchronization Services.
```

crsctl pin css

多数の特定ノードを固定します。

ノードの固定とは、ノード名とノード番号との関連付けを固定することです。固定されていないノードの場合、停止中にその期限が切れると、ノード番号が変更される可能性があります。固定されたノードの場合、期限切れになりません。

構文

```
crsctl pin css -n node_name [ node_name [...]]
```

使用上のノート

- サーバーを空白区切りのリストで指定できます。
- 固定されたサーバーには、12cリリース1 (12.1)より前のOracleソフトウェアが必要です。
- [crsctl unpin css](#)を使用するとノードの固定が解除されます。
- [crsctl delete node](#)コマンドを使用してノードを削除すると、ノードの固定も暗黙的に解除されます。

例

node2というノードを固定するには、次のコマンドを実行します。

```
# crsctl pin css -n node2
```

crsctl set css

クラスタ同期サービス・パラメータの値を設定します。

構文

```
crsctl set css parameter value
```

使用上のノート

- My Oracle Supportから指示された場合を除き、`crsctl set css`コマンドを使用して次のパラメータを設定しないでください。
- クラスタ同期サービスのパラメータは次のとおりです。

```
diagwait  
disktimeout  
logfilesize  
misscount  
priority  
reboottime
```

crsctl unpin css

多数のサーバーの固定を解除します。

固定されていないノードの場合、停止中にその期限が切れると、ノード番号が変更される可能性があります。

構文

```
crsctl unpin css -n node_name [node_name [...exit]]
```

使用上のノート

- ノードを空白区切りのリストで指定できます。
- 1週間より長い期間停止していた固定解除済サーバーは、`olsnodes`ではレポートされません。これらのサーバーはクラスタから動的に離れるため、ユーザーがクラスタから明示的に削除する必要はありません。
- [crsctl delete node](#)コマンドを使用してノードを削除すると、ノードの固定も暗黙的に解除されます。
- Oracle Clusterwareのアップグレード時にはすべてのサーバーが固定されますが、Oracle Clusterware 12c以降のリリースの新規インストール後は、クラスタに追加するすべてのサーバーの固定が解除されます。
- 12cリリース1 (12.1)より前のOracle RACのインスタンスが存在するサーバーにOracle Clusterware 12c以降のリリースをインストール済の場合、そのサーバーの固定を解除することはできません。

例

2つのノードの固定を解除するには、次のコマンドを実行します。

```
$ crsctl unpin css -n node1 node4
```

crsctl unset css

クラスタ同期サービス・パラメータの値を設定解除し、値をデフォルト値に戻します。

構文

```
crsctl unset css parameter
```

使用上のノート

- 次のクラスタ同期サービスのパラメータを指定できます。

- diagwait
- disktimeout
- misscount
- reboottime
- priority
- logfilesize

例

クラスタ同期サービスのパラメータreboottimeをデフォルト値にリストアするには、次のコマンドを実行します。

```
$ crsctl unset css reboottime
```

crsctl get css ipmiaddr

Intelligent Platform Management Interface(IPMI)デバイスのOracle Local Registryに格納されているアドレスを取得します。

構文

```
crsctl get css ipmiaddr
```

使用上のノート

- Oracle Clusterwareのインストールに使用されたユーザー・アカウントでコマンドを実行します。
 - このコマンドでは、Oracle Local Registryに格納されているIPアドレスのみが取得されます。これは、IPMIが実際に使用するIPアドレスではない場合があります。
- ローカル・サーバーで、rootとしてipmiutilまたはipmitoolを使用して、IPMIによって使用されているIPアドレスを取得します。

crsctl set css ipmiaddr

ローカルIntelligent Platform Management Interface(IPMI)デバイスのアドレスをOracle Local Registryに格納します。

構文

```
crsctl set css ipmiaddr ip_address
```

使用上のノート

- Oracle Clusterwareのインストールに使用されたユーザー・アカウントでコマンドを実行します。
- ローカル・サーバーで、rootとしてipmiutilまたはipmitoolを使用して、IPMIデバイスによって使用されるIPアドレスを取得します
- Oracle ClusterwareによってIPMIのIPアドレスが構成ストアに格納され、必要に応じてアドレスが配布されます。
- このコマンドでは、コマンドを実行したサーバーのIPMI IPアドレスのみが格納されます。
- 指定したアドレスのIPMIに別のサーバーがアクセスできない場合、このコマンドは失敗します。

例

IPMI IPアドレスをローカル・サーバーに格納して他のクラスタ・ノードに配布するには、次のコマンドを実行します。

```
$ crsctl set css ipmiaddr 192.0.2.244
```

crsctl set css ipmiadmin

Intelligent Platform Management Interface(IPMI)管理者のログイン資格証明をOracle Local Registryに格納します。

構文

```
crsctl set css ipmiadmin ipmi_administrator_name
```

使用上のノート

- このコマンドは、Oracle Clusterwareをインストールしたユーザー・アカウントで実行する必要があります。
- プロンプトに従い、新しい管理者アカウントの名前と関連付ける新しいパスワードを指定します。Oracle ClusterwareによってローカルのIPMIの名前およびパスワードが構成ストアに格納され、必要に応じて新しい資格証明が配布されます。
- このコマンドでは、このコマンドを実行したサーバーのIPMI管理者のみが変更されます。
- 指定したアドレスのローカルIPMIに別のサーバーがアクセスできない場合、このコマンドは失敗します。

例

IPMI管理者scottを変更するには、次のコマンドを実行します。

```
$ crsctl set css ipmiadmin scott
```

crsctl query css ipmiconfig

ローカル・サーバー上のOracle Clusterwareが障害分離にIPMIを使用するよう構成されているかどうかを確認します。

このコマンドは構成データの存在を検出しますが、その正しさは検証できないことに注意してください。

構文

```
crsctl query css ipmiconfig
```

使用上のノート

- このコマンドは、Oracle Cluster Registry (OCR)に格納されているIPMI構成を特定してアクセスしようとするため、Oracle Clusterwareのインストールに使用したアカウントで実行する必要があります。そうしないと、認可の失敗が報告される場合があります。
- レジストリにIPMI構成データが含まれていない場合、別のアカウントで実行すると、認可の失敗にならない場合があります。

例

このコマンドによって、次のような出力が返されます。

```
CRS-4236: Oracle Clusterware configured to use IPMI
```

または:

```
CRS-4237: Oracle Clusterware is not fully configured to use IPMI
```

crsctl unset css ipmiconfig

以前に格納されたIPMI構成(ログイン資格証明とIPアドレス)をOracle Local Registryからクリアします。

前提条件

このコマンドは、クラスタでIPMIを構成解除する場合、またはIPMI構成が不適切なユーザーによって以前に格納された場合に適しています。

構文

```
crsctl unset css ipmiconfig
```

使用上のノート

- このコマンドは、IPMIを構成するときに使用したユーザー・アカウントまたは権限を持つユーザーで実行する必要があります。
- このコマンドでは、このコマンドを実行したサーバーのIPMI構成のみがクリアされます。
- Oracle Clusterwareが、このコマンドで削除する構成データにアクセスして使用できていた場合、Oracle Clusterwareを再起動するまで、引き続きアクセスおよび使用が可能です。

例

Oracle Local RegistryからIPMI構成データをクリアし、IPMIが今後使用されないようにOracle Clusterwareを再起動するには、rootまたはWindowsの管理者グループのメンバーとしてログインし、次のコマンドを実行します。

```
# crsctl unset css ipmiconfig
# crsctl stop crs
# crsctl start crs
```

crsctl query css ipmidevice

ローカル・システム上のIntelligent Platform Management Interface (IPMI)ドライバの存在を確認します。

構文

```
crsctl query css ipmidevice
```

使用上のノート

- このコマンドは、IPMIインストール中に事前チェックを実行し、通常、インストーラによってのみ発行されます。
- このコマンドでは形式的なチェックが実行されるため、成功リターンは、IPMIハードウェアの構成が完了しており使用できることを保証するものではありません。
- このコマンドの実行に特別な権限は必要ありません。

例

このコマンドによって、次のような出力が返されます。

```
CRS-4231: IPMI device and/or driver found
```

または:

```
CRS-4218: Unable to access an IPMI device on this system
```

crsctl get css noautorestart

クラスタ同期サービスの自動再起動が無効になっている原因となる問題をリストします。

構文

```
crsctl get css noautorestart
```

使用上のノート

- このコマンドにパラメータはありません。
- CRSCTLでは、次の3つの可能な値のいずれかを戻します。
 - all: どのような問題が発生しても自動再起動は無効になります。
 - interconnect: インターコネクトの問題が発生した場合、自動再起動は無効になります。
 - vf: 投票ファイルの問題が発生した場合、自動再起動が無効になります。

crsctl set css noautorestart

特定の問題が発生したとき、クラスタ同期サービスの自動再起動を停止します。

構文

```
crsctl set css noautorestart {all | interconnect | vf | none}
```

使用上のノート

次のいずれかの値を指定します。

- all: どのような問題が発生したときも、自動再起動を無効化します。
- interconnect: インターコネクトの問題が発生したとき、自動再起動を無効化します。
- vf: 投票ファイルの問題が発生したとき、自動再起動を無効化します。
- none: クラスタ同期サービスの自動再起動を有効化します。

crsctl add css votedisk

Oracle ASMディスク・グループ以外のストレージ・デバイス上のクラスタに1つ以上の投票ファイルを追加するには、crsctl add css votediskコマンドを使用します。

構文

```
crsctl add css votedisk vdisk [...] [-purge]
```

パラメータ

表E-52 crsctl add css votediskコマンドのパラメータ

パラメータ	説明
vdisk [...]	追加する投票ファイルの完全修飾パスを指定します。複数の投票ファイルのパスを指定する場合は、空白区切りのリストを入力します。
-purge	既存のすべての投票ファイルを一度に削除するには、-

パラメータ	説明
	purge オプションを指定します。投票ファイルの既存のセットを 1 つの操作で置換できます。

使用上のノート

- 外部冗長を提供するディスク・アレイなどのストレージ・デバイスを持たないかぎり、3つ以上の投票ファイルを使用する必要があります。投票ファイルは6つ以上使用しないことをお勧めします。サポートされる投票ファイルの最大数は15です。

例

投票ファイルをクラスタに追加するには、次のコマンドを実行します。

```
$ crsctl add css votedisk /stor/grid/ -purge
```

crsctl delete css votedisk

Oracle Clusterware構成から投票ファイルを削除します。

構文

```
crsctl delete css votedisk {voting_disk_GUID [...] | vdisk [...] | +diskgroup}
```

パラメータ

表E-53 crsctl delete css votediskコマンドのパラメータ

パラメータ	説明
voting_disk_GUID [...]	削除対象の投票ファイルのファイル・ユニバーサル識別子 (GUID)を指定します。複数の GUID を指定する場合は、空白区切りのリストを入力します。
vdisk [...]	削除する投票ファイルのパスを指定できます。複数の投票ファイルのパスを指定する場合は、空白区切りのリストを入力します。
+diskgroup	削除する投票ファイルが含まれている Oracle ASM ディスク・グループの名前を指定できます。Oracle Clusterware が排他モードの場合のみ、パラメータを使用できます。

使用上のノート

- 削除する投票ファイルの1つ以上のGUID、または削除する投票ファイルの1つ以上のパス、または削除する投票ファイルを含むOracle ASMディスク・グループの名前を指定できます。
- [crsctl query css votedisk](#)コマンドを実行することによって、現在の各投票ファイルのGUIDを取得できます。

例

GUIDを使用して投票ファイルを削除するには、次のコマンドを実行します。

```
$ crsctl delete css votedisk 26f7271ca8b34fd0bfcdc2031805581e
```

crsctl query css votedisk

クラスタ同期サービスによって使用される投票ファイル、投票ファイルのステータス、ディスクの格納場所(Oracle ASMやその他の場所など)を表示します。

構文

```
crsctl query css votedisk
```

例

このコマンドによって、次のような出力が返されます。

##	STATE	File Universal Id	File Name	Disk group
1.	ONLINE	296641fd201f4f3fbf3452156d3b5881	(/ocfs2/host09_vd3)	[]
2.	ONLINE	8c4a552bdd9a4fd9bf93e444223146f2	(/netapp/ocrvf/newvd)	[]
3.	ONLINE	8afeee6ae3ed4fe6bfbb556996ca4da5	(/ocfs2/host09_vd1)	[]

Located 3 voting file(s).

crsctl check ctss

クラスタ時刻同期化サービスのステータスをチェックします。

構文

```
crsctl check ctss
```

例

このコマンドによって、次のような出力が返されます。

```
CRS-4700: The Cluster Time Synchronization Service is in Observer mode.
```

または:

```
CRS-4701: The Cluster Time Synchronization Service is in Active mode.  
CRS-4702: Offset from the reference node (in msec): 100
```

crsctl discover dhcp

特定のポートのネットワーク上でDHCP検出パケットを送信します。DHCPサーバーがネットワークに存在すると、そのサーバーは検出メッセージに応答し、コマンドが成功します。

構文

```
crsctl discover dhcp -clientid clientid [-port port] [-subnet subnet]
```

パラメータ

表E-54 crsctl discover dhcpコマンドのパラメータ

パラメータ	説明

パラメータ	説明
-clientid clientid	検出したいクライアント ID を指定します。クライアント ID は、 crsctl get clientid dhcp コマンドを実行して取得します。
-port port	CRSCTL が検出パケットを送信するポートを指定します。
-subnet subnet	必要に応じて、DHCP 検出パケットが送信されるサブネットを指定できます。

使用上のノート

このコマンドは、rootで実行する必要があります。

例

このコマンドによって、次のような出力が返されます。

```
CRS-10009: DHCP server returned server: 192.168.53.232,
loan address : 192.168.29.221/255.255.252.0, lease time: 43200
```

crsctl get clientid dhcp

DHCPサーバーから構成済クラスタ・リソースのIPアドレスを取得するためにOracle Clusterwareエージェントが使用するクライアントIDを表示します。VIPタイプは必須です。

構文

```
crsctl get clientid dhcp -cluname cluster_name -viptype vip_type
[-vip vip_res_name] [-n node_name]
```

パラメータ

表E-55 crsctl get clientid dhcpコマンドのパラメータ

パラメータ	説明
-cluname cluster_name	クラスタ・リソースが構成されるクラスタの名前を指定します。
-viptype vip_type	クライアント ID を表示する VIP リソースのタイプ(HOSTVIP、SCANVIP または APPVIP)を指定します。
-vip vip_res_name	VIP リソースの名前を指定します。APPVIP VIP タイプを指定する場合、このパラメータは必須です。
-n node_name	クライアント ID を取得するノードの名前を指定します。HOSTVIP VIP タイプを指定する場合、このパラメータは必須です。

例

このコマンドによって、次のような出力が返されます。

```
CRS-10007: stmjk0462clr-stmjk01-vip
```

crsctl release dhcp

特定のクライアントIDにDHCPリースの解放リクエストを送信し、ネットワーク上の特定のポートに解放パケットを送信します。

構文

```
crsctl release dhcp -clientid clientid [-port port]
```

パラメータ

表E-56 crsctl release dhcpコマンドのパラメータ

パラメータ	説明
-clientid clientid	解放したいクライアント ID を指定します。クライアント ID は、crsctl get clientid コマンドを実行して取得します。
-port port	CRSCTL が解放パケットを送信するポート。ポートを指定しないと、CRSCTL はデフォルト値の 67 を使用します。

例

このコマンドによって、次のような出力が返されます。

```
CRS-10012: released DHCP server lease for client ID stmjk0462clr-stmjk01-vip  
on port 67
```

crsctl request dhcp

ネットワーク上の指定されたポートにDHCPリクエスト・パケットを送信します。

DHCPサーバーに提供可能なIPアドレスがある場合は、クライアントIDのIPアドレスで応答します。

構文

```
crsctl request dhcp -clientid clientid [-port port]
```

パラメータ

表E-57 crsctl request dhcpコマンドのパラメータ

パラメータ	説明
-clientid clientid	要求したいクライアント ID を指定します。クライアント ID は、crsctl get clientid コマンドを実行して取得します。
-port port	必要に応じて、CRSCTL からのリクエスト・パケットの送信先となるポートを指定できます。ポートを指定しないと、CRSCTL

パラメータ	説明
	はデフォルト値の 67 を使用します。

例

このコマンドによって、次のような出力が返されます。

```
CRS-10009: DHCP server returned server: 192.168.53.232,
loan address : 192.168.29.228/255.255.252.0, lease time: 43200
```

crsctl replace discoverystring

投票ファイルの場所を特定するために使用される検出文字列を置き換えます。

構文

```
crsctl replace discoverystring "absolute_path[,...]"
```

使用上のノート

- 1つ以上の投票ファイルの場所に一致する絶対パスを、カンマ区切りで指定します。ワイルドカードを使用でき、パスのリストは、二重引用符("")で囲んで記述します。
- このコマンドを実行するには、root、Oracle Clusterwareインストール所有者または管理者グループのメンバーである必要があります。
- クラスタ内の任意のノードでこのコマンドを実行できます。
- 投票ファイルをOracle ASMディスク・グループに格納する場合は、検出文字列を変更できません。

例

現在の検出文字列が/oracle/css1/*だとします。/oracle/css2/ディレクトリの投票ファイルも使用するには、次のコマンドを使用して現在の検出文字列を置換します。

```
# crsctl replace discoverystring "/oracle/css1/*,/oracle/css2/*"
```

crsctl query dns

指定されたDNSサーバーの名前のDNS参照によって戻されるアドレスのリストを取得します。

構文

```
crsctl query dns -name name [-dnsserver DNS_server_address] [-type query_type]
[-port port] [-attempts number_of_attempts] [-timeout timeout_in_seconds] [-v]
```

パラメータ

表E-58 crsctl query dnsコマンドのパラメータ

パラメータ	説明
-name name	問い合わせる完全修飾ドメイン名を指定します。
-dnsserver DNS_server_address	必要に応じて、ドメイン名を問い合わせる DNS サーバーのアド

パラメータ	説明
	レスを指定できます。
-type query_type	必要に応じて IPv4 または IPv6 を選択でき、選択するには、問合せタイプとして A または AAAA をそれぞれ指定します。
-port port	必要に応じて、DNS サーバーがリスニングするポートを指定できます。ポートを指定しないと、デフォルトのポート 53 が使用されます。
-attempts number_of_attempts	必要に応じて、再試行の回数を指定できます。
-timeout timeout_in_seconds	必要に応じて、タイムアウトの時間(秒)を指定できます。
-v	冗長出力。

例

crsctl query dnsコマンドによって、stmjk07-vip.stmjk0462.foo.comというDNSサーバーについて、次のような出力が戻されます。

```
CRS-10024: DNS server returned 192.168.29.250 for name
stmjk07-vip.stmjk0462.foo.com
```

crsctl start ip

指定されたサブネット・マスクを使用して、指定されたインタフェース上で指定のIP名またはIPアドレスを起動します。

IPを開始するサーバーでこのコマンドを実行します。

構文

```
crsctl start ip -A {IP_name | IP_address}/netmask/interface_name
```

パラメータ

表E-59 crsctl start ipコマンドのパラメータ

パラメータ	説明
IP_name IP_address	前に -A を付けて、ドメイン名または IP アドレスを指定します。完全修飾ドメイン名を指定しないと、CRSCTL では標準の名前検索が使用されます。
/netmask/interface_name	起動する IP のサブネット・マスクとインタフェース名を、スラッシュ (/) で区切って指定します。このペアの前にも、スラッシュを付け

パラメータ	説明
	る必要があります。

例

ローカル・サーバーでIPを起動するには、次のようなコマンドを実行します。

```
$ crsctl start ip -A 192.168.29.220/255.255.252.0/eth0
```

crsctl status ip

指定のIPアドレスがネットワーク上で稼働しているかどうかをチェックします。

構文

```
crsctl status ip -A {IP_name | IP_address}
```

使用上のノート

前に-Aを付けて、ドメイン名またはIPアドレスを指定します。完全修飾ドメイン名を指定しないと、CRSCTLでは標準の名前検索が使用されます。

例

このコマンドによって、次のような出力が返されます。

```
CRS-10003: IP address 192.168.29.220 could be reached from current node
```

crsctl stop ip

指定されたサブネット・マスクを使用して、指定されたインタフェース上で指定のIP名またはIPアドレスを停止します。

このコマンドは、停止するIPが存在するサーバーで実行します。

構文

```
crsctl stop ip -A {IP_name | IP_address}/interface_name
```

パラメータ

表E-60 crsctl stop ipコマンドのパラメータ

パラメータ	説明
IP_name IP_address	前に-Aを付けて、ドメイン名またはIPアドレスを指定します。完全修飾ドメイン名を指定しないと、CRSCTLでは標準の名前検索が使用されます。
interface_name	IPを停止するインタフェースを指定します。

例

ローカル・サーバーでIPを停止するには、次のようなコマンドを実行します。

```
$ crsctl stop ip -A MyIP.domain.com/eth0
```

crsctl lsmodules

デバッグ可能なモジュールのコンポーネントをリストします。

構文

```
crsctl lsmodules {mdns | gpnp | css | crf | crs | ctss | evm | gipc}
```

使用上のノート

次のいずれかのモジュールのコンポーネントのリストを取得できます。

- mdns: マルチキャスト・ドメイン・ネーム・サーバー
- gpnp: グリッド・プラグ・アンド・プレイ・サービス
- css: クラスタ同期サービス
- crf: クラスタ状態モニター
- crs: クラスタ・レディ・サービス
- ctss: クラスタ時刻同期化サービス
- evm: Event Manager
- gipc: グリッド・プロセス間通信

例

このコマンドによって、次のような出力が返されます。

```
$ crsctl lsmodules evm
List EVMD Debug Module: CLSVER
List EVMD Debug Module: CLUCLS
List EVMD Debug Module: COMMCRS
List EVMD Debug Module: COMMNS
List EVMD Debug Module: CRSOCR
List EVMD Debug Module: CSSCLNT
List EVMD Debug Module: EVMAGENT
List EVMD Debug Module: EVMAPP
...
```

関連トピック

- [crsctl set logを使用した動的デバッグ](#)

crsctl create member_cluster_configuration

新しいメンバー・クラスタ用の構成デプロイメント・ファイルを作成します。

構文

```
crsctl create member_cluster_configuration member_cluster_name
  -file cluster_manifest_file -member_type database | application
  [-version member_cluster_version] [-domain_services [asm_storage local
  | direct | indirect] [rhp] [acfs]
```

パラメータ

表E-61 crsctl create member_cluster_configurationコマンドのパラメータ

パラメータ	説明

パラメータ	説明
member_cluster_name	新しいメンバー・クラスタの名前を指定します。
-file cluster_manifest_file	作成するクラスタ manifest ファイル(.xml 拡張子を含む)のパスを指定します。
-member_type database application	作成するメンバー・クラスタのタイプを指定します。database を選択した場合は、クラスタを使用して Oracle Database をデプロイできます。application を選択した場合は、クラスタを使用してアプリケーションのみをデプロイできます。
-version member_cluster_version	必要に応じて、5 桁のバージョン(12.2.0.1.0 など)の Oracle Grid Infrastructure を新しいメンバー・クラスタ上で指定できます(ドメイン・サービス・クラスタのバージョンとは異なる場合)。
-domain_services [asm_storage local direct indirect] [rhp] [acfs]	必要に応じて、このメンバー・クラスタに最初に構成されたサービスを指定できます。-domain_services パラメータを使用しない場合、CRSCTL はグリッド・インフラストラクチャ管理リポジトリおよび TFA のサービスのみを構成します。 また、必要に応じて、データベース・メンバー・クラスタの Oracle ASM 記憶域アクセス・パスを指定できます。 • local: 記憶域はクラスタに対してローカルです。 • direct または indirect: ドメイン・サービス・クラスタで指定された記憶域への直接アクセスまたは間接アクセスを選択します。 必要に応じて、rhp を選択して、フリー・パッチ適用およびプロビジョニング・クライアント・クラスタの資格証明および構成を生成できます。 オプションで、間接アクセス・メンバー・クラスタに対して acfs を選択すると、Oracle ACFS Remote を使用してメンバー・クラスタに記憶域を提供できます。

crsctl delete member_cluster_configuration

メンバー・クラスタの構成を削除します。

構文

```
crsctl delete member_cluster_configuration member_cluster_name [-f]
```

使用上のノート

削除するメンバー・クラスタの名前を指定する必要があり、必要に応じて、`-f`を使用して削除操作を強制できます。

crsctl query member_cluster_configuration

メンバー・クラスタの構成を表示します。

構文

```
crsctl query member_cluster_configuration [member_cluster_name]
```

使用上のノート

特定のメンバー・クラスタを名前で指定しないことを選択した場合、CRSCTLはすべてのメンバー・クラスタの詳細を表示します。

例

このコマンドによって、次のような出力が返されます。

```
$ crsctl query member_cluster_configuration
indirect-mc0      12.2.0.1.0 66cdb4efd2f77f7affe559cc56003a14 ASM, GIMR, TFA
indirect-mc1      12.2.0.1.0 2b85c6ac63434f30bfa6d074bb337621 ASM, GIMR, TFA
indirect-mc2      12.2.0.1.0 d883e65af9ffff1fbf67e922e99bd6e4 ASM, GIMR, TFA
indirect-mc3      12.2.0.1.0 60dcca80166aff0cffaf8e33213bfacb ASM, GIMR, TFA
indirect-mc4      12.2.0.1.0 78d64e1454505f3effda0136738f325e ASM, GIMR, TFA
```

crsctl delete node

クラスタからノードを削除します。

構文

```
crsctl delete node -n node_name
```

使用上のノート

- このコマンドを実行するには、rootである必要があります。
- `node_name`パラメータは必須です。

例

クラスタから`node06`というノードを削除するには、rootとして次のコマンドを実行します。

```
# crsctl delete node -n node06
```

crsctl get nodename

ローカル・ノードの名前を取得します。

構文

```
crsctl get nodename
```

例

このコマンドによって、次のような出力が返されます。

```
node2
```


crsctl get node role

クラスタ内のノードに構成されているノード・ロールを取得します。

構文

```
crsctl get node role {config | status} [node node_name | -all]
```

使用上のノート

- 特定のノードに構成されているノード・ロールを取得するには、configオプションを指定します。
- 特定のノードの現在のステータスを取得するには、statusオプションを指定します。
- ロール情報を取得する特定のノードを指定したり、-allを使用してクラスタ内のすべてのノードの情報を取得できます。特定のノードを指定しない場合、CRSCTLによってローカル・ノードに関する情報が戻されます。

例

このコマンドによって、次のような出力が返されます。

```
$ crsctl get node role config -all
Node 'mjkeenan_node5_0' configured role is 'hub'
Node 'mjkeenan_node5_1' configured role is 'hub'
Node 'mjkeenan_node5_2' configured role is 'hub'
Node 'mjkeenan_node5_3' configured role is 'hub'
$ crsctl get node role status -all
Node 'mjkeenan_node5_0' active role is 'hub'
Node 'mjkeenan_node5_2' active role is 'hub'
Node 'mjkeenan_node5_3' active role is 'hub'
$ crsctl get node role config -node mjkeenan_node5_1
Node 'mjkeenan_node5_1' configured role is 'hub'
```

crsctl add policy

構成ポリシーをポリシー・セットに追加します。

構文

```
crsctl add policy policy_name -attr
"attr_name=attr_value[,attr_name=attr_value[, ...]]"
```

パラメータ

表E-62 crsctl add policyコマンドのパラメータ

パラメータ	説明
policy_name	追加するポリシーの名前を指定します。
attr_name	前に -attr フラグを付けて、DESCRIPTION ポリシー属性を使用してポリシーの説明を指定します。
attr_value	ポリシーについて説明する DESCRIPTION ポリシー属性の値。 ノート: attr_name パラメータおよび attr_value パラメータは、カンマで区切って二重引用符("")で囲む必要があります。

パラメータ	説明
	たとえば:
	-attr "DESCRIPTION=daytime"

使用上のノート

- ポリシーを追加してもポリシーはアクティブ化されません
- policy_nameパラメータは必須です
- このコマンドの実行に必要な権限は、ポリシー・セットのACL属性の値によって異なります

例

ポリシーを追加するには、次のコマンドを実行します。

```
$ crsctl add policy nightTime -attr "DESCRIPTION=nighttime"
```

crsctl delete policy

ポリシー・セットから構成ポリシーを削除します。

構文

```
crsctl delete policy policy_name [policy_name [...]] [-i]
```

使用上のノート

- 削除するポリシーの名前または空白区切りのポリシー名のリストを指定します。policy_nameパラメータは必須です
- -iを指定すると、Oracle Clusterwareでリクエストをすぐに処理できない場合にコマンドが失敗します。
- このコマンドの実行に必要な権限は、ポリシー・セットのACL属性の値によって異なります

例

ポリシーを削除するには、rootまたはOracle Clusterwareインストール所有者として次のコマンドを実行します。

```
# crsctl delete policy nightTime
```

crsctl eval activate policy

システムを変更せずに、特定のポリシーをアクティブ化した場合の影響を予測します。

このコマンドはクラスタ管理者にとって役に立つ場合があります。

構文

```
crsctl eval activate policy policy_name [-f] [-admin [-l serverpools | resources  
| all] [-x] [-a]] [-explain]
```

パラメータ

表E-63 crsctl eval activate policyコマンドのパラメータ

パラメータ	説明

パラメータ	説明
-f	ポリシーを強制的にアクティブ化しようとした場合に起こる事象を評価するには、このパラメータを指定します。
-admin [-l serverpools resources all] [-x] [-a]	-l、-x および -a のなんらかの組合せを指定する場合、-admin を指定する必要があります。 -l パラメータを指定する場合、次の 3 つの出力レベルのいずれかを選択できます。 ● serverpools: 1 つのサーバー・プールで実行されているサーバーに出力を制限します ● resources: サーバー・プールのサーバーで実行されているリソースに出力を制限します ● all: 利用可能なすべての出力が表示されます -x パラメータを指定する場合、CRSCTL では差異が表示されます。 -a パラメータを指定する場合、CRSCTL ではすべてのリソースが表示されます。
-explain	妥当なコマンドの評価情報を表示するには、このパラメータを指定します。

例

このコマンドによって、次のような出力が返されます。

```
$ crsctl eval activate policy p2 -explain
Stage Group 1:
-----
Stage   Required   Action
-----
1       E   Starting to evaluate activation of policy 'p2' with server pools
      E   (MIN_SIZE, MAX_SIZE) 'Free(0, -1), sp1(2, -1), sp2(4, -1)'.
      E   Looking at other server pools to see whether MIN_SIZE value 4 of
      E   server pool 'sp2' can be met.
      E   Scanning server pools with MIN_SIZE or fewer servers in
      E   ascending order of IMPORTANCE.
      E   Considering server pools (IMPORTANCE): sp1(0) for suitable
      E   servers.
      E   Considering server pool 'sp1' because its MIN_SIZE is 2 and it
      E   has 0 servers above MIN_SIZE.
      E   Relocating server 'mjkeenan_node1_0' to server pool 'sp2'.
      E   Scanning server pools with MIN_SIZE or fewer servers in
      E   ascending order of IMPORTANCE.
      E   Considering server pools (IMPORTANCE): sp1(0) for suitable
      E   servers.
      E   Considering server pool 'sp1' because its MIN_SIZE is 2 and it
      E   has 0 servers above MIN_SIZE.
```

- サーバー・プールの動作

既存の構成ポリシーを変更します。

```
crsctl modify policy policy_name -attr "attr_name=attr_value"
```

表E-64 crsctl modify policyコマンドのパラメータ

使用上のノート

- policy_nameパラメータは必須です
- このコマンドの実行に必要な権限は、ポリシー・セットのACL属性の値によって異なります

既存のポリシーを変更するには、rootまたはOracle Clusterwareインストール所有者として次のコマンドを実行します。

crsctl status policy

構成ポリシーのステータスと定義を取得します。

構文

```
crsctl status policy [policy_name [...]] | -w "filter" | -active]
```

パラメータ

表E-65 crsctl status policyコマンドのパラメータ

パラメータ	説明
policy_name [...]	ステータスを表示するポリシーの名前または空白区切りのポリシー名のリストを指定します。
-w "filter"	または、前に -w フラグを付けて、ポリシー・フィルタを指定できます。 関連項目: 「 フィルタ 」を参照してください。
-active	または、このパラメータを指定して、アクティブなポリシーのステータスを表示できます。

使用上のノート

このコマンドの実行に必要な権限は、ポリシー・セットのACL属性の値によって異なります

crsctl create policyset

サーバー・プール構成を反映する単一のポリシー・セットを(テキスト・ファイルの形式で)作成します。

ポリシー・セットの作成後、テキスト・ファイルの内容をコピーして別のポリシー・セットを作成できます。

構文

```
crsctl create policyset -file path_to_file
```

使用上のノート

[crsctl modify policyset](#)を使用してポリシーの追加、削除、または更新を行って編集後に送り返すことができるファイルが、CRSCTLによって作成されるパスを指定します。

例

ポリシー・セットを作成するには、次のコマンドを実行します。

```
$ crsctl create policyset -file /tmp/ps
```

crsctl modify policyset

既存のポリシー・セットを変更します。

構文

```
crsctl modify policyset {-attr "attr_name=attr_value[,attr_name=attr_value[, ...]]"  
| -file file_name} [-ksp] [-f]
```

パラメータ

表E-66 crsctl modify policysetコマンドのパラメータ

パラメータ	説明
attr_name	-attr フラグの後の、変更するポリシー属性の名前。このコマンドでは、次の属性を指定できます。 • ACL • LAST_ACTIVATED_POLICY • SERVER_POOL_NAMES
attr_value	ポリシー属性の値。 ノート: attr_name パラメータおよび attr_value パラメータは、カンマで区切って二重引用符("")で囲む必要があります。たとえば: <pre>-attr "ACL='owner:mjkeenan:rwx,pgrp:svrtech:rwx, other::r--', SERVER_POOL_NAMES=sp1 sp2 Free"</pre>
-file file_name	-attr のかわりにこのパラメータを指定する場合、ポリシー・セット定義が含まれるファイルの名前を入力します。
-ksp	このパラメータを指定した場合、CRSCTL によってシステムのサーバー・プールが保持されるため、それらは独立した状態で、ポリシー・セットの管理対象にはなりません。
-f	ポリシー・セット構成を強制的に有効化するには、このパラメータを指定します。

使用上のノート

- このコマンドの実行に必要な権限は、ポリシー・セットのACL属性の値によって異なります
- -fileパラメータを使用するか、または[crsctl modify policy](#)コマンドを実行することでのみ、ポリシー定義を指定できます

例

既存のポリシー・セットを変更するには、rootまたはOracle Clusterwareインストール所有者として次のコマンドを実行します。

```
# crsctl modify policyset -file my_policy_set.def
```

crsctl status policyset

セットを変更できるユーザー、最後にアクティブ化されたポリシー、および現在有効な構成(現行ポリシー)を制御するポリシー・セット内の現在のポリシー(アクセス制御リストを含む)を取得します。

構文

```
crsctl status policyset [-file file_name]
```

使用上のノート

- このパラメータを指定することで、[crsctl modify policyset](#)を使用して複数のポリシーの追加、削除、または更新を行って編集後に送り返すことができるファイルを作成できます。このオプション・パラメータを指定しない場合、CRSCTLでは現在の構成が表示されます。
- このコマンドの実行に必要な権限は、ポリシー・セットのACL属性の値によって異なります

例

このコマンドによって、次のような出力が返されます。

```
ACL=owner:'mjkeenan:rwx,pgrp:g900:rwx,other::r--'
LAST_ACTIVATED_POLICY=DayTime
SERVER_POOL_NAMES=Free pool1 pool2 pool3
POLICY
NAME=DayTime
DESCRIPTION=Test policy
SERVERPOOL
  NAME=pool1
  IMPORTANCE=0
  MAX_SIZE=2
  MIN_SIZE=2
  SERVER_CATEGORY=
  SERVER_NAMES=
SERVERPOOL
  NAME=pool2
  IMPORTANCE=0
  MAX_SIZE=1
  MIN_SIZE=1
  SERVER_CATEGORY=
SERVERPOOL
  NAME=pool3
  IMPORTANCE=0
  MAX_SIZE=1
  MIN_SIZE=1
  SERVER_CATEGORY=
POLICY
NAME=NightTime
DESCRIPTION=Test policy
SERVERPOOL
  NAME=pool1
  IMPORTANCE=0
  MAX_SIZE=1
  MIN_SIZE=1
  SERVER_CATEGORY=
SERVERPOOL
  NAME=pool2
  IMPORTANCE=0
  MAX_SIZE=2
  MIN_SIZE=2
  SERVER_CATEGORY=
SERVERPOOL
  NAME=pool3
  IMPORTANCE=0
```

```

MAX_SIZE=1
MIN_SIZE=1
SERVER_CATEGORY=
POLICY
NAME=Weekend
DESCRIPTION=Test policy
SERVERPOOL
  NAME=pool1
  IMPORTANCE=0
  MAX_SIZE=0
  MIN_SIZE=0
  SERVER_CATEGORY=
SERVERPOOL
  NAME=pool2
  IMPORTANCE=0
  MAX_SIZE=1
  MIN_SIZE=1
  SERVER_CATEGORY=
SERVERPOOL
  NAME=pool3
  IMPORTANCE=0
  MAX_SIZE=3
  MIN_SIZE=3
  SERVER_CATEGORY=

```

crsctl check resource

特定のリソースのアプリケーション固有エージェント内でチェック・アクションを開始します。

リソース名に誤りがあるなど、何らかの理由でシステムがcheckリクエストを発行できない場合のみ、Oracle Clusterwareは出力を行います。

構文

```
crsctl check resource {resource_name [...] | -w "filter"} [-n node_name]
                    [-k cardinality_id] [-d degree_id]
```

パラメータ

表E-67 crsctl check resourceコマンドのパラメータ

パラメータ	説明
resource_name [...]	特定のリソースを指定します。リソース名の空白区切りリストを入力することで、複数のリソースをチェックできます。
-w "filter"	チェックされるリソースの数を制限するために Oracle Clusterware によって使用されるリソース・フィルタを指定します。フィルタは二重引用符("")で囲む必要があります。次にリソース・フィルタの例を示します。 ● "TYPE == cluster_resource": このフィルタは、cluster_resource タイプのリソースのみをチェックするように Oracle Clusterware を制限します。

パラメータ	説明
	● "CHECK_INTERVAL > 10": このフィルタは、CHECK_INTERVAL リソース属性に 10 より大きい値を持つリソースをチェックするように Oracle Clusterware を制限します。 ● "(CHECK_INTERVAL > 10) AND (NAME co 2)": このフィルタは、CHECK_INTERVAL リソース属性に 10 より大きい値を持ち、かつ、リソースの名前に数字 2 が含まれるリソースをチェックするように Oracle Clusterware を制限します。 関連項目: 「フィルタ」を参照してください。
-n node_name	特定のノードのリソース・インスタンスをチェックします。-n パラメータを指定しない場合、Oracle Clusterware ではローカル・サーバーのみのリソース・インスタンスがチェックされます。
-k cardinality_id	リソースのカーディナリティ ID を指定します。
-d degree_id	リソースの程度 ID を指定します。

使用上のノート

- 指定したリソースをチェックするには、それらのリソースに対する読取りおよび実行権限が必要です。
- oraで始まる名前を持つリソースはOracleリソースであるため、これらのリソースにはこのコマンドを使用しないでください。
- このコマンドの成功は、checkが発行されたということのみを意味します。CHECKアクションが完了したという意味ではありません。

例

特定のリソースのアプリケーション固有のエージェントでcheckアクションを開始するには、次のコマンドを実行します。

```
$ crsctl check resource appsvip
```

crsctl relocate resource

クラスタ内の別のサーバーにリソースを再配置します。

構文

```
crsctl relocate resource {resource_name [-k cid] | {resource_name | -all}
  -s source_server | -w "filter"} [-n destination_server] [-env
"env1=val1,env2=val2,..."]
  [-i] [-f]
```

パラメータ

表E-68 crsctl relocate resourceコマンドのパラメータ

パラメータ	説明
<code>resource_name [-k cid]</code>	再配置するリソースの名前を指定します。 オプションで、リソースのカーディナリティ ID を指定することもできます。このパラメータを指定すると、指定したカーディナリティのリソース・インスタンスが Oracle Clusterware によって再配置されます。
<code>resource_name -all -s source_server</code>	1 つの特定リソースを指定するか、リソースの再配置元として指定した特定のサーバー上に存在するすべてのリソースを指定できます。
<code>-w "filter"</code>	再配置されるリソースの数を制限するために Oracle Clusterware によって使用されるリソース・フィルタを指定できます。フィルタは二重引用符(" ")で囲む必要があります。次にリソース・フィルタの例を示します。 ● <code>"TYPE == cluster_resource"</code>: このフィルタは、<code>cluster_resource</code> タイプのリソースのみを再配置するように Oracle Clusterware を制限します。 ● <code>"CHECK_INTERVAL > 10"</code>: このフィルタは、<code>CHECK_INTERVAL</code> リソース属性に 10 より大きい値を持つリソースを再配置するように Oracle Clusterware を制限します。 ● <code>"(CHECK_INTERVAL > 10) AND (NAME co 2)"</code>: このフィルタは、<code>CHECK_INTERVAL</code> リソース属性に 10 より大きい値を持ち、かつ、リソースの名前に数字 2 が含まれるリソースを再配置するように Oracle Clusterware を制限します。 関連項目: 「フィルタ」を参照してください。
<code>-n destination_server</code>	リソースの再配置先サーバーの名前を指定します。宛先サーバーを指定しない場合、Oracle Clusterware では各リソースの属性プロファイルに従い、最適なサーバーにリソースが再配置されます。
<code>-env "env1=val1, env2=val2, ..."</code>	必要に応じて、このコマンドの 1 つ以上のリソース・プロファイル属性値を上書きできます。複数の環境名/値ペアを指定する場合、各ペアをカンマで区切り、リスト全体を二重引用符(" ")で

パラメータ	説明
	囲む必要があります。
-i	-i を指定すると、このコマンドの処理で Oracle Clusterware がリソースまたはその依存リソースをロック解除するのを待機する必要がある場合、コマンドはエラーを戻します。コマンドが互いに影響しないように、Oracle Clusterware によってリソースまたは他のオブジェクトがロックされることがあります。
-f	再配置対象のリソースに依存している他のリソースが実行中である場合に再配置対象のリソースを強制的に再配置するには、-f パラメータを指定します。このパラメータを使用すると、依存リソースは再配置または停止されます。 ノート: 1 より大きいカーディナリティを持つリソースを再配置する場合、-k または -s のいずれかを使用して、再配置対象のリソース・インスタンスを絞り込む必要があります。

使用上のノート

- 再配置対象のリソースを指定するには、次の3つの方法のいずれかを使用する必要があります。
 - 再配置対象の1つの特定のリソースを指定します。
 - または、特定のソース・サーバーから再配置する1つの特定のリソースまたはすべてのリソースを指定できます。
 - 3つ目の方法として、Oracle Clusterwareによって再配置対象のリソースの照合に使用されるリソース・フィルタを指定します。
- リソースの程度IDが1より大きい場合、このリソースのすべてのインスタンスがOracle Clusterwareによって再配置されます。
- 指定したリソースを再配置するには、それらのリソースに対する読取りおよび実行権限が必要です。
- oraで始まる名前を持つリソースはOracleリソースであるため、これらのリソースにはこのコマンドを使用しないでください。

例

あるサーバーから別のサーバーに1つの特定のリソースを再配置するには、次のコマンドを実行します。

```
# crsctl relocate resource myResource1 -s node1 -n node3
```

crsctl get resource use

RESOURCE_USE_ENABLEDサーバー構成パラメータの現在の設定値をチェックします。

構文

```
crsctl get resource use
```

使用上のノート

可能な値は1または0です。この属性の値が1の場合(デフォルト)、サーバーをリソースの配置に使用できます。値が0の場合、Oracle Clusterwareによってサーバーでのサーバー・プール・リソースの起動が禁止されます。サーバーは空き領域サーバー・プールに属したままになります。

例

このコマンドによって、次のような出力が返されます。

```
CRS-4966: Current resource use parameter value is 1
```

crsctl set resource use

このコマンドを実行するサーバーの、RESOURCE_USE_ENABLEDサーバー構成パラメータの値を設定します。

構文

```
crsctl set resource use [1 | 0]
```

使用上のノート

- 可能な値は1または0です。この属性の値を1に設定すると(デフォルト)、サーバーをリソースの配置に使用できます。値を0に設定すると、Oracle Clusterwareによってサーバーでのサーバー・プール・リソースの起動が禁止されます。サーバーは空き領域サーバー・プールに属したままになります。
- このコマンドは、rootまたはクラスタ管理者として、またはWindowsシステムの場合、管理者として実行する必要があります。

例

RESOURCE_USE_ENABLEDサーバー構成パラメータの値を設定するには、次のコマンドを実行します。

```
# crsctl set resource use 1
```

crsctl start rollingpatch

Oracle ClusterwareおよびOracle ASMをローリング・パッチ・モードに移行します。

ノート:



crsctl start rollingpatch および crsctl stop rollingpatch は、Oracle サポートの指示がないかぎり、直接実行しないでください。これらのコマンドは、各ノードでパッチ適用を編成する rootcrs.pl -prepatch/-postpatch の一部として起動されます。

このモードでは、ノード間で異なるパッチ・レベルが許可されます。

構文

```
crsctl start rollingpatch
```

使用上のノート

- このコマンドでは、Oracle Clusterwareのローリング・パッチ状態とOracle ASMのクラスタ状態が問い合されます。いずれか一方がローリング・パッチ・モードでない場合は、適切な方法を使用してOracle ClusterwareまたはOracle ASMがローリング・パッチ・モードに移行されます。
- このコマンドの実行時にOracle ClusterwareとOracle ASMが両方ともローリング・パッチ・モードの場合、このコマン

ドは何も行いません。

- ローリング・パッチ・モードは永続的ではありません。クラスタ内のすべてのノードが停止して再起動すると、クラスタは再起動時にローリング・パッチ・モードを終了します。同様に、クラスタ内のすべてのノードでOracle Clusterwareが停止して再起動すると、ローリング・パッチ・モードは終了します。
- このコマンドをOracle ASMクライアント・クラスタ内で発行した場合、Oracle ASMはローリング・パッチ・モードに移行されません。

crsctl stop rollingpatch

Oracle ClusterwareおよびOracle ASMをローリング・パッチ・モードから移行します。

ノート:



crsctl start rollingpatch および crsctl stop rollingpatch は、Oracle サポートの指示がないかぎり、直接実行しないでください。これらのコマンドは、各ノードでパッチ適用を編成する rootcrs.pl -prepatch/-postpatch の一部として起動されます。

ローリング・パッチ・モードから移行すると、ノード間での異なるパッチ・レベルは許可されません。

構文

```
crsctl stop rollingpatch
```

使用上のノート

- このコマンドでは、Oracle Clusterwareのローリング・パッチ状態とOracle ASMのクラスタ状態が問い合わせられます。いずれか一方がローリング・パッチ・モードの場合、適切な方法を使用してOracle ClusterwareまたはOracle ASMでローリング・パッチ・モードが終了されます。
- このコマンドでは、クラスタ内のすべてのノードでパッチ・レベルが一貫していることが確認され、そうでない場合はエラーが戻されます。
- このコマンドの発行時にOracle ClusterwareとOracle ASMが両方ともローリング・パッチ・モードでない場合、このコマンドは何も行いません。
- このコマンドをOracle ASMクライアント・クラスタ内で発行した場合、Oracle ASMでローリング・パッチ・モードは終了されません。

crsctl start rollingupgrade

Oracle ClusterwareおよびOracle ASMをローリング・アップグレード・モードに移行します。

ノート:



このコマンドは、Oracle サポートの指示がないかぎり、直接呼び出さないでください。アップグレードの一部として rootupgrade スクリプトによって起動されます。

構文

```
crsctl start rollingupgrade version
```

使用上のノート

- このコマンドは、Oracle Clusterwareローリング・アップグレードの状態およびOracle ASMのクラスタ状態を問い合わせます。いずれか一方がローリング・アップグレード・モードでない場合は、適切な方法を使用してOracle ClusterwareまたはOracle ASMがローリング・アップグレード・モードに移行されます。
- このコマンドの実行時にOracle ClusterwareとOracle ASMが両方ともローリング・アップグレード・モードである場合、このコマンドは何も行いません。
- このコマンドをOracle ASMクライアント・クラスタ内で実行した場合、Oracle ASMはローリング・アップグレード・モードに移行されません。

crsctl eval add server

システムを変更せずに、サーバーの追加をシミュレートします。

このコマンドはクラスタ管理者にとって役に立つ場合があります。

構文

```
crsctl eval add server server_name [-file file_path] | [-attr
"attr_name=attr_value[,...]"
[-admin [-l level [-x] [-a]] [-explain]
```

パラメータ

表E-69 crsctl eval add serverコマンドのパラメータ

パラメータ	説明
server_name	追加するサーバーの名前。
-file file_path	サーバー属性が含まれるファイルへの完全修飾されたパス。
-attr "attr_name=attr_value[,...]"	変更するサーバー属性の名前を指定できます。名前の前には -attr フラグを付け、名前の後にはサーバー属性の値を指定します。 ノート: attr_name=attr_value の名前/値ペアは、カンマで区切って二重引用符(" ")で囲む必要があります。名前と値の複数のペアはカンマで区切ります。たとえば: -attr "MAX_SIZE=30, IMPORTANCE=25"
-admin [-l level] [-x] [-a]	このパラメータを指定する場合、CRSCTL ではクラスタ管理者用に出力が表示されます。 -l パラメータを指定する場合、次の 3 つの出力レベルのいずれかを選択できます。 ● serverpools: 1 つのサーバー・プールで実行されているサーバーに出力を制限します

パラメータ	説明
	● resources: サーバー・プールのサーバーで実行されているリソースに出力を制限します ● all: 利用可能なすべての出力が表示されます -x パラメータを指定する場合、CRSCTL では差異が表示されます。 -a パラメータを指定する場合、CRSCTL ではすべてのリソースが表示されます。 ノート: -l、-x、-a パラメータのいずれか、またはこの 3 つのなんらかの組合せを指定するには、-admin パラメータを指定する必要があります。
-explain	妥当なコマンドの評価情報を表示するには、このパラメータを指定します。

例

このコマンドによって、次のような出力が返されます。

```
$ crsctl eval add server scott_node5_0
Stage Group 1:
-----
Stage   Required   Action
-----
1       E   Looking for a suitable server pool for server 'scott_node5_0'.
      E   Scanning server pools with MIN_SIZE or more servers in
      E   descending order of IMPORTANCE.
      E   Considering server pools (IMPORTANCE): Free(0) for relocating
      E   servers.
      E   Relocating server 'scott_node5_0' to server pool 'Free'.
      Y   Server 'scott_node5_0' will be added to pools 'Free'.
-----
```

関連トピック

- [サーバー・カテゴリ属性](#)

crsctl eval delete server

システムを変更せずに、サーバーを削除した場合の影響を予測します。

このコマンドはクラスタ管理者にとって役に立つ場合があります。

構文

```
crsctl eval delete server server_name [-admin [-l level] [-x] [-a]] [-f] [-explain]
```

パラメータ

表E-70 crsctl eval delete serverコマンドのパラメータ

パラメータ	説明
server_name	削除の前に評価するサーバーの名前を指定します。
-admin [-l level] [-x] [-a]	このパラメータを指定する場合、CRSCTL ではクラスタ管理者用に出力が表示されます。 -l パラメータを指定する場合、次の 3 つの出力レベルのいずれかを選択できます。 ● serverpools: 1 つのサーバー・プールで実行されているサーバーに出力を制限します ● resources: サーバー・プールのサーバーで実行されているリソースに出力を制限します ● all: 利用可能なすべての出力が表示されます -x パラメータを指定する場合、CRSCTL では差異が表示されます。 -a パラメータを指定する場合、CRSCTL ではすべてのリソースが表示されます。 ノート: -l、-x、-a パラメータのいずれか、またはこの 3 つのなんらかの組合せを指定するには、-admin パラメータを指定する必要があります。
-explain	妥当なコマンドの評価情報を表示するには、このパラメータを指定します。

例

このコマンドによって、次のような出力が返されます。

```
$ crsctl eval delete server scott_node1_0 -explain
Stage Group 1:
-----
Stage Number    Required          Action
-----
1               Y      Resource 'cs1' (1/1) will be in state 'OFFLINE'.
               E      Server 'scott_node1_0' is removed from server pool 'sp1'.
               E      Server pool 'sp1' is below the MIN_SIZE value of 2 with 1
               E      servers.
               E      Looking at other server pools to see whether MIN_SIZE value 2 of
               E      server pool 'sp1' can be met.
               E      Scanning server pools with more than MIN_SIZE servers in
               E      ascending order of IMPORTANCE.
               E      Considering server pools (IMPORTANCE): sp2(4) for suitable
               E      servers.
               E      Considering server pool 'sp2' because its MIN_SIZE is 1 and it
               E      has 1 servers above MIN_SIZE.
```



```

E Relocating server 'scott_node1_2' to server pool 'sp1'.
Y Server 'scott_node1_2' will be moved from pool 'sp2' to pool
  'sp1'.
Y Server 'scott_node1_0' will be removed from pools 'sp1'.
Y Resource 'cs2' (1/1) will be in state 'OFFLINE'.
2 Y Resource 'cs1' (1/1) will be in state 'ONLINE|INTERMEDIATE' on
  server 'scott_node1_2'.
-----

```

関連トピック

- [サーバー・プールの動作](#)

crsctl eval relocate server

システムを変更せずに、別のサーバー・プールにサーバーを再配置した場合の影響を予測します。

このコマンドはクラスタ管理者にとって役に立つ場合があります。

構文

```
crsctl eval relocate server server_name -to server_pool_name [-f]
  [-admin [-l level] [-x] [-a]]
```

パラメータ

表E-71 crsctl eval relocate serverのパラメータ

パラメータ	説明
server_name	再配置するサーバーの名前。複数のサーバーの再配置を評価するには、空白区切りのリストでサーバーを指定できます。
-to	サーバーの再配置先のサーバー・プールの名前を指定します。
-f	このパラメータを指定する場合、CRSCTL ではサーバーを強制的に再配置する影響が予測されます。
-admin [-l level] [-x] [-a]	このパラメータを指定する場合、CRSCTL ではクラスタ管理者用に出力が表示されます。 -l パラメータを指定する場合、次の 3 つの出力レベルのいずれかを選択できます。 ● serverpools: 1 つのサーバー・プールで実行されているサーバーに出力を制限します ● resources: サーバー・プールのサーバーで実行されているリソースに出力を制限します ● all: 利用可能なすべての出力が表示されます

パラメータ	説明
	-x パラメータを指定する場合、CRSCTL では差異が表示され ます。
	-a パラメータを指定する場合、CRSCTL ではすべてのリソースが 表示されます。
	ノート: -l、-x、-a パラメータのいずれか、またはこの 3 つのなんら かの組合せを指定するには、-admin パラメータを指定する必要が あります。

例

このコマンドによって、次のような出力が返されます。

```
$ crsctl eval relocate server scott-node4-0 -to sp2
Stage Group 1:
-----
Stage   Required Action
-----
1       Y Server 'scott-node4-0' will be moved from pool 'sp1' to pool 'sp2'.
2       Y Resource 'cs2' (2/1) will be in state 'ONLINE|INTERMEDIATE' on
        server 'scott-node4-0'.
        Y Resource 'cs1' (2/1) will be in state 'ONLINE|INTERMEDIATE' on
        server 'scott-node4-0'.
-----
```

crsctl modify server

特定のサーバーの構成を変更します。

構文

```
crsctl modify server server_name -attr "attr_name=attr_value[,...]"
```

パラメータ

表E-72 crsctl modify serverコマンド・パラメータ

パラメータ	説明
server_name	変更するサーバーの名前を指定します。
-attr "attr_name=attr_value	変更するサーバー属性の名前を指定します。名前の中には- attr フラグを付け、名前の後にはサーバー属性の値を指定 します。
	ノート: attr_name=attr_value の名前/値ペアは、カン マで区切って二重引用符("")で囲む必要があります。名前 と値の複数のペアはカンマで区切ります。たとえば:
	-attr

パラメータ	説明
	"CHECK_INTERVAL=30, START_TIMEOUT=25"

使用上のノート

属性名/値ペアの属性値にカンマが含まれる場合、値は一重引用符(' ')で囲む必要があります。たとえば:

```
"START_DEPENDENCIES='hard(res1, res2, res3)'"
```

例

特定のサーバーの構成を変更するには、次のようなコマンドを入力します。

```
$ crsctl modify server email_server_2 -attr "WORKLOAD_VALUE=1"
```

crsctl relocate server

異なるサーバー・プールにサーバーを再配置します。

構文

```
crsctl relocate server server_name [...] -c server_pool_name [-i] [-f]
```

パラメータ

表E-73 crsctl relocate serverコマンドのパラメータ

パラメータ	説明
server_name [...]	再配置するサーバーの名前を指定します。複数のサーバーを再配置するには、空白区切りのリストでサーバーを指定できます。
-c server_pool_name	サーバーの再配置先のサーバー・プールの名前を指定します。
-i	-i を指定すると、Oracle Clusterware でリクエストをすぐに処理できない場合にコマンドが失敗します。
-f	-f パラメータを指定すると、Oracle Clusterware では別のサーバー・プール内のサーバーで実行中のリソースが停止され、指定したサーバー・プールにそのサーバーが再配置されます。 -f パラメータを指定しない場合、Oracle Clusterware では停止する必要があるリソースが再配置対象のサーバーにあるかどうかチェックされます。該当するものが検出された場合、Oracle Clusterware では crsctl relocate server コマンドが拒否されます。

パラメータ	説明
	ノート: サーバー・プールのサーバーの数が MIN_SIZE サーバー・プール属性の値を超えていない場合、CRSCTL によって構成違反が発生しないため、強制パラメータには効果はありません。

使用上のノート

server_nameパラメータおよび-c server_pool_nameパラメータは必須です。

例

node6サーバーおよびnode7サーバーを、それらのノード上のアクティブなリソースを阻害することなくsp1サーバー・プールに移動するには、次のコマンドを使用します。

```
$ crsctl relocate server node6 node7 -c sp1
```

crsctl status server

1つ以上の特定サーバーのステータスと構成情報を取得します。

構文

```
crsctl status server {server_name [...] | -w "filter"} [-g | -p | -v | -f] |
[-category category_name | -w "filter"]
```

パラメータ

表E-74 crsctl status serverコマンドのパラメータ

パラメータ	説明
server_name [...]	1 つ以上のサーバー名を空白区切りで指定します。
-w "filter"	表示対象のサーバーを決定するフィルタを指定できます。フィルタは二重引用符("")で囲む必要があります。値にカッコまたは空白が含まれる場合、一重引用符(' ')で囲む必要があります。たとえば、"STATE = ONLINE"と指定すると、オンラインのサーバーのみが表示されます。 関連項目: 「フィルタ」を参照してください。
-g -p -v -f	Oracle Clusterware で特定のサーバーのステータスをチェックする場合には、次のパラメータのいずれかを指定できます。 ● -g: 指定したサーバーが登録されているかどうかをチェックする場合にこのパラメータを使用します。 ● -p: 指定したサーバーの静的な構成を表示する場合

パラメータ	説明
	にこのパラメータを使用します。 ● -v: 指定したサーバーの実行時の構成を表示する場合にこのパラメータを使用します。 ● -f: 指定したサーバーのすべての構成を表示する場合にこのパラメータを使用します。
-category category_name	必要に応じて、ステータスを取得するサーバーのカテゴリを指定できます。
-w "filter"	表示対象のカテゴリを決定するフィルタを指定できます。フィルタは二重引用符("")で囲む必要があります。値にカッコまたは空白が含まれる場合、一重引用符(' ')で囲む必要があります。たとえば、"STATE = ONLINE"と指定すると、オンラインのサーバーのみが表示されます。 関連項目: 「フィルタ」を参照してください。

例

例E-7 例1

このコマンドによって、次のような出力が返されます。

```
NAME=node1
STATE=ONLINE
NAME=node2
STATE=ONLINE
```

例E-8 例2

特定のサーバーの構成全体は次のようになります。

```
NAME=node2
MEMORY_SIZE=72626
CPU_COUNT=12
CPU_CLOCK_RATE=1711
CPU_HYPERTHREADING=0
CPU_EQUIVALENCY=1000
DEPLOYMENT=other
CONFIGURED_CSS_ROLE=hub
RESOURCE_USE_ENABLED=1
SERVER_LABEL=
PHYSICAL_HOSTNAME=
STATE=ONLINE
ACTIVE_POOLS=ora.pool1
STATE_DETAILS=
ACTIVE_CSS_ROLE=hub
```

crsctl get server css_critical

CSS_CRITICALサーバー属性の現在の構成済の値を取得します。

構文

```
crsctl get server css_critical
```

使用上のノート

このコマンドにパラメータはありません。

crsctl set server css_critical

CSS_CRITICALサーバー属性の構成値を設定します。

構文

```
crsctl set server css_critical {yes | no}
```

使用上のノート

サーバーをCSS_CRITICALとして設定するには、yesを選択します。

crsctl get server label

SERVER_LABELサーバー属性の現在の設定値をチェックします。

構文

```
crsctl get server label
```

例

このコマンドによって、次のような出力が返されます。

```
CRS-4972: Current SERVER_LABEL parameter value is hubserver
```

crsctl set server label

このコマンドを実行するサーバーのSERVER_LABELサーバー構成属性の構成値を設定します。

構文

```
crsctl set server label value
```

使用上のノート

- サーバーの値を指定します。この値はbuilding_Aなどの物理的な場所や、hubserverなどのサーバーの識別情報などに影響を与える場合があります。
- 変更内容を有効にするには、ノードでOracle Clusterware技術スタックを再起動する必要があります。

crsctl add serverpool

非データベース・リソース(アプリケーション・サーバーなど)をホストするためのサーバー・プールをOracle Clusterwareに追加します。

構文

```
crsctl add serverpool server_pool_name {-file file_path |  
-attr "attr_name=attr_value[,...]" } [-i] [-f]
```

パラメータ

表E-75 crsctl add serverpoolコマンドのパラメータ

パラメータ	説明
server_pool_name	短く、わかりやすいサーバー・プール名。サーバー・プール名には、254 文字の制限があり、プラットフォームでサポートされているすべての文字(感嘆符(!)、チルダ(~)および空白を除く)を使用できます。サーバー・プール名の先頭をピリオドまたは ora にすることはできません。
-file file_path	サーバー・プールを定義する属性ファイルへの完全修飾されたパス。
-attr "attr_name=attr_value[,...]"	追加するサーバー・プールのサーバー・プール属性の名前を指定できます。名前の前には -attr フラグを付け、名前の後にはサーバー・プール属性の値を指定します。 ノート: attr_name=attr_value の名前/値ペアは、カンマで区切って二重引用符("")で囲む必要があります。名前と値の複数のペアはカンマで区切ります。たとえば: -attr "MAX_SIZE=30, IMPORTANCE=3" 使用可能な属性名は次のとおりです。 ● IMPORTANCE ● MIN_SIZE ● MAX_SIZE ● SERVER_NAMES ● PARENT_POOLS ● EXCLUSIVE_POOLS ● ACL ● SERVER_CATEGORY
-i	-i を指定すると、Oracle Clusterware でリクエストをすぐに処理できない場合にコマンドが失敗します。

パラメータ	説明
-f	-f パラメータを指定すると、Oracle Clusterware では別のサーバー・プール内のサーバーで実行中のリソースが停止され、追加対象のサーバー・プールにそのサーバーが再配置されます。-f パラメータを指定しない場合、Oracle Clusterware では、サーバー・プールを作成した結果、追加対象のサーバー・プールにサーバーを渡す別のサーバー・プール内のサーバーで、リソースが停止されるかどうかチェックされます。その場合、Oracle Clusterware では <code>crsctl add serverpool</code> コマンドが拒否されます。

使用上のノート

- `server_pool_name`パラメータは必須です。
- 属性名/値ペアの属性値にカンマが含まれる場合、値は一重引用符(' ')で囲む必要があります。
- `ora`で始まる名前を持つサーバー・プールはOracleのサーバー・プールであるため、これらのサーバー・プールにはこのコマンドを使用しないでください。
- このコマンドを実行すると、新しい構成に適合するように、Oracle Clusterwareによって他のサーバーがサーバー・プール間で再配置されることがあります。
- このコマンドは、`root`またはクラスタ管理者で実行する必要があります。
- データベース以外のリソースをホスティングするサーバー・プールを作成するには、`crsctl add serverpool`コマンドを使用します。Oracle Databaseをホスティングするサーバー・プールを作成するには、`SRVCTL`コマンド・ユーティリティを使用します。

関連項目:

SRVCTLコマンド・ユーティリティを使用したサーバー・プールの作成の詳細は、[『Oracle Real Application Clusters管理およびデプロイメント・ガイド』](#)を参照してください。

例

サーバーの最大サイズが5である`testsp`というサーバー・プールを追加するには、`root`またはOracle Clusterwareインストール所有者として次のコマンドを実行します。

```
# crsctl add serverpool testsp -attr "MAX_SIZE=5"
```

`sp1`サーバー・プールの属性値を行ごとに記載した`sp1_attr`ファイルを作成します。次に例を示します。

```
IMPORTANCE=1
MIN_SIZE=1
MAX_SIZE=2
SERVER_NAMES=node3 node4 node5
PARENT_POOLS=Generic
EXCLUSIVE_POOLS=testsp
ACL=owner:oracle:rwx,pgrp:oinstall:rwx,other::r--
```

入力として`sp1_attr`ファイルを使用して`sp1`サーバー・プールを作成するには、次のコマンドを使用します。

```
$ crsctl add serverpool sp1 -file /tmp/sp1_attr
```


crsctl delete serverpool

Oracle Clusterware構成からサーバー・プールを削除します。

構文

```
crsctl delete serverpool server_pool_name [server_pool_name [...]] [-i]
```

関連項目:

サーバー・プールおよびサーバー・プール属性の詳細は、「[サーバー・プールの動作方法](#)」を参照してください。

使用上のノート

- server_pool_nameパラメータは必須です。
複数のサーバー・プール名を区切るには、スペースを使用します。
- -iを指定すると、Oracle Clusterwareでリクエストをすぐに処理できない場合にコマンドが失敗します。
- oraで始まる名前を持つサーバー・プールはOracleのサーバー・プールであるため、これらのサーバー・プールにはこのコマンドを使用しないでください。
- このコマンドはどの環境でも使用できますが、Oracle RAC環境でのみ役立ちます。

例

サーバー・プールを削除するには、rootまたはOracle Clusterwareインストール所有者として次のコマンドを実行します。

```
# crsctl delete serverpool sp1
```

crsctl eval add serverpool

システムを変更せずに、サーバー・プールを追加した場合の影響を予測します。

このコマンドはクラスタ管理者にとって役に立つ場合があります。

構文

```
crsctl eval add serverpool server_pool_name [-file file_path | -attr  
"attr_name=attr_value[,...]"  
[-admin [-l level [-x] [-a]] [-explain] [-f]
```

パラメータ

表E-76 crsctl eval add serverpoolコマンドのパラメータ

パラメータ	説明
server_pool_name	短く、わかりやすいサーバー・プール名を指定します。
-file file_path	サーバー・プールを定義する属性ファイルへの完全修飾パスを指定します。
-attr "attr_name=attr_value[,...]"	追加するサーバー・プールのサーバー・プール属性の名前を指定できます。名前の前には-attr フラグを付け、名前の後に

パラメータ	説明
	はサーバー・プール属性の値を指定します。 ノート: attr_name=attr_value の名前/値ペアは、カンマで区切って二重引用符("")で囲む必要があります。名前と値の複数のペアはカンマで区切ります。たとえば: <pre>-attr "MAX_SIZE=30, IMPORTANCE=3"</pre> サーバー・プール属性には次のものがあります。 ● IMPORTANCE ● MIN_SIZE ● MAX_SIZE ● SERVER_NAMES ● PARENT_POOLS ● EXCLUSIVE_POOLS ● ACL
-admin [-l level] [-x] [-a]	このパラメータを指定する場合、CRSCTL ではクラスタ管理者用に出力が表示されます。 -l パラメータを指定する場合、次の 3 つの出力レベルのいずれかを選択できます。 ● serverpools: 1 つのサーバー・プールで実行されているサーバーに出力を制限します ● resources: サーバー・プールのサーバーで実行されているリソースに出力を制限します ● all: 利用可能なすべての出力が表示されます -x パラメータを指定する場合、CRSCTL では差異が表示されます。 -a パラメータを指定する場合、CRSCTL ではすべてのリソースが表示されます。 ノート: -l、-x、-a パラメータのいずれか、またはこの 3 つの

パラメータ	説明
	なんらかの組合せを指定するには、-admin パラメータを指定する必要があります。
-explain	妥当なコマンドの評価情報を表示するには、このパラメータを指定します。
-f	このパラメータを指定する場合、CRSCTL ではサーバー・プールを強制的に追加する影響が予測されます。

例

次の例では、sp1というサーバー・プールの追加を評価します。

```
$ crsctl eval add serverpool sp1 -explain
Stage Group 1:
-----
Stage Required Action
-----
1      E   Starting to look for server pools that will donate servers to be
      E   reallocated to the new pool 'sp1'.
      E   Scanning server pools with more than MIN_SIZE servers in
      E   ascending order of IMPORTANCE.
      E   Considering server pools (IMPORTANCE): Free(0) for suitable
      E   servers.
      E   Considering server pool 'Free' because its MIN_SIZE is 0 and it
      E   has 4 servers above MIN_SIZE.
      E   Relocating server 'scott_node1_0' to server pool 'sp1'.
      E   Scanning server pools with more than MIN_SIZE servers in
      E   ascending order of IMPORTANCE.
      E   Considering server pools (IMPORTANCE): Free(0) for suitable
      E   servers.
      E   Considering server pool 'Free' because its MIN_SIZE is 0 and it
      E   has 3 servers above MIN_SIZE.
      E   Relocating server 'scott_node1_1' to server pool 'sp1'.
      E   Scanning server pools with more than MIN_SIZE servers in
      E   ascending order of IMPORTANCE.
      E   Considering server pools (IMPORTANCE): Free(0) for suitable
      E   servers.
      E   Considering server pool 'Free' because its MIN_SIZE is 0 and it
      E   has 2 servers above MIN_SIZE.
      E   Relocating server 'scott_node1_2' to server pool 'sp1'.
      E   Scanning server pools with more than MIN_SIZE servers in
      E   ascending order of IMPORTANCE.
      E   Considering server pools (IMPORTANCE): Free(0) for suitable
      E   servers.
      E   Considering server pool 'Free' because its MIN_SIZE is 0 and it
      E   has 1 servers above MIN_SIZE.
      E   Relocating server 'scott_node1_3' to server pool 'sp1'.
      E   Scanning server pools with MIN_SIZE or fewer servers in
      E   ascending order of IMPORTANCE.
      Y   Server 'scott_node1_2' will be moved from pool 'Free' to pool
      Y   'sp1'.
      Y   Server 'scott_node1_0' will be moved from pool 'Free' to pool
      Y   'sp1'.
      Y   Server 'scott_node1_1' will be moved from pool 'Free' to pool
      Y   'sp1'.
      Y   Server 'scott_node1_3' will be moved from pool 'Free' to pool
      Y   'sp1'.
-----
```

関連トピック

- [サーバー・プールの動作](#)

crsctl eval delete serverpool

システムを変更せずに、サーバー・プールの削除をシミュレートします。

このコマンドはクラスタ管理者にとって役に立つ場合があります。

構文

```
crsctl eval delete serverpool server_pool_name [-admin [-l level] [-x] [-a]] [-explain]
```

パラメータ

表E-77 crsctl eval delete serverpoolのパラメータ

パラメータ	説明
server_pool_name	削除するサーバー・プールの名前。
-admin [-l level] [-x] [-a]	このパラメータを指定する場合、CRSCTL ではクラスタ管理者用に出力が表示されます。 -l パラメータを指定する場合、次の 3 つの出力レベルのいずれかを選択できます。 ● serverpools: 1 つのサーバー・プールで実行されているサーバーに出力を制限します ● resources: サーバー・プールのサーバーで実行されているリソースに出力を制限します ● all: 利用可能なすべての出力が表示されます -x パラメータを指定する場合、CRSCTL では差異が表示されます。 -a パラメータを指定する場合、CRSCTL ではすべてのリソースが表示されます。 ノート: -l、-x、-a パラメータのいずれか、またはこの 3 つのなんらかの組合せを指定するには、-admin パラメータを指定する必要があります。
-explain	妥当なコマンドの評価情報を表示するには、このパラメータを指定します。

例

次のサーバー・プール構成について考えてみます。

```
NAME=Free
ACTIVE_SERVERS=
NAME=Generic
ACTIVE_SERVERS=
NAME=sp1
ACTIVE_SERVERS=scott-node2-0 scott-node2-3
NAME=sp2
ACTIVE_SERVERS=scott-node2-1
```

次のコマンドは、サーバー・プールsp1を削除した場合の影響を評価します。

```
$ crsctl eval delete sp sp1
Stage Group 1:
-----
Stage Required Action
-----
1 Y Server 'scott-node2-0' will be moved from pool 'sp1' to pool
   'sp2'.
   Y Server 'scott-node2-3' will be moved from pool 'sp1' to pool
   'sp2'.
-----
```

-explainパラメータを前述のコマンドに追加すると、次のような妥当なコマンドの評価が返されます。

```
$ crsctl eval delete sp sp1 -explain
Stage Group 1:
-----
Stage Required Action
-----
1 E Starting to look for server pools to reallocate servers from
   pool 'sp1'.
   E Looking for a suitable server pool for server
   'scott-node2-0'.
   E Scanning server pools with fewer than MIN_SIZE servers in
   descending order of IMPORTANCE.
   E Considering server pools (IMPORTANCE): sp2(1) for relocating
   servers.
   E Relocating server 'scott-node2-0' to server pool 'sp2'.
   E Looking for a suitable server pool for server
   'scott-node2-3'.
   E Scanning server pools with fewer than MIN_SIZE servers in
   descending order of IMPORTANCE.
   E Considering server pools (IMPORTANCE): sp2(1) for relocating
   servers.
   E Relocating server 'scott-node2-3' to server pool 'sp2'.
2 Y Server 'scott-node2-0' will be moved from pool 'sp1' to pool
   'sp2'.
   Y Server 'scott-node2-3' will be moved from pool 'sp1' to pool
   'sp2'.
-----
```

関連トピック

- [サーバー・プールの動作](#)

crsctl eval modify serverpool

システムに変更を加えずに、サーバー・プールの追加による影響を予測します。

このコマンドはクラスタ管理者にとって役に立つ場合があります。

構文

```
crsctl eval modify serverpool server_pool_name {-file file_path  
| -attr "attr_name=attr_value[,...]}  
[-f] [-admin [-l level [-x] [-a]] [-explain]
```

パラメータ

表E-78 crsctl eval modify serverpoolのパラメータ

パラメータ	説明
server_pool_name	変更するサーバー・プールの名前。
-file file_path	変更対象のサーバー・プールを定義する属性ファイルへの完全修飾されたパス。
-attr "attr_name=attr_value[,...]"	追加するサーバー・プールのサーバー・プール属性の名前を指定できます。名前の前には -attr フラグを付け、名前の後にはサーバー・プール属性の値を指定します。 ノート: attr_name=attr_value の名前/値ペアは、カンマで区切って二重引用符("")で囲む必要があります。名前と値の複数のペアはカンマで区切ります。たとえば: -attr "MAX_SIZE=30, IMPORTANCE=3" サーバー・プール属性には次のものがあります。 ● IMPORTANCE ● MIN_SIZE ● MAX_SIZE ● SERVER_NAMES ● PARENT_POOLS ● EXCLUSIVE_POOLS ● ACL
-f	このパラメータを指定する場合、CRSCTL ではサーバー・プールを強制的に変更する影響が予測されます。
-admin [-l level] [-x] [-a]	このパラメータを指定する場合、CRSCTL ではクラスタ管理者

パラメータ	説明
	用に表示されます。 -l パラメータを指定する場合、次の 3 つの出力レベルのいずれかを選択できます。 ● serverpools: 1 つのサーバー・プールで実行されているサーバーに出力を制限します ● resources: サーバー・プールのサーバーで実行されているリソースに出力を制限します ● all: 利用可能なすべての出力が表示されます -x パラメータを指定する場合、CRSCTL では差異が表示されます。 -a パラメータを指定する場合、CRSCTL ではすべてのリソースが表示されます。 ノート: -l、-x、-a パラメータのいずれか、またはこの 3 つのなんらかの組合せを指定するには、-admin パラメータを指定する必要があります。
-explain	妥当なコマンドの評価情報を表示するには、このパラメータを指定します。

使用上のノート

- server_pool_nameパラメータは必須です。
- 属性名/値ペアの属性値にカンマが含まれる場合、値は一重引用符(' ')で囲む必要があります。たとえば:

```
"START_DEPENDENCIES='hard(res1,res2,res3)'"
```
- このコマンドを実行すると、新しい構成に適合するように、Oracle Clusterwareによって他のサーバーがサーバー・プール間で再配置されることがあります。
- oraで始まる名前を持つサーバー・プールはOracleのサーバー・プールであるため、これらのサーバー・プールにはこのコマンドを使用しないでください。
- このコマンドはどの環境でも使用できますが、Oracle RAC環境でのみ役立ちます。

例

次の例では、sp1およびsp2という2つのサーバー・プールについて考えます。サーバー・プールsp1は次のように構成されます。

```
ACTIVE_SERVERS=mjkeenan-node2-0 mjkeenan-node2-3
```

サーバー・プールsp2は次のように構成されます。

```
ACTIVE_SERVERS=mjkeenan-node2-1
```

現在、sp1の重要性はsp2よりも高くなっています。次のコマンドは、sp2のIMPORTANCE属性とMAX_SIZE属性を変更した場合に何が発生するかを評価します。

```
$ crsctl eval modify serverpool sp2 -attr "MAX_SIZE=3,IMPORTANCE=5"
Stage Group 1:
-----
Stage Required Action
-----
1 Y Server 'scott-node2-0' will be moved from pool 'sp1' to pool
    'sp2'.
    Y Server 'scott-node2-3' will be moved from pool 'sp1' to pool
    'sp2'.
-----
```

-explainパラメータを前述のコマンドに追加すると、次のような妥当なコマンドの評価が返されます。

```
$ crsctl eval modify serverpool sp2 -attr "MAX_SIZE=3,IMPORTANCE=5" -explain
Stage Group 1:
-----
Stage Required Action
-----
1 E Starting to look at server pools due to a configuration change
    in the pool 'sp2'.
    E Looking at other server pools to see whether MIN_SIZE value 3 of
    server pool 'sp2' can be met.
    E Scanning server pools with MIN_SIZE or fewer servers in
    ascending order of IMPORTANCE.
    E Considering server pools (IMPORTANCE): sp1(3) for suitable
    servers.
    E Considering server pool 'sp1' because its MIN_SIZE is 2 and it
    has 0 servers above MIN_SIZE.
    E Relocating server 'scott-node2-0' to server pool 'sp2'.
    E Scanning server pools with MIN_SIZE or fewer servers in
    ascending order of IMPORTANCE.
    E Considering server pools (IMPORTANCE): sp1(3) for suitable
    servers.
    E Considering server pool 'sp1' because its MIN_SIZE is 2 and it
    has 0 servers above MIN_SIZE.
    E Relocating server 'scott-node2-3' to server pool 'sp2'.
    Y Server 'scott-node2-0' will be moved from pool 'sp1' to pool
    'sp2'.
    Y Server 'scott-node2-3' will be moved from pool 'sp1' to pool
    'sp2'.
-----
```

関連トピック

- [サーバー・プールの動作](#)

crsctl getperm serverpool

特定のサーバー・プールの権限を取得します。

構文

```
crsctl getperm serverpool server_pool_name [-u user_name | -g group_name]
```

関連項目:

サーバー・プールおよびサーバー・プール属性の詳細は、「[サーバー・プールの動作方法](#)」を参照してください。

パラメータ

表E-79 crsctl getperm serverpoolコマンドのパラメータ

パラメータ	説明
server_pool_name	権限を取得するサーバー・プールの名前を指定します。
-u user_name	-u を指定すると、Oracle Clusterware では特定のユーザーの権限が取得されます。
-g group_name	-g を指定すると、Oracle Clusterware では特定のグループの権限が取得されます。

使用上のノート

- server_pool_nameパラメータは必須です。
- oraで始まる名前を持つサーバー・プールはOracleのサーバー・プールであるため、これらのサーバー・プールにはこのコマンドを使用しないでください。
- このコマンドはどの環境でも使用できますが、Oracle RAC環境でのみ役立ちます。

例

このコマンドによって、次のような出力が返されます。

```
$ crsctl getperm serverpool sp1
NAME: sp1
owner:root:rwx,pgrp:root:r-x,other::r--
```

crsctl modify serverpool

既存のサーバー・プールを変更します。

構文

```
crsctl modify serverpool server_pool_name -attr "attr_name=attr_value[, ...]"
[-policy policyName | -all_policies] [-i] [-f]
```

パラメータ

表E-80 crsctl modify serverpoolコマンドのパラメータ

パラメータ	説明
server_pool_name	変更するサーバー・プールの名前を指定します。
-attr "attr_name=attr_value[, ...]"	変更するサーバー・プール属性の名前を指定します。名前の前には -attr フラグを付け、名前の後にはサーバー・プール属性の値を指定します。
	ノート: attr_name=attr_value の名前/値ペアは、カンマ

パラメータ	説明
	で区切って二重引用符(" ")で囲む必要があります。名前と値の複数のペアはカンマで区切ります。たとえば: <pre>-attr "MAX_SIZE=30, IMPORTANCE=3"</pre> 使用可能な属性名は次のとおりです。 ● IMPORTANCE ● MIN_SIZE ● MAX_SIZE ● SERVER_NAMES ● PARENT_POOLS ● EXCLUSIVE_POOLS ● ACL ● SERVER_CATEGORY
-policy policyName -all_policies	サーバー・プール定義を変更する特定のポリシーまたはすべてのポリシーを指定します。
-i	-i を指定すると、Oracle Clusterware でリクエストをすぐに処理できない場合にコマンドが失敗します。
-f	-f パラメータを指定すると、Oracle Clusterware では別のサーバー・プール内のサーバーで実行中のリソースが停止され、変更対象のサーバー・プールにそのサーバーが再配置されます。 -f パラメータを指定しない場合、Oracle Clusterware では、サーバー・プールを作成した結果、変更対象のサーバー・プールにサーバーを渡す別のサーバー・プール内のサーバーで、リソースが停止されるかどうかチェックされます。その場合、Oracle Clusterware では crsctl modify serverpool コマンドが拒否されます。

関連項目:

サーバー・プールおよびサーバー・プール属性の詳細は、「[サーバー・プールの動作方法](#)」を参照してください。

使用上のノート

- server_pool_nameパラメータは必須です。
- 属性名/値ペアの属性値にカンマが含まれる場合、値は一重引用符(' ')で囲む必要があります。たとえば:

```
"START_DEPENDENCIES='hard(res1,res2,res3)'"
```
- このコマンドを実行すると、新しい構成に適合するように、Oracle Clusterwareによって他のサーバーがサーバー・プール間で再配置されることがあります。
- oraで始まる名前を持つサーバー・プールはOracleのサーバー・プールであるため、これらのサーバー・プールにはこのコマンドを使用しないでください。
- このコマンドはどの環境でも使用できますが、Oracle RAC環境でのみ役立ちます。

例

既存のサーバー・プールを変更するには、rootまたはOracle Clusterwareインストール所有者として次のコマンドを実行します。

```
# crsctl modify serverpool sp1 -attr "MAX_SIZE=7"
```

crsctl setperm serverpool

特定のサーバー・プールの権限を設定します。

構文

```
crsctl setperm serverpool server_pool_name {-u acl_string | -x acl_string |  
-o user_name | -g group_name}
```

パラメータ

表E-81 crsctl setperm serverpoolコマンドのパラメータ

パラメータ	説明
server_pool_name	権限を設定するサーバー・プールの名前を指定します。
{-u -x -o -g}	サーバー・プールには次のパラメータのうち 1 つのみを指定できます。 ● -u acl_string: サーバー・プールのアクセス制御リスト(ACL)を更新できます。 ● -x acl_string: サーバー・プールの ACL を削除できます。 ● -o user_name: ユーザー名を入力することによってサーバー・プールの所有者を変更できます。 ● -g group_name: グループ名を入力することによってサーバー・プールのプライマリ・グループを変更できます。

パラメータ	説明
	ユーザー、グループ、他の ACL 文字列は次のように指定します。 <pre>user:user_name[:readPermwritePermexecPerm] group:group_name[:readPermwritePermexecPerm] other[:readPermwritePermexecPerm]</pre> ● user: ユーザー ACL ● group: グループ ACL ● other: その他の ACL ● readPerm: サーバー・プールに対する読取り権限。文字 r はユーザー、グループ、その他の読取り権限を付与し、マイナス記号(-)は読取り権限を付与しません。 ● writePerm: サーバー・プールに対する書込み権限。文字 w はユーザー、グループ、その他の書込み権限を付与し、マイナス記号(-)は書込み権限を付与しません。 ● execPerm: サーバー・プールに対する実行権限。文字 x はユーザー、グループ、その他の実行権限を付与し、マイナス記号(-)は実行権限を付与しません。

使用上のノート

- **server_pool_name**パラメータは必須です。
- **ora**で始まる名前を持つサーバー・プールはOracleのサーバー・プールであるため、これらのサーバー・プールにはこのコマンドを使用しないでください。
- このコマンドはどの環境でも使用できますが、Oracle RAC環境でのみ役立ちます。

例

サーバー・プールに対する読取り、書込み、および実行権限をユーザーJane Doeに付与するには、次のコマンドを実行します。

```
$ crsctl setperm serverpool sp3 -u user:jane.doe:rwX
```

crsctl status serverpool

1つ以上の特定サーバー・プールのステータスと構成情報を取得します。

構文

```
crsctl status serverpool [server_pool_name [...] | -w "filter"] [-p | -v | -f]
crsctl status serverpool {[server_pool_name [...]]} -g
```

パラメータ

表E-82 crsctl status serverpoolコマンドのパラメータ

パラメータ	説明

パラメータ	説明
server_pool_name [...] -g	特定のサーバー・プールを識別する場合には、1 つ以上のサーバー・プール名を空白区切りで指定します。 -g: 指定したサーバー・プールが登録されているかどうかをチェックする場合にこのパラメータを使用します。 ノート: サーバー・プール名を指定した後に、-g パラメータをその他のパラメータとともに使用することはできません
-w "filter"	このパラメータを使用すると、サーバー・プールを名前で指定するかわりに、MIN_SIZE > 3 などのフィルタを二重引用符(" ")で囲んで指定できます。特定の文字を使用してサーバー・プールを識別する場合にこのパラメータを使用します。 関連項目: 「フィルタ」を参照してください。
-p -v -f	必要に応じて、次のパラメータのいずれかを指定できます。 ● -p: 指定したサーバー・プールの静的な構成を表示する場合にこのパラメータを使用します。 ● -v: 指定したサーバー・プールの実行時の構成を表示する場合にこのパラメータを使用します。 ● -f: 指定したサーバー・プールのすべての構成を表示する場合にこのパラメータを使用します。

使用上のノート

- server_pool_nameパラメータまたはフィルタは必須です。
- oraで始まる名前を持つサーバー・プールはOracleのサーバー・プールであるため、これらのサーバー・プールにはこのコマンドを使用しないでください。
- このコマンドはどの環境でも使用できますが、Oracle RAC環境でのみ役立ちます。

例

例E-9 例1

サーバー・プールsp1のすべての構成を表示するには、次のコマンドを実行します。

```
$ crsctl status serverpool sp1 -f
NAME=sp1
IMPORTANCE=1
MIN_SIZE=0
MAX_SIZE=-1
SERVER_NAMES=node3 node4 node5
PARENT_POOLS=Generic
EXCLUSIVE_POOLS=
```

```
ACL=owner:oracle:rwx,pgrp:oinstall:rwx,other::r--
SERVER_CATEGORY=ora.hub.category
ACTIVE_SERVERS=node3 node4
```

例E-10 例2

すべてのサーバー・プールおよびそれらに関連付けられたサーバーを表示するには、次のコマンドを使用します。

```
$ crsctl status serverpool
NAME=Free
ACTIVE_SERVERS=
NAME=Generic
ACTIVE_SERVERS=node1 node2
NAME=ora.orcl
ACTIVE_SERVERS=node1 node2
NAME=sp1
ACTIVE_SERVERS=node3 node4
```

例E-11 例3

特定の基準を満たすサーバー・プールを検索するには、次のコマンドを使用します。

```
$ crsctl status serverpool -w "MAX_SIZE > 1"
NAME=sp2
ACTIVE_SERVERS=node3 node4
```

crsctl query socket udp

指定されたアドレスとポートでデーモンがリスニングできることを検証します。

構文

```
crsctl query socket udp [-address address] [-port port]
```

パラメータ

表E-83 crsctl query socket udpコマンドのパラメータ

パラメータ	説明
-address address	ソケットを作成する IP アドレスを指定します。アドレスを指定しない場合、CRSCTL はデフォルトとしてローカル・ホストを想定します。
-port port	ソケットを作成するポートを指定します。ポート番号を指定しない場合、CRSCTL はデフォルトとして 53 を想定します。

使用上のノート

1024より小さいポート番号を確認するには、このコマンドをrootとして実行する必要があります。

例

コマンドは以下のような出力を返します。

```
$ crsctl query socket udp
CRS-10030: could not verify if port 53 on local node is in use
# crsctl query socket udp
CRS-10026: successfully created socket on port 53 on local node
```

前述の2つのコマンドのうち最初のコマンドはrootとして実行されておらず、いずれのコマンドでもポートが指定されていないため、CRSCTLは、デフォルトを53として想定しました(1024未満)。この条件では、コマンドをrootとして実行する必要があります。

```
$ crsctl query socket udp -port 1023
CRS-10030: could not verify if port 1023 on local node is in use
# crsctl query socket udp -port 1023
CRS-10026: successfully created socket on port 1023 on local node
```

最初の2つの例と同様に、前述の2つのコマンドのうち最初のコマンドはrootとして実行されておらず、ポート番号は指定されていますが1024未満のため、root権限でコマンドを実行する必要があります。

次の最後の例では、1024より大きいポート番号が指定されているため、コマンドをrootとして実行する必要はありません。

```
$ crsctl query socket udp -port 1028
CRS-10026: successfully created socket on port 1028 on local node
```

crsctl start testdns

指定されたIPアドレスとポートでリスニングするテストDNSサーバーを起動します。

テストDNSサーバーは、受信したパケットを表示しますが、受信パケットに応答することはありません。通常、このコマンドは、ドメイン転送がGNSドメイン用に正しく設定されていることを確認するために使用します。

構文

```
crsctl start testdns [-address address [-port port]] [-once] [-v]
```

パラメータ

表E-84 crsctl start testdnsコマンドのパラメータ

パラメータ	説明
-address address	サーバー・アドレスを IP_address/netmask[/interface_name]の書式 で指定します。
-port port	サーバーがリスニングするポート。ポートを指定しないと、デフォルトのポート 53 が使用されます。
-once	DNS サーバーが 1 つの DNS 問合せパケットを受信した後に 終了する必要がある場合、このパラメータを指定します。
-v	冗長出力。

例

ローカル・サーバーでテストDNSサーバーを起動するには、次のようなコマンドを実行します。

```
$ crsctl start testdns -address 192.168.29.218 -port 63 -v
```

crsctl status testdns

指定されたアドレスとローカル・ホスト名で実行されているテストDNSサーバーを問い合わせます。

構文

```
crsctl status testdns [-server DNS_server_address] [-port port] [-v]
```

パラメータ

表E-85 crsctl status testdnsコマンドのパラメータ

パラメータ	説明
-server DNS_server_address	必要に応じて、ステータスを確認する DNS サーバーのアドレスを指定できます。
-port port	必要に応じて、DNS サーバーがリスニングするポートを指定できます。ポートを指定しないと、デフォルトのポート 53 が使用されます。
-v	冗長出力。

例

このコマンドによって、次のような出力が返されます。

```
CRS-10024: DNS server returned 192.168.28.74 for name  
stmjk07-vip.stmjk0462.foo.com
```

crsctl stop testdns

テストDNSサーバーを停止します。

構文

```
crsctl stop testdns [-address address] [-port port] [-v]
```

パラメータ

表E-86 crsctl stop testdnsコマンドのパラメータ

パラメータ	説明
-address address	必要に応じて、テスト DNS サーバーを起動したサーバー・アドレスを、IP_address/netmask[/interface_name]の形式で指定できます。
-port port	必要に応じて、DNS サーバーがリスニングするポートを指定できます。ポートを指定しないと、デフォルトのポート 53 が使用されます。

パラメータ	説明
-v	冗長出力。

例

このコマンドによって、次のような出力が返されます。

```
CRS-10032: successfully stopped the DNS listening daemon running on port 53 on
local node
```

crsctl replace votedisk

既存の投票ファイルを移動または置換します。

このコマンドを使用すると、Oracle ASMであれ他の記憶域オプションであれ、指定した場所に投票ファイルが作成されます。Oracle Clusterwareによって既存の投票ファイル情報が新しい場所にコピーされ、以前の場所からその投票ファイルが削除されます。

構文

```
crsctl replace votedisk [+asm_disk_group | path_to_voting_disk [...]]
```

パラメータ

表E-87 crsctl replace votediskコマンドのパラメータ

パラメータ	説明
+asm_disk_group	投票ファイルの配置先となる Oracle ASM ディスク・グループを(前にプラス記号(+))を付けて)指定します。
path_to_voting_disk [...]	Oracle ASM の外部に存在する投票ファイルのパスを空白区切りのリストで指定できます。

使用上のノート

- このコマンドを実行するには、root、Oracle Clusterwareインストール所有者または管理者グループのメンバーである必要があります。
- 投票ファイルを置換する場合は、Oracle ASMディスク・グループや他のストレージ・デバイスに存在するものを指定します。
- クラスタ内の任意のノードでこのコマンドを実行できます。

例

例E-12 例1

Oracle ASM内に配置されている投票ファイルを置換するには、次のコマンドを実行します。

```
# crsctl replace votedisk +diskgroup1
```

例E-13 例2

共有ファイル・システムに配置されている投票ファイルを置換するには、次のコマンドを実行します。

```
# crsctl replace votedisk /mnt/nfs/disk1 /mnt/nfs/disk2
```

crsctl add wallet

ユーザーを作成してウォレットに追加します。

構文

```
crsctl add wallet -type wallet_type [-name name] [-user user_name -passwd]
```

パラメータ

表E-88 crsctl add walletコマンドのパラメータ

パラメータ	説明
-type wallet_type	作成するウォレットのタイプを指定します(APPQOSADMIN、APPQOSUSER、APPQOSDB、MGMTDB、OSUSER、CVUDB など)。 ● OSUSER: このウォレット・タイプは、ポリシー管理データベースで Windows サービスを作成する際、または Windows サービスのパスワードを更新するために一般的にエージェントによって使用される、権限の低い Windows ユーザーのユーザー名とパスワードを格納します。 ● CVUDB: このウォレット・タイプは、CVU のヘルス・チェック・コンポーネントがデータベースに接続してデータベース・チェックを実行するために使用するデータベース・ユーザー名とパスワードを格納します。
-name name	APPQOSDB ウォレットや CVUDB ウォレットを作成するには、ウォレットの名前を指定する必要があります。
-user user_name -passwd	ウォレットに追加するユーザー名を指定して、標準入力からパスワードを指定します。OSUSER ウォレットを作成するにはユーザー名が必要です。

使用上のノート

ポリシー管理データベースを使用している場合、ウォレットが必要です。それ以外の場合はウォレットはオプションです。

例

ウォレットを追加するには、次のコマンドを実行します。

```
$ crsctl add wallet -type OSUSER -user lp_oracle_home_user -passwd
```

前述の例で、lp_oracle_home_userは、ポリシー管理データベースを作成したホームを所有する権限が低いOracleホーム・ユーザーです。

crsctl delete wallet

ウォレットまたはユーザーをウォレットから削除します。

構文

```
crsctl delete wallet -type wallet_type [-name name] [-user user_name]
```

パラメータ

表E-89 crsctl delete walletコマンドのパラメータ

パラメータ	説明
-type wallet_type	削除するウォレットのタイプ(APPQOSADMIN、APPQOSUSER、APPQOSDB、MGMTDB、OSUSER、CVUDB など)。 ● OSUSER: このウォレット・タイプは、ポリシー管理データベースで Windows サービスを作成する際、または Windows サービスのパスワードを更新するために一般的にエージェントによって使用される、権限の低い Windows ユーザーのユーザー名とパスワードを格納します。 ● CVUDB: このウォレット・タイプは、CVU のヘルス・チェック・コンポーネントがデータベースに接続してデータベース・チェックを実行するために使用するデータベース・ユーザー名とパスワードを格納します。
-name name	APPQOSDB ウォレットを削除するには、ウォレットの名前を指定する必要があります。
-user user_name	OSUSER ウォレットからユーザーを削除するには、ユーザー名を指定する必要があります。

例

OSUSERウォレットからユーザーを削除するには、次のコマンドを実行します。

```
$ crsctl delete wallet -type OSUSER -user lp_oracle_home_user
```

前述の例で、lp_oracle_home_userは、ポリシー管理データベースを作成したホームを所有する権限が低いOracleホーム・ユーザーです。また、ウォレットに他のユーザーが含まれている場合、コマンドを実行してもウォレットは削除されません。

crsctl modify wallet

特定のウォレット内の特定のユーザーのパスワードを変更します。

構文

```
crsctl modify wallet -type wallet_type [-name name] [-user user_name -passwd]
```

パラメータ

表E-90 crsctl modify walletコマンドのパラメータ

パラメータ	説明
-type wallet_type	変更するウォレットのタイプを指定します(APPQOSADMIN、APPQOSUSER、APPQOSDB、MGMTDB、OSUSER、CVUDB など)。 ● OSUSER: このウォレット・タイプは、ポリシー管理データベースで Windows サービスを作成する際、または Windows サービスのパスワードを更新するために一般的にエージェントによって使用される、権限の低い Windows ユーザーのユーザー名とパスワードを格納します。 ● CVUDB: このウォレット・タイプは、CVU のヘルス・チェック・コンポーネントがデータベースに接続してデータベース・チェックを実行するために使用するデータベース・ユーザー名とパスワードを格納します。
-name name	APPQOSDB ウォレットを変更するには、ウォレット名を指定する必要があります。
-user user_name -passwd	パスワードを変更するユーザーの名前を指定する必要があります。標準入力からパスワードを変更します。

使用上のノート

このコマンドを使用してユーザー名を変更することはできません。

例

権限が低いOracleホーム・ユーザーのパスワードを変更するには、次のコマンドを実行します。

```
$ crsctl modify wallet -type OSUSER -user lp_oracle_home_user -passwd
```

crsctl query wallet

ウォレットから権限の低いユーザーを問い合わせます。

構文

```
crsctl query wallet -type wallet_type [-name name] [-user user_name] [-all]
```

パラメータ

表E-91 crsctl query walletコマンドのパラメータ

パラメータ	説明
-type wallet_type	問い合わせるウォレットのタイプ(APPQOSADMIN、APPQOSUSER、APPQOSDB、MGMTDB、OSUSER、CVUDB など)。 ● OSUSER: このウォレット・タイプは、ポリシー管理データベースで Windows サービスを作成する際、または Windows サービスのパスワードを更新するために一般的にエージェントによって使用される、権限の低い Windows ユーザーのユーザー名とパスワードを格納します。 ● CVUDB: このウォレット・タイプは、CVU のヘルス・チェック・コンポーネントがデータベースに接続してデータベース・チェックを実行するために使用するデータベース・ユーザー名とパスワードを格納します。
-name name	APPQOSDB ウォレットを問い合わせるには、ウォレットの名前を指定する必要があります。
-user user_name	OSUSER ウォレットのユーザーを問い合わせるには、ユーザー名を指定する必要があります。
-all	-all を指定すると、特定のウォレットのすべてのユーザーが表示されます。

例

OSUSERウォレット内のすべてのユーザーを表示するには、次のコマンドを実行します。

```
$ crsctl query wallet -type OSUSER -all
```

Oracle Restart環境のCRSCTLコマンド

次のコマンドは、Oracle Restart環境でのみ使用できます。

この項に示すコマンドは、Oracle高可用性サービスを制御します。これらのコマンドは、Oracle高可用性サービス・デーモン(ohasd)、Oracle ASM(インストールされている場合)およびクラスタ同期サービス(Oracle ASMがインストールされている場合)で構成される、Oracle Restart環境のOracle高可用性サービス・スタックを管理します。これらのコマンドは、コマンドを実行するローカル・サーバーにのみ影響します。

ノート:



Oracle では、Oracle Restart 環境での `crs_*` コマンドの使用はサポートされていません。

クラスタ内の各サーバーの状態は、次の2つのいずれかです。

- スタック全体が起動している状態。Oracle高可用性サービスがアクティブであることを意味します。
- スタック全体が停止している状態。Oracle高可用性サービスが非アクティブであることを意味します。

crsctl check has

ohasdのステータスをチェックします。

構文

```
crsctl check has
```

例

このコマンドによって、次のような出力が返されます。

```
CRS-4638: Oracle High Availability Services is online
```

crsctl config has

サーバー上のOracle高可用性サービス・スタックの自動起動構成を表示します。

構文

```
crsctl config has
```

使用上のノート

このコマンドにパラメータはありません。

例

このコマンドによって、次のような出力が返されます。

```
CRS-4622 Oracle High Availability Services autostart is enabled.
```

crsctl disable has

サーバー起動時のOracle高可用性サービス・スタックの自動起動を無効化します。

構文

```
crsctl disable has
```

例

このコマンドによって、次のような出力が返されます。

```
CRS-4621 Oracle High Availability Services autostart is disabled.
```

crsctl enable has

サーバー起動時のOracle高可用性サービス・スタックの自動起動を有効化します。

構文

```
crsctl enable has
```

例

このコマンドによって、次のような出力が返されます。

```
CRS-4622 Oracle High Availability Services autostart is enabled.
```

crsctl query has releaseversion

ローカル・ノード上のバイナリに格納されているOracle Clusterwareソフトウェアのリリース・バージョンを表示します。

構文

```
crsctl query has releaseversion
```

例

このコマンドによって、次のような出力が返されます。

```
Oracle High Availability Services release version on the local node is [12.2.0.1.0]
```

crsctl query has softwareversion

ローカル・ノード上のソフトウェア・バージョンを表示します。

構文

```
crsctl query has softwareversion [server_name]
```

使用上のノート

サーバー名を指定しない場合、Oracle Clusterwareではローカル・サーバーで実行中のOracle Clusterwareのバージョンが表示されます。

例

このコマンドによって、次のような出力が返されます。

```
Oracle High Availability Services version on the local node is [12.2.0.1.0]
```

crsctl start has

ローカル・サーバーでOracle高可用性サービスを起動します。

構文

```
crsctl start has [-nowait] [-noautostart] [-wait]
```

使用上のノート

- -nowaitパラメータを使用して、Oracle高可用性サービスの起動の待機をスキップします。

- Oracle高可用性サービスのみを開始するには、-noautorestartパラメータを使用します。
- -waitパラメータを使用して、Oracle高可用性サービスの起動の待機します。

例

ローカル・サーバーのOracle高可用性サービスを起動するには、次のコマンドを実行します。

```
# crsctl start has
```

crsctl stop has

ローカル・サーバーでOracle高可用性サービスを停止します。

このコマンドでは、Oracle高可用性サービスの停止試行中、Oracle Clusterwareによって管理されるリソースの正常な停止が試行されます。

構文

```
crsctl stop has [-f]
```

使用上のノート

Oracle Clusterwareによって管理される任意のリソースが、crsctl stop hasコマンドの実行後も実行されている場合、コマンドは失敗します。すべてのリソースを無条件で停止し、Oracle高可用性サービスを停止するには、-fパラメータを使用します。

例

ローカル・サーバーのOracle高可用性サービスを停止するには、次のコマンドを実行します。

```
# crsctl stop has
```

トラブルシューティングおよび診断出力

rootユーザーとしてcrsctl set logコマンドを使用して、クラスタ・レディ・サービス(CRS)、クラスタ同期サービス(CSS)、イベント・マネージャ(EVM)およびクラスタウェア・サブコンポーネントの動的デバッグを有効にできます。

crsctl debugコマンドを使用して、デバッグ・レベルを動的に変更できます。デバッグ情報は、次回の起動で使用するために、Oracle Cluster Registry(OCR)に保存されます。リソースのデバッグを有効にすることもできます。

この項には次のトピックが含まれます：

- [crsctl set logを使用した動的デバッグ](#)
- [コンポーネント・レベルのデバッグ](#)

crsctl set logを使用した動的デバッグ

Oracle Clusterwareのログ・レベルを設定します。

構文

```
crsctl set log module_name "component1=level1,..."
```

使用上のノート

- 次のモジュールの様々なコンポーネントにログ・レベルを設定できます。

- mdns: マルチキャスト・ドメイン・ネーム・サーバー
- gpnp: グリッド・プラグ・アンド・プレイ・サービス
- css: クラスタ同期サービス
- crf: クラスタ状態モニター
- crs: クラスタ・レディ・サービス
- ctss: クラスタ時刻同期化サービス
- evm: Event Manager
- gipc: グリッド・プロセス間通信

各モジュールのコンポーネントのリストを取得するには、`crsctl lsmodules`コマンドを使用します。

- 二重引用符("")で囲まれたコンポーネント名/ログ・レベルの複数のペアをカンマ区切りリストで指定できます。
- Oracle Clusterware 11g リリース2 (11.2.0.3)以上では、コンポーネント名/ログ・レベルのペアは等号(=)で分割します。以前のバージョンのOracle Clusterwareでは、コロン(:)が使用されていました。
- `log_level`は、コンポーネントまたはリソースのログ・レベルを設定する1から5までの番号です。1は最小ログ出力、5は最も詳細なログ出力を表します。デフォルトのログ・レベルは2です。

例

CRSモジュールのCRSRTIコンポーネントおよびCRSCOMMコンポーネントのログ・レベルを設定するには、次のコマンドを実行します。

```
$ crsctl set log crs "CRSRTI=1,CRSCOMM=2"
```

コンポーネント・レベルのデバッグ

rootユーザーとして`crsctl set log`および`crsctl set trace`コマンドを使用して、様々なOracle Clusterwareモジュールの動的デバッグを有効にできます。

Oracle Clusterwareモジュールのデバッグの有効化

環境変数を設定するか、`crsctl set log`コマンドを実行して、Oracle Clusterwareモジュールとそれらのコンポーネント、およびリソースのデバッグを有効にできます。

様々なOracle Clusterwareモジュールのデバッグを有効にするには、次の構文を使用します。

```
crsctl set {log | trace} module_name "component:debugging_level[,...]"
```

`crsctl set`コマンドをrootユーザーとして実行し、次の情報を指定します。

- `module_name`: 次のいずれかのモジュールの名前を指定します。
 - mdns: マルチキャスト・ドメイン・ネーム・サーバー
 - gpnp: グリッド・プラグ・アンド・プレイ・サービス
 - css: クラスタ同期サービス
 - crf: クラスタ状態モニター
 - crs: クラスタ・レディ・サービス
 - ctss: クラスタ時刻同期化サービス
 - evm: Event Manager
 - gipc: グリッド・プロセス間通信
- `component`: 次のいずれかのモジュールのコンポーネントの名前を指定します。すべてのコンポーネントのリストについ

ては、[表E-92](#)を参照してください。

- `debugging_level`: デバッグ・コマンドで戻す詳細レベルを、1から5までの番号で指定します。1は最小デバッグ出力、5は最も詳細なデバッグ出力を表します。デフォルトのデバッグ・レベルは2です。

次の例では、様々なモジュールに対してデバッグを有効にする方法を示します。

- Oracle Clusterwareのデバッグを有効にするには:

```
# crsctl set log crs "CRSRTI:1,CRSCOMM:2"
```

- OCRのデバッグを有効にするには:

```
# crsctl set log crs "CRSRTI:1,CRSCOMM:2,OCRSRV:4"
```

- EVMのデバッグを有効にするには:

```
# crsctl set log evm "EVMCOMM:1"
```

- リソースのデバッグを有効にする方法

```
# crsctl set log res "resname:1"
```

デバッグに使用可能なコンポーネントのリストを取得するには、次の構文を使用して、`crsctl lsmodules`コマンドを実行します。

```
$ crsctl lsmodules {mdns | gnpn | css | crf | crs | ctss | evm | gipc}
```

ノート:



`crsctl lsmodules` コマンドを実行するために、root ユーザーである必要はありません。

[表E-92](#)に、モジュールCRS、CSSおよびEVMのそれぞれのコンポーネントを示します。一部のコンポーネント名はCRS、EVMおよびCSSのデーモン間で共通であり、特定のデーモンで有効な場合があることに注意してください。たとえば、`COMMNS`はNSレイヤーで、NSレイヤーは各デーモンで使用されるため、いずれかのデーモンでこの特定のモジュール・コンポーネントを有効にして、特定のデバッグ情報を取得できます。

表E-92 CRS、CSSおよびEVMモジュールのコンポーネント

CRSコンポーネント(1)	CSSコンポーネント(2)	EVMコンポーネント(3)
CRSUI CRSCOMM CRSRTI CRSMAIN CRSPLACE CRSAPP CRSRES CRSCOMM CRSOCR CRSTIMER CRSEVT CRSD CLUCLS CSSCLNT COMMCRS COMMNS	CSSD COMMCRS COMMNS	EVMD EVMDMAIN EVMCOMM EVMEVT EVMAPP EVMAGENT CRSOCR CLUCLS CSSCLNT COMMCRS COMMNS

脚注1: `crsctl lsmodules crs`コマンドを使用して、CRSコンポーネントのリストを取得します。

脚注2: `crsctl lsmodules css`コマンドを使用して、CSSコンポーネントのリストを取得します。

脚注3: `crsctl lsmodules evm`コマンドを使用して、EVMコンポーネントのリストを取得します。

たとえば、次のコマンドを使用して、特定のクラスタ・ノードにデバッグ・レベルを設定するには、`-node list`キーワードおよびノードの名前を次のように含めます。

```
crsctl set log crs "CRSRTI:1,CRSCOMM:2" -nodelist node1,node2
```

[表E-93](#)に、クラスタ同期サービス・モジュールを示します。

表E-93 クラスタ同期サービス(CSS)・モジュールおよびその機能

モジュール	説明
CSS	CSS クライアント・コンポーネント
CSSD	CSS デモン・コンポーネント

[表E-94](#)に、各通信(COMM)モジュールの機能を示します。

表E-94 通信(COMM)モジュールおよびその機能

モジュール	説明
COMMCRS	Clusterware 通信レイヤー
COMMNS	NS 通信レイヤー

[表E-95](#)に、各CRSモジュールで実行される機能を示します。

表E-95 Oracle Clusterware (CRS)モジュールおよびその機能

モジュール	説明
CRSUI	ユーザー・インタフェース・モジュール
CRSCOMM	通信モジュール
CRSRTI	リソース管理モジュール
CRSMAIN	メイン・モジュール/ドライバ
CRSPLACE	CRS 配置モジュール
CRSAPP	CRS アプリケーション
CRSRES	CRS リソース
CRSOCR	Oracle Cluster Registry インタフェース
CRSTIMER	CRS に関連する様々なタイマー

モジュール	説明
CRSEVT	CRS EVM/イベント・インタフェース・モジュール
CRSD	CRS デーモン

crsctl set log crsコマンドを使用して、[表E-96](#)にリストされているOCRコンポーネントをデバッグできます。[表E-96](#)にリストされているコンポーネントは、OCRMASおよびOCRASMを除き、Oracle Local Registry (OLR)にも使用できます。また、OCRMASおよびOCRSRVを除くOCRクライアントおよびOLRクライアントにも使用できます。一部のOCRクライアントおよびOLRクライアントにはOCRCONFIG、OCRDUMPなどがあります。

表E-96 Oracle Cluster Registry (OCR)コンポーネント名

モジュール	説明
OCRAPI	OCR 抽象化コンポーネント
OCRCLI	OCR クライアント・コンポーネント
OCRSRV	OCR サーバー・コンポーネント
OCRMAS	OCR マスター・スレッド・コンポーネント
OCRMSG	OCR メッセージ・コンポーネント
OCRCAC	OCR キャッシュ・コンポーネント
OCRRAW	OCR RAW デバイス・コンポーネント
OCRUTL	OCR ユーティリティ・コンポーネント
OCROSD	OCR オペレーティング・システム固有(OSD)レイヤー
OCRASM	OCR ASM コンポーネント

[表E-97](#)に、OCRツール・モジュールを示します。

表E-97 OCRCONFIGモジュールおよびその機能

モジュール	説明
OCRCONFIG	OCR を構成するための OCRCONFIG コンポーネント

モジュール	説明
OCRDUMP	Oracle Cluster Registry の内容を表示する OCRDUMP コンポーネント
OCRCHECK	すべての構成済 OCR を検証する OCRCHECK コンポーネント

Oracle Clusterwareリソースのデバッグの有効化

次の構文を使用して `crsctl set log` コマンドを実行することによって、Oracle Clusterware のリソースのデバッグを有効にできます。

```
crsctl set log res "resource_name=debugging_level"
```

`crsctl set log` コマンドを root ユーザーとして実行し、次の情報を指定します。

- `resource_name`: デバッグするリソースの名前。
- `debugging_level`: 1 から 5 までの番号は、デバッグ・コマンドで戻す詳細レベルを示します。1 は最小デバッグ出力、5 は最も詳細なデバッグ出力を表します。

ノート:

追加のデバッグが必要な問題を解決した後、以下のようにデバッグ・レベルをデフォルトにリセットできます。

```
# crsctl set log crs resource_name=default
```

デバッグに使用可能なリソースのリストを取得するには、`crsctl status resource` コマンドを実行します。

例

node1 上の VIP リソースのデバッグ・ログを生成するには、次のコマンドを発行します。

```
# crsctl set log res "ora.node1.vip:1"
```

F サーバー制御(SRVCTL)コマンド・リファレンス

サーバー制御(SRVCTL)ユーティリティを使用して、クラスタ内の様々なコンポーネントおよびアプリケーションを管理します。

この付録には、SRVCTLコマンドのユーティリティの使用法に関する情報と包括的なリストを含むサーバー制御コマンドの参照情報が記載されています。

ノート:

Oracle RAC管理操作に固有のSRVCTLコマンドは『[Oracle Real Application Clusters管理およびデプロイメント・ガイド](#)』に記載されています

SRVCTLの使用法

SRVCTLは、デフォルトでクラスタの各ノードにそれぞれインストールされます。SRVCTLを使用するには、ノードのオペレーティング・システムにログインし、大/小文字が区別される構文を使用して、SRVCTLコマンドとそのパラメータを入力します。

- 自分が管理しているデータベースのOracleホームから、現行のOracle Databaseリリースとともに提供されたバージョンのSRVCTLを使用します。SRVCTLのバージョンは、管理対象のオブジェクト(リスナー、Oracle ASMインスタンス、Oracle Database、Oracle DatabaseインスタンスおよびOracle Databaseサービス)のバージョンと同じである必要があります。
- SRVCTLでは、同じオブジェクトに対する複数コマンドの同時実行はサポートされていません。したがって、各データベース、サービスまたは他のオブジェクトに対して、1つずつSRVCTLコマンドを実行します。
- カンマ区切りリストをSRVCTLコマンドの一部として指定する場合、リスト内の項目の間に空白をしないでください。たとえば:

```
srvctl add database -serverpool "serverpool1,serverpool3"
```

Windows環境でカンマ区切りリストを指定する場合は、リストを二重引用符("")で囲む必要があります。LinuxまたはUNIX環境で、カンマ区切りリストを二重引用符内で囲むことは可能ですが、二重引用符は無視されます。

- SRVCTLコマンドの入力時に、新しい行で入力続ける場合、オペレーティング・システムの継続文字を使用できます。Linuxでは、バックスラッシュ(\\$)記号です。
- 出力が生成されないSRVCTLコマンドは、正常なコマンドです。完了(成功)時にすべてのSRVCTLコマンドがメッセージを返すわけではありません。しかし、SRVCTLコマンドが失敗した場合は、常にエラー・メッセージが返されます。
- SRVCTLは、成功時には0、失敗時には1、警告時には2を返します。start、stop、enable、disableなどの一部のコマンドでは、リクエストによって何も変更されない場合の警告に対して2が返されることがあります。つまり、コマンドのオブジェクトはすでに起動されているか、すでに停止されているか、すでに無効化されているかなどです。警告の場合は、SRVCTLによって、すでに行われた処理に関するメッセージも出力されます。
- -evalパラメータは、複数のSRVCTLコマンドで使用できます。このパラメータを使用すると、システムに変更を加えることなく、コマンドの実行をシミュレートできます。SRVCTLによって戻される出力には、特定のコマンドを実行した場合の結果が示されます。たとえば、サーバーを再配置した場合に想定される結果を確認するには、次のようにします。

```
$ srvctl relocate server -servers "rac1" -eval -serverpool pool2
Database db1
    will stop on node rac1
    will start on node rac7
```

```
Service mySrv1
    will stop on node rac1, it will not run on any node
Service myServ2
    will stop on node rac1
    will start on node rac6
Server rac1
    will be moved from pool myPoolX to pool pool2
```

-evalパラメータは、次のコマンドで使用できます。

- srvctl add database
- srvctl add service
- srvctl add srvpool
- srvctl modify database
- srvctl modify service
- srvctl modify srvpool
- srvctl relocate server
- srvctl relocate service
- srvctl remove srvpool
- srvctl start database
- srvctl start service
- srvctl stop database
- srvctl stop service

単一文字ではなくキーワードとしてのコマンド・パラメータの指定

Oracle Database 12cより前のリリースでは、SRVCTLコマンドライン・インタフェースのパラメータとして単一文字が使用されていました。ただし、この方法では、SRVCTLコマンドで使用できる一意のパラメータの数に制限が発生します。Oracle Database 12cで導入されたSRVCTLコマンド・パラメータは、単一文字ではなく完全な単語になっています(-multicastportや-subdomainなど)。

下位互換性をサポートするために、単一文字パラメータと新しいキーワード・パラメータを組み合わせで使用できます。新たに導入されたキーワード・パラメータは、単一文字パラメータと併用可能です。

ノート:



Oracle Database 12c以降では、キーワード・パラメータが優先されるために、単一文字パラメータは非推奨になりました(異なる機能の実装に、コマンドに応じて同じ文字を使用することを回避するためです)。

該当する場合は、-help パラメータの後に-compatible パラメータを追加することによって、等価の単一文字を取得できます。

SRVCTLオブジェクトの値の文字セットおよび大文字小文字の区別

SRVCTLは、様々なタイプの多くのオブジェクトとやり取りします。文字セットと名前の長さの制限、およびオブジェクト名で大/小文字が区別されるかどうかは、オブジェクト・タイプによって異なります。

表F-1 SRVCTLオブジェクト名の文字列制限

オブジェクト・タイプ	文字セット制限	大/小文字の区別	最大長
db_domain	英数字、アンダースコア(_)およびシャープ記号(#)		128 文字
db_unique_name	英数字、アンダースコア(_)、シャープ記号(#)およびドル記号(\$)。最初の 8 文字はポリシー管理データベースのインスタンス名に使用されるため、これらは一意である必要があります。	いいえ	30 文字(ただし、最初の 8 文字は、同じクラスタの他のデータベースに対して一意である必要があります)
diskgroup_name	ディスク・グループのネーミングには、他のデータベース・オブジェクトのネーミングと同じ制限があります。 関連項目: データベース・オブジェクトのネーミング規則の詳細は、 『Oracle Database SQL 言語リファレンス』 を参照してください。	いいえ(すべての名前は大文字に変換されます)	
instance_name	英数字	プラットフォームに依存	15 文字
listener_name			
node_name		いいえ	
scan_name	最初の文字はアルファベットである必要があります	いいえ	
server_pool	英数字、アンダースコア(_)、シャープ記号(#)、ピリオド(.)およびドル記号。ただし、名前をピリオドで始めること、一重引用符(' ')を含むこと、または「Generic」または「Free」という名前にすることはできません。この 2 つの名前は、組み込みサーバー・プールで予約され		250 文字

オブジェクト・タイプ	文字セット制限	大/小文字の区別	最大長
	ています。		
service_name			250 文字
volume_name	英数字(ダッシュ(-)で始めることはできません。最初の文字はアルファベットにする必要があります)。	いいえ	11 文字

SRVCTLヘルプの使用

この項には、SRVCTLに関する状況依存ヘルプの使用についての情報があります。

すべてのSRVCTLコマンドに関するヘルプを表示するには、コマンドラインから次のように入力します。

```
srvctl -help
```

各SRVCTLコマンドのコマンド構文およびパラメータのリストを表示するには、コマンドラインから次のように入力します。

```
srvctl command (or verb) object (or noun) -help
```

SRVCTLで `-help` を使用してコマンドのオンライン・ヘルプを要求すると、各パラメータの完全な単語が出力されます。該当する場合は、`-help` パラメータの後に `-compatible` パラメータを追加することによって、等価の単一文字を取得できます。たとえば:

```
$ srvctl config database -help -compatible
```

前述のコマンドを実行すると、`srvctl config database` コマンドの使用方法が出力され、すべてのパラメータが完全な単語として一覧表示され、その後に、該当する場合は等価の単一文字がカッコで囲まれて表示されます。

SRVCTLのバージョン・ナンバーを表示するには、次のように入力します。

```
$ srvctl -version
```

SRVCTLの権限とセキュリティ

SRVCTLを使用してデータベース構成を変更するには、管理するホームのソフトウェア所有者としてオペレーティング・システムにログインします。

たとえば、様々なユーザーがOracle DatabaseおよびOracle Grid Infrastructureをインストールした場合、データベースを管理するにはデータベース・ソフトウェア所有者(`ora_db`など)としてログインし、Oracle ASMインスタンスを管理するにはOracle Grid Infrastructureソフトウェア所有者(`ora_asm`など)としてログインします。

OSDBAオペレーティング・システム・グループのメンバーであるユーザーは、データベースを起動および停止できます。Oracle ASMインスタンスを停止および起動するには、ユーザーがOSASMオペレーティング・システム・グループのメンバーであることが必要です。

リスナー、Oracle Notification Services、サービスなどのオブジェクトを作成または登録するには、Oracleホームのソフトウェア所有者としてオペレーティング・システムにログインする必要があります。そのOracleホームに作成または登録したオブジェクトは、

Oracleホームの所有者のユーザー・アカウントで実行されます。データベースは、実行元になるデータベース・ホームのデータベース・インストール所有者として実行されます。

オブジェクト上で `srvctl add` 操作を実行するには、そのオブジェクトが実行されているホームのOracleアカウント所有者としてログインする必要があります。

一部のSRVCTLコマンドでコマンドを実行するには、LinuxシステムおよびUNIXシステムではrootでログインし、Windowsシステムでは管理者権限を持つユーザーとしてログインする必要があります。この付録のコマンド例では、それらのコマンドの前にrootプロンプト(#)を付けてあります。

追加のSRVCTLトピック

- SRVCTLはリスナー、インスタンス、ディスク・グループ、ネットワークなど、Oracleから提供されるリソースの管理に使用し、CRSCTLはOracle Clusterwareおよびそのリソースの管理に使用します。



ノート:

Oracle が提供するリソース(ora という名前で始まるリソース)は、CRSCTL では直接操作しないでください。直接操作することはクラスタ構成に悪影響を与える場合があります。

- [Ctrl]キーを押しながら[C]キーを押すと、SRVCTLコマンドの実行をキャンセルできますが、その結果、構成データが破損される場合があります。

この方法ではSRVCTLを中断しないでください。

非推奨のSRVCTLサブプログラムまたはコマンド

いくつかのSRVCTLコマンドおよびパラメータは、このリリースでは非推奨になりました。

すべてのSRVCTLコマンドの単一文字パラメータ

単一文字パラメータは、Oracle Database 12cでは非推奨になりました。

かわりに各パラメータの完全なキーワードを使用してください。引き続き単一文字パラメータが使用される従来のツールやスクリプトをサポートするために、現在のバージョンのSRVCTLでは、単一文字パラメータと完全なキーワード・パラメータの両方がサポートされています。

この付録のコマンド・リファレンスに、各SRVCTLコマンドのキーワードが示されています。[表F-2](#)に、非推奨になった単一文字パラメータを示します。

表F-2 SRVCTLコマンドで非推奨になった単一文字パラメータ

単一文 字	詳細名	値	説明	関連コマンド
A	address	{VIP_name IP}/netmask/[if1[if2...]]	ノード・アプリケーションのVIP アドレス指定	ノード・アプリケーション、 VIP、ネットワーク、リスナー、 SCAN VIP および SCAN リ

単一文 字	詳細名	値	説明	関連コマンド
				スナーのコマンド
a	all		その種のすべてのリソース	srvctl config database 共通
a	diskgroup	diskgroup_list	Oracle ASM ディスク・グループのカンマ区切りリスト	データベース、インスタンス、Oracle ASM、ディスク・グループおよびファイル・システムのコマンド
a	detail		詳細な構成情報の出力	共通
a	available	available_list	使用可能インスタンスのカンマ区切りリスト	サービスおよびサーバー・プールのコマンド
a	abort		失敗したオンライン再配置の停止	データベースの再配置
a	viponly		VIP 構成の表示	ノード・アプリケーション、VIP、ネットワーク、リスナー、SCAN VIP および SCAN リスナーのコマンド
B	rlbgoal	{NONE SERVICE_TIME THROUGHPUT}	サービスのランタイム・ロード・バランシングの目標	サービスおよびサーバー・プールのコマンド
c	currentnode	current_node	サービスの再配置元となるノードの名前	サービスおよびサーバー・プールのコマンド
c	cardinality	{UNIFORM SINGLETON}	サービスをサーバー・プール内のすべてのアクティブ・サーバーで実行するか (UNIFORM)、1 台のサーバーでのみ実行するか (SINGLETON)	サービスおよびサーバー・プールのコマンド
c	dbtype	type	データベースのタイプ: Oracle RAC One Node、	データベース、インスタンス、Oracle ASM、ディスク・グ

単一文 字	詳細名	値	説明	関連コマンド
			Oracle RAC またはシング ル・インスタンス	ループおよびファイル・システム のコマンド
d	db または database	db_unique_name	データベースの一意の名前	共通
d	device	volume_device	ボリューム・デバイスのパス	データベース、インスタンス、 Oracle ASM、ディスク・グ ループおよびファイル・システム のコマンド
d	domain		GNS から供給されるサブドメ インの表示	OC4J、ホーム、CVU および GNS のコマンド
e	emport	em_port_number	Oracle Enterprise Manager のローカル・リスニ ング・ポート	ノード・アプリケーション、 VIP、ネットワーク、リスナー、 SCAN VIP および SCAN リ スナーのコマンド
e	failovertype	{NONE SESSION BASIC TRANSACTION}	サービスのフェイルオーバー・タ イプ	サービスおよびサーバー・プー ルのコマンド
e	server	server_list	Oracle RAC One Node データベースの候補サーバー のリスト	データベース、インスタンス、 Oracle ASM、ディスク・グ ループおよびファイル・システム のコマンド
f	force		強制削除	共通
g	diskgroup	diskgroup_name	ディスク・グループ名	ファイル・システム、ディスクグ ループのコマンド
g	gsdonly		GSD 構成の表示	ノード・アプリケーション、 VIP、ネットワーク、リスナー、 SCAN VIP および SCAN リ スナーのコマンド
g	serverpool	server_pool_name server_pool_list	サーバー・プール名 データベース・サーバー・プー	サービスおよびサーバー・プー ルのコマンド

単一文 字	詳細名	値	説明	関連コマンド
			ル名のカンマ区切りリスト	データベース、インスタンス、Oracle ASM、ディスク・グループおよびファイル・システムのコマンド
h	help			共通
i	importance	number	サーバー・プールの重要度を表す番号	サービスおよびサーバー・プールのコマンド
i	instance	instance_name instance_list	管理者管理 Oracle RAC One Node データベースのインスタンス名の接頭辞 インスタンス名のカンマ区切りリスト	データベース、インスタンス、Oracle ASM、ディスク・グループおよびファイル・システムのコマンド
I	ip	ip_address	GNS がリスニングする VIP アドレス	OC4J、ホーム、CVU および GNS のコマンド
i	oldinst	instance_name	元のインスタンス名	サービスおよびサーバー・プールのコマンド
i	scannumber	scan_ordinal_number	SCAN 用の IP アドレスの序数	ノード・アプリケーション、VIP、ネットワーク、リスナー、SCAN VIP および SCAN リスナーのコマンド
i	vip	vip_name または "vip_name_list"	VIP 名	ノード・アプリケーション、GNS、VIP、ネットワーク、リスナー、SCAN VIP および SCAN リスナーのコマンド
j	acfspath	acfs_path_list	データベースの依存性を設定する Oracle ACFS パスのカンマ区切りリスト	データベース、インスタンス、Oracle ASM、ディスク・グループおよびファイル・システムのコマンド
j	clbgoal	{SHORT LONG}	サービスの接続時ロード・バラ	サービスおよびサーバー・プールのコマンド

単一文 字	詳細名	値	説明	関連コマンド
			リスニングの目標	ルのコマンド
k	netnum	network_number	ネットワーク番号	サービスおよびサーバー・プールのコマンド ノード・アプリケーション、VIP、ネットワーク、リスナー、SCAN VIP および SCAN リスナーのコマンド OC4J、ホーム、CVU および GNS のコマンド
l	list		GNS のすべてのレコードのリスト	OC4J、ホーム、CVU および GNS のコマンド
l	listener	listener_name	リスナーの名前	ASM のコマンド
l	loglevel	log_level	GNS が実行するロギング・レベル(0 から 6)の指定	OC4J、ホーム、CVU および GNS のコマンド
l	min	number	サーバー・プールの最小サイズ	サービスおよびサーバー・プールのコマンド
l	onslocalport	port_number	ローカル・クライアント接続用の Oracle Notification Service リスニング・ポート	ノード・アプリケーション、VIP、ネットワーク、リスナー、SCAN VIP および SCAN リスナーのコマンド
l	role	service_role	二重引用符("")で囲んだサーバー・ロールのカンマ区切りリスト(各ロールは PRIMARY、PHYSICAL_STANDBY、LOGICAL_STANDBY または SNAPSHOT_STANDBY のいずれか)	サービスおよびサーバー・プールのコマンド
m	domain	domain_name	データベースのドメイン	データベース、インスタンス、

単一文 字	詳細名	値	説明	関連コマンド
				Oracle ASM、ディスク・グループおよびファイル・システムのコマンド
m	failovermethod	{NONE BASIC}	サービスのフェイルオーバー・メソッド	サービスおよびサーバー・プールのコマンド
m	multicastpost		GNS デーモンがマルチキャスト・リクエストをリスニングしているポート	OC4J、ホーム、CVU および GNS のコマンド
m	path	mountpoint_path	マウントポイント・パス	データベース、インスタンス、Oracle ASM、ディスク・グループおよびファイル・システムのコマンド
n	name		特定のアドレスを使用した GNS による名前の通知	OC4J、ホーム、CVU および GNS のコマンド
n	node	node_name	特定のノードの名前	共通
n	nodes	node_list	ノード名のカンマ区切りリスト	ファイル・システムのコマンド
n	dbname	database_name	データベース名 (DB_NAME)、-db パラメータで指定する一意の名前と異なる場合	データベース、インスタンス、Oracle ASM、ディスク・グループおよびファイル・システムのコマンド
n	scanname	scan_name	完全修飾 SCAN 名(ドメインを含む)	ノード・アプリケーション、VIP、ネットワーク、リスナー、SCAN VIP および SCAN リスナーのコマンド
n	servers	server_list	候補としてのサーバー名のカンマ区切りリスト	サービスおよびサーバー・プールのコマンド
n	targetnode	node_name	サービスの再配置先となるノードの名前	サービスおよびサーバー・プールのコマンド

単一文 字	詳細名	値	説明	関連コマンド
o	oraclehome	oracle_home	\$ORACLE_HOME のパス	データベースのコマンド
p	endpoints	[TCP:]port_number[/IPC:key[/NMP:pipe_name[/TCPS:s_port[/SDP:port]	SCAN リスナー・エンドポイント	ノード・アプリケーション、VIP、ネットワーク、リスナー、SCAN VIP および SCAN リスナーのコマンド
p	port		GNS デーモンが DNS サーバーとの通信に使用するポート	OC4J、ホーム、CVU および GNS のコマンド
p	rmiport	port_number	OC4J RMI ポート番号	OC4J、ホーム、CVU および GNS のコマンド
P	tafpolicy	{NONE BASIC}	TAF ポリシーの指定	サービスおよびサーバー・プールのコマンド
p	spfile	spfile_location	サーバー・パラメータ・ファイルのパス	データベース、インスタンス、Oracle ASM、ディスク・グループおよびファイル・システムのコマンド
q	notification	{TRUE FALSE}	FAN が OCI 接続に対して有効かどうか	サービスのコマンド
q	query		GNS への、名前に属しているレコードの問合せ	OC4J、ホーム、CVU および GNS のコマンド
r	preferred	preferred_list	優先インスタンスのカンマ区切りリスト	サービスおよびサーバー・プールのコマンド
r	onsremoteport	port_number	リモート・ホストからの接続用の Oracle Notification Service リスニング・ポート	ノード・アプリケーション、VIP、ネットワーク、リスナー、SCAN VIP および SCAN リスナーのコマンド
r	relocate		VIP の再配置	ノード・アプリケーション、VIP、ネットワーク、リスナー、SCAN VIP および SCAN リ

単一文 字	詳細名	値	説明	関連コマンド
				スナーのコマンド
r	revert		管理者管理 Oracle RAC One Node データベースの候補サーバー・リストからの、失敗したオンライン再配置リクエストのターゲット・ノードの削除	データベースの再配置
r	role	role_type	スタンバイ・データベースのロール: PRIMARY、PHYSICAL_STANDBY、LOGICAL_STANDBY または SNAPSHOT_STANDBY	データベース、インスタンス、Oracle ASM、ディスク・グループおよびファイル・システムのコマンド
s	ononly		Oracle Notification Service デモン構成の表示	ノード・アプリケーション、VIP、ネットワーク、リスナー、SCAN VIP および SCAN リスナーのコマンド
s	skip		ポートの確認のスキップ	リスナー、SCAN および SCAN リスナー。
s	statfile	file_name	前に実行した srvctl stop home コマンドによって作成された state_file のファイル・パス	OC4J、ホーム、CVU および GNS のコマンド
s	status		GNS のステータスの表示	OC4J、ホーム、CVU および GNS のコマンド
S	subnet	subnet/net_mask/[if1[if2...]]	ネットワークのネットワーク・アドレス指定	ノード・アプリケーション、VIP、ネットワーク、リスナー、SCAN VIP および SCAN リスナーのコマンド
s	service	service_name service_name_list	サービスの名前	サービスおよびサーバー・ポート

単一文 字	詳細名	値	説明	関連コマンド
			サービス名のカンマ区切りリスト	ルのコマンド
s	startoption	start_options	データベースの起動オプション (mount、open、read only)	データベース、インスタンス、Oracle ASM、ディスク・グループおよびファイル・システムのコマンド
t	checkinterval	time_interval	チェック間隔(分)	OC4J、ホーム、CVU および GNS のコマンド
t	edition	edition_name	サービスの初期セッション・エディション	サービスおよびサーバー・プールのコマンド
t	envs	"name_list"	環境変数のリスト	共通
t	namevals	"name=value,..."	環境変数の名前および値	共通
T	nameval	"name=value"	単一環境変数の名前と値	共通
t	update	instance_name	新しいインスタンス名	サービスおよびサーバー・プールのコマンド
t	remoteservers	host_name[:port_number][,host_name[:port_number]...]	このクラスターの外部にある Oracle Notification Service デモン用のリモート・ホスト名とポート番号のペアのリスト	ノード・アプリケーション、VIP、ネットワーク、リスナー、SCAN VIP および SCAN リスナーのコマンド
t	stopoption	stop_options	データベースの停止オプション (NORMAL、TRANSACTIONAL、IMMEDIATE または ABORT)	データベース、インスタンス、Oracle ASM、ディスク・グループおよびファイル・システムのコマンド
t	toversion	target_version	ダウングレード先のバージョン	データベース、インスタンス、Oracle ASM、ディスク・グループおよびファイル・システム

単一文 字	詳細名	値	説明	関連コマンド
				のコマンド
u	max	number	サーバー・プールの最大サイ ズ	サービスおよびサーバー・プ ールのコマンド
u	nettype	network_type	ネットワーク・サーバー・タイ プ (STATIC、DHCP または MIXED)	ノード・アプリケーション、 VIP、ネットワーク、リスナー、 SCAN VIP および SCAN リ スナーのコマンド
u	newinst		サービス構成への新しいイン スタンスの追加	サービスのコマンド
u	update		SCAN VIP の数に合わせた SCAN リスナーの更新	ノード・アプリケーション、 VIP、ネットワーク、リスナー、 SCAN VIP および SCAN リ スナーのコマンド
u	user	oracle_user	ファイル・システムのマウントお よびアンマウントが許可された Oracle ユーザーまたはその 他のユーザー	データベース、インスタンス、 Oracle ASM、ディスク・グ ループおよびファイル・システム のコマンド
v	verbose		冗長出力	共通
v	volume	volume_name	ボリュームの名前	データベース、インスタンス、 Oracle ASM、ディスク・グ ループおよびファイル・システム のコマンド
V	versions			共通
w	failoverdelay	number	フェイルオーバーの遅延	サービスおよびサーバー・プ ールのコマンド
w	nettype	network_type	ネットワーク・サーバー・タイ プ (STATIC、DHCP または MIXED)	ノード・アプリケーション、 VIP、ネットワーク、リスナー、 SCAN VIP および SCAN リ

単一文 字	詳細名	値	説明	関連コマンド
				スナークのコマンド
w	timeout	timeout	オンライン再配置のタイムアウト(分)	データベース、インスタンス、Oracle ASM、ディスク・グループおよびファイル・システムのコマンド
x	dtp	{TRUE FALSE}	分散トランザクション処理を有効化するかどうか	サービスおよびサーバー・プールのコマンド
x	node	node_name	ノード名(このパラメータは非クラスタ・データベースでのみ使用します)	共通
y	noprompt		確認プロンプトの抑止	共通
y	policy	{AUTOMATIC MANUAL}	リソースの管理ポリシー	データベース、インスタンス、Oracle ASM、ディスク・グループ、ファイル・システム、サービスおよびサーバー・プールのコマンド
z	failoverretry	number	フェイルオーバー再試行回数	サービスおよびサーバー・プールのコマンド
z	rmdepondisk		ディスク・グループへのデータベースの依存性の削除	データベース、インスタンス、Oracle ASM、ディスク・グループおよびファイル・システムのコマンド

その他のSRVCTLコマンドおよびパラメータ

次のコマンド・パラメータは、このリリースでは非推奨になりました。

表F-3 SRVCTLで非推奨になったコマンドおよびパラメータ

コマンド	非推奨になったパラメータ
srvctl modify asm	-node node_name

コマンド	非推奨になったパラメータ
srvctl modify instance -z	かわりに、-node オプションを使用し、その値を""に設定してください。
srvctl modify gns	[-ip ip_address] [-advertise host_name -address address] [-delete host_name -address address] [-createalias name -alias alias] [-deletealias alias] かわりに、srvctl update gns コマンドを使用してください。
srvctl * oc4j	oc4j の名詞は非推奨になり、qosmserver に置き換えられました。SRVCTL では、サポート対象外になるまで、引き続き oc4j の名詞が受け入れられます。
srvctl add service	-tafpolicy パラメータを含む PRECONNECT オプションは非推奨です。
srvctl modify service	-failovermethod {NONE BASIC}は非推奨です。 -tafpolicy パラメータを含む PRECONNECT オプションは非推奨です。

SRVCTLのコマンド・リファレンス

Oracle Grid Infrastructure環境で使用するSRVCTLコマンドの包括的なリスト。

SRVCTLでは次のコマンド構文を使用します。

```
srvctl command object [parameters]
```

このSRVCTL構文の各要素の意味は次のとおりです。

- commandは、start、stop、removeなどの動詞です。
- object (名詞とも呼ばれる)は、SRVCTLがコマンドを実行するターゲットまたはオブジェクト(Oracle Automatic Storage Management (Oracle ASM)やファイル・システムなど)です。オブジェクトの短縮形も使用できます。
- parametersは、コマンドの追加パラメータを使用できるようにすぐ前のコマンドの組合せの使用範囲を拡大します。一部のパラメータは必須ですが、省略可能なパラメータもあります。

ノート:



Windows でコマンド・パラメータにカンマ区切りリストを指定する場合は、リストを二重引用符("")で囲む必要があります。

[表F-4](#)は、SRVCTLコマンドのobjectの部分に使用できるキーワードのリストです。各オブジェクト・キーワードとして、完全な名前または短縮形のいずれかを使用できます。「目的」列に、オブジェクトとそのオブジェクトに実行できるアクションを記述します。

表F-4 オブジェクト・キーワードおよび短縮形

オブジェクト	キーワード	目的
Oracle ASM	asm	Oracle ASM に関する環境変数の追加、変更および管理、構成の表示、Oracle ASM の有効化、無効化、起動、停止、ステータス取得
Oracle ASM ネットワーク	asmnetwork	Oracle ASM ネットワーク構成の追加、構成、変更および削除
クラスタ検証ユーティリティ	cvu	クラスタ検証ユーティリティに関する追加、変更、有効化、無効化、削除、再配置、起動、停止、およびステータスの取得
クラスタ間依存性	cdp	クラスタ間依存性サービスに関する追加、変更、有効化、無効化、削除、再配置、起動、停止、およびステータス取得。
クラスタ間依存性プロキシ	cdpproxy	クラスタ間依存性プロキシ・リソースに関する追加、変更、有効化、無効化、削除、再配置、起動、停止、およびステータス取得。
エクスポート・ファイル・システム	exportfs	エクスポート・ファイル・システムに関する追加、変更、構成表示、有効化、無効化、起動、停止、ステータス取得、および削除
Oracle Automatic Storage Management クラスタ・ファイル・システム	filesystem	Oracle ACFS に関する追加、変更、構成表示、削除、起動、停止、およびステータス取得
グリッド・ネーミング・サービス	gns	グリッド・ネーミング・サービスに関する追加、変更、構成表示、インポート、エクスポート、有効化、無効化、起動、停止、ステータス取得、再配置、削除、および更新
高可用性 VIP	havip	高可用性 VIP に関する追加、構成表示、有効化、起動、ステータス取得、停止、無効化、再配置、および削除
Oracle ASM IO Server	ioserver	Oracle ASM IO Server に関する追加、構成表示、変更、有効化、無効化、起動、停止、再配置、ステータス取得、環境変数管理、および削除
管理データベース	mgmtadb	管理データベース(Cluster Health Monitor リポジトリ)に関する追加、変更、構成表示、有効化、無効化、起

オブジェクト	キーワード	目的
		動、停止、ステータス取得、再配置、環境変数管理、および削除
管理リスナー	mgmtlsnr	Cluster Health Monitor の管理リスナー・リソースに関する追加、変更、構成表示、有効化、無効化、起動、停止、ステータス取得、再配置、環境変数管理、および削除
ネットワーク接続ストレージ (NAS)	mountfs	NAS に対する追加、構成表示、有効化、起動、ステータス取得、停止、無効化、再配置、および削除
ネットワーク・ストレージ・サービス・リソース	netstorageservice	NetStorageService リソースに関する追加、構成表示、有効化、起動、ステータス取得、停止、無効化、および削除
仮想マシン・マネージャ	ovmm	仮想マシン・マネージャに関する追加、構成表示、有効化、変更、および削除
Oracle Database Quality of Service Management サーバー	qosmserver	Oracle Database Quality of Service Management サーバーに関する追加、変更、構成表示、有効化、無効化、起動、停止、ステータス取得、再配置、および削除
フリー・パッチ適用およびプロビジョニング・クライアント	rhpclient	フリー・パッチ適用およびプロビジョニング・クライアントに関する追加、変更、構成表示、有効化、無効化、起動、停止、ステータス取得、再配置、および削除
フリー・パッチ適用およびプロビジョニング・サーバー	rhpserver	フリー・パッチ適用およびプロビジョニング・サーバーに関する追加、変更、構成表示、有効化、無効化、起動、停止、ステータス取得、再配置、および削除
仮想マシン	vm	仮想マシンに関する追加、変更、構成表示、有効化、無効化、起動、停止、ステータス取得、再配置、および削除

ノート:

Oracle RAC管理操作に固有のSRVCTLコマンドは『[Oracle Real Application Clusters管理およびデプロイメント・ガイド](#)』に記載されています

srvctl config all

クラスタおよびデータベース・コンポーネントの構成詳細を表示します。

構文

```
srvctl config all [-help]
```

例

このコマンドによって、次のような出力が返されます。

```
$ srvctl config all
Oracle Clusterware configuration details
=====
Oracle Clusterware basic information
-----
Operating system      Linux
Name                  stbm000025-rac3
Class                 STANDALONE
Cluster nodes         stbm000025-vm11, stbm000025-vm10, stbm000025-vm9,
                      stbm000025-vm12
Version               19.0.0.0.0
Groups                SYSOPER:oinstall SYSASM:oinstall SYSRAC:oinstall
                      SYSDBA:oinstall
OCR locations         +DATA2
Voting disk locations DATA2
Voting disk file paths AFD:DATA1

Cluster network configuration details
-----
Interface name  Type  Subnet      Classification
ens3            IPV4  100.70.0.0/20  PUBLIC
ens4            IPV4  192.168.1.0/24 PRIVATE, ASM
SCAN configuration details
-----
SCAN "stbm000025-scan3" details
+++++
Name                  stbm000025-scan3
IPv4 subnet          100.70.0.0/20
DHCP server type     static
End points           TCP:1521

SCAN listeners
-----
Name                  VIP address
LISTENER_SCAN1       100.70.5.26
LISTENER_SCAN2       100.70.5.20
LISTENER_SCAN3       100.70.5.22
ASM configuration details
-----
Mode                  remote
Password file         +DATA2
SPFILE                +DATA2
ASM disk group details
+++++
Name    Redundancy
DATA2   EXTERN
Database configuration details
=====
Database "ora.orcl.db" details
-----
Name                  ora.orcl.db
Type                  RAC
Version               19.0.0.0.0
Role                  PRIMARY
```


Management policy	AUTOMATIC
SPFILE	+DATA2
Password	+DATA2
file Groups	OSDBA:oinstall OSOPER:oinstall OSBACKUP:oinstall OSDG:oinstall OSKM:oinstall OSRAC:oinstall
Oracle home	/scratch/app/19c/220523/dbhome

srvctl add asm

クラスタ全体にOracle ASMインスタンスに関するレコードを追加します。

このコマンドはOracle Grid Infrastructureホームから1回だけ実行される必要があります。

構文

```
srvctl add asm [-listener listener_name] [-pwfile password_file_path  
[-flex [-count {number_of_instances | ALL}] | -proxy [-spfile  
server_parameter_file_path]]
```

パラメータ

表F-5 srvctl add asmコマンドのパラメータ

パラメータ	説明
-listener listener_name	リスナーの名前を入力します。このパラメータを指定しない場合、リスナー名はデフォルトで LISTENER になります。
-pwfile password_file_path	Oracle ASM のパスワード・ファイルの場所へのフルパスを入力します。
-flex [-count {number_of_instances ALL}] -proxy [-spfile server_parameter_file_path]	Oracle ASM インスタンスが Oracle Flex ASM インスタンスとして構成されることを指定します。必要に応じて、作成する Oracle ASM インスタンスの数を指定したり、すべてのノードに Oracle ASM インスタンスを作成するよう指定できます。 または、-proxy パラメータを指定して、Oracle ASM 動的ボリューム・マネージャのプロキシ・インスタンス・リソースを構成できます。必要に応じて、サーバー・パラメータ・ファイルのパスを指定できます。

使用上のノート

Oracle Database 12c以降のインストールでOracle ASMを管理するには、クラスタのOracle Grid Infrastructureホーム(Gridホーム)にあるSRVCTLバイナリを使用します。Oracle RACまたはOracle Databaseがすでにインストールされている場合、データベース・ホームでSRVCTLバイナリを使用してOracle ASMを管理することはできません。

例

Oracle ASMのクラスタウェア・リソースをクラスタ内のすべてのノードに追加するには、次のコマンドを使用します。

```
$ srvctl add asm
```

srvctl config asm

すべてのOracle ASMインスタンスの構成を表示します。

構文

```
srvctl config asm [-proxy] [-detail]
```

パラメータ

表F-6 srvctl config asmコマンドのパラメータ

パラメータ	説明
-proxy	必要に応じて、このパラメータを使用して、表示する Oracle ASM 構成がプロキシ・インスタンスかどうかを指定できます。
-detail	必要に応じて、このパラメータを使用して、詳細な構成情報を出力できます。

使用上のノート

Oracle Database 12c以降のインストールでOracle ASMを管理するには、クラスタのOracle Grid Infrastructureホーム(Gridホーム)にあるSRVCTLバイナリを使用します。Oracle RACまたはOracle Databaseがすでにインストールされている場合、データベース・ホームでSRVCTLバイナリを使用してOracle ASMを管理することはできません。

例

このコマンドによって、次のような出力が返されます。

```
$ srvctl config asm
ASM Home: /u01/app/19.0.0/grid
ASM Listener: LISTENER
ASM instance count: 3
Password file:
```

srvctl disable asm

Oracle ASMプロキシ・リソースを無効にします。

Oracle ASMにOracle Clusterwareデータが含まれる場合、またはノードでの障害発生前に実行されていたノードが再起動した場合、Oracle ASMは再起動します。また、srvctl disable asmコマンドを使用すると、srvctl start asmコマンドでOracle ASMインスタンスを起動できなくなります。

Oracle ASMプロキシ・リソースを無効にすると、データベースおよびディスク・グループを起動できなくなります。データベースおよびディスク・グループがOracle ASMプロキシ・リソースに依存しているためです。

構文

```
srvctl disable asm [-proxy] [-node node_name]
```

パラメータ

表F-7 srvctl disable asmコマンドのパラメータ

パラメータ	説明
-proxy	必要に応じて、このパラメータを使用して、プロキシの Oracle ASM 構成を無効にするかどうかを指定できます。
-node node_name	必要に応じて、Oracle ASM を無効にする特定のノードを指定できます。

使用上のノート

- Oracle Database 12c以降のインストールでOracle ASMを管理するには、クラスタのOracle Grid Infrastructureホーム(Gridホーム)にあるSRVCTLバイナリを使用します。Oracle RACまたはOracle Databaseがすでにインストールされている場合、データベース・ホームでSRVCTLバイナリを使用してOracle ASMを管理することはできません。
- Oracle ASMはOracle Clusterwareスタックの一部であり、Oracle ASMにOCRと投票ファイルが格納されていると、OHASDがOracle Clusterwareスタックを起動するときにOracle ASMが起動します。srvctl disable asmコマンドを使用しても、OHASDで管理されるOracle ASMインスタンスの起動は行われます。したがって、Oracle ASMは、Oracle Clusterwareスタックが必要とする場合に起動します。

srvctl disable asmコマンドを使用すると、Oracle ASM Oracle Clusterwareプロキシ・リソースとそれに依存するすべてのリソースを起動できなくなります。このため、Oracle Clusterwareの管理対象オブジェクト(データベース、ディスク・グループ、Oracle ASMに依存するファイル・システムなど)を起動できなくなります。たとえば、srvctl start database | diskgroup | filesystemコマンドを実行しても、Oracle ASM Oracle Clusterwareプロキシ・リソースが無効になっているノードのオブジェクトを起動することはできません。また、srvctl start asmコマンドでリモート・ノードのOracle ASMの起動することもできません。

例

次の例は、crmnode1という名前のノード上のOracle ASMを無効にします。

```
$ srvctl disable asm -node crmnode1
```

srvctl enable asm

Oracle ASMインスタンスを有効化します。

構文

```
srvctl enable asm [-proxy] [-node node_name]
```

パラメータ

表F-8 srvctl enable asmコマンドのパラメータ

パラメータ	説明
-proxy	必要に応じて、このパラメータを使用して、プロキシの Oracle ASM 構成を有効にするかどうかを指定できます。
-node node_name	必要に応じて、Oracle ASM を有効にする特定のノードを指

パラメータ	説明
	定できます。

使用上のノート

Oracle Database 12c以降のインストールでOracle ASMを管理するには、クラスタのOracle Grid Infrastructureホーム(Gridホーム)にあるSRVCTLバイナリを使用します。Oracle RACまたはOracle Databaseがすでにインストールされている場合、データベース・ホームでSRVCTLバイナリを使用してOracle ASMを管理することはできません。

例

次に、このコマンドの例を示します。

```
$ srvctl enable asm -node crmnode1
```

srvctl getenv asm

Oracle ASMに関連付けられている環境変数の値を表示します。

構文

```
srvctl getenv asm [-envs "name_list"]
```

使用上のノート

- -envパラメータを使用しなかった場合は、Oracle ASMに関連付けられているすべての環境変数の値が表示されます。
- 値を取得する環境変数の名前をカンマ区切りのリストで指定できます。

例

次の例は、Oracle ASMで使用されるすべての環境変数の現在の値を表示します。

```
$ srvctl getenv asm
```

srvctl modify asm

Oracle ASMで使用するリスナー、Oracle ASMで使用するディスク・グループ検出文字列、または非クラスタ・データベースまたはクラスタ・データベース用にOracle ASMで使用するSPFILEを変更します。

ノート:



Oracle Database 12c 以降のインストールで Oracle ASM を管理するには、クラスタの Oracle Grid Infrastructure ホーム(Grid ホーム)にある SRVCTL バイナリを使用します。Oracle RAC または Oracle Database がすでにインストールされている場合、データベース・ホームで SRVCTL バイナリを使用して Oracle ASM を管理することはできません。

構文

```
srvctl modify asm [-listener listener_name] [-pwfile password_file_path]
[-pwfilebackup backup_password_file_path] [-count {number_of_instances | ALL}] [-diskstring asm_diskstring]
[-spfile spfile_path_name [-proxy] [-force]]
```

パラメータ

パラメータ	説明
-listener listener_name	Oracle ASM が登録するリスナーの名前を指定します。
-pwfile password_file_path	Oracle ASM のパスワード・ファイルの場所へのフルパスを指定します。
-pwfilebackup backup_password_file_path	Oracle ASM パスワード・バックアップ・ファイルの場所を更新する場合は、このオプションを指定します。 ノート: このオプションは、Oracle ASM パスワード・ファイルをバックアップしません。
-count {number_of_instances ALL}	作成する Oracle ASM インスタンスの数を指定したり、すべてのノードに Oracle ASM インスタンスを作成するよう指定できます。
-diskstring asm_diskstring	新しい Oracle ASM ディスク・グループ検出文字列を指定します。
-spfile spfile_path_name	Oracle ASM で使用される新規 SPFILE のパスを指定します。
-proxy	このパラメータを使用して、Oracle ASM プロキシ・インスタンスを指定します。
-force	このパラメータを使用して、Oracle ASM を強制的に変更します。

例

Oracle ASMの構成を変更するこのコマンドの例を示します。

```
$ srvctl modify asm -listener lsnr1
```

srvctl predict asm

Oracle ASM障害の結果の予測

構文

```
srvctl predict asm [-node node_name] [-verbose]
```

使用上のノート

- 必要に応じて、Oracle ASM障害の結果を予測する特定のノードを指定できます。
- 冗長出力を選択することもできます。

例

次のコマンド例は、特定のノードでのOracle ASM障害の結果を予測します。

```
$ srvctl predict asm -node crmnode2
```

srvctl relocate asm

Oracle Flex ASMインスタンスをOracle Flex Clusterのハブ・ノードから別のハブ・ノードに再配置。

構文

```
srvctl relocate asm -currentnode current_node [-targetnode target_node] [-force]
```

パラメータ

表F-9 srvctl relocate asmのパラメータ

パラメータ	説明
-currentnode current_node	Oracle Flex ASM インスタンスが現在存在するノードの名前
-targetnode target_node	Oracle Flex ASM インスタンスを再配置する先のノードの名前
-force	Oracle Flex ASM インスタンスの再配置の強制

使用上のノート

- 実行しているOracle Flex ASMインスタンスが1つのみの場合、このコマンドは使用できません。
- Oracle ASMとOracle Clusterwareの間にブートストラップの問題があるため、Oracle ClusterwareによってOracle ASMが使用されるノードでは、Oracle ASMインスタンスを再配置する前に、`crsctl stop crs`または`crsctl stop cluster`を実行してノードのクラスタ・レディ・サービスを停止します。

例

Oracle Flex ASMインスタンスを現在のノードからhubnode2という名前のノードに再配置するこのコマンドの例を次に示します。

```
$ srvctl relocate asm -currentnode hubnode1 -targetnode hubnode2
```

srvctl remove asm

Oracle Clusterware管理からのOracle ASMリソースの削除。

ノート:



Oracle Database 12c 以降のインストールで Oracle ASM を管理するには、クラスタの Oracle Grid Infrastructure ホーム(Grid ホーム)にある SRVCTL バイナリを使用します。Oracle RAC または Oracle Database がすでにインストールされている場合、データベース・ホームで SRVCTL バイナリを使用して Oracle ASM を管理することはできません。

構文

```
srvctl remove asm [-proxy] [-force]
```

パラメータ

表F-10 srvctl remove asmコマンドのパラメータ

パラメータ	説明
-proxy	必要に応じて、このパラメータを使用して、プロキシの Oracle ASM 構成を削除するかどうかを指定できます。
-force	Oracle ASM リソースを強制的に削除します(すべての依存性を無視します)。

例

次に、このコマンドの例を示します。

```
$ srvctl remove asm -force
```

srvctl setenv asm

Oracle ASM環境構成を設定します。

構文

このコマンドは、次のいずれかの構文で使用します。

```
srvctl setenv asm -envs "name=value, [...]"  
srvctl setenv asm -env "name=value"
```

パラメータ

表F-11 srvctl setenv asmコマンドのパラメータ

パラメータ	説明
-envs "name=value, [...]"	このパラメータを使用して、環境変数の名前/値ペアを二重引用符("")で囲んで指定します。名前/値ペアが複数ある場合は、カンマ区切りのリストで指定します。
-env "name=value"	このパラメータを使用して、単一の環境変数を二重引用符("")で囲み、カンマまたは他の特殊文字を含んだ値に

パラメータ	説明
	設定できるようにします。

例

次の例は、Oracle ASMの言語環境構成を設定します。

```
$ srvctl setenv asm -envs "LANG=en"
```

srvctl start asm

Oracle ASMインスタンスを起動します。

ノート:



Oracle Database 12c 以降のインストールで Oracle ASM を管理するには、クラスタの Oracle Grid Infrastructure ホーム(Grid ホーム)にある SRVCTL バイナリを使用します。Oracle RAC または Oracle Database がすでにインストールされている場合、データベース・ホームで SRVCTL バイナリを使用して Oracle ASM を管理することはできません。

構文

```
srvctl start asm [-proxy] [-node node_name [-startoption start_options]]
```

パラメータ

表F-12 srvctl start asmコマンドのパラメータ

パラメータ	説明
-proxy	必要に応じて、このパラメータを使用して、プロキシの Oracle ASM 構成を起動するかどうかを指定できます。
-node node_name	必要に応じて、Oracle ASM を起動する特定のノードを指定できます。 ノート: このパラメータは、Oracle Clusterware でのみ使用できます。
-startoption start_options	必要に応じて、起動コマンドにオプションを指定できます (OPEN、MOUNT、NOMOUNT など) ノート: 起動オプションに複数の語を指定する場合(read only、read write など)、語をスペースで区切り、二重引用符("")で囲みます。たとえば"read only"とします。

パラメータ	説明
	関連項目: 起動オプションの詳細は、 『SQL*Plus ユーザーズ・ガイドおよびリファレンス』 を参照してください。

例

次のコマンド例は、クラスタのシングル・ノードでASMインスタンスを起動します。

```
$ srvctl start asm -node crmnode1
```

次の例は、クラスタまたは非クラスタ・データベースのすべてのノードでOracle ASMインスタンスを起動します。

```
$ srvctl start asm
```

srvctl status asm

Oracle ASMインスタンスのステータスを表示します。

ノート:



Oracle Database 12c 以降のインストールで Oracle ASM を管理するには、クラスタの Oracle Grid Infrastructure ホーム(Grid ホーム)にある SRVCTL バイナリを使用します。Oracle RAC または Oracle Database がすでにインストールされている場合、データベース・ホームで SRVCTL バイナリを使用して Oracle ASM を管理することはできません。

構文

```
srvctl status asm [-proxy] [-node node_name] [-detail] [-verbose]
```

パラメータ

表F-13 srvctl status asmコマンドのパラメータ

パラメータ	説明
-proxy	必要に応じて、このパラメータを使用して、Oracle ASM プロキシ・インスタンスのステータスを表示できます。
-node node_name	必要に応じて、特定のノードを指定できます。このパラメータを指定しない場合、SRVCTL によってすべての Oracle ASM インスタンスのステータスが表示されます。
-detail	詳細なステータス情報の出力。
-verbose	冗長出力を表示します。

例

このコマンドによって、次のような出力が返されます。

```
$ srvctl status asm -verbose
ASM is running on node2, node3, node4
Detailed state on node node2: Started
Detailed state on node node3: Started
Detailed state on node node4: Started
```

srvctl stop asm

Oracle ASMインスタンスを停止します。

構文

```
srvctl stop asm [-proxy] [-node node_name] [-stopoption stop_options] [-force]
```

パラメータ

表F-14 srvctl stop asmコマンドのパラメータ

パラメータ	説明
-proxy	必要に応じて、このパラメータを使用して、プロキシの Oracle ASM 構成を停止するかどうかを指定できます。
-node node_name	必要に応じて、Oracle ASM を停止する特定のノードを指定できます。 このパラメータを指定しなかった場合、SRVCTL はクラスタ内にあるすべてのアクティブなノードで Oracle ASM インスタンスを停止します。 ノート: このパラメータは、Oracle Clusterware でのみ使用できます。
-stopoption stop_options	必要に応じて、停止コマンドにオプションを指定できます (NORMAL、TRANSACTIONAL、IMMEDIATE、ABORT など)。 関連項目: 停止オプションの詳細は、『SQL*Plus ユーザーズ・ガイドおよびリファレンス』も参照してください。
-force	必要に応じて、このパラメータを使用して、Oracle ASM に依存するディスク・グループ、ファイル・システムおよびデータベースを停止できます。

使用上のノート

- Oracle Database 12c以降のインストールでOracle ASMを管理するには、クラスタ用Oracle Grid Infrastructureホーム(Gridホーム)でSRVCTLを使用します。Oracle RACまたはOracle Databaseがすでにインストールされている場合、データベース・ホームでSRVCTLを使用してOracle ASMを管理することはできません。

- このコマンドはOracle ASMを停止しないため、Oracle ASMにOCRが格納されているときには使用できません。Oracle ASMを停止するには、Oracle Clusterwareを停止する必要があります。

例

次に、このコマンドの例を示します。

```
$ srvctl stop asm -node crmnode1 -stopoption IMMEDIATE
```

srvctl unsetenv asm

Oracle ASM環境構成の設定を解除します。

構文

```
srvctl unsetenv asm -envs "name_list"
```

使用上のノート

-envsパラメータを使用して、二重引用符(" ")で囲まれた環境変数名のカンマ区切りリストを指定します。

例

次の例は、Oracle ASM環境変数の環境構成の設定を解除します。

```
$ srvctl unsetenv asm -envs "CLASSPATH"
```

srvctl add asmnetwork

Oracle ClusterwareにOracle ASMネットワーク構成を追加します。

構文

```
srvctl add asmnetwork [-netnum asm_network_number] [-deptype {optional | mandatory}] -subnet subnet[/netmask/iface1[|iface2...]]
```

パラメータ

表F-15 srvctl add asmnetworkコマンドのパラメータ

パラメータ	説明
-netnum asm_network_number	必要に応じて、追加する Oracle ASM ネットワーク番号(デフォルトは 1)を指定できます。
-deptype {optional mandatory}	Oracle ASM 依存性タイプがオプションか必須かを指定します。デフォルトは optional です。
-subnet subnet[/netmask/iface1[iface2...]]	追加する Oracle ASM ネットワークのサブネットを指定します。インタフェース名を指定しない場合、ネットワークは指定されたサブネットの任意のインタフェースを使用します。 IPv6 の場合、netmask は接頭辞の長さ(64 など)です。

srvctl config asmnetwork

Oracle Clusterwareに登録されているOracle ASMネットワークの構成情報を表示します。

構文

```
srvctl config asmnetwork [-netnum asm_network_number]
```

使用上のノート

必要に応じて、構成情報を表示するOracle ASMネットワーク番号(デフォルトは1)を指定できます。

srvctl modify asmnetwork

Oracle ClusterwareでOracle ASMネットワーク構成を変更します。

構文

```
srvctl modify asmnetwork [-netnum asm_network_number] [-deptype {optional | mandatory}]
```

パラメータ

表F-16 srvctl modify asmnetworkコマンドのパラメータ

パラメータ	説明
-netnum asm_network_number	必要に応じて、変更する Oracle ASM ネットワーク番号(デフォルトは 1)を指定できます。
-deptype {optional mandatory}	Oracle ASM 依存性タイプがオプションか必須かを指定します。

srvctl remove asmnetwork

Oracle ClusterwareからOracle ASMネットワーク構成を削除します。

構文

```
srvctl remove asmnetwork {-netnum asm_network_number | -all} [-force]
```

パラメータ

表F-17 srvctl remove asmnetworkコマンドのパラメータ

パラメータ	説明
-netnum asm_network_number -all	削除する Oracle ASM ネットワーク番号(デフォルトは 1)を指定するか、-all を指定して Oracle Clusterware に構成されているすべての Oracle ASM ネットワーク番号を削除します。
-force	依存性に関係なく Oracle ASM ネットワークを強制的に削除

パラメータ	説明
するには、このパラメータを使用します。	

srvctl add cdp

クラスタ間依存性プロキシ・リソースをクラスタに追加します(最初のパブリック・ネットワーク上のみ)。

構文

```
srvctl add cdp [-port port_number] [-passfile_admin afile] [-passfile_readonly rfile]
```

パラメータ

表F-18 srvctl add cdpコマンドのパラメータ

パラメータ	説明
-port port_number	必要に応じて、クラスタ内のすべてのクラスタ間依存性プロキシ・リソースがリスニングするポートを指定できます。ポートを指定しない場合、SRVCTL によって、ドメイン・サービス・クラスタのすべてのノードで使用可能なポートが選択されます。
-passfile_admin afile	管理者用のパスワード・ファイルがある場合は、このパラメータを使用して指定できます。パスワード・ファイルが存在しない場合、SRVCTL によって、パスワードがランダムに生成されます。
-passfile_readonly rfile	読取り専用ユーザーのパスワード・ファイルがある場合は、このパラメータを使用して指定できます。パスワード・ファイルが存在しない場合、SRVCTL によって、ランダムに生成されたパスワードがファイルに書き込まれます。

srvctl disable cdp

クラスタ間依存性プロキシ・リソースを無効にします。

構文

```
srvctl disable cdp [-cdpnumber cdp_ordinal_number]
```

使用上のノート

必要に応じて、無効化する個別のクラスタ間依存性プロキシ・リソースの序数を指定できます。これを行わない場合、SRVCTL によって、特定のクラスタ内のすべてのクラスタ間依存性プロキシ・リソースが無効化されます。

srvctl enable cdp

クラスタ間依存性プロキシ・リソースを有効にします。

構文

```
srvctl enable cdp [-cdpnumber cdp_ordinal_number]
```

使用上のノート

必要に応じて、有効化する個別のクラスタ間依存性プロキシ・リソースの序数を指定できます。これを行わない場合、SRVCTLによって、特定のクラスタ内のすべてのクラスタ間依存性プロキシ・リソースが有効化されます。

srvctl modify cdp

クラスタ間依存性プロキシ・リソースのポートとパスワードを変更します。

構文

```
srvctl modify cdp [-port port_number] [-passfile_admin afile] [-passfile_readonly rfile]
```

パラメータ

表F-19 srvctl modify cdpコマンドのパラメータ

パラメータ	説明
-port port_number	必要に応じて、クラスタ内のすべてのクラスタ間依存性プロキシ・リソースがリスニングするポートを変更できます。
-passfile_admin afile	必要に応じて、管理者のパスワード情報を変更できます。
-passfile_readonly rfile	必要に応じて、読取り専用ユーザーのパスワード情報を変更できます。

使用上のノート

変更を行う前に、クラスタ間依存性プロキシ・リソースを停止する必要があります。

srvctl relocate cdp

クラスタ内のクラスタ間依存性プロキシ・リソースを再配置します。

構文

```
srvctl relocate cdp -cdpnumber cdp_ordinal_number [-node node_name] [-force]
```

パラメータ

表F-20 srvctl relocate cdpコマンドのパラメータ

パラメータ	説明
-cdpnumber cdp_ordinal_number	識別序数を指定して、再配置するクラスタ間依存性プロキシ・リソースを指定します。

パラメータ	説明
-node node_name	必要に応じて、再配置するクラスタ間依存性プロキシ・リソースが存在するノードの名前を指定できます。
-force	必要に応じて、このパラメータを使用して、実行中のVIPでクラスタ間依存性プロキシ・リソースを再配置できます。実行中のVIPでこのパラメータを使用しないでクラスタ間依存性プロキシ・リソースを再配置しようとすると、コマンドは失敗します。

srvctl remove cdp

クラスタ間依存性プロキシ・リソースを削除せずに、クラスタからクラスタ間依存性プロキシ・リソースを削除します。

構文

```
srvctl remove cdp [-force]
```

使用上のノート

- -forceパラメータを使用して、実行中のクラスタ間依存性プロキシ・リソースを削除します。
- このコマンドは、すべてのクラスタ間依存性プロキシ構成を完全に削除します。プロキシが存在する場合に-forceを使用すると、プロキシは再作成されるまで機能しなくなります。

srvctl start cdp

クラスタ上でクラスタ間依存性プロキシ・リソースを起動します。

構文

```
srvctl start cdp [-cdpnumber cdp_ordinal_number] [-node node_name]
```

パラメータ

表F-21 srvctl start cdpコマンドのパラメータ

パラメータ	説明
-cdpnumber cdp_ordinal_number	必要に応じて、このパラメータを使用して、識別序数を指定すると、個別のクラスタ間依存性プロキシ・リソースを起動できます。
-node node_name	必要に応じて、起動するクラスタ間依存性プロキシ・リソースが存在するノードの名前を指定できます。

srvctl status cdp

クラスタ間依存性プロキシ・リソースのステータスを取得します。

構文

```
srvctl status cdp [-cdpnumber cdp_ordinal_number]
```

使用上のノート

必要に応じて、個別のクラスタ間依存性プロキシ・リソースの序数を指定できます。これを行わない場合、SRVCTLによって、特定のクラスタ内のすべてのクラスタ間依存性プロキシ・リソースのステータスが返されます。

srvctl stop cdp

クラスタ上のクラスタ間の依存性プロキシ・リソースを停止します。

構文

```
srvctl stop cdp [-cdpnumber cdp_ordinal_number]
```

使用上のノート

必要に応じて、識別序数を指定して、特定のクラスタ間依存性プロキシ・リソースを停止できます。

srvctl add cdpproxy

クラスタ間依存性プロキシ・リソースを特定のクラスタに追加します(資格証明の存在、または資格証明の事前インポートが必要です)。

構文

```
srvctl add cdpproxy -client_type client_type -client_name client_name  
[-remote_start {YES|NO}]
```

パラメータ

表F-22 srvctl add cdpproxyコマンドのパラメータ

パラメータ	説明
-client_type client_type	プロキシするクラスタ・リソース・タイプを指定します。
-client_name client_name	プロキシするクラスタ・リソースの名前を指定します。
-remote_start {YES NO}	リモート・リソースの起動時にプロキシ・リソースを起動するかどうかを指定します。

使用上のノート

- Oracle Clusterware 18cの場合、クライアント・タイプは常にdiskgroupです。
- Oracle Clusterware 18cの場合、クライアント名は、リモート・ディスク・グループの名前です。

srvctl config cdpproxy

クラスタ間依存性プロキシ・リソースの構成情報を表示します。

構文


```
srvctl config cdpproxy -client_type client_type [-client_name client_name]
```

パラメータ

表F-23 srvctl config cdpproxyコマンドのパラメータ

パラメータ	説明
-client_type client_type	構成情報を表示するプロキシのクラスタ・リソース・タイプ。
-client_name client_name	構成情報を表示するクラスタ・リソース・プロキシの名前を指定します。

使用上のノート

- Oracle Clusterware 18cの場合、クライアント・タイプは常にdiskgroupです。
- Oracle Clusterware 18cの場合、クライアント名は、リモート・ディスク・グループの名前です。

srvctl disable cdpproxy

クラスタ間依存性プロキシ・リソースを無効にします。

構文

```
srvctl disable cdpproxy -client_type client_type -client_name client_name
```

パラメータ

表F-24 srvctl disable cdpproxyコマンドのパラメータ

パラメータ	説明
-client_type client_type	無効化するプロキシのクラスタ・リソース・タイプ。
-client_name client_name	無効化するクラスタ・リソース・プロキシの名前を指定します。

使用上のノート

- Oracle Clusterware 18cの場合、クライアント・タイプは常にdiskgroupです。
- Oracle Clusterware 18cの場合、クライアント名は、リモート・ディスク・グループの名前です。

srvctl enable cdpproxy

クラスタ間依存性プロキシ・リソースを有効にします。

構文

```
srvctl enable cdpproxy -client_type client_type -client_name client_name
```

パラメータ

表F-25 srvctl enable cdpproxyコマンドのパラメータ

パラメータ	説明
-client_type client_type	有効化するプロキシのクラスタ・リソース・タイプ。
-client_name client_name	有効化するクラスタ・リソース・プロキシの名前を指定します。

使用上のノート

- Oracle Clusterware 18cの場合、クライアント・タイプは常にdiskgroupです。
- Oracle Clusterware 18cの場合、クライアント名は、リモート・ディスク・グループの名前です。

srvctl modify cdpproxy

クラスタ間依存性プロキシ・リソースの構成を変更します。

構文

```
srvctl modify cdpproxy -client_type client_type -client_name client_name
[-remote_start {YES|NO}]
```

パラメータ

表F-26 srvctl modify cdpproxyコマンドのパラメータ

パラメータ	説明
-client_type client_type	プロキシするクラスタ・リソース・タイプ。
-client_name client_name	プロキシするクラスタ・リソースの名前を指定します。
-remote_start {YES NO}	リモート・リソースの起動時にプロキシ・リソースを起動するかどうかを指定します。

使用上のノート

- Oracle Clusterware 18cの場合、クライアント・タイプは常にdiskgroupです。
- Oracle Clusterware 18cの場合、クライアント名は、リモート・ディスク・グループの名前です。

srvctl relocate cdpproxy

クラスタのノード間でクラスタ間依存性プロキシを再配置します。

構文

```
srvctl relocate cdpproxy -client_type client_type -client_name client_name
[-node node_name] [-force]
```

パラメータ

表F-27 srvctl relocate cdpproxyコマンドのパラメータ

パラメータ	説明
-client_type client_type	再配置するプロキシのクラスタ・リソース・タイプを指定します。
-client_name client_name	再配置するクラスタ・リソース・プロキシの名前を指定します。
-node node_name	必要に応じて、クラスタ間依存性プロキシの再配置先のノードの名前を指定できます。
-force	クラスタ間依存性プロキシに依存するリソースが実行されている場合は、このパラメータを使用する必要があります。

使用上のノート

- Oracle Clusterware 18cの場合、クライアント・タイプは常にdiskgroupです。
- Oracle Clusterware 18cの場合、クライアント名は、リモート・ディスク・グループの名前です。

srvctl remove cdpproxy

クラスタ間依存性プロキシ・リソースを削除します。

構文

```
srvctl remove cdpproxy -client_type client_type [-client_name client_name]
[-force]
```

パラメータ

表F-28 srvctl remove cdpproxyコマンドのパラメータ

パラメータ	説明
-client_type client_type	削除するプロキシのクラスタ・リソース・タイプを指定します。
-client_name client_name	必要に応じて、削除するクラスタ・リソース・プロキシの名前を指定できます。
-force	削除するクラスタ間依存性プロキシ・リソースが実行中の場合は、このパラメータを使用する必要があります。

使用上のノート

- Oracle Clusterware 18cの場合、クライアント・タイプは常にdiskgroupです。
- Oracle Clusterware 18cの場合、クライアント名は、リモート・ディスク・グループの名前です。

srvctl start cdpproxy

クラスタ間依存性プロキシ・リソースを起動します。

構文

```
srvctl start cdpproxy -client_type client_type -client_name client_name  
[-node node_name]
```

パラメータ

表F-29 srvctl start cdpproxyコマンドのパラメータ

パラメータ	説明
-client_type client_type	起動するプロキシのクラスタ・リソース・タイプを指定します。
-client_name client_name	起動するクラスタ・リソース・プロキシの名前を指定します。
-node node_name	必要に応じて、クラスタ間依存性プロキシを起動するノードの名前を指定できます。

使用上のノート

- Oracle Clusterware 18cの場合、クライアント・タイプは常にdiskgroupです。
- Oracle Clusterware 18cの場合、クライアント名は、リモート・ディスク・グループの名前です。

srvctl status cdpproxy

クラスタ間依存性プロキシ・リソースのステータスを取得します。

構文

```
srvctl status cdpproxy -client_type client_type -client_name client_name]
```

パラメータ

表F-30 srvctl status cdpproxyコマンドのパラメータ

パラメータ	説明
-client_type client_type	ステータスをチェックするプロキシのクラスタ・リソース・タイプを指定します。
-client_name client_name	ステータスをチェックするクラスタ・リソース・プロキシの名前を指定します。

使用上のノート

- Oracle Clusterware 18cの場合、クライアント・タイプは常にdiskgroupです。
- Oracle Clusterware 18cの場合、クライアント名は、リモート・ディスク・グループの名前です。

srvctl stop cdpproxy

クラスタ間依存性プロキシ・リソースを停止します。

構文

```
srvctl stop cdpproxy -client_type client_type -client_name client_name  
[-node node_name]
```

パラメータ

表F-31 srvctl stop cdpproxyコマンドのパラメータ

パラメータ	説明
-client_type client_type	停止するプロキシのクラスタ・リソース・タイプを指定します。
-client_name client_name	停止するクラスタ・リソース・プロキシの名前を指定します。
-node node_name	必要に応じて、クラスタ間依存性プロキシ・リソースを停止するノードの名前を指定できます。

使用上のノート

- Oracle Clusterware 18cの場合、クライアント・タイプは常にdiskgroupです。
- Oracle Clusterware 18cの場合、クライアント名は、リモート・ディスク・グループの名前です。

srvctl add cvu

Oracle Clusterware構成へのクラスタ検証ユーティリティ(CVU)の追加。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文およびパラメータ

srvctl add cvuコマンドは、次の構文で使します。

```
srvctl add cvu [-checkinterval time_in_minutes]
```

使用上のノート

- 必要に応じて、チェック間隔を変更できます。デフォルトでは、CVUは6時間に1回実行され、クラスタの健全性を検証します。CVUチェックに指定できる最も短い間隔は10分です。
- Gridホーム・ユーザーとしてログインしている間にこのコマンドを実行してください。

例

チェック間隔が6時間のCVUを追加するには、次のコマンドを使用します。

```
$ srvctl add cvu -checkinterval 360
```

srvctl config cvu

CVUチェックの間隔(分数)を表示します。

構文およびパラメータ

srvctl config cvuコマンドは、次の構文で使⽤します。

```
srvctl config cvu
```

このコマンドにパラメータはありません。

例

このコマンドによって、次のような出力が返されます。

```
CVU is configured to run once every 360 minutes
```

srvctl disable cvu

Oracle Clusterware管理用のクラスタ検証ユーティリティ(CVU)を無効にします(有効になっている場合)。

構文

```
srvctl disable cvu [-node node_name]
```

パラメータ

必要に応じて、CVUを無効にする特定のノードの名前を指定できます。指定しなかった場合、SRVCTLはすべてのノードでCVUを無効化します。

例

次のコマンド例は、crmnode1というノードのCVUを無効化します。

```
$ srvctl disable cvu -node crmnode1
```

srvctl enable cvu

Oracle Clusterware管理用のクラスタ検証ユーティリティ(CVU)を有効にします(無効になっている場合)。

構文

```
srvctl enable cvu [-node node_name]
```

パラメータ

必要に応じて、CVUを有効にする特定のノードの名前を指定できます。指定しなかった場合、SRVCTLはすべてのノードでCVUを有効化します。

例

次のコマンド例は、crmnode1というノードのCVUを有効化します。

```
$ srvctl enable cvu -node crmnode1
```

srvctl modify cvu

CVUにチェック間隔を変更します。

構文

```
srvctl modify cvu [-checkinterval time_in_minutes]
```

パラメータ

デフォルトでは、CVUは360分(6時間)に1回実行され、クラスタの健全性を検証します。間隔を変更する場合、CVUチェックに指定できる最も短い間隔は10分です。

例

次のコマンド例は、CVUのチェック間隔を4時間に変更します。

```
$ srvctl modify cvu -checkinterval 240
```

srvctl relocate cvu

クラスタの別のノードにCVUを一時的に再配置します。

構文

```
srvctl relocate cvu [-node node_name]
```

使用上のノート

必要に応じて、CVUの再配置先となる特定のノードを指定できます。

例

次のコマンド例は、ローカル・ノードのCVUをcrmnod2というクラスタ・ノードに一時的に再配置します。

```
$ srvctl relocate cvu -node crmnod2
```

srvctl remove cvu

Oracle Clusterware構成からCVUを削除します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl remove cvu [-force]
```

使用上のノート

-forceパラメータを使用して、実行中または起動中のCVUリソースを削除します。

srvctl start cvu

クラスタの1つのノードでCVUリソースを起動します。ノード名を指定した場合、CVUはそのノードで起動します。

ノート:



このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl start cvu [-node node_name]
```

使用上のノート

必要に応じて、CVUを起動する特定のノードを指定できます。指定しなかった場合、SRVCTLはすべてのノードでCVUを起動します。

例

次のコマンド例は、クラスタのシングル・ノードでCVUを起動します。

```
$ srvctl start cvu -node crmnode1
```

srvctl status cvu

クラスタの1つのノードでCVUリソースの現在の状態を表示します。ノード名を指定すると、コマンドはそのノードのCVUステータスを確認します。

ノート:



このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl status cvu [-node node_name]
```

使用上のノート

必要に応じて、CVUリソースが実行されている特定のノードを指定できます(ローカル・ノードでない場合)。

例

次のコマンド例は、クラスタ内のcrmnode1というノードでCVUのステータスを確認します。

```
$ srvctl status cvu -node crmnode1
```

srvctl stop cvu

実行中または起動中のいずれかのステータスのクラスタ検証ユーティリティ(CVU)を停止します。

ノート:



このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl stop cvu [-force]
```


使用上のノート

必要に応じて、-forceパラメータを指定してCVUを停止できます(それが依存リソースである場合)。

例

次のコマンド例は、CVUを停止します。

```
$ srvctl stop cvu
```

srvctl add exportfs

Oracle Clusterwareでエクスポート・ファイル・システムの構成を作成します。

構文

```
srvctl add exportfs -name unique_name -id havip_name -path path_to_export  
[-clients export_clients] [-options export_options] [-type NFS | SMB]
```

パラメータ

表F-32 srvctl add exportfsコマンドのパラメータ

パラメータ	説明
-name unique_name	作成するエクスポート・ファイル・システムの一意の名前を入力します。このパラメータは必須です。
-id havip_name	srvctl add havip コマンドで指定した既存の HAVIP を識別する番号または名前を入力します。このパラメータは必須です。
-path path_to_export	エクスポートする Oracle Automatic Storage Management Cluster File System (Oracle ACFS)へのパスを入力します。このパラメータは必須です。
-clients export_clients -options export_options	必要に応じて、ファイル・システムのエクスポート先クライアントのカンマ区切りリストを入力します。ネット・グループおよびワイルドカードを入力できます。たとえば: -clients @netgroup1, *.us.domain.com, 16.140.0.0, hostname1 ノート: -clients パラメータは、Linux システムにのみ使用できます。Solaris システムおよび AIX システムには、-options パラメータを使用します。
-type NFS SMB	エクスポート・タイプ(ネットワーク・ファイル・システムの NFS または Oracle ACFS サーバー・メッセージ・ブロックの SMB)を指定します。デフォルトは、Linux および UNIX プラットフォームの場合は NFS、Windows の場合は SMB です。

使用上のノート

- -type SMBとともに-cclientsパラメータを使用することはできません。

- LinuxおよびUNIXプラットフォームで、rootとしてこのコマンドを実行する必要があります。

例

オペレーティング・システムのデフォルト値を使用してエクスポートするには:

```
# srvctl add exportfs -name cifsExport -path "/acfs1" -options "browsable = yes,read  
only = no,comment = acfs1"
```

特定のクライアントに読み取り専用でエクスポートするには:

```
# srvctl add exportfs -name export1 -id havip1 -path /u01/db1 -clients node1 -options  
ro
```

srvctl config exportfs

Oracle Clusterwareのエクスポート・ファイル・システムの構成情報を表示します。

構文

```
srvctl config exportfs [-name unique_name | -id havip_name]
```

パラメータ

表F-33 srvctl config exportfsコマンドのパラメータ

パラメータ	説明
-name unique_name	構成を表示する NFS エクスポートの一意の名前を入力します。
-id havip_name	特定の HAVIP の名前を入力し、その HAVIP に関連付けられているすべての NFS エクスポートの構成を表示します。

例

-nameパラメータを指定する場合、このコマンドでは、-nameと-idのどちらを指定するかに関係なく、次のような出力が戻されます。

```
$ srvctl config exportfs -id havip1  
export file system kep1 is configured  
Exported path: /scratch/ghnew/base/mymount3  
Export options:  
Export Type:  
Exported clients:  
export file system kexp2 is configured  
Exported path: /scratch/ghnew/base/mymount3  
Export options:  
Export Type:  
Exported clients:
```

srvctl disable exportfs

Oracle Clusterwareで管理されるエクスポート・ファイル・システムを無効化します。

構文

```
srvctl disable exportfs -name filesystem_name
```

使用上のノート

- このコマンドは、無効にするファイル・システムの一意の名前のみを取ります。
- LinuxおよびUNIXプラットフォームで、rootユーザーとしてこのコマンドを実行する必要があります。

例

次の例では、export1というエクスポート・ファイル・システムを無効にします。

```
# srvctl disable exportfs -name export1
```

srvctl enable exportfs

Oracle Clusterwareでエクスポート・ファイル・システムの構成を有効化します。

構文

```
srvctl enable exportfs -name filesystem_name
```

このコマンドは、必須パラメータとして、有効にするファイル・システムの名前のみを取ります。

使用上のノート

このコマンドは、LinuxおよびUNIXプラットフォームではrootユーザーとして実行する必要があります。

例

次のコマンド例では、export1というファイル・システムを有効にします。

```
# srvctl enable exportfs -name export1
```

srvctl modify exportfs

Oracle Clusterwareでエクスポート・ファイル・システムの構成を変更します。

構文

```
srvctl modify exportfs -name unique_name [-path path_to_export]
[-clients node_list] [-options nfs_options_string]
```

パラメータ

表F-34 srvctl modify exportfsコマンドのパラメータ

パラメータ	説明
-name unique_name	変更する NFS エクスポート・ファイル・システムの一意の名前を入力します。このパラメータは必須です。
-path path_to_export	エクスポートが配置されている Oracle Automatic Storage Management Cluster File System (Oracle ACFS)へのパスを変更します。

パラメータ	説明
<code>-clients node_list</code>	ファイル・システムのエクスポート先クライアント(ホスト)のカンマ区切りリストを変更できます。ネット・グループおよびワイルドカードを入力できます。たとえば: <pre>-clients @netgroup1, *.us.domain.com, 16.140.0.0, hostname1</pre> ノート: <code>-clients</code> パラメータは、Linux システムにのみ使用できます。Solaris システムおよび AIX システムには、<code>-options</code> パラメータを使用します。
<code>-options nfs_client_string</code>	このパラメータを使用して、エクスポート・ファイル・システムのエクスポート・オプションを変更します。

使用上のノート

- LinuxおよびUNIXプラットフォームで、rootユーザーとしてこのコマンドを実行する必要があります。
- 変更を有効にするには、事前にNFSエクスポートを開始し、停止する必要があります。

例

次のコマンド例では、`exportfs1`というエクスポート・ファイル・システムのパスを変更します。

```
# srvctl modify exportfs -name exportfs1 -path /mnt/racdb1
```

srvctl remove exportfs

指定したエクスポート・ファイル・システムの構成を削除します。

構文

```
srvctl remove exportfs -name exportfs_name [-force]
```

パラメータ

表F-35 srvctl remove exportfsコマンドのパラメータ

パラメータ	説明
<code>-name exportfs_name</code>	削除するエクスポート・ファイル・システムの一意の名前を入力します。
<code>-force</code>	このパラメータを使用して、すべての依存性を無視して、エクスポート・ファイル・システムを強制的に削除します。

使用上のノート

- LinuxおよびUNIXプラットフォームで、rootユーザーとしてこのコマンドを実行する必要があります。

- -forceパラメータを使用しない場合は、エクスポート・ファイル・システムを事前に停止してから削除を実行してください。
- -forceパラメータを使用して得られた構成においては、Oracle Clusterwareスタックをシャットダウンしようとしたときにエラーが発生することがあります。

例

export1という停止済のエクスポート・ファイル・システムを削除するには:

```
# srvctl remove exportfs -name export1
```

実行中のエクスポート・ファイル・システムをそのままの状態に削除するには:

```
# srvctl remove exportfs -name export1 -force
```

srvctl start exportfs

Oracle Clusterwareでエクスポート・ファイル・システムの構成を起動します。

構文

```
srvctl start exportfs {-name unique_name | -id havip_name}
```

パラメータ

表F-36 srvctl start exportfsコマンドのパラメータ

パラメータ	説明
-name unique_name	起動する NFS エクスポートの一意の名前を入力します。
-id havip_name	HAVIP リソースに関連付けられている一意の ID。

例

エクスポート・ファイル・システムを起動します。

```
$ srvctl start exportfs -name export1
```

srvctl status exportfs

エクスポート・ファイル・システムの構成のステータスを表示します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl status exportfs [-name unique_name | -id havip_name]
```

パラメータ

表F-37 srvctl status exportfsコマンドのパラメータ

パラメータ	説明
-name unique_name	必要に応じて、ステータスを表示する NFS エクスポートの一意の名前を指定できます。
-id havip_name	あるいは、HAVIP リソースの名前を指定し、指定した HAVIP リソースに関連付けられているすべての NFS エクスポートのステータスを表示することもできます。

使用上のノート

どのパラメータも指定しない場合、クラスタ内のすべての NFS エクスポートのステータスが SRVCTL により表示されます。

例

このコマンドによって、次のような出力が返されます。

```
$ srvctl status exportfs
export file system export1 is enabled
export file system export1 is not exported
export file system export2 is enabled
export file system export2 is exported on node node1
```

srvctl stop exportfs

Oracle Clusterware で管理されるエクスポート・ファイル・システムの構成を停止します。

構文

```
srvctl stop exportfs {-name unique_name | -id havip_name} [-force]
```

パラメータ

表F-38 srvctl stop exportfs コマンドのパラメータ

パラメータ	説明
-name unique_name	停止する NFS エクスポートの一意の名前を入力します。
-id havip_name	または、HAVIP の名前を指定できます。このパラメータを指定した場合、指定した HAVIP リソースに関連付けられているすべての NFS エクスポートが SRVCTL により停止されます。
-force	エラーを無視して NFS エクスポートを停止します。

使用上のノート

Linux および UNIX プラットフォームで、root ユーザーとしてこのコマンドを実行する必要があります。

例

次に、このコマンドの例を示します。

```
# srvctl stop exportfs -name export1
```

srvctl add filesystem

ファイル・システム(Oracle Automatic Storage Management Cluster File System (Oracle ACFS)またはその他)を含むデバイスをOracle Clusterwareスタックに追加して、自動マウントおよび高可用性を実現します。

Oracle ACFSファイル・システム・リソースは通常、アプリケーション・リソースの依存性リストで使用するために作成されます。たとえば、Oracle Databaseホームとして使用するためにOracle ACFSファイル・システムが構成されている場合、ファイル・システム用に作成されたリソースは、Oracle Databaseアプリケーションのリソース依存性リストに含まれます。これによって、データベース・アプリケーションの起動アクションのためにファイル・システムおよびスタックが自動的にマウントされます。

構文

```
srvctl add filesystem {-device volume_device | -volume volume_name -diskgroup disk_group_name}
    -path mountpoint_path [-node "node_list" | -serverpool "server_pool_list"]
    [-user "user_list"] [-mountowner user_name] [-mountgroup group_name] [-mountperm octal_permission]
    [-fstype {ACFS | EXT3 | EXT4}] [-fsoptions options] [-description description]
    [-appid application_id] [-autostart {ALWAYS | NEVER | RESTORE}] [-acceleratorvols volume_name]
```

ノート:

-fstype パラメータの値 EXT3 および EXT4 は、Linux に対してのみサポートされています。

パラメータ

表F-39 srvctl add filesystemコマンドのパラメータ

パラメータ	説明
-device volume_device	ファイル・システム・ボリューム・デバイスへのパスを指定します。
-volume volume_name	ボリュームの名前を指定します。
-diskgroup disk_group_name	-volume があるディスク・グループの名前を指定します。
-path mountpoint_path	マウントをネストするために他のリソースへの依存性を設定する場合に使用する、マウント・ポイント・パスを指定します。これは、絶対的パスである必要があります。
-node node_list -serverpool server_pool_list	必要に応じて、ファイル・システム・デバイスのマウント先ノードのカンマ区切りリストを指定できます。ノードまたはサーバー・プールのリストを指定した場合は、SRVCTL によって単一ノードのファイル・システム・リソースが作成されます。ノードまたはサーバー・プールを指定しない場合、SRVCTL はすべて

パラメータ	説明
	のクラスタ・ノードで実行されるファイル・システム・リソースを作成します。 あるいは、ファイル・システムが実行されるサーバー・プールのカンマ区切りリストを指定することもできます。サーバー・プールのリストを指定する場合、ファイル・システムが実行されるノードの数またはタイプを動的に制限します。これは、HANFS などのクラスタ全体の Oracle ACFS リソースを必要とする可能性のある今後の使用に対しては、クラスタ全体のファイル・システムとして考慮されません。 ノート: ノードとサーバー・プールは相互に排他的です。
-user user_list	必要に応じて、ファイル・システムのマウントおよびアンマウントを許可されたユーザーのカンマ区切りリストを指定できます。 ファイル・システムが Oracle Database ソフトウェア用の場合、Oracle ホームの所有者の名前を指定します。 ユーザーを指定しない場合は、root のみがファイルシステムをマウントおよびアンマウントでき、ファイルシステムで <code>srvctl start filesystem</code> を実行できます。
-mountowner user_name	オプションで、マウント・ポイント所有者であるユーザーを指定することもできます。
-mountgroup group_name	オプションで、マウント・ポイント所有者を含むグループを指定することもできます。
-mountperm octal_permission	オプションで、マウント・ポイントの権限(OCTAL 形式)を指定することもできます。
-fstype {ACFS EXT3 EXT4}	必要に応じて、マウントするファイル・システムのタイプを指定できます。デフォルト値は ACFS です。 マウントするファイル・システムのタイプが ACFS ではない場合、単一ノードのファイル・システムである必要があり、-node または -serverpool を指定する必要があります。
-fsoptions options	必要に応じて、ファイル・システムのマウントに使用するオプションを指定できます。オプションはファイル・システムおよびオ

パラメータ	説明
	ペレーティング・システムによって異なります。
-description description	必要に応じて、ファイル・システムの説明を指定できます。これは、管理者が後で参照できるよう属性として格納されます。
-appid application_id	必要に応じて、アプリケーション ID (ファイル・システムを論理グループに分類するために使用する一意の識別子)を指定できます。この方法は、各種ノードによって異なる可能性があるファイル・システムのうち、リソース・タイプ <code>ora.id.fs.type</code> への依存が必要なアプリケーションで使用されるファイル・システムを分類する目的で使用できます。
-autostart {ALWAYS NEVER RESTORE}	オプションで、ファイル・システム・リソースの autostart ポリシーを指定することもできます。 • ALWAYS: 常にファイル・システム・リソースが自動的に開始されます。 • ALWAYS: ファイル・システム・リソースは自動的に開始されません。 • RESTORE: ファイル・システムは、最後の状態にリストアされます。このオプションがデフォルトです。
-acceleratorvols volume_name	オプションで、アクセラレータ・ボリュームを指定することもできます。アクセラレータ・ボリュームは、Oracle ACFS によって重要なメタデータの格納に使用されるセカンダリ Oracle ACFS ボリュームです。

使用上のノート

- 追加するファイル・システム・リソースへのパスを -device パラメータで指定するか、-volume volume_name -diskgroup disk_group_name を指定します。
- このコマンドは、Linux および UNIX プラットフォームでは root ユーザーとして実行し、Windows プラットフォームでは管理者ユーザーとして実行する必要があります。
- このコマンドは Oracle Grid Infrastructure ホームから 1 回だけ実行される必要があります。
- Oracle Database インストールで Oracle ACFS を管理するには、クラスタの Oracle Grid Infrastructure ホーム (Grid ホーム) にある SRVCTL バイナリを使用します。Oracle RAC または Oracle Database がすでにインストールされている場合、データベース・ホームで SRVCTL バイナリを使用して Oracle ACFS を管理することはできません。
- Standard Edition High Availability (SEHA) データベースは、Oracle Database 19c (19.7) リリース以降

でのみサポートされています。

- このコマンドはOracle Clusterwareでのみ使用可能です。

例

ディスク・グループRAC_DATAにd1volume1-295ディスク・デバイスをOracle ACFSボリュームVOLUME1として追加し、マウント・ポイントは/oracle/cluster1/acfsとします。

```
# srvctl add filesystem -device /dev/asm/d1volume1-295
-path /oracle/cluster1/acfs1
```

動的ボリューム・デバイスasm-test-55にOracle ACFSファイル・システムを追加し、指定したノードの1つに同時にこのファイル・システムをマウントにします。

```
# srvctl add filesystem -fstype ACFS -device asm-test-55 -path myacfs
-node node1,node2,node3
```

マウント・ポイントが/u01/app/oracleであるDATAディスク・グループのsehomeボリュームでStandard Edition High Availability (SEHA)を構成するためにOracle ACFSファイル・システムを追加するには、次のようにします。

```
# srvctl add filesystem -volume sehome -diskgroup DATA -path /u01/app/oracle -fstype
ACFS
-autostart ALWAYS -user oracle
```

srvctl config filesystem

特定のファイル・システム・リソースの構成を表示します。

構文

```
srvctl config filesystem [-device volume_device | -volume volume_name -diskgroup
disk_group_name]
```

パラメータ

表F-40 srvctl config filesystemコマンドのパラメータ

パラメータ	説明
-device volume_device	ファイル・システム・ボリューム・デバイスへのパスを指定します。
-volume volume_name	ボリュームの名前を指定します。
-diskgroup disk_group_name	-volume があるディスク・グループの名前を指定します。

使用上のノート

- 構成を表示するファイル・システム・リソースへのパスを-deviceパラメータで指定するか、-volume volume_name -diskgroup disk_group_nameを指定します。
- このコマンドはOracle Clusterwareでのみ使用可能です。

例

次の例は、すべてのファイル・システムの構成の一覧を表示します。

```
$ srvctl config filesystem
```

次の例は、特定のデバイスの構成を表示します。

```
$ srvctl config filesystem -device /dev/asm/d1volume1-295
```

srvctl disable filesystem

特定のOracle Clusterware管理ファイル・システム・ボリュームを無効化します。

構文

```
srvctl disable filesystem {-device volume_device | -volume volume_name -diskgroup disk_group_name}
```

パラメータ

表F-41 srvctl disable filesystemコマンドのパラメータ

パラメータ	説明
-device volume_device	ファイル・システム・ボリューム・デバイスへのパスを指定します。
-volume volume_name	ボリュームの名前を指定します。
-diskgroup disk_group_name	-volume があるディスク・グループの名前を指定します。

使用上のノート

- 無効にするファイル・システム・リソースへのパスを-deviceパラメータで指定するか、-volume volume_name -diskgroup disk_group_nameを指定します。
- このコマンドはOracle Clusterwareでのみ使用可能です。

例

次に、このコマンドの例を示します。

```
# srvctl disable filesystem -device /dev/asm/racvol1
```

srvctl enable filesystem

Oracle ClusterwareのOracle ACFSボリュームまたは汎用ファイル・システムを有効化します。

構文

```
srvctl enable filesystem {-device volume_device | -volume volume_name -diskgroup disk_group_name}
```

パラメータ

表F-42 srvctl enable filesystemコマンドのパラメータ

パラメータ	説明

パラメータ	説明
-device volume_device	ファイル・システム・ボリューム・デバイスへのパスを指定します。
-volume volume_name	ボリュームの名前を指定します。
-diskgroup disk_group_name	-volume があるディスク・グループの名前を指定します。

使用上のノート

- 有効にするファイル・システム・リソースへのパスを -device パラメータで指定するか、-volume volume_name -diskgroup disk_group_name を指定します。
- このコマンドは Oracle Clusterware でのみ使用可能です。

例

次に、このコマンドの例を示します。

```
# srvctl enable filesystem -device /dev/asm/racvol1
```

srvctl modify filesystem

ファイル・システム・リソースの構成を変更します。

構文

```
srvctl modify filesystem
  {-device volume_device | -volume volume_name -diskgroup disk_group_name}
  [-user {/+|/-}user_name | "user_list"]
  [-mountowner user_name] [-mountgroup group_name] [-mountperm octal_permission]
  [-path mountpoint_path] [-node "node_list" | -serverpool "server_pool_list"]
  [-fsoptions options] [-description description]
  [-autostart {ALWAYS | NEVER | RESTORE}] [-force]
```

パラメータ

表F-43 srvctl modify filesystem コマンドのパラメータ

パラメータ	説明
-device volume_device	ファイル・システム・ボリューム・デバイスへのパスを指定します。
-volume volume_name	ボリュームの名前を指定します。
-diskgroup disk_group_name	-volume があるディスク・グループの名前を指定します。
-user user_name "user_list"	ファイル・システムをマウントおよびアンマウントする権限を持つユーザーを指定します。単一ユーザーを追加(/+)または削除(/-)するか、一連のユーザー全体をカンマ区切りリストで置き換えることができます。このパラメータを指定しない場合、SRVCTL では、デフォルトでコマンドの実行ユーザーになります。

パラメータ	説明
	す。
-mountowner user_name	マウント・ポイントの所有者を指定した値に変更します。
-mountgroup group_name	マウント・ポイントのグループを指定した値に変更します。
-mountperm octal_permission	ユーザー、グループおよびその他の権限を指定された値に変更します。
-path mountpoint_path	マウントをネストするために他のリソースへの依存性を設定する場合に使用する、マウント・ポイント・パスを変更します。これは、絶対的パスである必要があります。
-node "node_list" -serverpool "server_pool_list"	ファイル・システム・デバイスのマウント先ノードのカンマ区切りリストを変更します。ノードまたはサーバー・プールのリストを指定した場合は、SRVCTL によって単一ノードのファイル・システム・リソースが作成されます。ノードまたはサーバー・プールを指定しない場合、SRVCTL はすべてのクラスタ・ノードで実行されるファイル・システム・リソースを作成します。 あるいは、ファイル・システムが実行されるサーバー・プールのカンマ区切りリストを変更することもできます。サーバー・プールのリストを指定する場合、ファイル・システムが実行されるノードの数またはタイプを動的に制限します。これは、HANFS などのクラスタ全体の Oracle ACFS リソースを必要とする可能性のある今後の使用に対しては、クラスタ全体のファイル・システムとして考慮されません。 ノート: ノードとサーバー・プールは相互に排他的です。
-fsoptions options	ファイル・システムのマウントに使用するオプションを変更します。オプション(カンマ区切りリスト)は、ファイル・システムおよびオペレーティング・システムによって異なります。
-description description	ファイル・システムの説明を変更します。
-autostart {ALWAYS NEVER RESTORE}	ファイル・システム・リソースの自動起動ポリシーを変更します。 • ALWAYS: 常にファイル・システム・リソースが自動的に開始されます。 • ALWAYS: ファイル・システム・リソースは自動的に開

パラメータ	説明
	始されません。 RESTORE: ファイル・システムは、最後の状態にリストアされます。このオプションがデフォルトです。
-force	変更の強制。このオプションは依存関係を無視します。

使用上のノート

- 変更するファイル・システム・リソースへのパスを-deviceパラメータで指定するか、-volume volume_name -diskgroup disk_group_nameを指定します。
- ファイル・システム・リソースに対してオンライン変更を実行した場合、次にリソースを起動または再起動した後でのみ変更は有効になります。
- LinuxおよびUNIXプラットフォームで、rootユーザーとしてこのコマンドを実行する必要があります。
- このコマンドはOracle Clusterwareでのみ使用可能です。

例

次の例は、RACVOL1ボリュームについて許可されたユーザーをsysadに変更します。

```
# srvctl modify filesystem -device /dev/asm/racvol1 -user /+sysad
```

srvctl predict filesystem

ファイル・システム障害の結果を予測します。

構文

```
srvctl predict filesystem {-device volume_device | -volume volume_name -diskgroup disk_group_name} [-verbose]
```

パラメータ

表F-44 srvctl predict filesystemコマンドのパラメータ

パラメータ	説明
-device volume_device	ファイル・システム・ボリューム・デバイスへのパスを指定します。
-volume volume_name	ボリュームの名前を指定します。
-diskgroup disk_group_name	-volume があるディスク・グループの名前を指定します。
-verbose	必要に応じて、-verbose パラメータを使用して、詳細な出力を表示できます。

使用上のノート

- ファイル・システム障害の結果を予測するファイル・システム・リソースへのパスを -device パラメータで指定するか、 -volume volume_name -diskgroup disk_group_name を指定します。
- 必要に応じて、冗長出力を選択することもできます。
- このコマンドは Oracle Clusterware でのみ使用可能です。

例

次に、このコマンドの例を示します。

```
$ srvctl predict filesystem -device /dev/asm/volume1-123
```

srvctl remove filesystem

クラスタから特定のファイル・システム・リソースを削除します。

構文

```
srvctl remove filesystem {-device volume_device | -volume volume_name -diskgroup disk_group_name} [-force]
```

パラメータ

表F-45 srvctl remove filesystem コマンドのパラメータ

パラメータ	説明
-device volume_device	ファイル・システム・ボリューム・デバイスへのパスを指定します。
-volume volume_name	ボリュームの名前を指定します。
-diskgroup disk_group_name	-volume があるディスク・グループの名前を指定します。
-force	オプションで、-force パラメータを使用してクラスタからリソースを強制的に削除することもできます。

使用上のノート

- 削除するファイル・システム・リソースへのパスを -device パラメータで指定するか、 -volume volume_name -diskgroup disk_group_name を指定します。
- 必要に応じて、-force パラメータを使用し、リソースの依存性を無視して、クラスタからリソースを強制的に削除することもできます。
- このコマンドは、Linux および UNIX プラットフォームでは root ユーザーとして実行し、Windows プラットフォームでは管理者ユーザーとして実行する必要があります。
- このコマンドは Oracle Clusterware でのみ使用可能です。

例

次に、このコマンドの例を示します。

```
# srvctl remove filesystem -device /dev/asm/racvol1
```

srvctl start filesystem

ファイル・システム・リソースを起動(マウント)します。

構文

```
srvctl start filesystem {-device volume_device | -volume volume_name -diskgroup disk_group_name} [-node node_name]
```

パラメータ

表F-46 srvctl start filesystemコマンドのパラメータ

パラメータ	説明
-device volume_device	起動するファイル・システム・リソース・デバイスのパスを指定します。
-volume volume_name	ボリュームの名前を指定します。
-diskgroup disk_group_name	-volume があるディスク・グループの名前を指定します。
-node node_name	必要に応じて、ファイル・システム・リソースが起動されるノードの名前を指定できます。 このパラメータを指定しなかった場合、SRVCTL はクラスタ内にあるすべての使用可能なノードでファイル・システム・リソースを起動します。

使用上のノート

- 起動するファイル・システム・リソースへのパスを -device パラメータで指定するか、-volume volume_name -diskgroup disk_group_name を指定します。
- このコマンドは、Linux および UNIX プラットフォームでは root ユーザーとして実行し、Windows プラットフォームでは管理者ユーザーとして、または構成済ファイル・システム・ユーザーとして実行する必要があります。

ファイル・システムのマウントおよびアンマウントを許可されたユーザーを識別するために、srvctl add filesystem を -user パラメータを指定して実行した場合は、構成されているファイル・システム・ユーザーの1人として srvctl start filesystem を実行できます。

関連項目:

ファイル・システム・ユーザーの構成の詳細は、[\[srvctl add filesystem\]](#) を参照してください。

- このコマンドは Oracle Clusterware でのみ使用可能です。

例

クラスタに構成されているすべてのノードでファイル・システムを起動します。

```
$ srvctl start filesystem -device /dev/asm/data_db1-68
```

node1でファイル・システムを起動します。

```
$ srvctl start filesystem -device /dev/asm/data_db1-68 -node node1
```

srvctl status filesystem

ファイル・システム・リソースのステータスを表示します。

構文

```
srvctl status filesystem [-device volume_device | -volume volume_name -diskgroup disk_group_name] [-verbose]
```

パラメータ

表F-47 srvctl status filesystemコマンドのパラメータ

パラメータ	説明
-device volume_device	ファイル・システム・ボリューム・デバイスへのパスを指定します。
-volume volume_name	ボリュームの名前を指定します。
-diskgroup disk_group_name	-volume があるディスク・グループの名前を指定します。
-verbose	必要に応じて、-verbose パラメータを使用して、詳細な出力を表示できます。

使用上のノート

- オプションで、ステータスを取得するファイル・システム・リソースへのパスを -device パラメータで指定するか、-volume volume_name -diskgroup disk_group_name を指定できます。-device または -volume -diskgroup を指定しない場合は、SRVCTL によってすべてのファイル・システムのステータスが表示されます。
- このコマンドは Oracle Clusterware でのみ使用可能です。

例

デバイス、またはボリュームとディスク・グループを指定できます。

たとえば、デバイス名を指定する場合は、次のようになります。

```
$ srvctl status filesystem -d /dev/asm/fs1vol-358
ACFS file system /acfs/fs1 is not mounted
$ srvctl start filesystem -d /dev/asm/fs1vol-358
...
$ srvctl status filesystem -d /dev/asm/fs1vol-358
ACFS file system /acfs/fs1 is mounted on nodes nd1-4n-01, nd1-4n-02, nd1-4n-03, nd1-4n-04
```

たとえば、ボリュームとディスク・グループを指定する場合は、次のようになります。

```
$ srvctl status filesystem -volume fs1vol -diskgroup DATA
ACFS file system /acfs/fs1 is mounted on nodes nd1-4n-01,nd1-4n-02,nd1-4n-03,nd1-4n-04
```

デバイスもボリュームとディスク・グループも指定しない場合は、次のようになります。

```
$ srvctl status filesystem
ACFS file system /acfs/fs1 is mounted on nodes nd1-4n-01,nd1-4n-02,nd1-4n-03,nd1-4n-04
```

ノート:



前述の例では、ファイル・システムは Oracle ACFS です。他のファイル・システムを使用している場合、ファイル・システムは EXT3 または EXT4 として表示されます。

srvctl stop filesystem

Oracle ACFSファイル・システムまたは汎用ファイル・システムを停止(アンマウント)します。

構文

```
srvctl stop filesystem {-device volume_device | -volume volume_name -diskgroup disk_group_name} [-node node_name] [-force]
```

パラメータ

表F-48 srvctl stop filesystemコマンドのパラメータ

パラメータ	説明
-device volume_device	停止するファイル・システム・ボリューム・デバイスへのパスを指定します。
-volume volume_name	ボリュームの名前を指定します。
-diskgroup disk_group_name	-volume があるディスク・グループの名前を指定します。
-node node_name	必要に応じて、ノードの名前を指定できます。このパラメータを指定しなかった場合、SRVCTL はクラスタ内にあるすべてのアクティブなノードでボリューム・リソースを停止します。
-force	必要に応じて、-force を使用してファイル・システムを停止し、このファイル・システムに依存するデータベースまたはその他のリソースも停止できます。

使用上のノート

- 停止するファイル・システム・リソースへのパスを-deviceパラメータで指定するか、-volume volume_name -diskgroup disk_group_nameを指定します。

- このコマンドは、LinuxおよびUNIXプラットフォームではrootユーザーとして実行し、Windowsプラットフォームでは管理者ユーザーとして、または構成済ファイル・システム・ユーザーとして実行する必要があります。

ファイル・システムのマウントおよびアンマウントを許可されたユーザーを識別するために、`srvctl add filesystem` を `-user` パラメータを指定して実行した場合は、構成されているファイル・システム・ユーザーの1人として `srvctl stop filesystem` を実行できます。

関連項目:

ファイル・システム・ユーザーの構成の詳細は、[\[srvctl add filesystem\]](#) を参照してください。

- このコマンドはOracle Clusterwareでのみ使用可能です。

例

次に、このコマンドの例を示します。

```
# srvctl stop filesystem -device /dev/asm/racvol_1 -force
```

srvctl add gns

DHCPパブリック・ネットワークを使用している場合や、クライアント・クラスタを作成する場合に、クラスタにグリッド・ネーミング・サービス(GNS)を追加します。

構文

`srvctl add gns` コマンドは、次の構文モデルの1つで使します。

現在GNSを実行していないクラスタにGNSを追加するには:

```
srvctl add gns [-domain domain_name] -vip {vip_name | ip_address} [-skip] [-verbose]
```

GNSを実行していないクラスタから、GNSを実行している別のクラスタのクライアント・クラスタに切り替えます。

```
srvctl add gns -clientdata file_name
```

セカンダリGNSをクラスタに追加するには:

```
srvctl add gns -vip vip_name | ip_address -clientdata file_name
```

パラメータ

表F-49 `srvctl add gns` コマンドのパラメータ

パラメータ	説明
<code>-domain domain_name</code>	クラスタ IP アドレスのフォワード DNS 参照に使用されるネットワーク・サブドメイン。このパラメータは、 <code>-vip</code> を指定する場合にのみ使用でき、GNS の追加先クラスタに GNS ゾーン委任があり、DHCP を使用している場合は必須です。 <code>-vip</code> と一緒に <code>-domain</code> パラメータを使用して GNS を追加する場合、追加先クラスタはサーバー・クラスタになります。
	<code>-domain</code> を指定しない場合、ドメインなしで GNS が追加さ

パラメータ	説明
	れます。
<code>-vip {vip_name ip_address}</code>	GNS が DNS リクエストをリスニングする仮想 IP (VIP) 名または IP アドレスのいずれかを指定します。<code>-vip</code> を <code>-clientdata</code> と一緒に使用することはできません。 GNS の追加先クラスタをサーバー・クラスタにするには、<code>-vip</code> と一緒に <code>-domain</code> パラメータを使用してください。
<code>-skip</code>	VIP アドレスの到達可能性の確認をスキップするには、このパラメータを指定します。
<code>-clientdata path_to_file</code>	<code>srvctl export gns</code> コマンドで作成した GNS 資格証明ファイルへのパスを指定します。<code>srvctl add gns</code> コマンドを実行する前に、追加先クラスタのノードにこのファイルをコピーしておく必要があります。 <code>-clientdata</code> を使用して GNS を追加する場合、追加先クラスタはクライアント・クラスタまたはセカンダリ GNS になります。 セカンダリ GNS を追加する場合にのみ、<code>-vip</code> パラメータとともに <code>-clientdata</code> パラメータを使用できます。
<code>-verbose</code>	冗長出力。

使用上のノート

- このコマンドは、LinuxおよびUNIXプラットフォームではrootユーザーとして実行し、Windowsプラットフォームでは管理者ユーザーとして実行する必要があります。
- GNSをクラスタに初めて追加する場合は、`-vip`パラメータと一緒に`-domain`パラメータを使用してください。
- `-clientdata`パラメータを使用してGNSをクラスタに追加する場合は、まずGNSデータをファイルにエクスポートし、そのファイルを追加先クラスタのノードに手動でコピーする必要があります。`-clientdata`パラメータを使用し、クラスタ内の任意のノードに対してこのコマンドを実行します。
- セカンダリGNSを追加する場合にのみ、`-vip`パラメータと`-clientdata`パラメータを同時に指定できます。

例

GNSをクラスタに追加し、そのクラスタをサーバー・クラスタにします。

```
# srvctl add gns -vip 192.168.16.17 -domain cluster.mycompany.com
```

GNSをクラスタに追加し、そのクラスタをクライアント・クラスタにします。

```
# srvctl add gns -clientdata /tmp/gnsdata
```

srvctl config gns

グリッド・ネーミング・サービス(GNS)の構成を表示します。

構文

```
srvctl config gns [-detail] [-subdomain] [-multicastport] [-node node_name] [-port]
[-status]
[-version] [-query name] [-list] [-clusterguid] [-clustername] [-clustertype] [-
loglevel]
[-network] [-role] [-instances]
```

パラメータ

表F-50 srvctl config gnsコマンドのパラメータ

パラメータ	説明
-detail	GNS に関する詳細な構成情報の印刷。
-subdomain	GNS から供給されるサブドメインの表示。
-multicastport	GNS デーモンがマルチキャスト・リクエストをリスニングしているポートの表示。
-node node_name	指定されたノードの GNS の構成情報の表示。
-port	GNS デーモンが DNS サーバーとの通信に使用するポートの表示。
-status	GNS のステータスの表示。
-version	GNS のバージョンの表示。
-query name	GNS への、特定の名前に属しているレコードの問合せ。
-list	GNS のすべてのレコードのリスト。
-clusterguid	GNS が実行されているクラスタのグローバル一意識別子の表示。
-clustername	GNS が実行されているクラスタの名前の表示。
-clustertype	このクラスタで使用される GNS の構成のタイプの表示。

パラメータ	説明
-loglevel	GNS のログ・レベルの印刷。
-network	GNS がリスニングしているネットワークの表示。
-role	GNS インスタンスのロールを表示します。
-instances	インスタンス・リストを表示します。

srvctl disable gns

特定のノードまたはクラスタ内で使用可能なすべてのノードのGNSを無効化します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl disable gns [-node node_name] [-verbose]
```

使用上のノート

- 必要に応じて、GNSを無効にする、クラスタ内のノードの名前を指定できます。このパラメータを指定しない場合、SRVCTLによってクラスタ全体のGNSが無効化されます。
- 必要に応じて、-verboseパラメータを使用して、詳細な出力を表示できます。

例

次のコマンド例は、crm7というノードのGNSを無効化します。

```
$ srvctl disable gns -node crm7
```

srvctl enable gns

すべてのノードまたは特定のノードのGNSを有効化します。



ノート:

このコマンドは Oracle Clusterware でのみ使用できます。

構文

```
srvctl enable gns [-node node_name] [-verbose]
```

使用上のノート

- 必要に応じて、GNSを有効にするノードの名前を指定できます。このパラメータを指定しなかった場合、SRVCTLはク

ラスタ内にあるすべてのノードでGNSを有効化します。

- 必要に応じて、`-verbose`パラメータを使用して、詳細な出力を表示できます。

例

次に、このコマンドの例を示します。

```
$ srvctl enable gns -node node1
```

srvctl export gns

グリッド・ネーミング・サービス(GNS)・インスタンス・データをファイルにエクスポートし、異なるサーバー・クラスタ間でGNSを移動する場合またはクライアント・クラスタを作成する場合にそのファイルを使用できるようにします。

SRVCTLによって、データはOCRから抽出されます。エクスポートされるデータは次のとおりです。

- ユーザーの許可に使用される資格証明(サーバーがリスニングしているVIPアドレスを含む)
- GNSに保持される名前およびDNSレコード

構文

```
srvctl export gns {-instance path_to_file | {-clientdata path_to_file -role {CLIENT|SECONDARY} [-version]}}
```

パラメータ

表F-51 srvctl export gnsコマンドのパラメータ

パラメータ	説明
<code>-instance path_to_file</code>	異なるサーバー・クラスタ間で GNS を移動する場合に、このパラメータを使用して、GNS インスタンス・データが書き込まれるファイルを指定します。
<code>-clientdata path_to_file -role {CLIENT SECONDARY}</code>	このパラメータは、クライアント・クラスタを追加して、そのクライアント・クラスタがサーバー・クラスタで名前管理操作を実行するときにアクセスする資格証明ファイルを作成する場合に、GNS 資格証明データが書き込まれるファイルを指定するために使用します。 クライアント・データがエクスポートされるロールを指定するには、<code>-role</code> パラメータを使用する必要もあります。
<code>-version</code>	必要に応じて、このパラメータを使用して、クライアント・データが生成されるバージョンを指定できます。

例

次のコマンドでは、GNS資格証明データを`gns.txt`というファイルにエクスポートします。

```
$ srvctl export gns -clientdata /temp/gnsdata/gns.txt
```

srvctl import gns

srvctl import gnsコマンドでは、srvctl export gnsコマンドの実行時に作成するファイルからデータをインポートします。このコマンドは、GNSを別のサーバー・クラスタに配置する場合に使用します。

GNSデータをインポートすると、SRVCTLによって資格証明が格納され、レコード・データがOCRに配置されます。インポート・プロセスの実行中に別のGNSインスタンスが実行されていたり、別のインスタンスのデータが見つかった場合は、エラーが発生します。

また、このコマンドを実行すると、実行したクラスタがサーバー・クラスタとなります。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

srvctl modify gns

グリッド・ネーミング・サービス(GNS)で使用されるIPアドレス、ドメインまたはその他の構成パラメータを変更します。

構文

srvctl modify gnsコマンドは、次の構文モデルの1つで使用します。

```
srvctl modify gns -loglevel log_level
```

または

```
srvctl modify gns [-resolve name] [-verify name]
[-parameter name:value[,name:value...]]
[-vip {vip_name | ip} [-skip]] [-clientdata file_name]
[-role {PRIMARY} [-force]] [-verbose]
```

パラメータ

表F-52 srvctl modify gnsコマンドのパラメータ

パラメータ	説明
-loglevel log_level	GNS 診断ロギング・レベルを指定します(許容値は 1 から 6 です)。
-resolve name	GNS を介して指定した名前を解決
-verify name	指定した名前が GNS によって通知されるかどうかを確認します。
-parameter name:value[,name:value...]	1 つ以上の構成パラメータの値を設定します。
-vip {vip_name ip} [-skip]	GNS がリスニングする VIP 名または IP アドレスを指定

パラメータ	説明
	ノート: この属性はオンライン・リソース属性変更を使用して変更できます。 必要に応じて、-skip パラメータを指定して、VIP アドレスの到達可能性の確認をスキップできます。
-clientdata file_name	指定したファイル内に含まれるクライアント・データで GNS クライアント・データを変更
-role {PRIMARY} [-force]	セカンダリ GNS インスタンスをプライマリに変換します。 セカンダリ GNS インスタンスをプライマリに強制的に変換するには、-force パラメータを使用します。

例

次に、このコマンドの例を示します。

```
$ srvctl modify gns -vip 192.0.2.15
```

srvctl relocate gns

現行のホスティング・ノードからクラスタ内の別のノードにGNSを再配置します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl relocate gns [-node node_name] [-verbose]
```

使用上のノート

- 必要に応じて、GNSの移動先ノードの名前を指定できます。
- 必要に応じて、-verboseパラメータを使用して、冗長出力を表示できます。
- このコマンドを実行するには、LinuxシステムおよびUNIXシステムではrootでログインし、Windowsでは管理者権限を持つユーザーとしてログインする必要があります。

例

次に、このコマンドの例を示します。

```
# srvctl relocate gns -node node1
```

srvctl remove gns

クラスタからGNSを削除します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl remove gns [-force] [-verbose]
```

使用上のノート

- 必要に応じて、-forceパラメータを使用し、エラーの発生に関係なく、クラスタからGNSを強制的に削除することもできます。
- 必要に応じて、-verboseパラメータを使用して、冗長出力を表示できます。

例

次に、このコマンドの例を示します。

```
$ srvctl remove gns
```

srvctl start gns

特定のノードでGNSを起動します。

構文

```
srvctl start gns [-loglevel log_level] [-node node_name] [-verbose]
```

パラメータ

表F-53 srvctl start gnsコマンドのパラメータ

パラメータ	説明
-loglevel log_level	必要に応じて、GNS が実行するロギング・レベルを指定できます。ログのレベルは 1(最小のトレース)から 6(すべてをトレースし、処理時間が長い)の間で変わります。
-node node_name	必要に応じて、GNS を起動する、クラスタ内のノードの名前を指定できます。
-verbose	必要に応じて、このパラメータを使用して冗長出力を表示できます。

使用上のノート

このコマンドは、[サーバー・クラスタ](#)でのみ実行できます。このコマンドを[クライアント・クラスタ](#)で実行しようとすると、エラーが発生し

ます。

例

次のコマンド例は、crmnnode1というクラスター・ノードのGNSを起動します。

```
$ srvctl start gns -node crmnnode1
```

srvctl status gns

GNSの現在の状態を表示します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl status gns [-node node_name] [-verbose]
```

使用上のノート

- 必要に応じて、状態を表示するGNSの実行元ノードを指定できます。
- 必要に応じて、-verboseパラメータを使用して、詳細な出力を表示できます。

srvctl stop gns

クラスターのGNSを停止します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl stop gns [-node node_name] [-force] [-verbose]
```

使用上のノート

- 必要に応じて、停止するGNSの実行元ノードの名前を指定できます。
- 必要に応じて、-forceと-verboseを使用し、GNSを強制的に停止したり、詳細な出力を表示することができます。

例

次のコマンド例は、ローカル・ノードのGNSを停止します。

```
$ srvctl stop gns
```

srvctl update gns

グリッド・ネーミング・サービス(GNS)インスタンスを変更します。

ノート:



このコマンドは Oracle Clusterware でのみ使用可能です。

構文

srvctl update gnsコマンドは、次の構文モデルの1つで使します。

```
srvctl update gns -advertise name -address ip_address [-timetolive time_to_live]
[-verbose]
srvctl update gns -delete name [-address address] [-verbose]
srvctl update gns -alias alias -name name [-timetolive time_to_live] [-verbose]
srvctl update gns -deletealias alias [-verbose]
srvctl update gns -createsrv service -target target -protocol protocol
[-weight weight] [-priority priority] [-port port_number]
[-timetolive time_to_live] [-instance instance_name] [-verbose]
srvctl update gns -deletesrv service_name -target target -protocol protocol
[-verbose]
srvctl update gns -createtxt name -target target [-timetolive time_to_live]
[-namettl name_ttl] [-verbose]
srvctl update gns -deletetxt name -target target [-verbose]
srvctl update gns -createptr name -target target [-timetolive time_to_live]
[-namettl name_ttl] [-verbose]
srvctl update gns -deleteptr name -target target [-verbose]
```

パラメータ

表F-54 srvctl update gnsコマンドのパラメータ

パラメータ	説明
-advertise name	名前を GNS 全体に通知するには、このパラメータを使用します。
-address ip_address	通知対象の名前に対する IP アドレスを指定します。
-timetolive time_to_live	必要に応じて、レコードの存続時間(秒数)を指定できます。
-delete name	GNS からの名前通知を解除するには、このパラメータを使用します。
-alias alias	通知対象の名前の別名を作成するには、このパラメータを使用します。
-name name	別名に関連付ける名前を指定する必要があります。
-deletealias alias	別名を削除するには、このパラメータを使用します。
-createsrv service	レコードによって表されるサービスを作成するには、このパラ

パラメータ	説明
	メータを使用します。
-target target	サービスに名前を関連付けるには、このパラメータを使用します。
-protocol protocol	サービスへの接続に使用するプロトコルを指定するには、このパラメータを使用します。
-weight weight	必要に応じて、レコードの重みを指定できます。
-priority priority	必要に応じて、0 から 65535 までの範囲の値を使用して、レコードの優先順位を指定できます。
-port port_number	必要に応じて、0 から 65535 までの範囲の値を使用して、サービスへの接続に使用するポート番号を指定できます。
-instance instance_name	必要に応じて、サービスのインスタンス名を指定できます。
-deletesrv service_name	削除するサービス・レコードの名前を指定するには、このパラメータを使用します。
-createtxt name	指定した名前を使用してテキスト(TXT)・レコードを追加するには、このパラメータを使用します。
-namettl name_ttl	必要に応じて、名前の存続時間(秒数)を指定できます。
-deletetxt name	特定の名称のテキスト(TXT)・レコードを削除するには、このパラメータを使用します。
-createptr name	特定の名称のポインタ(PTR)・レコードを追加するには、このパラメータを使用します。
-deleteptr name	特定の名称のポインタ(PTR)・レコードを削除するには、このパラメータを使用します。

使用上のノート

このコマンドを実行するには、LinuxおよびUNIXプラットフォームではrootユーザーとして、Windowsプラットフォームでは管理者ユーザーとしてログインする必要があります。

例

次のコマンドは、GNSを使用して名前を通知します。

```
# srvctl update gns -advertise myname -address 192.168.1.45
```

srvctl add havip

高可用性VIP (HAVIP) (高可用性NFSおよびSMBエクスポートに使用)をクラスタに追加します。

構文

```
srvctl add havip -id havip_name -address {host_name | ip_address}  
[-netnum network_number] [-description text] [-skip] [-homenode node_name]
```

パラメータ

表F-55 srvctl add havipコマンドのパラメータ

パラメータ	説明
-id havip_name	HAVIP リソースの一意的 ID を指定します。
-address {host_name ip_address}	作成する HAVIP 用のホスト名または IPv4 IP アドレスを指定します。 ノート: 非 DHCP で非ラウンド・ロビン DNS アドレスを使用して、IPv4 IP アドレスを構成する必要があります。
-netnum network_number	必要に応じて、HAVIP が依存するネットワーク・リソースを指定できます。このパラメータのデフォルト値は 1 です。
-description text	必要に応じて、HAVIP のテキスト説明を指定できます。
-skip	必要に応じて、このパラメータを指定して、IP アドレスの到達可能性の確認をスキップできます。
-homenode node_name	必要に応じて、優先ノードまたは空の文字列を指定して、ホーム・ノードをクリアできます。

例

次に、このコマンドの例を示します。

```
# srvctl add havip -id myhavip -address 192.0.2.1 -netnum 2 -description  
"My HA VIP" -skip -homenode myNode2
```

srvctl config havip

特定の高可用性VIP (HAVIP) (高可用性NFSおよびSMBエクスポートに使用)またはクラスタ内のすべてのHAVIPリソースの構成情報を表示します。

構文

srvctl config havipコマンドは、次の構文で使います。

```
srvctl config havip [-id havip_name]
```

パラメータ

表F-56 srvctl config havipコマンドのパラメータ

パラメータ	説明
-id havip_name	表示する HAVIP リソースの一意の名前を指定します。このパラメータを指定しない場合、SRVCTL によって、Oracle Clusterware に知られているすべての HAVIP の構成情報が表示されます。

例

このコマンドによって、次のような出力が返されます。

```
$ srvctl config havip -id hrexports
VIP: /mjk-vm3-vip/10.149.237.196/10.149.236.0/255.255.252.0/eth0
Description: HR Exports
```

srvctl disable havip

指定したいいくつかのノードでの特定の高可用性VIP (HAVIP) (高可用性NFSおよびSMBエクスポートに使用)リソースの実行を禁止します。

構文

```
srvctl disable havip -id havip_name [-node node1,node2,...,noden]
```

パラメータ

表F-57 srvctl disable havipコマンドのパラメータ

パラメータ	説明
-id havip_name	無効化する HAVIP リソースの一意の名前を指定します。
-node node1,node2,...,noden	HAVIP リソースを実行しないノードの名前またはノード名のカンマ区切りリストを指定します。 クラスタ内の 1 つを除くすべてのノードを指定した場合、HAVIP はフェイルオーバーしません。

使用上のノート

LinuxおよびUNIXプラットフォームで、rootユーザーとしてこのコマンドを実行する必要があります。

例

次に、このコマンドの例を示します。

```
# srvctl disable havip -id myhavip -node myNode1,myNode3
```

srvctl enable havip

指定したいいくつかのノードでの特定の高可用性VIP (HAVIP) (高可用性NFSおよびSMBエクスポートに使用)の実行を許可します。

構文

```
srvctl enable havip -id havip_name [-node node_name]
```

パラメータ

表F-58 srvctl enable havipコマンドのパラメータ

パラメータ	説明
-id havip_name	有効化する HAVIP リソースの一意の名前を指定します。
-node node_name	HAVIP リソースを実行できるようにするノードの名前を指定します。

使用上のノート

LinuxおよびUNIXプラットフォームで、rootユーザーとしてこのコマンドを実行する必要があります。

例

次に、このコマンドの例を示します。

```
# srvctl enable havip -id myhavip -node myNode1
```

srvctl modify havip

高可用性VIP (HAVIP) (高可用性NFSおよびSMBエクスポートに使用)を変更します。

構文

```
srvctl modify havip -id havip_name [-address {host_name | ip_address}  
[-netnum network_number] [-skip]] [-description text] [-homenode node_name]
```

パラメータ

表F-59 srvctl modify havipコマンドのパラメータ

パラメータ	説明
-id havip_name	変更する HAVIP リソースの一意の名前を指定します。
-address {host_name ip_address [-netnum network_number] [-skip]]	変更する HAVIP 用のホスト名または IPv4 IP または IPv6 アドレスを指定します。
	ノート: 非 DHCP で非ラウンド・ロビン DNS アドレスを使用して、IPv4 IP アドレスを構成する必要があります。

パラメータ	説明
-netnum network_number	必要に応じて、HAVIP が依存するネットワーク・リソースを変更できます。このオプション・パラメータのデフォルト値は 1 です。
-skip	IP アドレスの到達可能性の確認をスキップするには、このパラメータを指定します。
-description text	HAVIP のテキスト説明を指定します。
-homenode node_name	必要に応じて、優先ノードまたは空の文字列を指定して、ホーム・ノードをクリアできます。

使用上のノート

LinuxおよびUNIXプラットフォームで、rootユーザーとしてこのコマンドを実行する必要があります。

例

次に、このコマンドの例を示します。

```
# srvctl modify havip -id myhavip -address 192.168.16.17 -netnum 2
```

srvctl relocate havip

高可用性VIP (HAVIP) (高可用性NFSおよびSMBエクスポートに使用)をクラスタ内の別のノードに再配置します。

構文

```
srvctl relocate havip -id havip_name [-node node_name] [-force]
```

パラメータ

表F-60 srvctl relocate havipコマンドのパラメータ

パラメータ	説明
-id havip_name	再配置する HAVIP リソースの一意の名前を指定します。
-node node_name	HAVIP リソースの再配置先ノードの名前を指定します。
-force	必要に応じて、HAVIP リソースの再配置を強制できます。

例

次に、このコマンドの例を示します。

```
$ srvctl relocate havip -id myhavip -node node3
```

srvctl remove havip

高可用性VIP (HAVIP) (高可用性NFSおよびSMBエクスポートに使用)を削除します。

構文

```
srvctl remove havip -id havip_name [-force]
```

パラメータ

表F-61 srvctl remove havipコマンドのパラメータ

パラメータ	説明
-id havip_name	削除する HAVIP リソースの一意の名前を指定します。
-force	このパラメータを使用して、クラスタからリソースを強制的に削除して、すべての依存性を無視します。

使用上のノート

- HAVIPリソースを事前に停止してから削除を実行してください。
- LinuxおよびUNIXプラットフォームで、rootユーザーとしてこのコマンドを実行する必要があります。

例

次に、このコマンドの例を示します。

```
# srvctl remove havip -id myhavip -force
```

srvctl start havip

特定のノードの特定の高可用性VIP (HAVIP) (高可用性NFSおよびSMBエクスポートに使用)を起動します。

構文

```
srvctl start havip -id havip_name [-node node_name]
```

パラメータ

表F-62 srvctl start havipコマンドのパラメータ

パラメータ	説明
-id havip_name	起動する HAVIP リソースの一意の名前を指定します。
-node node_name	必要に応じて、-node パラメータを使用して、HAVIP リソースを起動するノードの名前を指定できます。

使用上のノート

LinuxおよびUNIXプラットフォームで、rootユーザーとしてこのコマンドを実行する必要があります。

例

次に、このコマンドの例を示します。

```
# srvctl start havip -id myhavip -node myNode1
```

srvctl status havip

クラスタ内のすべての高可用性VIP (HAVIP) (高可用性NFSおよびSMBエクスポートに使用)または特定の1つのHAVIPのステータスを表示します。

構文

```
srvctl status havip [-id havip_name]
```

使用上のノート

表示するHAVIPリソースの一意の名前を指定します。このパラメータを指定しない場合、SRVCTLによって、Oracle Clusterwareに知られているすべてのHAVIPのステータスが表示されます。

例

このコマンドによって、次のような出力が返されます。

```
$ srvctl status havip
HAVIP ora.ha1.havip is enabled
HAVIP ora.ha1.havip is not running
```

srvctl stop havip

特定のノードの高可用性VIP (HAVIP) (高可用性NFSおよびSMBエクスポートに使用)を停止します。

構文

```
srvctl stop havip -id havip_name [-node node_name] [-force]
```

パラメータ

表F-63 srvctl stop havipコマンドのパラメータ

パラメータ	説明
-id havip_name	停止する HAVIP リソースの一意の名前を指定します。
-node node_name	必要に応じて、停止する HAVIP リソースが存在するノードの名前を指定できます。
-force	このパラメータを使用して、HAVIP を強制的に停止します。

使用上のノート

LinuxおよびUNIXプラットフォームで、rootユーザーとしてこのコマンドを実行する必要があります。

例

次に、このコマンドの例を示します。

```
# srvctl stop havip -id myhavip -node myNode1 -force
```

srvctl add ioserver

クラスタにOracle ASM IO Server構成を追加します。

標準クラスタをOracle Flex Clusterに構成、アップグレードまたは変換する場合、クラスタのハブ・ノードでOracle ASM IO Serverを作成できます。

構文

```
srvctl add ioserver [-spfile spfile] [-count number_of_ioserver_instances| ALL] [-listener listener_name]
```

パラメータ

表F-64 srvctl add ioserverコマンドのパラメータ

パラメータ	説明
-spfile spfile	サーバー・パラメータ・ファイルのパスを指定します。
-count number_of_ioserver_instances ALL	追加する Oracle ASM IO Server インスタンスの数を指定するか、ALL を指定してすべてのノードにインスタンスを追加します。
-listener listener_name	リスナーの名前を指定します。

例

次に、このコマンドの例を示します。

```
# srvctl add ioserver -count 2
```

srvctl config ioserver

Oracle ASM IO Serverの構成の詳細を表示します。

構文

```
srvctl config ioserver
```

使用上のノート

このコマンドにパラメータはありません。

例

このコマンドでは、次のような出力が表示されます。

```
$ srvctl config ioserver
ASM I/O Server instance count: 3
ASM I/O Server is enabled.
ASM I/O server is enabled on nodes:
ASM I/O server is disabled on nodes:
```

srvctl disable ioserver

Oracle ASM IO Serverを無効化します。

構文

```
srvctl disable ioserver -node node_name
```

使用上のノート

Oracle ASM IOServerを無効化するハブ・ノードの名前を指定するには、-nodeパラメータを使用します。

srvctl enable ioserver

特定のノードのOracle ASM IOServerを有効化します。

構文

```
srvctl enable ioserver [-node node_name]
```

使用上のノート

Oracle ASM IOServerを有効化するハブ・ノードの名前を指定するには、-nodeパラメータを使用します。

srvctl getenv ioserver

Oracle ASM IOServerに関連付けられている環境変数の値の表示

構文

```
srvctl getenv ioserver [-envs "name_list"]
```

使用上のノート

環境変数の名前のカンマ区切りリストを指定します。このパラメータを指定しない場合、SRVCTLによって、データベースに関連付けられているすべての環境変数の値が表示されます。

srvctl modify ioserver

クラスタのOracle ASM IOServer構成を変更します。

構文

```
srvctl modify ioserver [-spfile spfile] [-count number_of_ioserver_instances | ALL]
[-listener listener_name] [-force]
```

パラメータ

表F-65 srvctl modify ioserverコマンドのパラメータ

パラメータ	説明
-spfile spfile	必要に応じて、サーバー・パラメータ・ファイルのパスを指定します
-count number_of_ioserver_instances ALL	必要に応じて、変更するインスタンスの数を指定するか、ALL (すべてのノードのインスタンスを変更する)を指定します。数を変更すると、クラスタ・リソースのカーディナリティが変更されるため、すぐに変更が有効になります。

パラメータ	説明
-listener listener_name	必要に応じて、リスナーの名前を指定します。
-force	必要に応じて、依存関係を無視し、Oracle ASM IOServer を強制的に変更できます。

srvctl relocate ioserver

現行のホスティング・ノードからクラスタ内の別のノードにOracle ASM IOServerインスタンスを再配置します。

構文

```
srvctl relocate ioserver -currentnode node_name [-targetnode node_name] [-force]
```

パラメータ

表F-66 srvctl relocate ioserverコマンドのパラメータ

パラメータ	説明
-currentnode node_name	Oracle ASM IOServer インスタンスが現在存在するノードの名前
-targetnode node_name	IOServer インスタンスの再配置先ノードの名前
-force	必要に応じて、Oracle ASM IOServer インスタンスの再配置を強制できます。

例

次のコマンドでは、Oracle ASM IOServerがcrsNode2というノードからcrsNode5というノードに再配置されます。

```
$ srvctl relocate ioserver -currentnode crsNode2 -targetnode crsNode5
```

srvctl remove ioserver

このコマンドを使用して、クラスタからOracle ASM IOServerインスタンスを削除します。

構文

```
srvctl remove ioserver [-force]
```

使用上のノート

すべての依存性を無視し、Oracle ASM IOServerインスタンスを強制的に削除するには、-forceパラメータを使用します。

srvctl setenv ioserver

Oracle ASM IOServerの環境構成を管理します。

構文

```
srvctl setenv ioserver {-envs "name=val[,name=val][...]" | -env "name=val"}
```

表F-67 srvctl setenv ioserverコマンドのパラメータ

パラメータ	説明
-envs "name=val,..."	二重引用符("")で囲まれた環境変数の名前と値のペアのカンマ区切りリスト
-env "name=val"	または、このパラメータを使用して、二重引用符("")で囲まれた単一環境変数をカンマや他の特殊文字を含む値に設定します。

srvctl start ioserver

クラスタのOracle ASM IOServerを起動します。

ノード名を指定して、特定のハブ・ノードのOracle ASM IOServerを起動することもできます。

構文

```
srvctl start ioserver [-node node_name]
```

使用上のノート

特定のハブ・ノードのOracle ASM IOServerを起動する場合は、-nodeパラメータを使用してノードの名前を指定します。特定のハブ・ノードを指定しない場合、SRVCTLによって、クラスタ内のすべてのOracle ASM IOServerが起動されます。

例

次のコマンドでは、mjkhub46というハブ・ノードのOracle ASM IOServerが起動します。

```
$ srvctl start ioserver -node mjkhub46
```

srvctl status ioserver

クラスタ内で実行されているOracle ASM IOServerのステータスを表示します。

クラスタを実行しているすべてのOracle ASM IOServerのステータス、特定のハブ・ノードで実行されているOracle ASM IOServerのステータスまたは特定のデータベースからのクライアントに関する情報を表示できます。

構文

```
srvctl status ioserver [-node node_name] [-detail]
```

パラメータ

表F-68 srvctl status ioserverのパラメータ

パラメータ	説明
-node node_name	特定のハブ・ノードを指定して、そこで実行されている Oracle ASM IOServer リソース・インスタンスのステータスを表示します。ハブ・ノードを指定しない場合は、クラスタ内で実行されているすべての Oracle ASM IOServer リソース・インスタンス

パラメータ	説明
	のステータスが戻されます。
-detail	このパラメータを使用して、Oracle ASM IO Server のステータスに関する詳細情報を表示します。

例

-detail パラメータを指定してコマンドを実行すると、SRVCTLによって次のような出力が表示されます。

```
$ srvctl status ioserver -detail
ASM I/O Server is enabled
ASM I/O Server is running on nodes: mjk376, mjk377
ASM I/O Server instance +IOS1 running on node mjk376 is connected to ASM instance
+ASM2:
Number of clients connected: 2
Names of clients: later14, alus16, later12, alus9, db0010
ASM I/O Server instance +IOS2 running on node mjk377 is connected to ASM instance
+ASM3:
Number of clients connected: 3
Number of requests completed per sec: 110
Names of networks: eth1, eth2
Names of clients: later11, later3, later14, later15, later16, alus19, alus17, alus13
```

srvctl stop ioserver

クラスタのOracle ASM IO Serverを停止します。

構文

```
srvctl stop ioserver [-node node_name] [-force]
```

使用上のノート

- 特定のハブ・ノードのOracle ASM IO Serverを停止するには、-nodeパラメータを使用します。特定のハブ・ノードを指定しない場合、SRVCTLによって、クラスタ内のすべてのOracle ASM IO Serverが停止されます。
- 依存性に関係なく、Oracle ASM IO Serverを停止するには、-forceパラメータを使用します。

srvctl unsetenv ioserver

Oracle ASM IO Server環境構成の設定を解除します。

構文

```
srvctl unsetenv ioserver -envs "name_list"
```

使用上のノート

-envsパラメータを使用して、二重引用符(" ")で囲まれた環境変数名のカンマ区切りリストを指定します。

srvctl add mgmtadb

管理データベース(CHMリポジトリ)・リソースをクラスタに追加します。

構文


```
srvctl add mgmtdb [-domain domain]
```

使用上のノート

必要に応じて、データベースのドメインを指定できます。DB_DOMAINデータベース初期化パラメータを設定している場合は、このパラメータの値を指定する必要があります。

例

管理データベースをクラスタに追加するには:

```
$ srvctl add mgmtdb -domain example.com
```

srvctl config mgmtdb

管理データベース(CHMリポジトリ)・リソースの構成情報を表示します。



ノート:

このコマンドは Oracle Clusterware でのみ使用できます。

構文

```
srvctl config mgmtdb [-verbose] [-all]
```

使用上のノート

- 必要に応じて、-verboseパラメータを使用して、詳細な出力を表示できます。
- 必要に応じて、-allパラメータを使用して、詳細な構成情報を出力できます。

srvctl disable mgmtdb

指定したノードの管理データベース(CHMリポジトリ)・リソースを無効化します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl disable mgmtdb [-node node_name]
```

使用上のノート

必要に応じて、無効にする管理データベースが実行されているクラスタ・ノードの名前を指定できます。

例

次に、このコマンドの例を示します。

```
$ srvctl disable mgmtdb
```

srvctl enable mgmtdb

管理データベース(CHMリポジトリ)・リソースを有効化します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl enable mgmtdb [-node node_name]
```

使用上のノート

必要に応じて、管理データベース・リソースを有効にするクラスタ・ノードの名前を指定できます。

例

node5というノードの管理データベースを有効化するには、次のコマンドを使用します。

```
$ srvctl enable mgmtdb -node node5
```

srvctl getenv mgmtdb

管理データベース(CHMリポジトリ)の環境変数を取得します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl getenv mgmtdb [-envs "name_list"]
```

使用上のノート

環境変数の名前のカンマ区切りリストを、二重引用符(" ")で囲んで指定します。このパラメータを使用しなかった場合、SRVCTLはリスナーに関連付けられているすべての環境変数の値を表示します。

例

次の例は、管理データベースに指定されたすべての環境変数を一覧表示します。

```
$ srvctl getenv mgmtdb
```

srvctl modify mgmtdb

管理データベース(CHMリポジトリ)の構成を変更します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl modify mgmtdb [-pwfile password_file_path] [-spfile spfile_path]  
[-startoption start_option] [-stopoption stop_option] [-diskgroup "diskgroup_list"]
```

パラメータ

表F-69 srvctl modify mgmtdbコマンドのパラメータ

パラメータ	説明
-pwfile password_file_path	管理データベースのパスワード・ファイルの場所へのフルパスを変更します。
-spfile spfile_path	管理データベースで使用される新しい spfile のパス名を変更します。
-startoption start_option	管理データベースの起動オプション(OPEN、MOUNT、NOMOUNT など)を変更します。 ノート: 起動オプションに複数の語を指定する場合(read only、read write など)、語をスペースで区切り、二重引用符("")で囲みます。たとえば"read only"とします。 関連項目: 起動オプションの詳細は、 『SQL*Plus ユーザーズ・ガイドおよびリファレンス』 を参照してください
-stopoption stop_option	管理データベースの停止オプション(NORMAL、TRANSACTIONAL、IMMEDIATE、ABORT など)を変更します。 関連項目: 停止オプションの詳細は、 『SQL*Plus ユーザーズ・ガイドおよびリファレンス』 を参照してください。
-diskgroup "diskgroup_list"	管理データベースによって使用される Oracle ASM ディスク・グループのカンマ区切りリスト(二重引用符("")で囲んだリスト)を変更します。

例

次の例は、SYSFILES Oracle ASMディスク・グループを使用するように管理データベースに指示します。

```
$ srvctl modify mgmtdb -diskgroup "SYSFILES"
```

srvctl relocate mgmtdb

クラスタのノード間で管理データベース(CHMリポジトリ)・リソースを再配置します。

ノート:



このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl relocate mgmtdb [-node node_name]
```

使用上のノート

管理データベースの再配置先ノードを指定します。

例

次に、このコマンドの例を示します。

```
$ srvctl relocate mgmtdb -node crsNode2
```

srvctl remove mgmtdb

Oracle Clusterware管理から管理データベース(CHMリポジトリ)を削除します。

ノート:



このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl remove mgmtdb [-force] [-noprompt] [-verbose]
```

パラメータ

表F-70 srvctl remove mgmtdbコマンドのパラメータ

パラメータ	説明
-force	必要に応じて、このパラメータを使用し、エラーの発生に関係なく、クラスタから管理データベースを強制的に削除することもできます。
-noprompt	必要に応じて、このパラメータを使用してプロンプトを抑止できます。
-verbose	必要に応じて、このパラメータを使用して冗長出力を表示できます。

使用上のノート

削除を試行する前に、まず管理データベースをシャットダウンする必要があります。

例

次に、このコマンドの例を示します。

```
$ srvctl remove mgmtldb -noprompt
```

srvctl setenv mgmtldb

管理データベース(CHMリポジトリ)の環境構成を管理します。

 ノート:
このコマンドは Oracle Clusterware でのみ使用可能です。

構文

このコマンドは、次のいずれかの構文モデルで使⽤します。

```
srvctl setenv mgmtldb -envs "name=val[,...]"  
srvctl setenv mgmtldb -env "name=val"
```

パラメータ

表F-71 srvctl setenv mgmtldbコマンドのパラメータ

パラメータ	説明
-envs "name=val[,...]"	環境変数の名前/値ペアのカンマ区切りリストを、二重引用符("")で囲んで指定します。
-env "name=val"	このパラメータを使用して、単一の環境変数を二重引用符("")で囲み、カンマまたは他の特殊文字を含んだ値に設定できるようにします。

例

次の例は、管理データベースの言語環境構成を設定します。

```
$ srvctl setenv mgmtldb -env LANG=en
```

srvctl start mgmtldb

管理データベース(CHMリポジトリ)・リソースを起動します。

 ノート:
このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl start mgmtldb [-startoption start_options] [-node node_name]
```

パラメータ

表F-72 srvctl start mgmtldbコマンドのパラメータ

パラメータ	説明
-startoption start_options	必要に応じて、起動コマンドにオプションを指定できます (OPEN、MOUNT、NOMOUNT など)。 ノート: 起動オプションに複数の語を指定する場合(read only、read write など)、語をスペースで区切り、二重引用符(" ")で囲みます。たとえば"read only"とします。 関連項目: 起動オプションの詳細は、『SQL*Plus ユーザーズ・ガイドおよびリファレンス』を参照してください
-node node_name	必要に応じて、管理データベース・リソースを起動する特定のノードを指定できます。 ノート: このパラメータは、Oracle Clusterware でのみ使用できます。

例

次のコマンド例は、crmnnode1というクラスター・ノードの管理データベースを起動します。

```
$ srvctl start mgmtldb -node crmnnode1
```

srvctl status mgmtldb

管理データベース(CHMリポジトリ)・リソースの現在の状態を表示します。

 ノート:
このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl status mgmtldb [-verbose]
```

使用上のノート

必要に応じて、-verboseパラメータを使用して、詳細な出力を表示できます。

例

次のコマンド例は、管理データベースのステータスを確認します。

```
$ srvctl status mgmtldb
```

srvctl stop mgmtdb

管理データベース(CHMリポジトリ)・リソースを停止します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl stop mgmtdb [-stopoption stop_options] [-force]
```

パラメータ

表F-73 srvctl stop mgmtdbコマンドのパラメータ

パラメータ	説明
-stopoption stop_options	必要に応じて、このパラメータを使用して、shutdown コマンドのオプション(NORMAL、TRANSACTIONAL、IMMEDIATE、ABORT など)を指定できます。 関連項目: 停止オプションの詳細は、 『SQL*Plus ユーザーズ・ガイドおよびリファレンス』 を参照してください。
-force	データベース、関連付けられたサービス、および依存リソースを強制的に停止するには、このパラメータを使用します。

例

次に、このコマンドの例を示します。

```
$ srvctl stop mgmtdb -stopoption NORMAL
```

srvctl unsetenv mgmtdb

管理データベース(CHMリポジトリ)の環境構成の設定を解除します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl unsetenv mgmtdb -envs "name_list"
```

使用上のノート

-envsパラメータを使用して、二重引用符(" ")で囲まれた環境変数名のカンマ区切りリストを指定します。

例

次の例は、管理データベース環境変数の環境構成の設定を解除します。

```
$ srvctl unsetenv mgmtldb -envs "LANG"
```

srvctl add mgmtlsnr

管理リスナー・リソース(Cluster Health Monitor用)をクラスタに追加します。

構文

```
srvctl add mgmtlsnr [-endpoints "[TCP:]port_list[/IPC:key][/NMP:pipe_name  
[/TCPS:s_port][/SDP:port][EXADIRECT:port]" ] [-skip]]
```

パラメータ

表F-74 srvctl add mgmtlsnrコマンドのパラメータ

パラメータ	説明
-endpoints "[TCP:]port_list [/IPC:key][/NMP:pipe_name [/TCPS:s_port][/SDP:port][EXADIRECT:port]"]	リスナーのプロトコル仕様。port_list は TCP ポートま たはリスナー・エンドポイントのカンマ区切りリストです
-skip	ポートの確認をスキップすることを示します。

例

次のコマンドでは、ポート1341をリスニングする管理リスナーをクラスタに追加します。

```
$ srvctl add mgmtlsnr -endpoints "TCP:1341"
```

srvctl config mgmtlsnr

管理リスナー・リソース(CHM用)の構成情報を表示します。



ノート:

このコマンドは Oracle Clusterware でのみ使用できます。

構文

```
srvctl config mgmtlsnr [-all]
```

使用上のノート

詳細な構成情報を印刷するには、-allパラメータを使用します。

srvctl disable mgmtlsnr

指定したノードの管理リスナー・リソース(CHM用)を無効化します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl disable mgmtlsnr [-node node_name]
```

使用上のノート

必要に応じて、無効にする管理リスナーが実行されているクラスタ・ノードの名前を指定できます。

例

次に、このコマンドの例を示します。

```
$ srvctl disable mgmtlsnr
```

srvctl enable mgmtlsnr

管理データベース・リスナー・リソース(CHM用)を有効化します。

ノート:



このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl enable mgmtlsnr [-node node_name]
```

使用上のノート

必要に応じて、管理リスナーを有効にするクラスタ・ノードの名前を指定できます。

例

node5というノードの管理リスナーを有効化するには、次のコマンドを使用します。

```
$ srvctl enable mgmtlsnr -node node5
```

srvctl getenv mgmtlsnr

管理リスナー・リソース(CHM用)の環境変数を取得します。

ノート:



このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl getenv mgmtlsnr [-envs "name_list"]
```

使用上のノート

必要に応じて、環境変数名のカンマ区切りリストを指定できます。このパラメータを使用しなかった場合、SRVCTLはリスナーに関連付けられているすべての環境変数の値を表示します。

例

次の例は、管理リスナーに指定されたすべての環境変数を一覧表示します。

```
$ srvctl getenv mgmtlsnr
```

srvctl modify mgmtlsnr

管理リスナー・リソース(CHM用)の構成を変更します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl modify mgmtlsnr [-endpoints "[TCP:]port_list[/IPC:key][/NMP:pipe_name]  
[/TCPS:s_port][/SDP:port]"]
```

使用上のノート

必要に応じて、管理リスナーのプロトコル仕様を指定できます。port_listはポート番号のカンマ区切りリストです。この属性は、オンライン・リソース属性変更を使用して変更することもできます。

例

次の例は、管理リスナーのTCPポートを変更します。

```
$ srvctl modify mgmtlsnr -endpoints "TCP:2521,2522"
```

srvctl remove mgmtlsnr

Oracle Clusterwareから管理リスナー・リソース(CHM用)を削除します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl remove mgmtlsnr [-force]
```

使用上のノート

すべての依存性を無視し、管理リスナーを強制的に削除するには、-forceパラメータを使用します。

例

次に、このコマンドの例を示します。

```
$ srvctl remove mgmtlsnr
```

srvctl setenv mgmtlsnr

管理リスナー・リソース(CHM用)の環境構成を管理します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

このコマンドは、次のいずれかの構文モデルで 사용합니다。

```
srvctl setenv mgmtlsnr -envs "name=val[,...]"
srvctl setenv mgmtlsnr -env "name=val"
```

パラメータ

表F-75 srvctl setenv mgmtlsnrコマンドのパラメータ

パラメータ	説明
-envs "name=val[,...]"	設定する環境変数の名前/値ペアのカンマ区切りリストを、二重引用符(" ")で囲んで指定します。
-env "name=val"	このパラメータを使用して、単一の環境変数を二重引用符(" ")で囲み、カンマまたは他の特殊文字を含んだ値に設定できるようにします。

例

次の例は、管理リスナーの言語環境構成を設定します。

```
$ srvctl setenv mgmtlsnr -env LANG=en
```

srvctl start mgmtlsnr

管理リスナー・リソース(CHM用)を起動します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl start mgmtlsnr [-node node_name]
```

使用上のノート

管理リスナーを起動するノードを指定します。

例

crmnode1というノードの管理リスナーを起動するには、次のコマンドを実行します。

```
$ srvctl start mgmtlsnr -node crmnode1
```

srvctl status mgmtlsnr

管理リスナー・リソース(CHM用)のステータスを表示します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl status mgmtlsnr [-node node_name] [-verbose]
```

使用上のノート

- 必要に応じて、クラスタ・ノードの名前を指定できます。
- 必要に応じて、-verboseパラメータを使用して、詳細な出力を表示できます。

例

node2というノードの管理リスナーのステータスを表示するには、次のコマンドを使用します。

```
$ srvctl status mgmtlsnr -node node2
```

srvctl stop mgmtlsnr

すべてのノードまたは指定したノードの管理リスナー・リソース(CHM用)を停止します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl stop mgmtlsnr [-node node_name] [-force]
```

使用上のノート

- 必要に応じて、管理リスナーを実行している単一ノードの名前を指定できます。
- 必要に応じて、-forceパラメータを使用して、管理リスナーを強制的に停止できます。

例

次のコマンドは、ノードmynode1の管理リスナーを停止します。

```
$ srvctl stop mgmtlsnr -node mynode1
```

srvctl unsetenv mgmtlsnr

管理リスナー・リソース(CHM用)の環境構成の設定を解除します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl unsetenv mgmtlsnr -envs "name_list"
```

使用上のノート

-envsパラメータを使用して、二重引用符("")で囲まれた環境変数名のカンマ区切りリストを指定します。

例

次の例は、管理リスナー環境変数の環境構成の設定を解除します。

```
$ srvctl unsetenv mgmtlsnr -envs "LANG"
```

srvctl add mountfs

Network Attached Storage (NAS)構成をOracle Clusterwareに追加します。

ノート:



このコマンドは Oracle Clusterware でのみ使用できます。

構文

```
srvctl add mountfs -name mountfs_name -path mount_path -exportserver server_name  
-exportpath path [-mountoptions mount_options] [-user user]
```

パラメータ

表F-76 srvctl add mountfsコマンドのパラメータ

パラメータ	説明
-name mountfs_name	NAS の一意の名前を指定します。
-path mount_path	NAS のマウント・パスを指定します。
-exportserver server_name	エクスポート・サーバーの名前を指定します。
-exportpath path	エクスポート・ファイル・パスを指定します。
-mountoptions mount_options	必要に応じて、NAS のマウント・オプションを指定できます。
-user user	必要に応じて、NAS のマウントとマウント解除の権限を持つユーザーを指定できます。

srvctl config mountfs

特定のNASの構成情報を表示します。



ノート:

このコマンドは Oracle Clusterware でのみ使用できます。

構文

```
srvctl config mountfs [-name mountfs_name]
```

使用上のノート

構成情報を表示するNASの一意の名前を指定します。

srvctl disable mountfs

Oracle Clusterware管理において指定したNASを無効化します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl disable mountfs -name mountfs_name [-node node_list]
```

パラメータ

表F-77 srvctl disable mountfsコマンドのパラメータ

パラメータ	説明
-name mountfs_name	無効化する NAS の一意の名前を指定します。
-node node_list	必要に応じて、NAS を無効化するノードのカンマ区切りリストを指定できます。

srvctl enable mountfs

指定されたNetwork Attached Storage (NAS)をOracle Clusterware管理に対して有効化します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl enable mountfs -name mountfs_name [-node node_list]
```

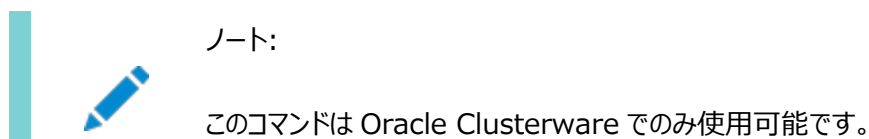
パラメータ

表F-78 srvctl enable mountfsコマンドのパラメータ

パラメータ	説明
-name mountfs_name	無効化する NAS の一意の名前を指定します。
-node node_list	必要に応じて、NAS を有効化するノードのカンマ区切りリストを指定できます。

srvctl modify mountfs

Oracle Clusterware内のNetwork Attached Storage (NAS)構成を変更します。



構文

```
srvctl modify mountfs -name mountfs_name [-path mount_path] [-exportserver  
server_name]  
[-exportpath path] [-mountoptions mount_options] [-user user_name]
```

パラメータ

表F-79 srvctl modify mountfsコマンドのパラメータ

パラメータ	説明
-name mountfs_name	変更する NAS の一意の名前を指定します。
-path mount_path	NAS のマウント・パスを変更します。
-exportserver server_name	エクスポート・サーバーの名前を変更します。
-exportpath path	エクスポート・ファイル・パスを変更します。
-mountoptions mount_options	NAS のマウント・オプションを変更します。
-user user_name	NAS のマウントとマウント解除の権限を持つユーザーを変更します。

srvctl remove mountfs

特定のNAS構成を削除します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl remove mountfs -name mountfs_name [-force]
```

使用上のノート

- マウントするNASの一意の名前を指定します。
- 必要に応じて、-forceパラメータを使用し、NASの削除中にリソースの依存性を無視することもできます。

srvctl start mountfs

特定のNASをマウントします。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl start mountfs -name mountfs_name [-node node_list]
```

パラメータ

表F-80 srvctl start mountfsコマンドのパラメータ

パラメータ	説明
-name mountfs_name	マウントする NAS の一意の名前を指定します。
-node node_list	必要に応じて、NAS をマウントするノードのカンマ区切りリストを指定できます。

srvctl status mountfs

特定のNASの現在の状態を表示します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl status mountfs -name mountfs_name
```

使用上のノート

状態を表示するNASの一意の名前を指定します。

srvctl stop mountfs

特定のNASをマウント解除します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl stop mountfs -name mountfs_name [-node node_list] [-force]
```

パラメータ

表F-81 srvctl stop mountfsコマンドのパラメータ

パラメータ	説明
-name mountfs_name	マウント解除する NAS の一意の名前を指定します。
-node node_list	必要に応じて、NAS をマウント解除するノードのカンマ区切りリストを指定できます。
-force	必要に応じて、-force パラメータを使用し、NAS の停止中にリソースの依存性を無視することもできます。

srvctl add ovmm

仮想マシンを管理するために、資格証明ストアにマネージャ資格証明を追加します。

構文

```
srvctl add ovmm -wallet certificate_wallet_file -username user_name -ovmmhost  
host_name | ip_address  
-ovmmport port_number [-autologin]
```

パラメータ

表F-82 srvctl add ovmmコマンドのパラメータ

パラメータ	説明
-wallet certificate_wallet_file	仮想マシン・マネージャで使用される CA 証明書が含まれている証明ウォレットへのパスを指定します。
-autologin	必要に応じて、ウォレット資格証明の自動ログインを示すこのパラメータを含めることができます。このパラメータを指定しない場合、ユーザーは、ウォレットのパスワードの入力を

パラメータ	説明
	常に求められます。
<code>-username user_name</code>	マネージャ資格証明を追加する対象のユーザー名を指定します。入力を要求されたら、マネージャのパスワードを指定する必要があります。
<code>-ovmmhost host_name ip_address</code>	仮想マシン・マネージャとして指定されたホストのホスト名または IP アドレスを指定します。
<code>-ovmmport port_number</code>	仮想マシン・マネージャで使用するポート番号を指定します。

使用上のノート

このコマンドは、rootまたは同等のユーザーとして実行する必要があります。

例

次に、このコマンドの例を示します。

```
# srvctl add ovmm -username scott -wallet /tmp/ovmmcred/cwallet.sso -ovmmhost node1 -ovmmport 80
```

srvctl config ovmm

仮想マシン・マネージャの現在の構成を表示します。

構文

```
srvctl config ovmm
```

使用上のノート

このコマンドはパラメータを取りません。

例

このコマンドによって、次のような出力が返されます。

```
$ srvctl config ovmm
OVMM Host: host1
OVMM Port: 80
```

srvctl modify ovmm

仮想マシン・マネージャの構成を変更します。

構文

```
srvctl modify ovmm [-wallet certificate_wallet_file [-autologin]] [-username user_name]
                  [-ovmmhost host_name | ip_address [-ovmmport port_number]]
```

パラメータ

表F-83 `srvctl modify ovmm` コマンドのパラメータ

パラメータ	説明
<code>-wallet certificate_wallet_file</code>	仮想マシン・マネージャで使用される CA 証明書が含まれている証明ウォレットへのパスを指定します。
<code>-autologin</code>	必要に応じて、ウォレット資格証明の自動ログインを示すこのパラメータを含めることができます。このパラメータを指定しない場合、ユーザーは、ウォレットのパスワードの入力を常に求められます。
<code>-username user_name</code>	マネージャ資格証明を追加する対象のユーザー名を指定します。入力を要求されたら、マネージャのパスワードを指定する必要があります。
<code>-ovmmhost host_ip</code>	仮想マシン・マネージャとして指定されたホストのホスト名または IP アドレスを指定します。
<code>-ovmmpport port_number</code>	仮想マシン・マネージャで使用されるポート番号を指定します。

使用上のノート

- このコマンドは、root または同等のユーザーとして実行する必要があります。
- `srvctl modify ovmm` コマンドでは、新しい資格証明を書き込む前に、すべての資格証明を削除します。

例

次に、このコマンドの例を示します。

```
$ srvctl modify ovmm -username scott -ovmport 81
```

srvctl remove ovmm

仮想マシン・マネージャの現在の構成を削除します。

構文

```
srvctl remove ovmm
```

使用上のノート

- このコマンドは、root または同等のユーザーとして実行する必要があります。
- このコマンドはパラメータを取りません。

例

次に、このコマンドの例を示します。

```
# srvctl remove ovmm
```

srvctl add qosmserver

クラスタ内のすべてのノードに、Oracle Database QoS Managementサーバー・インスタンスを追加します。

構文

```
srvctl add qosmserver [-verbose]
```

使用上のノート

必要に応じて、-verboseパラメータを使用して、冗長出力を表示できます。

例

次のコマンドでは、Oracle Database QoS Managementサーバー・インスタンスを追加します。

```
$ srvctl add qosmserver
```

srvctl config qosmserver

qosmserverリソースの構成情報を表示します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl config qosmserver
```

パラメータ

このコマンドにパラメータはありません。

例

次のコマンドでは、qosmserverリソースの構成情報を表示します。

```
$ srvctl config qosmserver
```

srvctl disable qosmserver

すべてのノードまたは特定のノードで、Oracle Database QoS Managementサーバー・インスタンスを無効化します。

構文

```
srvctl disable qosmserver [-node node_name] [-verbose]
```

使用上のノート

このコマンドでは、必要に応じて、Oracle Database QoS Managementサーバー・インスタンスを無効にする特定のノードを指定でき、さらに冗長出力を選択することもできます。

例

次の例では、crsNode3というノードのOracle Database QoS Managementサーバーを無効にします。

```
$ srvctl disable qosmserver -node crsNode3
```

srvctl enable qosmserver

すべてのノードまたは特定のノードで、Oracle Database QoS Managementサーバー・インスタンスを有効化します。

構文

```
srvctl enable qosmserver [-node node_name] [-verbose]
```

使用上のノート

このコマンドでは、必要に応じて、Oracle Database QoS Managementサーバー・インスタンスを有効にする特定のノードを指定でき、さらに冗長出力を選択することもできます。

例

次のコマンド例では、crsNode3というノードのOracle Database QoS Managementサーバー・インスタンスを有効にします。

```
$ srvctl enable qosmserver -node crsNode3
```

srvctl modify qosmserver

Oracle Database QoS Managementサーバー・インスタンスのRMIポートを変更します。

構文

```
srvctl modify qosmserver
[-rmiport qosmserver_rmi_port]
[-httpport qosmserver_http_port]
[-secure '{YES|NO}']
[-enableHTTPS '{YES|NO}']
[-verbose]
[-force]
```

表F-84 srvctl modify qosmserverコマンドのパラメータ

パラメータ	説明
-rmiport qosmserver_rmi_port	Oracle Database QoS Management サーバー・インスタンスによって使用される QoS Management RMI ポート番号を指定します。 ノート: この属性はオンライン・リソース属性変更を使用して変更できます。
-httpport qosmserver_http_port	HTTP ポート番号を指定します。
-secure {YES NO}	QoS Management サーバーのセキュリティを有効にするかどうかを指定します。
-enableHTTPS {YES NO}	QoS Management サーバーのトランスポート層セキュリティを有効にするかどうかを指定します。

パラメータ	説明
-force	変更を反映するためにリソースを停止して再起動します。
-verbose	冗長出力

例

次のコマンド例では、QoS Management RMIポート番号を5385に設定します。

```
$ srvctl modify qosmserver -rmiport 5385
```

srvctl predict qosmserver

srvctl predict qosmserverコマンドでは、Oracle Database QoS Managementサーバー障害の結果を予測します。

構文

```
srvctl predict qosmserver [-verbose]
```

このコマンドで利用できるパラメータは、冗長出力を表示する1つのオプション・パラメータのみです。

例

このコマンドによって、次のような出力が返されます。

```
$ srvctl predict qosmserver
Resource ora.qosmserver will be started on node crsNode1
```

srvctl relocate qosmserver

Oracle Database QoS Managementサーバー・インスタンスを、現行のホスティング・ノードからクラスタ内の別のノードに再配置します。

構文

```
srvctl relocate qosmserver [-node node_name] [-verbose]
```

使用上のノート

- 必要に応じて、Oracle Database QoS Managementサーバー・インスタンスを再配置する特定のノードを指定できます。
- 必要に応じて、-verboseパラメータを使用して、冗長出力を表示できます。

例

次のコマンド例では、crsNode1というノードにOracle Database QoS Managementサーバー・インスタンスを再配置して、冗長出力を表示します。

```
$ srvctl relocate qosmserver -node crsNode1 -verbose
```

srvctl remove qosmserver

Oracle Clusterware構成からOracle Database QoS Managementサーバー・インスタンスを削除します。

構文

```
srvctl remove qosmserver [-force] [-verbose]
```

使用上のノート

必要に応じて、-forceパラメータと-verboseパラメータを使用して、Oracle Database QoS Managementサーバー・インスタンスを強制的に削除したり、冗長出力を表示することができます。

例

次のコマンドでは、Oracle Database QoS Managementサーバー・インスタンスを強制的に削除します。

```
$ srvctl remove qosmserver -force
```

srvctl start qosmserver

Oracle Database QoS Managementサーバー・インスタンスを起動します。

構文

```
srvctl start qosmserver [-node node_name] [-verbose]
```

使用上のノート

- 必要に応じて、Oracle Database QoS Managementサーバー・インスタンスを起動する特定のノードを指定できます。
- 必要に応じて、-verboseパラメータを使用して、冗長出力を表示できます。

例

次のコマンド例では、crsNode1というノードのOracle Database QoS Managementサーバー・インスタンスを起動します。

```
$ srvctl start qosmserver -node crsNode1
```

srvctl status qosmserver

Oracle Database QoS Managementサーバー・インスタンスを実行しているノードを特定します。

構文

```
srvctl status qosmserver [-node node_name] [-verbose]
```

必要に応じて、Oracle Database QoS Managementサーバー・インスタンスのステータスを取得できる対象を指定したり、-verboseパラメータを使用して冗長出力を表示することもできます。

例

このコマンドによって、次のような出力が返されます。

```
$ srvctl status qosmserver
QoS Management Server is enabled.
QoS Management Server is running on node crsNode1.
```

srvctl stop qosmserver

Oracle Database QoS Managementサーバー・インスタンスを停止します。

構文

```
srvctl stop qosmserver [-force] [-verbose]
```

使用上のノート

- 必要に応じて、Oracle Database QoS Managementサーバー・インスタンスを停止する特定のノードを指定できます。
- 必要に応じて、-verboseパラメータを使用して、冗長出力を表示できます。

例

次のコマンド例では、Oracle Database QoS Managementサーバー・インスタンスを強制的に停止します。

```
$ srvctl stop qosmserver -force
```

srvctl add rhpclient

フリー・パッチ適用およびプロビジョニング・クライアントの構成ファイルをOracle Clusterware構成に追加します。

構文

```
srvctl add rhpclient -clientdata path_to_file [-diskgroup disk_group_name -storage base_path] [-email email_address -mailserver mail_server_address -mailserverport mail_server_port] [-verbose]
```

パラメータ

表F-85 srvctl add rhpclientコマンドのパラメータ

パラメータ	説明
-clientdata path_to_file	フリー・パッチ適用およびプロビジョニング・クライアントのデータを含むファイルのパスを指定します。
-diskgroup disk_group_name	イメージの記憶域のための Oracle ACFS ファイル・システムの作成元となる Oracle ASM ディスク・グループを指定します。
-storage base_path	すべてのクラスタ・ノードで利用できる場所を指定します。この場所は、フリー・パッチ適用およびプロビジョニング・クライアント上のイメージの記憶域として作成する Oracle ACFS をマウントするために使用されます。
-email email_address	電子メール・アドレスを指定します。

パラメータ	説明
-mailserver mail_server_address	メール・サーバーのアドレスを指定します。
-mailserverport mail_server_port	メール・サーバーのポート番号を指定します。
-verbose	冗長出力を表示します。

使用上のノート

- このコマンドは、フリー・パッチ適用およびプロビジョニング・サーバーでは実行できません。
- LinuxおよびUNIXプラットフォームで、rootユーザーとしてこのコマンドを実行する必要があります。
- このコマンドではOracle Clusterwareリソースは作成されません。
- クライアントのRHP_MANAGED_STORAGEに作業用コピーを格納するには、このコマンド(または)とともに、-diskgroupおよび-storageパラメータを指定する必要があります。これらのオプションを使用する他のメリットは、マルチノード・クライアント・クラスタに適用されます。フリー・パッチ適用およびプロビジョニング・サーバーが作業用コピーをクライアント・クラスタにプロビジョニングすると、イメージは1つのノードに転送され、クライアントがそれを内部的に残りのノードに配布します。

例

次のコマンドでは、フリー・パッチ適用およびプロビジョニング・クライアントを作成します。

```
# srvctl add rhpcclient -clientdata /tmp/gnsdata
```

srvctl config rhpcclient

フリー・パッチ適用およびプロビジョニング・クライアント・インスタンスの構成情報を表示します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl config rhpcclient
```

例

このコマンドによって、次のような出力が返されます。

```
$ srvctl config rhpcclient
Fleet Patching and Provisioning Client is configured
Cluster name: mjk9394
Storage base path: /scratch/aime/storage
Disk group: MJKRHPCDG
Fleet Patching and Provisioning Server (RHPS): mjk8990clust
Fleet Patching and Provisioning Server discovery string: 198.51.100.191
Port number: 8896
Fleet Patching and Provisioning Client is enabled
Fleet Patching and Provisioning Client is individually enabled on nodes:
Fleet Patching and Provisioning Client is individually disabled on nodes:
```

srvctl disable rhpcient

Oracle Clusterware管理において特定のノード上のフリート・パッチ適用およびプロビジョニング・クライアント・インスタンスを無効化します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl disable rhpcient [-node node_name]
```

使用上のノート

必要に応じて、フリート・パッチ適用およびプロビジョニング・クライアントを無効にする特定のノードを指定できます。

例

次に、このコマンドの例を示します。

```
$ srvctl disable rhpcient -node clusterNode01
```

srvctl enable rhpcient

特定のノードのフリート・パッチ適用およびプロビジョニング・クライアント・インスタンスを有効化します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl enable rhpcient [-node node_name]
```

使用上のノート

必要に応じて、フリート・パッチ適用およびプロビジョニング・クライアント・インスタンスを有効化するノードを指定できます。ノードを指定しない場合、Oracle Clusterware構成のすべてのフリート・パッチ適用およびプロビジョニング・クライアントがSRVCTLにより有効化されます。

例

次に、このコマンドの例を示します。

```
$ srvctl enable rhpcient -node clusterNode03
```

srvctl modify rhpcient

フリート・パッチ適用およびプロビジョニング・クライアントのクライアント・データとイメージの記憶域の場所を変更します。

構文

```

srvctl modify rhpclient [-clientdata path_to_client_data] [-port rmi_port] [-
diskgroup dg_name -storage base_path] [-email email_address -mailserver
mail_server_address
-mailserverport mail_server_port]

```

パラメータ

表F-86 srvctl modify rhpclientコマンドのパラメータ

パラメータ	説明
-clientdata path_to_client_data	フリート・パッチ適用およびプロビジョニング・クライアント・データのパスを変更します。
-port rmi_port	フリート・パッチ適用およびプロビジョニング・クライアントで使用される RMI ポート番号を変更します。
-diskgroup dg_name	イメージの記憶域としての Oracle ACFS ファイル・システムの作成元となる Oracle ASM ディスク・グループを変更します。
-storage base_path	すべてのクラスタ・ノードで使用できるが必ずしも共有されない場所を変更します。フリート・パッチ適用およびプロビジョニング・クライアントを追加するときにこの場所が存在する必要はありません。すべてのローカルの Oracle ACFS 記憶域で、すべてのイメージが base_path/images にマウントされます。
-email email_address	電子メール・アドレスを変更します。
-mailserver mail_server_address	メール・サーバーのアドレスを変更します。
-mailserverport mail_server_port	メール・サーバーのポート番号を変更します。

使用上のノート

- このコマンドは、LinuxおよびUNIXプラットフォームではrootユーザーとして実行する必要があります。
- このコマンドは、フリート・パッチ適用およびプロビジョニング・クライアントでのみ実行します。
- フリート・パッチ適用およびプロビジョニング・サーバーに変更が通知されるように、このコマンドを実行した後で、フリート・パッチ適用およびプロビジョニング・クライアントでsrvctl stop rhpclientとsrvctl start rhpclientを実行する必要があります。

srvctl relocate rhpclient

フリート・パッチ適用およびプロビジョニング・クライアント・インスタンスを別のノードに再配置します。

ノート:



このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl relocate rhpclient -node node_name
```

使用上のノート

フリー・パッチ適用およびプロビジョニング・クライアント・インスタンスの再配置先ノードを指定します。

例

次に、このコマンドの例を示します。

```
$ srvctl relocate rhpclient -node crsNode03
```

srvctl remove rhpclient

フリー・パッチ適用およびプロビジョニング・クライアント・インスタンスを削除します。

ノート:



このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl remove rhpclient [-force]
```

使用上のノート

必要に応じて、-forceパラメータを使用し、すべての依存性を無視して、フリー・パッチ適用およびプロビジョニング・クライアント・インスタンスを強制的に削除することもできます。

srvctl start rhpclient

1つ以上のノードでフリー・パッチ適用およびプロビジョニング・クライアント・インスタンスを起動します。

ノート:



このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl start rhpclient [-node node_name]
```

使用上のノート

フリー・パッチ適用およびプロビジョニング・クライアント・インスタンスを起動するノードを指定します。ノードを指定しない場合、フリー・パッチ適用およびプロビジョニング・クライアントが起動されるノードがOracle Clusterwareによって決定されます。フリー・パッチ適用およびプロビジョニング・クライアント・リソースのカーディナリティは1です。Oracle Clusterwareは、フリー・パッチ適用およびプロビジョニング・クライアント・リソースが起動されるときに配置を決定します。

例

このコマンドの例では、crs3という名前のノードでフリート・パッチ適用およびプロビジョニング・サーバーが起動されます。

```
$ srvctl start rhpclient -node crs3
```

srvctl status rhpclient

フリート・パッチ適用およびプロビジョニング・クライアント・インスタンスの現在の状態を表示します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl status rhpclient
```

使用上のノート

このコマンドには、-help以外のパラメータは指定できません。

例

このコマンドでは、次のような出力が表示されます。

```
$ srvctl status rhpclient
Fleet Patching and Provisioning Client is enabled
Fleet Patching and Provisioning Client is running on node mjk1270093
```

srvctl stop rhpclient

実行中または起動中のフリート・パッチ適用およびプロビジョニング・クライアント・インスタンスを停止します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl stop rhpclient
```

使用上のノート

このコマンドには、-help以外のパラメータは指定できません。

srvctl add rhpserver

フリート・パッチ適用およびプロビジョニング・サーバーをOracle Clusterware構成に追加します。

構文

```
srvctl add rhpserver -storage base_path -diskgroup disk_group_name
[-email email_address -mailserver mail_server_address -mailserverport
mail_server_port]
[-pl_port RHP_progress_listener_port] [-clport RHP_copy_listener_port]
```

```
[-enableTLS {YES|NO}] [-enableHTTPS '{YES|NO}'] [-port_range low_val-high_val] [-usegnsvip]
```

パラメータ

表F-87 srvctl add rhpserverコマンドのパラメータ

パラメータ	説明
-storage base_path	すべてのクラスタ・ノードで利用できる場所を指定します。この場所を共有する必要はありません。サーバーのカatalogはこの場所に保持され、イメージ・ファイル・システムはすべて base_path/images にマウントされます。
-diskgroup disk_group_name	イメージを格納するための Oracle ACFS ファイル・システムの作成元となる特定の Oracle ASM ディスク・グループを指定します。
-email email_address	電子メール・アドレスを指定します。
-mailserver mail_server_address	メール・サーバーのアドレスを指定します。
-mailserverport mail_server_port	メール・サーバーのポート番号を指定します。
-pl_port RHP_progress_listener_port	フリー・パッチ適用およびプロビジョニング進捗リスナーのポート番号を指定します。
-clport RHP_copy_listener_port	フリー・パッチ適用およびプロビジョニング・コピー・リスナーのポート番号を指定します。
-enableTLS {YES NO}	フリー・パッチ適用およびプロビジョニング・トランスポート・レベル・セキュリティを有効化します
-enableHTTPS '{YES NO}'	HTTP のフリー・パッチ適用およびプロビジョニング・トランスポート・レイヤー・セキュリティを有効にします。
-port_range low_val-high_val	ファイル転送に使用するポートについて、上下のポート値を指定します。
-usegnsvip	GNS ポート 53 がすでに使用されている場合、オプションで、Oracle FPP サーバーが仮想 IP アドレス(VIP)で通信できるようにします。

使用上のノート

- このコマンドを実行するには、LinuxおよびUNIXプラットフォームではrootユーザーである必要があります。
- Oracle Clusterwareをインストールしたユーザーがデフォルト・ユーザーになり、このユーザーがフリート・パッチ適用およびプロビジョニング・サーバー・リソースのユーザー・リストに追加されます。

srvctl config rhpserver

フリート・パッチ適用およびプロビジョニング・サーバーの構成情報を表示します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl config rhpserver
```

例

このコマンドによって、次のような出力が返されます。

```
$ srvctl config rhpserver
Storage base path: /scratch/aime/storagedg
Disk Groups: MJKRHPSDG
Port number: 8896
Fleet Patching and Provisioning Server is enabled
Fleet Patching and Provisioning Server is individually enabled on nodes:
Fleet Patching and Provisioning Server is individually disabled on nodes:
```

srvctl disable rhpserver

Oracle Clusterware管理において特定のノード上のフリート・パッチ適用およびプロビジョニング・サーバーを無効化します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl disable rhpserver [-node node_name]
```

使用上のノート

必要に応じて、フリート・パッチ適用およびプロビジョニング・サーバーを無効化するノードを指定できます。

例

次に、このコマンドの例を示します。

```
$ srvctl disable rhpserver -node clusterNode01
```

srvctl enable rhpserver

このコマンドを使用して、特定のノードのフリート・パッチ適用およびプロビジョニング・サーバーを有効化します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl enable rhpserver [-node node_name]
```

使用上のノート

必要に応じて、フリート・パッチ適用およびプロビジョニング・サーバーを有効化するノードを指定できます。ノードを指定しない場合、Oracle Clusterware構成のすべてのフリート・パッチ適用およびプロビジョニング・サーバーがSRVCTLにより有効化されます。

例

次に、このコマンドの例を示します。

```
$ srvctl enable rhpserver -node clusterNode03
```

srvctl modify rhpserver

フリート・パッチ適用およびプロビジョニング・サーバーの記憶域を変更します。

構文

```
srvctl modify rhpserver [-port rmi_port [-force]] [-email email_address  
-mailserver mail_server_address -mailserverport mail_server_port]  
[-pl_port RHP_progress_listener_port] [-clport RHP_copy_listener_port]  
[-enableTLS {YES|NO}] [-enableHTTPS '{YES|NO}'] [-port_range low_val-high_val] [-  
usegnsvip]
```

パラメータ

表F-88 srvctl modify rhpserverコマンドのパラメータ

パラメータ	説明
-port rmi_port	フリート・パッチ適用およびプロビジョニング・サーバーで使用する RMI ポート番号を変更します。
-force	このパラメータを指定すると、リソースが停止して再起動し、変更が有効になります。
-email email_address	電子メール・アドレスを変更します。
-mailserver mail_server_address	メール・サーバーのアドレスを変更します。
-mailserverport mail_server_port	メール・サーバーのポート番号を変更します。
-pl_port RHP_progress_listener_port	フリート・パッチ適用およびプロビジョニング進捗リスナーの

パラメータ	説明
	ポート番号を変更します。
<code>-clport RHP_copy_listener_port</code>	フリート・パッチ適用およびプロビジョニング・コピー・リスナーのポート番号を変更します。
<code>-enableTLS {YES NO}</code>	フリート・パッチ適用およびプロビジョニング・トランスポート・レベル・セキュリティを有効化します
<code>-enableHTTPS '{YES NO}'</code>	HTTP のフリート・パッチ適用およびプロビジョニング・トランスポート・レイヤー・セキュリティを有効にします。
<code>-port_range low_val-high_val</code>	ファイル転送に使用するポートについて、上下のポート値を変更します。
<code>-usegnsvip</code>	GNS ポート 53 がすでに使用されている場合、オプションで、Oracle FPP サーバーが仮想 IP アドレス(VIP)で通信できるようにします。

srvctl relocate rhpserver

フリート・パッチ適用およびプロビジョニング・サーバーを別のノードに再配置します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl relocate rhpserver [-node node_name]
```

使用上のノート

必要に応じて、フリート・パッチ適用およびプロビジョニング・サーバーを再配置するノードを指定できます。

例

次に、このコマンドの例を示します。

```
$ srvctl relocate rhpserver -node crsNode03
```

srvctl remove rhpserver

クラスタに構成されているフリート・パッチ適用およびプロビジョニング・サーバー・インスタンスを削除します。

構文

```
srvctl remove rhpserver [-resource] [-force] [-verbose]
```

表F-89 srvctl remove rhpserverコマンドのパラメータ

パラメータ	説明
-resource	必要に応じて、フリート・パッチ適用およびプロビジョニング・サーバーのリソースのみを削除して、そのリポジトリが影響を受けないようにすることができます。
-force	すべての依存性を無視し、フリート・パッチ適用およびプロビジョニング・サーバーを強制的に削除します。
-verbose	冗長出力を表示します。

srvctl start rhpserver

特定のノードでフリート・パッチ適用およびプロビジョニング・サーバーを起動します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl start rhpserver [-node node_name]
```

使用上のノート

必要に応じて、フリート・パッチ適用およびプロビジョニング・サーバーを起動するノードを指定できます。

例

次のコマンドを実行すると、clusterNode03という名前のノードでフリート・パッチ適用およびプロビジョニング・サーバーが起動します。

```
$ srvctl start rhpserver -node clusterNode03
```

srvctl status rhpserver

フリート・パッチ適用およびプロビジョニング・サーバーの現在の状態を表示します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl status rhpserver
```

使用上のノート

このコマンドには、-help以外のパラメータは指定できません。

例

このコマンドによって、次のような出力が返されます。

```
$ srvctl status rhpserver
Fleet Patching and Provisioning Server is enabled
Fleet Patching and Provisioning Server is running on node mjk1270089
```

srvctl stop rhpserver

実行中または起動中のフリート・パッチ適用およびプロビジョニング・サーバーを停止します。



ノート:

このコマンドは Oracle Clusterware でのみ使用可能です。

構文

```
srvctl stop rhpserver
```

使用上のノート

このコマンドには、-help以外のパラメータは指定できません。

srvctl add vm

仮想マシン・リソースをOracle Clusterware管理に追加します。

構文

```
srvctl add vm -name resource_name -vm "list_of_vms" [-serverpool server_pool_name | -category server_category | -node "node_list"] [-stoptimeout stop_timeout] [-checkinterval check_interval]
```

パラメータ

表F-90 srvctl add vmコマンドのパラメータ

パラメータ	説明
-name resource_name	仮想マシン・リソースの一意の名前を指定します。名前が一意でない場合、コマンドでエラーが発生します。
-vm "list_of_vms"	仮想マシン・リソースに追加する仮想マシン名または仮想マシン ID のカンマ区切りリストを指定します。リストは、二重引用符(" ")で囲んで記述します。 現在、Oracle Clusterware ではこのリストの先頭にある仮想マシンが優先されるため、最も重要な仮想マシン

パラメータ	説明
	を最初にリストする必要があります。
<code>-serverpool server_pool_name -category server_category -node "node_list"</code>	必要に応じて、仮想マシン・リソースを追加するサーバー・プールの名前、サーバー・カテゴリまたはノードのリストを指定できます。 ノードのリストは、二重引用符("")で囲んで記述します。 ノードのリストを指定する場合、ノードの数は指定する仮想マシン・リソースの数以上である必要があります。
<code>-stoptimeout stop_timeout</code>	必要に応じて、停止タイムアウトを秒単位で指定できます。パラメータに値を割り当てない場合のデフォルトは 180 です。
<code>-checkinterval check_interval</code>	必要に応じて、チェック間隔を秒単位で指定できます。パラメータに値を割り当てない場合のデフォルトは 10 です。

使用上のノート

仮想マシン・リソースを追加する前に、`srvctl add ovmm`を実行して、仮想マシン・マネージャの資格証明を追加する必要があります。

例

次に、このコマンドの例を示します。

```
$ srvctl add vm -name vmres -vm vm1 -node node1
```

srvctl check vm

特定の仮想マシンまたは特定のノードで実行されている仮想マシン・インスタンスのステータスを確認します。

構文

```
srvctl check vm -name resource_name [-vm vm_name | -node node_name]
```

パラメータ

表F-91 srvctl check vmコマンドのパラメータ

パラメータ	説明
<code>-name resource_name</code>	ステータスを確認する仮想マシン・リソースの名前を指定します。

パラメータ	説明
-vm vm_name	指定したリソースが存在する仮想マシンの名前または ID を指定します。
-node node_name	または、仮想マシンが存在するノードの名前を指定します。

srvctl config vm

特定の仮想マシンの現在の構成を表示します。

構文

```
srvctl config vm [-name unique_name]
```

使用上のノート

構成情報を表示する仮想マシン・リソースの一意の名前を指定します。特定の仮想マシン・リソースを指定しない場合、クラスタ内のすべての仮想マシン・リソースが表示されます。

例

このコマンドでは、使用されるコマンド構文に応じて、次のような出力が表示されます。

```
$ srvctl config vm -name vmres1
VM resource name: vmres1
VMs: vm1, y, vm3
VM Names: x, y, z
VM IDs: vm1, vm2, vm3
Server pool: vmpool
Server category:
Nodes:
Stop Timeout: 7
Check Interval: 5
VM resource "vmres1" is enabled.
VM resource is individually enabled on nodes:
VM resource is individually disabled on nodes:
VM resource is individually disabled for VMs: vm2,vm3
$
$ srvctl config vm
vmres1
vmres2
vmres3
$
```

srvctl disable vm

仮想マシン・リソースを無効にします。

構文

```
srvctl disable vm -name unique_name [-vm "vm_list" | -node node_name]
```

パラメータ

表F-92 srvctl disable vmコマンドのパラメータ

パラメータ	説明
-name unique_name	無効にする仮想マシン・リソースの一意の名前を指定します。
-vm "vm_list"	指定したリソースが存在する仮想マシンの名前または ID を指定するか、名前または ID のリストを二重引用符(" ")で囲んで指定します。このパラメータを指定すると、その名前または ID に一致するカーディナリティのインスタンスが有効になります。
-node node_name	または、仮想マシンが存在するノードの名前を指定します。このパラメータを指定すると、そのノードでは指定したリソースのインスタンスが実行されなくなります。

使用上のノート

オプション・パラメータを指定しない場合、仮想マシン・リソースが無効になり、すべてのノードの有効または無効の状態がパーズされますが、リソース内の仮想マシンの状態は変更されません。

例

次に、このコマンドの例を示します。

```
$ srvctl disable vm -name vmres -vm "vm1,vm2,vm3"
```

srvctl enable vm

仮想マシン・リソースを有効にします。

構文

```
srvctl enable vm -name unique_name [-vm "vm_list" | -node node_name]
```

パラメータ

表F-93 srvctl enable vmコマンドのパラメータ

パラメータ	説明
-name unique_name	有効にする仮想マシン・リソースの名前を指定します。
-vm vm_list	指定したリソースが存在する仮想マシンの名前または ID を指定するか、名前または ID のリストを二重引用符(" ")で囲んで指定します。このパラメータを指定すると、その名前または ID に一致するカーディナリティのインスタンスが無効になります。
-node node_name	または、仮想マシンが存在するノードの名前を指定します。このパラメータを指定すると、そのノードでは指定したリソースのインスタンスが実行されなくなります。

使用上のノート

オプション・パラメータを指定しない場合、仮想マシン・リソースが有効になり、すべてのノードの有効または無効の状態がパーズされますが、リソース内の仮想マシンの状態は変更されません。

例

次に、このコマンドの例を示します。

```
$ srvctl enable -name vmres -node node3
```

srvctl modify vm

仮想マシン・リソースの構成を変更します。

構文

```
srvctl modify vm -name unique_name [-addvm "list_of_vms" | -removevm "list_of_vms"  
  [-serverpool server_pool_name | -category server_category | -node "node_list"]  
  [-stoptimeout stop_timeout] [-checkinterval check_interval]
```

パラメータ

表F-94 srvctl modify vmコマンドのパラメータ

パラメータ	説明
-name unique_name	変更する仮想マシン・リソースの一意の名前を指定します。
-addvm "list_of_vms"	仮想マシン・リソースに追加する仮想マシン名または仮想マシン ID のカンマ区切りリストを二重引用符("")で囲んで指定します。仮想マシン・リソース・インスタンスが実行中かどうかに関係なく、いつでも仮想マシン・リソースに仮想マシンを追加できます。仮想マシンを追加すると、新しいカーディナリティの ID が作成されます。
-removevms "list_of_vms"	仮想マシン・リソースから削除する仮想マシン名または仮想マシン ID のカンマ区切りリストを指定します。まだ実行されている仮想マシン・リソースの既存のインスタンスがある場合、仮想マシンは削除できません。設定すると、エラーが発生します。
-serverpool server_pool_name -category server_category -node "node_list"	仮想マシン・リソースが属するサーバー・プール、サーバー・カテゴリまたはノードのリストを変更できます。 ノードのリストは、二重引用符("")で囲んで記述します。 ノードのリストを指定する場合、ノードの数は指定する仮

パラメータ	説明
	想マシン・リソースの数以上である必要があります。
<code>-stoptimeout stop_timeout</code>	停止タイムアウトを秒単位で変更できます。パラメータに値を割り当てない場合のデフォルトは 180 です。
<code>-checkinterval check_interval</code>	チェック間隔を秒単位で変更できます。パラメータに値を割り当てない場合のデフォルトは 10 です。

例

次に、このコマンドの例を示します。

```
$ srvctl modify vm -name vmres -addvm vm2 -removevm vm1 -stoptimeout 93
```

srvctl relocate vm

実行中の仮想マシン・リソースを別のノードに再配置するか、指定したノードで実行されている仮想マシン・リソースを別のノードに再配置します。

構文

```
srvctl relocate vm -name unique_name [-vm vm_name | -srcnode source_node_name]
-node destination_node_name]
```

パラメータ

表F-95 srvctl relocate vmコマンドのパラメータ

パラメータ	説明
<code>-name unique_name</code>	再配置する仮想マシン・リソースの一意の名前を指定します。
<code>-vm vm_name</code>	仮想マシン・リソースの再配置先のノードの名前とともに、仮想マシン・リソースの再配置先の指定したリソースが存在する仮想マシンの名前または ID を指定します。
<code>-srcnode source_node_name</code>	または、仮想マシン・リソースの再配置元のノードを指定できます。
<code>-node destination_node_name</code>	必要に応じて、仮想マシン・リソースの再配置先の宛先ノードの名前を指定できます。

例

次に、このコマンドの例を示します。

```
$ srvctl relocate -name vmres -vm vm1 -node node3
```


srvctl remove vm

特定の仮想マシン・リソースを削除します。

構文

```
srvctl remove vm -name unique_name [-force]
```

使用上のノート

削除する仮想マシン・リソースの一意の名前を指定し、必要に応じて、実行中の仮想マシン・リソースを強制的に削除できます。

srvctl start vm

ノードの特定の仮想マシン・リソースを起動します。

構文

```
srvctl start vm -name unique_name [-vm vm_name -node node_name]
```

パラメータ

表F-96 srvctl start vmコマンドのパラメータ

パラメータ	説明
-name resource_name	起動する仮想マシン・リソースの一意の名前を指定します。
-vm vm_name	必要に応じて、起動する仮想マシン・リソースが存在する仮想マシンを指定できます。
-node node_name	必要に応じて、起動する仮想マシン・リソースを含む仮想マシンが存在するノードの名前を指定できます。

使用上のノート

-vmパラメータと-nodeパラメータを一緒に使用して、特定のノードの特定の仮想マシン・リソースを起動できます。-vmパラメータのみを使用して特定の仮想マシン・リソースを起動することも、-nodeパラメータのみを使用して特定のノードに存在する仮想マシン・リソースを起動することもできます。

例

次に、このコマンドの例を示します。

```
$ srvctl start vm -name vmres -vm vm3 -node node3
```

srvctl status vm

仮想マシン・リソースの現在認識されている状態を表示します。

構文

```
srvctl status vm -name unique_name [-vm vm_name | -node node_name]
```

パラメータ

表F-97 `srvctl status vm` コマンドのパラメータ

パラメータ	説明
<code>-name unique_name</code>	ステータスを表示する仮想マシン・リソースの一意の名前を指定します。
<code>-vm vm_name</code>	必要に応じて、ステータスを表示する仮想マシン・リソースが存在する仮想マシンを指定できます。
<code>-node node_name</code>	必要に応じて、ステータスを表示する仮想マシン・リソースを含む仮想マシンが存在するノードの名前を指定できます。

例

このコマンドでは、指定したパラメータに応じて、次のような出力が返されます。

```
$ srvctl status vm -name vmres1 -vm vm1
Virtual machine "vm1" of VM resource "vmres1" is running on node aime1.
$
$ srvctl status vm -name vmres1
Virtual machine "vm1" of VM resource "vmres1" is running on node aime1.
Virtual machine "vm2" of VM resource "vmres1" is not running.
Virtual machine "vm3" of VM resource "vmres1" is running on node aime3
$
$ srvctl status vm -name vmres1 -vm vm1 -S 1
result: vm_resource={vmres1} inst_name={vm1} node_name={aime1} up={true}
$
```

srvctl stop vm

特定の仮想マシン・リソース、または特定の仮想マシンまたはノードの仮想マシン・リソースを停止します。

構文

```
srvctl stop vm -name unique_name [-vm vm_name | -node node_name]
```

パラメータ

表F-98 `srvctl stop vm` コマンドのパラメータ

パラメータ	説明
<code>-name unique_name</code>	停止する仮想マシン・リソースの一意の名前を指定します。
<code>-vm vm_name</code>	必要に応じて、停止する仮想マシン・リソースが存在する仮想マシンを指定できます。
<code>-node node_name</code>	必要に応じて、停止する仮想マシン・リソースを含む仮想マシンが存在するノードの名前を指定できます。

例

次に、このコマンドの例を示します。

```
$ srvctl stop vm -name vmres1 -node node1
```

G Oracle Clusterware C Application Program Interfaceエージェント・フレームワーク

Oracle Clusterwareは、リソースとしてOracle Clusterwareに登録されたアプリケーションを管理し、特定のリソースを起動、停止および監視できるアプリケーション独自のプリミティブに対するアクセス権限を持つ必要があります。Oracle Clusterwareでは、リソース固有のすべてのコマンドはエージェントと呼ばれるエンティティを介して実行されます。

エージェントは、リソースを管理するためのエージェント・フレームワークおよびユーザー・コードを含むプログラムです。エージェント・フレームワークは、アプリケーション固有のコードを組み込んで、カスタマイズされたアプリケーションを管理できるCライブラリです。アプリケーションの起動、停止、状態のチェックなど、CまたはC++での実際のアプリケーション管理機能のすべてをエージェントにプログラムします。これらの機能はエントリ・ポイントと呼ばれます。Oracle Clusterwareにかわって、エージェント・フレームワークによりこれらのエントリ・ポイント機能が起動されます。エージェントの開発者は、これらのエントリ・ポイントを使用して、リソースの起動、停止および監視方法に関する、機能を特定のリソースに必要な機能を組み込むことができます。エージェントで複数のリソースを管理できます。

この付録の内容は次のとおりです。

- [エージェント・フレームワークのデータ型](#)
- [エージェント・フレームワークのコンテキストの初期化および永続性](#)
- [CおよびC++エントリ・ポイント機能のプロトタイプ](#)
- [CおよびC++エントリ・ポイント・タイプおよびコード](#)
- [CおよびC++エントリ・ポイント・ファンクションの戻り値](#)
- [マルチスレッドに関する注意事項](#)
- [非推奨となったAPI](#)
- [APIリファレンス](#)
- [エージェントの例](#)

エージェント・フレームワークのデータ型

Oracle Clusterwareエージェント・フレームワークは、Oracleが定義したポータブル・データ型を使用して文字列および数字を表します。これらのデータ型を使用することで、複数のプラットフォームでエージェント・プログラムを構築および実行できます。

[表G-1](#)では、エージェント・フレームワークで使用されるOracleデータ型について説明しています。

表G-1 エージェント・フレームワークで使用されるOracleデータ型

Oracleデータ型	説明
ub4	符号なしの 32 ビット整数
sb4	符号付き 32 ビット整数

Oracleデータ型	説明
oratext*	Cスタイルの NULL で終了する文字列で、char *に明示的にキャストする必要があります

エージェント・フレームワークAPIによって戻されるclsagfwret型のステータス値は、次のとおりです。

- CLSAGFW_SUCCESS: ファンクションが正常に処理されました
- CLSAGFW_FAIL: パラメータ内または関数の実行でエラーがありました。
-

APIによって戻された文字列値は、現在のコマンド・エントリ・ポイントの範囲のみでアクセスできます。エージェントは戻された文字列値をエージェント独自のバッファにコピーして値を保持できます。

エージェント・フレームワークのコンテキストの初期化および永続性

エージェント・フレームワークは、clsagfw_init()ファンクション・コールによって初期化されます。維持する必要があるコンテキストはありません。アクション・エントリ・ポイント・ファンクションは、エージェント・フレームワーク・コンテキスト(type clsagfw_aectx)のポインタに渡されます。このコンテキストは、コールされているエントリ・ポイントに固有のもので、エントリ・ポイントの範囲を超えて使用することはできません。その他のエージェント・フレームワークAPIは、このコンテキスト・ポイントをパラメータとみなします。これは、エージェント・フレームワークが維持する不透明なデータ構造ですが、明示的に初期化されたり破壊されることはありません。

CおよびC++エントリ・ポイント機能のプロトタイプ

エージェントはリソース管理コマンドにエントリ・ポイント・ファンクションを実装し、エージェント・フレームワークにこれらのファンクションを登録します。

エージェント・フレームワークは、エントリ・ポイント・ファンクションをコールし、リソース上でコマンドやアクションを実行します。エントリ・ポイント・ファンクションは、次のプロトタイプおよび規則に従う必要があります。

```
typedef ub4 (*clsagfw_action_entry) (const clsagfw_aectx *aectx)
```

エントリ・ポイント・ファンクションは、エージェント・フレームワークが維持するコンテキスト・エリアへのポインタを使用してコールされます。このファンクションは、エントリ・ポイントの実行ステータスを示す値を戻します。CHECKエントリ・ポイントの場合、ファンクションはリソースの現在の状態を戻します。

C++を使用してエージェントを作成する場合、extern 'C' キーワードを使用してエントリ・ポイント・ファンクション・プロトタイプを宣言します。

CおよびC++エントリ・ポイント・タイプおよびコード

現在、エージェント・フレームワークは、リソース管理コマンドおよびリソース上で実行される操作に相当する7つのエントリ・ポイントを定義します。

エージェントは、START、STOP、CHECKおよびCLEANコマンドのエントリ・ポイントを登録する必要があります。その他のコマンド・エントリ・ポイントはオプションです。エージェントは、各リソース・タイプおよびコマンドにclsagfw_set_entrypoint() APIをコールすることでエントリ・ポイントを登録します。[表G-2](#)では、エントリ・ポイント・タイプおよびそれらに相当する型コードについて説明しています。

表G-2 エントリ・ポイント・タイプ

名前	説明	型コード
START	リソースを起動しオンラインにします	CLSAGFW_ACTION_START
STOP	リソースを停止しオフラインにします	CLSAGFW_ACTION_STOP
CLEAN	リソース上でコマンドが失敗した後にクリーンアップします	CLSAGFW_ACTION_CLEAN
CHECK	リソースの状態を確認します	CLSAGFW_ACTION_CHECK
ABORT	時間切れになった実行中のコマンドを終了します	CLSAGFW_ACTION_ABORT
DELETE	リソースは削除されます	CLSAGFW_ACTION_RES_DELETE D
MODIFY	リソースは変更されます	CLSAGFW_RESATTR_MODIFIED
ACTION	リソース上のカスタム・アクションを起動します	CLSAGFW_RES_ACTION

関連項目:

エントリ・ポイントの詳細は、「[Oracle Clusterwareリソースおよびエージェント](#)」を参照してください。

CおよびC++エントリ・ポイント・ファンクションの戻り値

CHECKエントリ・ポイントを除くすべてのエントリ・ポイント・ファンクションは、次の値のいずれかを戻して、エントリ・ポイントの実行状態を示します。

- CLSAGFW_AE_SUCCESS: エントリ・ポイントの実行が正常に完了した場合
- CLSAGFW_AE_FAIL: エントリ・ポイントの実行が失敗した場合

エージェント・フレームワークは、CHECKエントリ・ポイント・ファンクションをコールして、リソースの状態を監視します。このファンクションは、現在のリソースの状態を表す次のいずれかの値を戻します。

- CLSAGFW_ONLINE
- CLSAGFW_UNPLANNED_OFFLINE
- CLSAGFW_PLANNED_OFFLINE
- CLSAGFW_UNKNOWN
- CLSAGFW_PARTIAL
- CLSAGFW_FAILED

関連項目:

リソースの状態の評価およびリターンの詳細は、「[Oracle Clusterwareのエージェント](#)」を参照してください。

マルチスレッドに関する注意事項

Oracle Clusterwareエージェント・フレームワークはマルチスレッド対応に設計されています。また、エージェント・コードは、新しいスレッドを起動して、外部イベント・ソースの監視などのアプリケーション独自の機能を実行できます。

エージェント・フレームワークは、一度に1つのエントリ・ポイントのみを処理しますが、リソース上の現在有効なアクションを終了するために起動されるABORTエントリ・ポイントのみは例外です。CおよびC++の開発者は、すべてのエージェント・コードがマルチスレッド環境で正常に動作することを確認する必要があります。グローバル変数および他のリソースへのアクセスを、mutexesなどのロック機構を使用してシリアライズ化する必要があります。エージェントに含まれるCまたはC++の他の外部ライブラリも、マルチスレッド環境で正常に動作する必要があります。

非推奨のAPI

[表G-3](#)は、非推奨のエージェント・フレームワークAPIおよびそれに対応するOracle Clusterwareの代替APIを説明しています。

表G-3 非推奨のエージェント・フレームワークAPI

非推奨のAPI	代替
<code>clsagfw_exit()</code>	<code>clsagfw_exit2()</code>
<code>clsagfw_send_status()</code>	<code>clsagfw_send_status2()</code>

APIリファレンス

この項では、次のファンクションについて説明しています。

- [clsagfw_add_type\(\)](#)
- [clsagfw_check_resource\(\)](#)
- [clsagfw_create_attr_iterator\(\)](#)
- [clsagfw_delete_cookie\(\)](#)
- [clsagfw_exit2\(\)](#)
- [clsagfw_get_attr_from_iterator\(\)](#)
- [clsagfw_get_attrvalue\(\)](#)
- [clsagfw_get_check_type\(\)](#)
- [clsagfw_get_cmdid\(\)](#)
- [clsagfw_get_cookie\(\)](#)
- [clsagfw_get_request_action_name\(\)](#)
- [clsagfw_get_resource_id\(\)](#)
- [clsagfw_get_resource_name\(\)](#)

- [clsagfw_get_retry_count\(\)](#)
- [clsagfw_get_type_name\(\)](#)
- [clsagfw_init\(\)](#)
- [clsagfw_is_cmd_timedout\(\)](#)
- [clsagfw_log\(\)](#)
- [clsagfw_modify_attribute\(\)](#)
- [clsagfw_reset_attr_iterator\(\)](#)
- [clsagfw_send_status2\(\)](#)
- [clsagfw_set_cookie\(\)](#)
- [clsagfw_set_entrypoint\(\)](#)
- [clsagfw_set_exitchb\(\)](#)
- [clsagfw_set_resource_state_label\(\)](#)
- [clsagfw_startup\(\)](#)

clsagfw_add_type()

エージェントが管理するリソース・タイプを登録します。

構文

```
clsagfwret clsagfw_add_type(const oratext *type_name);
```

入力

type_name: エージェントが管理するリソース・タイプの名前です。

使用上のノート

起動時に、エージェントは管理するすべてのリソース・タイプを登録する必要があります。1つのエージェントで、複数のリソース・タイプおよび複数のリソースを管理できます。

clsagfw_check_resource()

特定のリソースに対してチェック・アクションを開始します。

構文

```
void clsagfw_check_resource(const oratext *pResId)
```

入力

pResId: リソースのID

使用上のノート

エージェントは、エントリ・ポイントまたは他のバックグラウンド・スレッドでこのファンクションを使用して、特定のリソースでチェック・アクションを開始できます。エージェント・フレームワークは、他のスレッドのチェック・アクションをスケジュールして実行します。このファンクションは、CHECKエントリ・ポイントから起動された場合、何の影響も与えません。

clsagfw_create_attr_iterator()

現在のリソースのすべての属性にアクセスするためのイテレータを作成します。

構文

```
clsagfwret clsagfw_create_attr_iterator(const clsagfw_aectx *pAeCtx,  
                                         clsagfw_ai_flags flags);
```

入力

- pAeCtx: エントリ・ポイントに渡されるエージェント・フレームワーク・コンテキスト・ポインタ
- flags:
 - CLSAGFW_ALL_ATTRS: リソースのすべての属性
 - CLSAGFW_MODIFIED_ATTRS: 変更された属性のみ

使用上のノート

エージェントはこのファンクションをコールしてイテレータを作成し、リソースのすべての属性にアクセスします。これは、エージェントが異なるタイプの複数のリソースを管理する場合に役立ちます。イテレータは、エージェント・フレームワーク・コンテキストに維持される内部イテレータです。各属性は、clsagfw_get_attr_from_iterator() APIをコールすることで取得されます。変更された属性はMODIFYエントリ・ポイント以外ではアクセスできません。エントリ・ポイントは指定のタイミングでイテレータを1つのみ作成できます。続いてイテレータの作成を試みると、ポインタはいずれの場合も、すでに作成されているイテレータになります。

clsagfw_delete_cookie()

エージェント・フレームワークに格納されているクッキーを削除します。

構文

```
clsagfwret clsagfw_delete_cookie(const oratext *key);
```

入力

key: Cookieのキーで、NULLで終了する文字列

使用上のノート

このファンクションは、Cookieを削除し、Cookieおよびその値が使用するすべてのメモリを解放します。

clsagfw_exit2()

リソース再試行フィードバックを使用してエージェント・プロセスを終了します。

構文

```
void clsagfw_exit2(const oratext *rid, sb4 exitCode, bool retry);
```

入力

- rid: 回復不能なエラーの発生原因であるリソースのリソースID
- exitCode: プロセス終了コード
- retry: リソースを再試行するオプション

使用上のノート

リソースのエントリ・ポイントを実行しているときに、エージェントで回復不能なエラーが発生し、現在のプロセスを終了することが必要な場合があります。エージェントはこのAPIを使用してエージェント・プロセスを終了し、リカバリ不能なエラーの原因となっているエージェント・フレームワークにフィードバックを提供できます。再試行パラメータをtrueに設定すると、エージェント・プロセスが再起動したときに、リソースの現在のコマンドが再試行されます。パラメータをfalseに設定すると、現在のリソースのコマンドは再試行されず、リソースの監視は無効になります。このオプションは、特定のリソースが、繰り返し発生するエージェントの失敗の原因である場合に使用できます。失敗する頻度が多い場合、Oracle Clusterwareサーバーは、エージェントが終了して自動的に無効になるリソースの記録を取ります。失敗する頻度が多いために無効化されるリソースを最有効化するには、リクエストを明示的に発行してリソースを起動する必要があります(たとえば、CRSCTLを使用します)。

Oracle Clusterwareサーバーはエージェントを再起動して、エージェントが管理するリソースの監視を続行します。

clsagfw_get_attr_from_iterator()

作成されたイテレータから次の属性名と値を戻します。

構文

```
clsagfwret clsagfw_get_attr_from_iterator(const clsagfw_aectx *pAeCtx,
                                           const oratext **argName,
                                           const oratext **argValue);
```

入力

pAeCtx: エントリ・ポイントに渡されるエージェント・フレームワーク・コンテキスト・ポインタ

戻り値

- argname: 属性名
- argValue: 属性値

使用上のノート

このファンクションを使用して、エージェント・フレームワーク・コンテキストにイテレータを作成した後に属性を取得します。このファンクションが値を戻すと、イテレータは次の属性に進みます。APIがCLSAGFW_FAILを戻した場合、取得する属性はリソースにこれ以上ありません。

clsagfw_get_attrvalue()

現在のリソースの属性の値を取得します。

構文

```
void clsagfw_get_attrvalue(const clsagfw_aectx *pAeCtx,
                           const oratext *pAttrName,
                           const oratext **pAttrValue);
```

入力

- pAeCtx: エントリ・ポイントに渡されるエージェント・フレームワーク・コンテキスト・ポインタ
- pAttrName: 取得する属性の名前

戻り値

pAttrValue: 指定した属性の値

使用上のノート

エージェントはこのAPIを使用して、現在のコマンド・リソースの属性1つの値を取得します。

clsagfw_get_check_type()

現在実行されているチェック・コマンドのタイプを返します。

構文

```
clsagfw_checktype  
clsagfw_get_check_type(const clsagfw_aectx *pAeCtx);
```

入力

pAeCtx: エントリ・ポイントに渡されるエージェント・フレームワーク・コンテキスト・ポインタ

使用上のノート

エージェント・フレームワークは、複数の条件でのリソースのCHECKエントリ・ポイントを起動します。このファンクションを使用することで、エージェントは現在のCHECKエントリ・ポイントの正確な条件を識別し、適切なアクションを実行します。

使用可能な条件は次のとおりです。

- CLSAGFW_INITIAL_CHECK: Oracle Clusterwareサーバーを起動または再起動したときに、最初のプローブ・ステージの一部として起動されるエントリ・ポイントを確認します。
- CLSAGFW_CHECK_AFTER_CMD: アクションが実行された後、結果のリソースを確認するために、起動されるエントリ・ポイントを確認して、以前のアクション(起動または停止など)を確認します。
- CLSAGFW_PERIODIC_CHECK: エージェント・フレームワークが定期的に起動するエントリ・ポイントを確認して、リソースの状態を監視します。時間間隔は、リソースのCHECK_INTERVAL属性で指定します。
- CLSAGFW_EXTERNAL_CHECK: CRSCTLなどの外部エンティティからの明示的な確認リクエストを受け取ると、エージェント・フレームワークが起動するエントリ・ポイントを確認します。
- CLSAGFW_CHECKFROM_EP: clsagfw_check_resource() APIをコールして、エージェントが開始するアクションを確認します。

clsagfw_get_cmdid()

エントリ・ポイントが起動されるコマンドのタイプを戻します。

構文

```
clsagfw_aecode clsagfw_get_cmdid(const clsagfw_aectx *pAeCtx);
```

入力

pAeCtx: エントリ・ポイントに渡されるエージェント・フレームワーク・コンテキスト・ポインタ

使用上のノート

エージェントは、ABORTエントリ・ポイントでこのAPIをコールして、終了中のリソース・コマンドを判断できます。

clsagfw_get_cookie()

エージェント・フレームワークに格納されているクッキーの値を取得します。

構文

```
clsagfwret clsagfwret clsagfw_set_cookie(const oratext *key,  
                                          const void **value);
```

入力

key: Cookieのキーで、NULLで終了する文字列

Value: クッキーの値を指すポインタ

clsagfw_get_request_action_name()

リソースで実行されるカスタム・アクションの名前を返します。

構文

```
const oratext *  
clsagfw_get_request_action_name(const clsagfw_aectx *pAeCtx);
```

入力

pAeCtx: ユーザーのエントリ・ポイントに渡されるエージェント・フレームワーク・コンテキスト

使用上のノート

このファンクションをACTIONエントリ・ポイントでコールすると、リクエスト・アクション・コマンドで指定したアクション名が取得されます。リソースのACTION属性に、リソースに対するアクションのリストを構成します。

clsagfw_get_resource_id()

エントリ・ポイントが処理されているリソースのIDを取得します。

構文

```
const oratext* clsagfw_get_resource_id(const clsagfw_aectx *pAeCtx);
```

入力

pAeCtx: エントリ・ポイントに渡されるエージェント・フレームワーク・コンテキスト・ポインタ

使用上のノート

このファンクションは、Oracle Clusterwareが生成するリソースIDを戻します。リソースのパブリックな表示名を取得するには、[clsagfw_get_resource_name\(\)](#) APIを使用します。

clsagfw_get_resource_name()

現在のコマンドのリソース名を取得します。

構文

```
const oratext*  
clsagfw_get_resource_name(const clsagfw_aectx *pAeCtx);
```

入力

pAeCtx: エントリ・ポイントに渡されるエージェント・フレームワーク・コンテキスト・ポインタ

使用上のノート

このファンクションは、現在のコマンドおよびエントリ・ポイントを実行するリソースの名前を取得します。

clsagfw_get_retry_count()

現在のコマンドの再試行回数を取得します。

構文

```
ub4 clsagfw_get_retry_count(const clsagfw_aectx *pAeCtx)
```

入力

pAeCtx: ユーザーのエントリ・ポイントに渡されるエージェント・フレームワーク・コンテキスト・ポインタ

使用上のノート

エージェントは、エントリ・ポイントでこのファンクションをコールして、現在のコマンドが再試行されているか確認できます。エージェント・プロセスやOracle Clusterwareサーバー・プロセスは、リソースでコマンドを実行している途中で終了する可能性があります。Oracle Clusterwareは、エージェントまたはサーバー・プロセスが再起動したときに、失敗したコマンドを再試行します。コマンドは、前回の試行で正常に実行された場合と一部しか実行されていない場合があります。エージェントは、Oracle Clusterwareがコマンドを再試行するときに、再開またはクリーンアップ・アクションを実行できます。コマンドを再試行できる回数はOracle Clusterwareの内部パラメータです。

clsagfw_get_type_name()

エントリ・ポイントが実行されているリソースのタイプ名を取得します。

構文

```
const oratext* clsagfw_get_type_name(const clsagfw_aectx *pAeCtx);
```

入力

pAeCtx: エントリ・ポイントに渡されるエージェント・フレームワーク・コンテキスト・ポインタ

使用上のノート

このファンクションは、Oracle Clusterwareが生成するリソースIDを戻します。リソースのパブリックな表示名を取得するには、[clsagfw_get_resource_name\(\)](#) APIを使用します。

clsagfw_init()

エージェント・フレームワークを初期化します。

構文

```
clsagfwret clsagfw_init(sb4 argc,  
                        oratext **argv,  
                        ub4 flags,  
                        const oratext *logName,  
                        void *reserved);
```

入力

argc: エージェント処理に渡されたコマンドライン引数の数

argv: エージェント処理に渡されたコマンドライン引数

flags:

logName: 対象エージェントに対するログ・ファイルの名前

reserved: NULLを渡します

使用上のノート

リソースでアクションを実行する必要がある場合、Oracle Clusterwareは、リソースに構成されたエージェントを自動的に起動します。起動時、エージェント・プロセスは最初にこのファンクションをコールして、エージェント・フレームワーク・ランタイムを初期化します。Oracle Clusterwareサーバーによってエージェント・プロセスに渡されるコマンドライン引数は、clsagfw_init()に渡される必要があります。初期化が完了すると、エージェントは、エージェントが管理するリソース・タイプを登録できます。

clsagfw_is_cmd_timedout()

ABORTエントリ・ポイントの原因を戻します。

構文

```
bool clsagfw_is_cmd_timedout(const clsagfw_aectx *pAeCtx);
```

入力

pAeCtx: エントリ・ポイントに渡されるエージェント・フレームワーク・コンテキスト・ポインタ

使用上のノート

エージェントは、ABORTエントリ・ポイントでこのファンクションを使用して、ABORTエントリ・ポイントの起動の原因を特定できます。ABORTエントリ・ポイントは、コマンドの実行時間がそのタイムアウト値を超過した場合か、Oracle Clusterwareの管理者がコマンドを中断した場合のいずれかの場合に起動できます。

clsagfw_log()

エージェント・ログ・ファイルの診断メッセージを出力します。

構文

```
void clsagfw_log(const clsagfw_aectx *pAeCtx,  
                const ub4 log_level,  
                const oratext *fmt, ...);
```

入力

pAeCtx: ユーザーのエントリ・ポイントに渡されるエージェント・フレームワーク・コンテキスト

log_level: 1から5までの範囲のメッセージのログ・レベル

fmt: 標準Cライブラリ・プリント仕様に基づくメッセージ・フォーマット文字列

...: フォーマット文字に対応する変数パラメータ

使用上のノート

エージェントは、このファンクションをコールして、エージェント実行のデバッグや理解に役立つ診断メッセージのログを記録します。LOG_LEVELパラメータを使用して、ログ・メッセージの量を制御できます。CRSCTLを使用して現在のロギング・レベルを次のように構成できます。

```
$ crsctl set log level res "myResource1=3"
```

誤用のログは、TODOディレクトリのエージェント・ログ・ファイルに記録されます。デフォルトでは、ログ・ファイルの名前は、エージェント実行ファイル名と同じです。これを変更するには、logNameパラメータをclsagfw_init() APIに渡します。

clsagfw_modify_attribute()

現在のリソースの属性を変更します。

構文

```
void clsagfw_modify_attribute(const clsagfw_aectx *pAeCtx,
                             const oratext *pAttrName,
                             const oratext *pAttrValue);
```

入力

pAeCtx: エントリ・ポイントに渡されるエージェント・フレームワーク・コンテキスト・ポインタ

pAttrName: 変更対象の属性の名前

pAttrValue: 属性の新しい値

使用上のノート

エージェントは、APIをコールすると、現在のコマンド・リソースの属性の値を変更できます。リソースの変更リクエストは、検証のためにOracle Clusterwareサーバーに送信され、更新された属性値は、Oracle Clusterwareサーバーが確認するまでエージェントに反映されません。このファンクションのコール元はSTART、STOPおよびCLEANエントリ・ポイントのみです。

clsagfw_reset_attr_iterator()

属性イテレータをリセットします。

構文

```
clsagfwret clsagfw_reset_attr_iterator(const clsagfw_aectx *pAeCtx,
                                       clsagfw_ai_flags flags);
```

入力

pAeCtx: エントリ・ポイントに渡されるエージェント・フレームワーク・コンテキスト・ポインタ

flags: アクセス対象の属性

CLSAGFW_ALL_ATTRS: リソースのすべての属性

CLSAGFW_MODIFIED_ATTRS: 変更された属性のみ

使用上のノート

このファンクションはエージェント・フレームワーク・コンテキストのイテレータをリセットするため、エージェントは属性のナビゲーションを再起動できます。

clsagfw_send_status2()

Oracle Clusterwareサーバーとユーザーに、進捗、警告およびエラー・メッセージをレポートします。

構文

```
void clsagfw_send_status2(const clsagfw_aectx* pAeCtx,
                         const oratext* pResId,
                         clsagfw_status_type type,
```

```
const oratext* fmtp,...);
```

入力

pAeCtx: エントリ・ポイントに渡されるエージェント・フレームワーク・コンテキスト・ポインタ

pResId: メッセージが報告されているリソース

type: 次に続くメッセージのタイプの内の1つ

CLSAGFW_STATUS_PROGRESS: 進捗メッセージ

CLSAGFW_STATUS_WARNING: 警告メッセージ

CLSAGFW_STATUS_ERROR: エラー・メッセージ

fmtp: メッセージ・フォーマット文字列

...: フォーマット文字列内の位置に対応する変数

使用上のノート

エージェントは、コマンドを実行している間、リソースで操作を実行しているOracle Clusterwareサーバーおよびユーザーに情報メッセージを送信できます。これらのメッセージで、ユーザーに実行の進捗状況および警告やエラーの有無を通知することができます。

clsagfw_set_cookie()

クッキーとその値をエージェント・フレームワークに格納します。

構文

```
clsagfwret clsagfwret clsagfw_set_cookie(const oratext *key,
                                          const void *value);
```

入力

key: Cookieのキーで、NULLで終了する文字列

Value: 格納対象のクッキーの値を指すポインタ

使用上のノート

clsagfw_get_cookie()をコールすると、保存されたCookieの値を後で取得できます。

clsagfw_set_entrypoint()

リソースに対する特定のアクションについてC/C++エントリ・ポイント・ファンクションが実行されるように設定します。

構文

```
void clsagfw_set_entrypoint(const oratext *type_name,
                           clsagfw_action_entry action_entry,
                           clsagfw_aecode action_entrycode)
```

入力

key: Cookieのキーで、NULLで終了する文字列

Value: 格納対象のクッキーの値を指すポインタ

使用上のノート

`clsagfw_get_cookie()` をコールすると、保存されたCookieの値を後で取得できます。

clsagfw_set_exitcb()

エージェント・プロセスの終了時に呼び出されるファンクションを登録します。

構文

```
typedef void (*clsagfw_exit_callback) (sb4 exitCode);  
void clsagfw_set_exitcb(clsagfw_exit_callback exit_func)
```

入力

`exit_func`: エージェント・プロセスが終了すると起動されるファンクション

使用上のノート

エージェントは、エージェント・フレームワークがエージェント・プロセスを終了すると、コールバック・ファンクションを登録して起動することができます。コールバック・ファンクションは、エージェントがインスタンス化した内部リソースを適切にクリーンアップできます。エージェント・プロセスがリカバリ不能なエラーから修復するために終了する場合、Oracle Clusterwareサーバーは、古いインカネーションが終了するまでエージェントを再起動しません。

clsagfw_set_resource_state_label()

現在のリソースにカスタム・ラベルを設定します。

構文

```
void clsagfw_set_resource_state_label(const clsagfw_aectx *pAeCtx,  
                                     const oratext *pLabel);
```

入力

`pAeCtx`: エントリ・ポイントに渡されるエージェント・フレームワーク・コンテキスト・ポインタ

`pLabel`: リソースに関連付けるラベルで、NULLで終了する文字列

使用上のノート

エージェントはこのファンクションを使用して、リソースの状態に関連付けられた説明ラベルを設定できます。このラベルはリソースのSTATE_DETAILS属性にレポートされます。

clsagfw_startup()

エージェント・フレームワークを起動します。

構文

```
clsagfwret clsagfw_startup(void)
```

使用上のノート

エージェント・フレームワークが初期化され、これらのエントリ・ポイントが登録された後、エージェントはこのAPIをコールしてエージェント・フレームワークを起動し、次にこれにより、マルチスレッド・インフラストラクチャが起動され、リソースのアクションが監視および実行されます。

成功すると、このファンクションがコール中のスレッドに戻ることはありません。リソース・アクションのエントリ・ポイントを起動する必要がある場合は、エージェント・フレームワークを使用します。

エージェントの例

[例-1](#)は、ファイルを表すリソースを管理するためのエージェント実装の詳細を説明しています。

例G-1 demoagent1.cpp

```
/* Copyright (c) 2004, 2012, Oracle and/or its affiliates.
All rights reserved. */
/*
 * NAME
 * Demoagent1.cpp: Demonstrates agent to monitor a file
 *
 * DESCRIPTION
 * This is a sample program that demonstrates an agent to monitor
 * a file. The agent has the following tasks:
 * - On startup      : Create the file.
 * - On shutdown    : Delete the file.
 * - On check command : Detect whether the file is present or not.
 * - On clean command : Delete the file.
 * This program can be used as a basis to build up more complicated
 * agents that manage real-world resources.
 *
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <clsagfw.h>
```

エージェント・フレームワークclsagfw.hヘッダー・ファイルを含めます。

```
#define TEST_TYPE1 (oratest *) "HOTFILE_TYPE"
#define TEST_PATHNAME_ATTR (oratest *) "PATH_NAME"
/*
 * NAME
 * test_agent_exit
 * DESCRIPTION
 * Function to clean up, called by framework whenever the agent
 * process is about to terminate
 * PARAMS
 * exitCode, which is an sb4 describing the reason for exit.
 * RETURNS
 * void
 */
void
test_agent_exit(sb4 exitCode)
{
    clsagfw_log(NULL, 1, (oratest *) "Demo Agent is exiting..¥n");
}
```

エージェント・フレームワークに登録されているコールバック・ファンクションを終了します。

```
/*
 * NAME
 * type1_start
 * DESCRIPTION
 * Action entry point for the 'start' command. Creates the file for
 * monitoring.
 * PARAMS
 * Pointer to the agent framework context
 * RETURNS
 * ub4 which could be CLSAGFW_AE_FAIL or CLSAGFW_AE_SUCCESS
 * depending on whether the action failed or succeeded.
```

```

*/
ub4
type1_start(const clsagfw_aectx *ectx)
{
    ub4 ret = CLSAGFW_AE_FAIL;
    const oratext *pResName = NULL;
    const oratext *pPathName = NULL;
    FILE *fp;
    clsagfw_log(ectx, 1, (oratext *)"Start action called..");
    /* Try to read the resource name */
    if (clsagfw_get_attrvalue(ectx, (oratext *)"NAME", &pResName) !=
        CLSAGFW_SUCCESS)
    {
        goto done;
    }
    /* Try to read the PATH_NAME attribute */
    if (clsagfw_get_attrvalue(ectx, TEST_PATHNAME_ATTR, &pPathName) !=
        CLSAGFW_SUCCESS)
    {
        goto done;
    }
}

```

clsagfw_get_attrvalue()を使用してリソース名およびPATH_NAME属性の名前を取得します。

```

    clsagfw_log(ectx, 1,
                (oratext *)"Start action arguments: resName = %s,
                pathName = %s", pResName, pPathName);
    /* Try to create the file */
    fp = fopen((char *)pPathName, "w");
    if(!fp)
    {
        /* Could not create the file */
        clsagfw_log(ectx, 1, (oratext *) "START action for resource %s:
        FAILED¥n", pResName);
        ret = CLSAGFW_AE_FAIL;
    }
    else
    {
        /* Created the file successfully */
        clsagfw_log(ectx, 1, (oratext *) "START action for resource %s:
        SUCCESS¥n", pResName);
        ret = CLSAGFW_AE_SUCCESS;
        fclose(fp);
    }
done:
    return ret;
}

```

clsagfw_log()をコールすると、診断メッセージおよびエラー・メッセージのログがエージェント・ログ・ファイルに記録されます。

```

/*
 * NAME
 *     type1_stop
 * DESCRIPTION
 *     Action entry point for the 'stop' and 'clean' commands. Deletes
 *     the file being monitored.
 * PARAMS
 *     Pointer to agent framework context
 * RETURNS
 *     ub4 which could be CLSAGFW_AE_FAIL or CLSAGFW_AE_SUCCESS
 *     depending on whether the action failed or succeeded.
 */
ub4
type1_stop (const clsagfw_aectx *ectx)
{
    ub4 ret = CLSAGFW_AE_FAIL;
}

```

```

const oratext    *pResName = NULL;
const oratext    *pPathName = NULL;
clsagfw_log(ectx, 1, (oratext *)"Stop action called..");
/* Try to read the resource name */
if (clsagfw_get_attrvalue(ectx, (oratext *)"NAME", &pResName) !=
    CLSAGFW_SUCCESS)
{
    clsagfw_log(ectx, 1, (oratext *)"STOP action: %s: Could not
        read attribute¥n", pResName);
    goto done;
}
/* Try to read the PATH_NAME attribute */
if (clsagfw_get_attrvalue(ectx, TEST_PATHNAME_ATTR, &pPathName) !=
    CLSAGFW_SUCCESS)
{
    pPathName = pResName;
}
clsagfw_log(ectx, 1,
    (oratext *)"Stop action arguments: resName = %s,
        pathName = %s", pResName, pPathName);
/* Try to delete the file */
if (remove((char *)pPathName))
{
    /* Could not delete the file */
    clsagfw_log(ectx, 1, (oratext *)"STOP action for resource %s:
        FAILED¥n", pResName);
    ret = CLSAGFW_AE_FAIL;
}
else
{
    /* Deleted the file successfully */
    clsagfw_log(ectx, 1, (oratext *)"STOP action for resource %s:
        SUCCESS¥n", pResName);
    ret = CLSAGFW_AE_SUCCESS;
}
done:
return ret;
}
/*
 * NAME
 *     type1_check
 * DESCRIPTION
 *     Action entry point for the 'check' command. Determines if the
 *     file exists.
 * PARAMS
 *     Pointer to agent framework context.
 * RETURNS
 *     ub4 which gives the status of the resource. Check the
 *     agent framework reference for details on this function's
 *     return codes.
 */
ub4
type1_check(const clsagfw_aectx *ectx)
{
    ub4          ret = CLSAGFW_UNKNOWN;
    const oratext    *pResName = NULL;
    const oratext    *pPathName = NULL;
    FILE *fp;
    clsagfw_log(ectx, 1, (oratext *)"Check action called..");
    /* Try to read the resource name */
    if (clsagfw_get_attrvalue(ectx, (oratext *) "NAME", &pResName) !=
        CLSAGFW_SUCCESS)
    {
        goto done;
    }
    /* Try to read the PATH_NAME attribute */
    if (clsagfw_get_attrvalue(ectx, TEST_PATHNAME_ATTR, &pPathName) !=

```

```

        CLSAGFW_SUCCESS)
    {
        clsagfw_log(ectx, 1, (oratest *)"CHECK action: %s: Could not
            read attribute¥n", pResName);
        goto done;
    }
    clsagfw_log(ectx, 1,
        (oratest *)"Check action arguments: resName = %s,
            pathName = %s", pResName, pPathName);
    /* Check if the file is accessible */
    fp = fopen((char *)pPathName, "r");
    if (!fp)
    {
        /* Could not open file */
        clsagfw_log(ectx, 1, (oratest *)"CHECK action: %s status -
            UNPLANNED_OFFLINE¥n", pResName);
        ret = CLSAGFW_UNPLANNED_OFFLINE;
    }
    else
    {
        /* Opened file successfully */
        clsagfw_log(ectx, 1, (oratest *)"CHECK action: %s status -
            ONLINE¥n", pResName);
        fclose(fp);
        ret = CLSAGFW_ONLINE;
    }
done:
    return ret;
}
/*
 * Initialization of the agent framework and registration of types is
 * done in main.
 */
int main(sb4 argc, oratest **argv)
{
    clsagfw_log(NULL, 1, (oratest *)" *** Agent Framework Demo Agent
        Started *** ¥n");
    /*
     * Initialize the agent framework
     */
    if (clsagfw_init(argc, argv, 0, NULL, 0)
        != CLSAGFW_SUCCESS)
    {
        clsagfw_log(NULL, 1, (oratest *)"Failed to initilize the agent
            framework¥n");
        clsagfw_exit(-1);
    }
    /*
     * Set the exit callback function
     */
    clsagfw_set_exitcb(test_agent_exit);
    /*
     * Add the type definition to the framework
     */
    if (clsagfw_add_type(TEST_TYPE1) != CLSAGFW_SUCCESS)
    {
        clsagfw_log(NULL, 1, (oratest *)"Failed in adding type %s to the
            framework¥n", TEST_TYPE1);
        clsagfw_exit(-1);
    }
    /*
     * Set all entry points for for HOTFILE_TYPE
     */
    clsagfw_set_entrypoint(TEST_TYPE1, type1_start,
        CLSAGFW_ACTION_START);
    clsagfw_set_entrypoint(TEST_TYPE1, type1_stop,
        CLSAGFW_ACTION_STOP);
}

```

```

    clsagfw_set_entrypoint(TEST_TYPE1, type1_check,
                           CLSAGFW_ACTION_CHECK);
    clsagfw_set_entrypoint(TEST_TYPE1, type1_stop,
                           CLSAGFW_ACTION_CLEAN);
    clsagfw_log(NULL, 1, (oratext *)"Added resource type [%s] to the
                           agent framework\n", TEST_TYPE1);
/*
 * All set to go, Start the framework. This function does not
 * return if the framework starts successfully.
 */
clsagfw_startup();
/** NOT REACHED **/
return 0;
}

```

H Oracle Clusterware C Application Program Interface

この付録では、Oracle Clusterware C Application Program Interface(API)について説明します。この付録の内容は次のとおりです。

- [Oracle Clusterwareのプログラミング・インタフェース\(C API\)について](#)
- [対話形式CLSCRS API](#)
- [非対話形式CLSCRS API](#)
- [コマンド評価API](#)
- [サーバーのカテゴリ化API](#)
- [STAT3 API](#)
- [その他のAPI](#)

関連項目:

Oracle Clusterwareを使用したアプリケーションの高可用性の実現の詳細は、[「Oracle Clusterwareを使用したアプリケーションの高可用性の実現」](#)を参照してください。

Oracle Clusterwareのプログラミング・インタフェース(C API)について

この章では、Oracle Clusterware (CLSCRS)のプログラミング・インタフェース(C API)の使用について説明します。

- [概要](#)
- [操作上のノート](#)
- [非推奨のCLSCRS API](#)
- [既存のCLSCRS APIの変更点](#)

概要

CLSCRSは、Oracle ClusterwareのCベースAPIのセットです。CLSCRS APIを使用すると、Oracle Clusterwareによって管理されるエンティティの操作を管理できます。これらのエンティティには、リソース、リソース・タイプおよびサーバー・プールが含まれます。APIを使用してOracle Clusterwareにユーザー・アプリケーションを登録することができるため、クラスタウェアでユーザー・アプリケーションを管理して高可用性を維持できます。アプリケーションが登録されると、そのアプリケーションを管理してステータスを問い合わせることができます。アプリケーションが不要になった場合には、そのアプリケーションを停止して、Oracle Clusterwareから登録解除することができます。

Oracle Clusterwareサービスは、Oracle Clusterwareの一部として実行されるクラスタ・レディ・サービスによって提供されます。CLSCRS APIでは、すべての関数コールで明示的に名前が指定されているコンテキストが使用されます。APIでは、処理レベルまたはスレッド・レベルでの格納は行われません。診断ロギングのコールバックを使用できます。

ノート:



Oracle Clusterware の高可用性 API は、Oracle Database クライアント・インストール・メディアからインストールできます。

操作上のノート

Oracle Clusterwareの構成および操作要件に対するプログラミング・インタフェース(C API)について説明します。

この項には次のトピックが含まれます:

- [コンテキストの初期化および永続性](#)
- [スレッドのサポート](#)
- [CLSCRS APIデータ構造](#)
- [メモリー管理](#)
- [エラーの処理およびトレース](#)
- [コールバック・メカニズム](#)
- [フィルタ](#)
- [スクリプト・エージェントの使用方法](#)
- [ヘルプ・インタフェース](#)

コンテキストの初期化および永続性

CLSCRS APIを使用する場合、最初にclscrsコンテキストを初期化する必要があります。このコンテキストを作成および終了するためのコールは次のとおりです。

- clscrs_init_crs: clscrsコンテキストを初期化します。
- clscrs_term_crs: clscrsコンテキストを終了します。

コンテキストが不要になると、コール元によってそのコンテキストは終了されます。

スレッドのサポート

CLSCRS_FLAG_USETHREADSフラグを指定して初期化した場合、CLSCRS APIによってスレッドが内部的に生成される場合があります。APIの各ファンクションは、コール元のスレッドのコンテキストで実行されます。APIコンテキスト・オブジェクトは、複数のスレッドによって同時に使用できない場合があります。ただし、クライアント側でのスレッド・アフィニティは必要ありません。一度に対応できるのはコンテキストごとに1つのスレッドであるというルールに従って、プロセスは複数のAPIコンテキストを作成し、様々なスレッドで使用する場合があります。

CLSCRS APIデータ構造

次のエンティティはAPIコールに渡され、APIコールからの戻り値が含まれます。

- clscrs_sp: 文字列ペア(sp)には、名前および値の文字列が含まれます。値はNULLの場合もあります。作成されたり破棄されたり、およびその内容を調べることで値を置換することが可能です。文字列ペアは、1つの文字列ペア・リスト(splist)のみのメンバーの場合もあります。
- clscrs_splist: 文字列ペア・リスト(splist)は、様々なコンテキストで使用される0(ゼロ)個以上の文字列ペア

アのリストです。APIで文字列ペアを文字列ペア・リストに追加したり、リストから削除することまたはAPIで文字列ペアを繰り返し適用することができます。

- `clscrs_entity_type`: Oracle Clusterwareエンティティのタイプの識別に使用できる列挙タイプ。

列挙タイプは次のように定義します。

```
/* enum to specify the entity type */
typedef enum
{
    clscrs_entity_res           = 1, /* resource */
    clscrs_entity_restype      = 2, /* resource type */
    clscrs_entity_serverpool   = 3, /* server pool */
    clscrs_entity_server       = 4, /* server */
    clscrs_entity_resinst      = 5, /* resource instances */
    clscrs_entity_config_policy = 6, /* configuration policy */
    clscrs_entity_config_policyset = 7, /* configuration policy
set */
    clscrs_entity_server_category = 8 /* server category */
} clscrs_entity_type;
```

- `clscrs_crsentity`: このデータ構造(`crsentity`)はOracle Clusterwareエンティティを示し、これには、名前、およびOracle Clusterwareエンティティが使用されるコンテキストに適した追加データが含まれます。APIには、Oracle Clusterwareエンティティの属性データが含まれる場合もあれば、操作に関するステータスおよび戻りメッセージが示される場合もあります。シングル・エンティティは、1つの`clscrs_crsentitylist`のみのメンバーの場合もあります。
- `clscrs_crsentitylist`: エンティティ・リスト(`crsentitylist`)は、`clscrs_crsentity`の0個以上のインスタンスが含まれるデータ構造です。APIによって、エンティティ・リストでエンティティを追加したり削除すること、またはエンティティを繰り返し適用することができます。

メモリー管理

CLSCRS APIは、要素およびリストに対して機能します。要素はリストに追加されます。要素およびリストの両方に対するメモリーの割当ておよび解放は、明示的なAPIコールを介して行われます。割り当てられたメモリーの解放は、コール元によって行われます。ただし、要素がリストに追加されている場合は、そのリストのみを破棄する必要があり、リストを破棄することで、その要素も暗黙的に破棄されます。要素がリストに追加されていない場合は、要素を破棄する必要があります。再帰的リストの場合、親リストを破棄すると、それに含まれるリストもすべて破棄されます。`clscrs_sp`および`clscrs_crsentity`要素は、コール元によって破棄される必要があります。これらが`clscrs_splist`または`clscrs_crsentitylist`に属する場合は、リストを破棄すると、`clscrs_sp`および`clscrs_crsentity`のエンティティがそれぞれ破棄されます。

たとえば、リソースが作成されてリソース・リストに追加された場合は、そのリソース・リストのみが破棄され、個々のリソースは破棄されません。また、リソース・リストを破棄すると、個々のリソースのメモリーが解放されます。

メモリーは、APIによって、次のコールを介して割り当てられます。

```
clscrs_sp_create()
clscrs_crsentity_create()
clscrs_serverpool_create()
clscrs_type_create()
clscrs_splist_create()
clscrs_crsentitylist_create()
clscrs_entity_id_create()
```

前述のリストの各コールには、対応する`clscrs_*_destroy()`コールがあります。

エラーの処理およびトレース

対話形式および非対話形式のCLSCRS APIは、それぞれ異なるエラー処理メカニズムを使用します。

非対話形式CLSCRS APIでは、エラー・コードは関数コールの戻り値として戻されます。たとえば：

```
clscrsret clscrs_sp_get_value(clscrs_sp *sp, oratext **value);
```

エラー・コードは、clscrsret値として戻されます。

対話形式CLSCRS APIでは、出力結果は次のように表示されます。

1. 関数コールの戻り値によって、リクエストの概要が出力されます。サーバーでリクエストが受信されたかどうかを示されます。また、正常に完了したか、完全に失敗したか、または一部で失敗したかどうかとも示されます。戻り値が正常であるということは、リクエストの受信および処理が行われ、リクエストされたすべてのエンティティに対する結果が正常であったことを示します。
2. リクエストが行われたエンティティごとに、op_statusリストにプログラムの完了コードが格納されます。値がsuccessではない場合は、特定のオブジェクトに対するリクエストの処理で高度な問題が発生したことを示します。
3. 必要に応じて、APIでは、コールバック・メカニズムを使用して、ローカライズされた判読可能なエラー、警告、ステータス・メッセージを受信するように示される場合があります。コールバックが起動されるたびに、コールバックの起動に関するメッセージ、メッセージ・タイプ(重大度)およびオブジェクトIDが表示されます。

たとえば：

```
CLSCRS_STAT clscrs_register_resource2(clscrs_crsentitylist *in_crsentitylist, uword flags,
                                       clscrs_msgf2 msgf, void *msgarg,
                                       clscrs_crsentitylist *op_status);
```

1. 関数によって、CLSCRS_STATという値のエラー・コードが戻されます。
2. CRSDによって、clscrs_msgf2コールバックを介して、エラー・メッセージ、警告メッセージ、進捗メッセージがクライアントに戻されます。クライアントには、CRSDによって戻されるこれらのメッセージを処理するコールバックが実装されている必要があります。
3. 以前のOracle Clusterwareリリースでは、APIに、op_statusリストの一部としてOracle Clusterwareエンティティの各操作の結果も含まれていました。次のAPIを使用してその情報にアクセスできます。

```
clscrsret clscrs_entity_get_op_status(clscrs_entity *res, CLSCRS_STAT *status,
                                       oratext **msg);
```

status引数には、Oracle ClusterwareエンティティのCRSD操作に関するステータス・コードが含まれます。また、msg引数には、操作の結果に関するCRSDからのメッセージが含まれます。msg引数の各Oracle Clusterwareエンティティに対するCRSD操作の結果は引き続きop_statusリストに含まれますが、msg引数の使用によるエラー・コードおよびメッセージの取得は非推奨となり、新しいエンティティのAPIではサポートされていません。既存の使用(特にリソースでの処理)でのみサポートされます。CRSDによって戻されるメッセージを処理する場合は、コールバック関数を使用してください。

コールバック・メカニズム

対話形式CLSCRS APIでは、クライアントがCRSDによって送信されたエラー・メッセージ、警告メッセージおよび進捗メッセージを処理するために使用するコールバック・メカニズムが提供されます。

コールバック・メカニズムのシグネチャは次のとおりです。

```
typedef void (*clscrs_msgf2)(void *usrp, const oratext *id, const oratext *msg,
                             clscrs_msgtype msgtype);
```

前の構文では:

- usrp: コールのコンテキストが含まれるユーザー指定のポインタ
- id: メッセージに対応するエンティティの識別子
- msg: 出力テキスト
- msgtype: メッセージのタイプ(エラー、警告または進捗)

[例H-1](#)に、コールバック・メカニズムの例を示します。

例H-1 コールバック・メカニズム

```
void myCallback(void *arg, const oratext *pId, const oratext *pMsg,
                clscrs_msgtype msgType)
{
    if (pMsg != NULL)
    {
        cout << pMsg << endl;
    }
}
```

[例H-2](#)に、対話形式APIでコールバック・メカニズムを使用する方法を示します。

例H-2 対話形式APIでのコールバック・メカニズムの使用

```
clscrs_start_resource2(pResIdList, NULL,
                      env, myCallback, NULL,
                      0, pOpStatus);
```

コンテキストの作成時にCLSCRS_FLAG_TRACEフラグを渡すことによって、APIのデバッグ・トレース・メッセージを出力することもできます。コンテキスト作成のシグネチャは次のとおりです。

```
CLSCRS_STAT clscrs_init_crs(clscrs_ctx **ctx, clscrs_msgf2 errf, void *errCtx,
                           ub4 flags);
```

トレース・メッセージを機能させるには、clscrs_init_crs APIで、CLSCRS_FLAG_TRACEフラグおよびclscrs_msgf2コールバック・メカニズムの両方を指定する必要があります。

clscrs_msgf2コールバック・メカニズムのシグネチャは次のとおりです。

```
typedef void (*clscrs_msgf)(void *usrp, const oratext *msg, sword msglen);
```

フィルタ

フィルタを使用して、CLSCRS APIが動作するOracle Clusterwareエンティティを絞り込むことができます。簡易フィルタは、演算子を使用した属性/値ペアです。演算子は、例に示すように空白で囲む必要があります。簡易フィルタは、Boolean演算子を使用した式フィルタと呼ばれる式に組み込むことができます。

サポートされるフィルタの演算子は次のとおりです。

- =
- >
- <
- !=
- co: 次を含む
- st: 次で始まる
- en: 次で終わる
-

サポートされるブール演算子はANDおよびORです。

次にフィルタの例を示します。

- TYPE = type1
- ((TYPE = type1) AND (CHECK_INTERVAL > 50))
- (TYPE = type1) AND ((CHECK_INTERVAL > 30) OR (AUTO_START co never))
- NAME en network.res
- TYPE st ora.db

関連項目:

clscrsx.hファイル([Oracle Clusterware](#)のOracle Clusterware Webページからダウンロード可能)にある clscrs_comparator_enum および clscrs_operator_enum を使用して、前述のコンパレータおよび演算子に対する正しいタイプをAPIコールでそれぞれ取得します。

フィルタは2種類あり、CLSCRSにはこれらのフィルタを作成するAPIのセットがあります。

- 比較フィルタ: 2つの値を比較する簡易フィルタ。たとえば:

```
TYPE = ora.db.type
```

比較フィルタを作成するには、clscrs_compfilter_create APIを使用します。たとえば、(TYPE = ora.db.type) 比較フィルタは次のように作成します。

```
clscrs_compfilter_create(ctx, clscrs_TYPE,
                        clscrs_comparator_eq, (const oratext *) "ora.db.type",
                        &myCompFilter);
```

- 式フィルタ: 比較フィルタまたは式フィルタ、あるいはその両方の組合せで作成されるフィルタ。たとえば:

```
((TYPE = ora.db.type) AND (CHECK_INTERVAL > 50))
```

比較フィルタを作成するには、clscrs_expfilter_create APIを使用します。たとえば、式フィルタは次のように作成します。

```
clscrs_exprfilter_create(myCompFilter1, clscrs_operator_or,
                        myCompFilter2, &myExprFilter);
```

関連項目:

clscrs_compfilter_create および clscrs_expfilter_create APIの使用の詳細は、clscrsx.hファイル([Oracle Clusterware](#)のOracle Clusterware Webページからダウンロード可能)を参照してください。

ノート:



clscrs_compfilter_create および clscrs_expfilter_create API によって割り当てられるメモリは、どちらも clscrs_filter_destroy() がコールされると解放されます。

エンティティ・リストのかわりに、次の対話形式CLSCRS APIでフィルタを使用できます。

- clscrs_start_resource2

- clscrs_stat2
- clscrs_stop_resource2
- clscrs_check_resource2
- clscrs_relocate_resource2
-

[例H-3](#)に、対話形式CLSCRS APIでのフィルタの使用を示します。

例H-3 対話形式CLSCRS APIのフィルタ

```
clscrs_start_resource2(myCompFilter, NULL,
                      env, msgf2, NULL,
                      0, pOpStatus);
```

スクリプト・エージェントの使用方法

スクリプト・エージェントのエントリ・ポイントでCLSCRS APIを使用する場合は、次の点を考慮してください。

1. start、stop、cleanなどの一部のアクションは、リソース・インスタンスでロックされた状態で実行されます。そのため、リソースに対する操作が直接行われるリクエストや、関係性が拡張されることで行われるリクエストをサーバーに発行すると、デッドロックの原因になります。
2. 通常、読取り専用の発行(clscrs_stat2)は初回のチェックでないかぎり安全で、この場合、デッドロックの原因にもなるため、スクリプト・エージェントではサーバー上でのコールバックは実行しないでください。clsagfwAPIを使用して、チェック・エントリ・ポイントを問い合わせます。

関連項目:

スクリプト・エージェントの例は、[「Oracle Clusterwareのリソース・リファレンス」](#)を参照してください。

ヘルプ・インタフェース

clscrsx.hファイル([Oracle Clusterware](#)のOracle Clusterware Webページからダウンロード可能)に、CLSCRS APIの完全なリスト(それぞれの使用方法を含む)、およびcrsapp.cという名前のデモがあります。

非推奨のCLSCRS API

[表H-1](#)に非推奨のCLSCRS APIおよびそれに対応するOracle Clusterwareの代替APIを示します。

表H-1 非推奨のCLSCRS API

非推奨のAPI	代替
clscrs_check_resource	clscrs_check_resource2
clscrs_entity	clscrs_entity_type
clscrs_fail_resource	代替なし
clscrs_msgf	clscrs_msgf2
clscrs_register_resource	clscrs_register_resource2
clscrs_relocate_resource	clscrs_relocate_resource2
clscrs_res_attr_count	clscrs_crsentity_attr_count
clscrs_res_create	clscrs_crsentity_create

非推奨のAPI	代替
clscrs_res_create	clscrs_crntitylist_create
clscrs_res_destroy	clscrs_crntitylist_destroy
clscrs_res_get_attr	clscrs_crntitylist_get_attr
clscrs_res_get_attr_list	clscrs_crntitylist_get_attr_list
clscrs_res_get_name	clscrs_crntitylist_get_name
clscrs_res_get_node_list	clscrs_crntitylist_get_node_list
clscrs_res_get_op_status	clscrs_crntitylist_get_op_status
clscrs_res_get_reslist	clscrs_crntitylist_get_crntitylist
clscrs_res_set_attr	clscrs_crntitylist_set_attr
clscrs_res_set_attr_list	clscrs_crntitylist_set_attr_list
clscrs_res_set_reslist	clscrs_crntitylist_set_crntitylist
clscrs_reslist_append	clscrs_crntitylist_append
clscrs_reslist_count	clscrs_crntitylist_count
clscrs_reslist_delete_res	clscrs_crntitylist_delete_crntity
clscrs_reslist_destroy	clscrs_crntitylist_destroy
clscrs_reslist_find	clscrs_crntitylist_find
clscrs_reslist_first	clscrs_crntitylist_first
clscrs_reslist_next	clscrs_crntitylist_next
clscrs_start_resource	clscrs_start_resource2
clscrs_stat	clscrs_stat2
clscrs_stop_resource	clscrs_stop_resource2
clscrs_unregister_resource	clscrs_unregister_resource2

既存のCLSCRS APIの変更点

Oracleによって、clscrs_stat2 APIに次のフラグが追加されています。

- CLSCRS_STATFLAG_SERVERBYCATEGORY: 特定のサーバー・カテゴリに一致するサーバーを問い合わせるには、このフラグを使用します。
- CLSCRS_STATFLAG_CATEGORYBYSERVER: 特定のサーバーに一致するサーバー・カテゴリを問い合わせるには、このフラグを使用します。

Oracleによって、clscrs_start_resource2およびclscrs_stop_resource2 APIに次のフラグが追加されています。

- CLSCRS_FLAG_T_HA_PREPARE: リソースを処理する外部管理インタフェースを準備するようにOracle Clusterwareデーモン(CRSD)に指定するには、透過的HA (start2/stop2 API)でこのフラグを使用します。コールが戻ってきた際、成功の場合は、インタフェースはリソースを起動または停止し、他方のフラグ (CLSCRS_FLAG_T_HA_FINISH)を使用してコールバックすると想定されます。
- CLSCRS_FLAG_T_HA_FINISH: このフラグは、透過的HAでCLSCRS_FLAG_T_HA_PREPAREとともに使用します。起動または停止アクション終了時のCRSD (start2/stop2 API)への2つ目のコールで、このフラグを使用する必要があります。CRSDはAPIの起動が適用されるインスタンスを認識する必要があるため、クライアントはこのフラグで、リソースではなくリソースのインスタンスを指定する必要があることに注意してください。

- CLSCRS_NO_STATE_RESTORATION: このフラグは、リソースAPIを起動および停止するため、および状態が影響を受けたリソースのリソース状態の復元をスキップするようにCRSDに指定するために使用できます。このフラグを使用しない場合、通常はこの手順が試行されます。

対話形式CLSCRS API

これらのAPIは、コマンドを実行するために、クラスタ・レディ・サービス・デーモン(CRSD)をコールします。

これらのAPIが機能するには、CRSDが起動され、実行されている必要があります。

表H-2 Oracle Clusterwareの対話形式CLSCRS APIのサマリー

C API	説明
clscrs_check_resource2	指定されたリソースのチェック・エントリ・ポイントを実行するように Oracle Clusterware に通知します。
clscrs_get_server_by_category	特定のサーバー・カテゴリに一致するサーバーのリストを取得します。
clscrs_is_crs_admin	ユーザーが Oracle Clusterware 管理者かどうかをチェックします。
clscrs_register_resource2	入力リソース・リストにリソースを登録します。
clscrs_register_servercategory	入力サーバー・カテゴリ・リストにサーバー・カテゴリを登録します。
clscrs_register_serverpool	サーバーの入力リストに使用するサーバー・プールを登録します。
clscrs_register_type	入力リソース・リストにリソース・タイプを登録します。
clscrs_relocate_resource2	リソース識別子のリストを再配置します。
clscrs_relocate_server	サーバーのリストを再配置します。
clscrs_request_action	特定のアクション・セットを実行するように Oracle Clusterware に通知します。
clscrs_restart_resource	名前が指定されたリソース・セットを再起動するように Oracle Clusterware に指定します。
clscrs_start_resource2	名前が指定されたリソース・セットを起動するように Oracle

C API	説明
	Clusterware に通知します。
clscrs_stat2	特定のリソースに関する情報を取得します。
clscrs_stat3	特定のエンティティに関する情報を取得します。
clscrs_stop_resource2	名前が指定されたリソース・セットを停止するように Oracle Clusterware に通知します。
clscrs_stop_resource_in_pools	サーバー・プールの特定のリソースを停止するように Oracle Clusterware に指定します。
clscrs_start_resource_in_pools	サーバー・プールの特定のリソースを起動するように Oracle Clusterware に指定します。
clscrs_unregister_resource2	リソース名の入カリストのリソースを登録解除します。
clscrs_unregister_servercategory	入力サーバー・カテゴリ・リストでサーバー・カテゴリを登録解除します。
clscrs_unregister_serverpool	指定されたサーバー・プールを登録解除します。
clscrs_unregister_type	入カリストのリソース・タイプを登録解除します。
clscrs_whatif_add_server	サーバーを追加した場合に発生する事象をシミュレーションします。
clscrs_whatif_delete_server	サーバーを削除した場合に発生する事象をシミュレーションします。
clscrs_whatif_fail_resource	リソースの障害が発生した場合に発生する事象をシミュレーションします。
clscrs_whatif_register_resource	リソースを登録した場合に発生する事象をシミュレーションします。
clscrs_whatif_register_serverpool	サーバー・プールを登録した場合に発生する事象をシミュレーションします。

C API	説明
clscrs_whatif_relocate_resource	リソースを再配置した場合に発生する事象をシミュレーションします。
clscrs_whatif_relocate_server	サーバーを再配置した場合に発生する事象をシミュレーションします。
clscrs_whatif_set_activepolicy	ポリシーをアクティブ化した場合に発生する事象をシミュレーションします。
clscrs_whatif_start_resource	リソースを起動した場合に発生する事象をシミュレーションします。
clscrs_whatif_stop_resource	リソースを停止した場合に発生する事象をシミュレーションします。
clscrs_whatif_unregister_serverpool	サーバー・プールを登録解除した場合に発生する事象をシミュレーションします。

非対話形式CLSCRS API

コンテキストの初期化、対話形式APIのリクエスト・ペイロードの準備、対話形式APIの処理後の出力などの機能には、非対話形式CLSCRS APIを使用できます。非対話形式CLSCRS APIはCRSDをコールしません。

コールバック・エラーのレポート・メカニズムは、非対話形式CLSCRS APIでは使用できません。clscrs_stat2、clscrs_stat3、およびすべてのclscrs_whatif_* APIを除くすべての対話形式CLSCRS APIでは、このコールバック・メカニズムが使用されます。これらのAPIのクライアントは、コールバック・メカニズムを使用して、エラー、警告および進捗メッセージをCRSDから受信します。

フィルタを使用して、Oracle Clusterwareエンティティのリストを減らすこともできます。また、対話形式APIのフィルタを使用して、Oracle Clusterwareエンティティのリストを減らすこともできます。

スレッド・セーフティ

APIデータ構造の管理のために提供されるルーチンは、複数のスレッドの同じAPIコンテキストで同時に使用できませんが、クライアント側でのスレッド・アフィニティは必要ありません。各インスタンスで個別のAPIが使用されている場合は、このプロセスにより、複数のスレッドでこれらのルーチンが起動する可能性があります。

次の表に、非対話形式CLSCRS APIを示します。

表H-3 非対話形式CLSCRS API

C API	説明
clscrs_action_getentity	アクションのエンティティを戻します。

C API	説明
clscrs_action_getparams	アクションのパラメータのリストを戻します。
clscrs_action_gettype	アクションのタイプを戻します。
clscrs_actionlist_count	アクション・リスト内のアクションの数を計算します。
clscrs_actionlist_create	アクション・リストを作成します。
clscrs_actionlist_destroy	アクション・リストを破棄します。
clscrs_actionlist_first	アクション・リスト内の最初のアクションを戻します。
clscrs_actionlist_next	アクション・リスト内の次のアクションを戻します。
clscrs_actionlist_print	アクション・リストを出力します。
clscrs_actionlist_seqid	アクション・リストの順序 ID を戻します。
clscrs_compfilter_create	2 つの値を比較する簡易フィルタを構成します。
clscrs_crntity_attr_count	エンティティの属性の数をカウントします。
clscrs_crntity_create	新しいエンティティを作成します(メモリーを割り当てます)。
clscrs_crntity_destroy	エンティティを破棄してメモリーを解放します。
clscrs_crntity_get_attr	エンティティ、サーバー・プール、またはサーバー属性の値を取得します。
clscrs_crntity_get_attr_list	エンティティ、リソース・タイプ、サーバー・プール、またはサーバーの属性リストを取得します。
clscrs_crntity_get_crntitylist	エンティティのエンティティ・リストを取得します。
clscrs_crntity_get_name	エンティティの名前を取得します。
clscrs_crntity_get_node_list	現在エンティティをホスティングしているノードのリストを取得します。

C API	説明
<code>clscrs_crsentity_get_op_status</code>	エンティティの操作のステータスを取得します。
<code>clscrs_crsentity_get_registered</code>	エンティティの登録ステータスを取得します。
<code>clscrs_crsentity_set_attr</code>	エンティティおよびサーバー・プールの属性を設定します。
<code>clscrs_crsentity_set_attr_list</code>	エンティティ、リソース・タイプ、サーバー・プール、またはサーバーの属性リストを設定します。
<code>clscrs_crsentity_set_crsentitylist</code>	エンティティのリソース・リストを設定します。
<code>clscrs_crsentitylist_append</code>	エンティティ・リストにエンティティを追加します。
<code>clscrs_crsentitylist_count</code>	エンティティ・リストのエンティティの数をカウントします。
<code>clscrs_crsentitylist_create</code>	エンティティのリストを作成します。
<code>clscrs_crsentitylist_delete_crsentity</code>	特定の名前に一致するエンティティをエンティティ・リストから削除します。
<code>clscrs_crsentitylist_destroy</code>	エンティティ・リストを破棄してメモリーを解放します。
<code>clscrs_crsentitylist_find</code>	特定の名前に一致するエンティティ・リストのエンティティを検索します。
<code>clscrs_crsentitylist_first</code>	エンティティ・リストの最初のエンティティを取得します。
<code>clscrs_crslist_next</code>	エンティティ・リストから現行の次のエンティティを取得します。
<code>clscrs_entity_id_create</code>	リソース、リソース・タイプ、サーバー・グループなどの Oracle Clusterware エンティティを識別するエンティティ識別子を作成します。
<code>clscrs_entity_id_destroy</code>	<code>clscrs_entity_id_create()</code> で作成されたエンティティ識別子に関連付けられたメモリーを解放します。
<code>clscrs_exprfilter_create</code>	比較フィルタまたは式フィルタ(あるいはその両方)で式フィルタを構成します。

C API	説明
<code>clscrs_filter_destroy</code>	フィルタのメモリーを解放します。
<code>clscrs_get_entity_type</code>	指定されたエンティティ識別子に対応するエンティティ・タイプを取得します。
<code>clscrs_get_fixed_attrlist</code>	属性グループ識別子に対応する属性のリストを取得します。
<code>clscrs_get_resource_instance_details</code>	使用されているリソース・インスタンス識別子から、リソース名、カーディナリティ、程度などのリソース・インスタンスの詳細を取得します。
<code>clscrs_getnodename</code>	ノード名を取得します。
<code>clscrs_init_crs</code>	Oracle Clusterware と通信するためにコンテキストを初期化します。
<code>clscrs_sp_get</code>	文字列ペアの名前要素および値要素を取得します。
<code>clscrs_sp_get_value</code>	文字列ペアの値要素を取得します。
<code>clscrs_sp_set</code>	文字列ペアの値の部分を変更します。
<code>clscrs_splist_append</code>	文字列ペア・リスト(splist)に新しい文字列ペア(sp)を追加します。
<code>clscrs_splist_count</code>	文字列ペア・リスト(splist)の文字列ペア(sp)の数をカウントします。
<code>clscrs_splist_create</code>	新しい文字列ペア・リストを作成します。
<code>clscrs_splist_create_and_set</code>	新しい文字列ペア・リスト(splist)を作成し、リストの最初の文字列ペアの名前および値を設定します。
<code>clscrs_splist_delete_sp</code>	文字列ペア・リスト(splist)から文字列ペア(sp)を削除します。
<code>clscrs_splist_destroy</code>	文字列ペア・リスト(splist)のメモリーを解放します。
<code>clscrs_splist_find</code>	文字列ペア・リスト(splist)の文字列ペア(sp)の値を検索

C API	説明
	します。
<code>clscrs_splist_first</code>	文字列ペア・リスト(splist)の最初の文字列ペア(sp)を取得します。
<code>clscrs_splist_next</code>	文字列ペア・リスト(splist)の現行の次の文字列ペア(sp)を取得します。現行の次の文字列ペアは、実質的に文字列ペア・リストの次の文字列ペアです。リスト・イテレータは API 内に格納され、公開されません。
<code>clscrs_splist_replace</code>	文字列ペア・リスト(splist)の文字列ペア(sp)の値を置換します。
<code>clscrs_term_crs</code>	Oracle Clusterware と通信するためにコンテキストを解放します。
<code>clscrs_type_create</code>	新しいリソース・タイプを作成します。
<code>clscrs_type_get_attr</code>	リソース・タイプの属性の値またはプロパティを取得します。
<code>clscrs_type_set_attr</code>	リソース・タイプに属性を追加します。

関連トピック

- [エラーの処理およびトレース](#)
- [コールバック・メカニズム](#)
- [フィルタ](#)

コマンド評価API

コマンド評価APIを使用して、計画済または計画外の仮定上のイベントに対するOracle Clusterwareの反応を予測できます。

Oracle Clusterwareでは、システムでのイベントに応じて、対応アクション計画を生成できます。このアクション計画は、一連のリソース状態の移行またはサーバー・プールの再構成(あるいはその両方)で構成されます。コマンド評価APIは、このアクション計画を外部に公開し、イベントの結果を実際の発生前にクライアントが予測できるようにするメカニズムを備えています。

また、以降の各項に示すAPIにフラグを指定することにより、Oracle Clusterwareがイベントに対するレスポンス・アクション計画を作成するために行った決定の理由を説明する、妥当なコマンド評価情報を取得できます。

コマンド評価レスポンス計画は、次のイベント・カテゴリで使用可能です。

- リソース: 起動、停止、再配置、追加および変更
- サーバー・プール: 追加、削除および変更

- サーバー：追加、削除および再配置
- ポリシー：アクティブ・ポリシーの変更
- サーバー・カテゴリ：変更

妥当なコマンド評価レスポンス計画は、次のイベント・カテゴリで使用可能です。

- サーバー・プール：追加、削除および変更
- サーバー：追加および削除
- ポリシー：アクティブ・ポリシーの変更

Oracle Clusterwareではコマンド評価の出力がアクションのリストとして提供され、各アクションがOracle Clusterwareによって実行される特定のステップを示します。各アクションは`clscrs_action`構造によってカプセル化され、`clscrs_actionlist`構造がアクションの順序全体を表します。また、Oracle Clusterwareでは、それらの構造を作成、調査、反復、および破棄するための関数セット(アクション構造用の`clscrs_action_*`とアクション・リスト用の`clscrs_actionlist_*`)が用意されています。使用 방법은、対応するエンティティ・リストおよびエンティティの関数と同じです。

また、コマンド評価APIでは、予想されるシステム状態を問い合わせる機能もクライアントに提供されています。

`clscrs_querylist`構造で`stat3`書式を使用して問合せを指定し、`clscrs_crsentitylist`構造によって結果が提供されます。使用方法的詳細は、`stat3`の項を参照してください。

Oracle Clusterwareによって提供される各コマンド評価レスポンスには、Oracle Clusterwareの現在の状態を示す順序IDが含まれます。順序IDは、Oracle Clusterwareによって管理される新しいイベントごとに増分されます。Oracle Clusterwareでは、順序IDが変更されないかぎり、提供されたアクション計画が対象イベントに対してそのまま実行されることが保証されます。たとえば、`whatif start resource F00`リクエストに対してOracle Clusterwareから提供されるアクション計画は、順序IDが変更されていないかぎり、実際の`start resource F00`リクエストが送信されたときにOracle Clusterwareによって実行されるアクションと同じです。

[例H-4](#)に、コマンド評価APIの使用方法を示します。

例H-4 コマンド評価APIの使用例

```
boolean          tracectx = TRUE;
oratext          *resid;
clscrs_ctx       *ctx;
clscrs_env       env;
clscrs_splist    *resid_list;
clscrs_action    *cur_actn;
clscrs_actionlist *alist;
clscrs_splist    *params;
// Init crs
clscrs_init_crs(&ctx, (clscrs_msgf)clsuslztrace, &tracectx, (ub4)0);
// Init parameters to the call
clscrs_entity_id_create(ctx, "MYRES", clscrs_entity_res, &resid);
clscrs_splist_create(ctx, &resid_list);
clscrs_splist_append(resid_list, resid, NULL);
clscrs_actionlist_create(ctx, &alist);
// Make call into the what-if API
clscrs_whatif_start_resource(resid_list, nodename, flags, NULL, NULL, alist);
// Process the resulting list of actions
for(clscrs_actionlist_first(alist,&cur_actn);cur_actn;clscrs_actionlist_next(alist,&cur_actn))
{
    params = clscrs_action_getparams(cur_actn);
    switch(clscrs_action_gettype(cur_actn))
    {
```

```

case clscrs_actiontype_resstate:
    // Read params and do something
    break;
case clscrs_actiontype_srvmove:
    // Read params and do something
    break;
case clscrs_actiontype_newgroup:
    // Read params and do something
    break;
case clscrs_actiontype_errorcase:
    // Read params and do something
    break;
}
}
clscrs_actionlist_destroy(alist);
clscrs_splist_destroy(resid_list);
clscrs_term_crs(&ctx);

```

この項では、APIのパラメータを、入力を指定するものと、関数が正常に完了した際に情報を表示するものとに分けて説明します。

clscrs_whatif_set_activepolicy

特定のポリシーをアクティブ化した場合にOracle Clusterwareが実行するアクションを決定します。

構文

```

clscrs_whatif_set_activepolicy(const oratext *name, uword flags,
                               clscrs_querylist *qlist,
                               clscrs_crsentitylist *status,
                               clscrs_actionlist *alist);

```

入力

- name: ポリシーの名前。
- flags: CLSCRS_FLAG_NONE
- CLSCRS_FLAG_WHATIF_VERBOSE
- CLSCRS_FLAG_WHYIF: 妥当なコマンド評価情報を表示します。
- CLSCRS_FLAG_FORCE
- qlist: Oracle Clusterwareエンティティ・ステータスのクライアント問合せを指定します。NULLは、問合せを指定しないことを示します。

出力

- status: 戻されるエンティティ情報が含まれるリスト。
- alist: Oracle Clusterwareによって実行されるアクションが含まれるアクション・リスト。

戻り値

- CLSCRS_STAT_SUCCESS: Oracle Clusterwareによってリクエストが正常に完了されたこと、またはエンティティがフィルタに一致しなかったことを示します。
- CLSCRS_STAT_AUTHORIZATION_FAILURE: 認可が失敗した場合に表示されます。
- CLSCRS_STAT_CONNECTION: 通信エラーが発生した場合に表示されます。
- CLSCRS_STAT_WRONG_ACTIVE_VERSION: Oracle Clusterwareをアップグレードする前にAPIを使用した場合に表示されます。
- CLSCRS_STAT_INTERNAL_ERROR: 予期しない非ユーザー・エラーが発生した場合に表示されます。
- CLSCRS_STAT_INVALID_ARGS: 正しくない引数が指定された場合に表示されます。

clscrs_whatif_fail_resource

特定のリソースで障害が発生した場合にOracle Clusterwareが実行するアクションを決定します。

構文

```
clscrs_whatif_fail_resource(clscrs_splist *name, const oratext *server,  
                           uword flags, clscrs_querylist *qlist,  
                           clscrs_crentitylist *status,  
                           clscrs_actionlist *alist);
```

入力

- name: リソースID、インスタンスIDまたはフィルタ。
- server: リソースの障害が発生するサーバーの名前。NULLが許可されます。
- flags: CLSCRS_FLAG_NONE
- CLSCRS_FLAG_WHATIF_VERBOSE
- qlist: Oracle Clusterwareエンティティ・ステータスのクライアント問合せを指定します。NULLは、問合せを指定しないことを示します。

出力

- status: 戻されるエンティティ情報が含まれるリスト。
- alist: Oracle Clusterwareによって実行されるアクションが含まれるアクション・リスト。

戻り値

- CLSCRS_STAT_SUCCESS: Oracle Clusterwareによってリクエストが正常に完了されたこと、またはエンティティがフィルタに一致しなかったことを示します。
- CLSCRS_STAT_AUTHORIZATION_FAILURE: 認可が失敗した場合に表示されます。
- CLSCRS_STAT_CONNECTION: 通信エラーが発生した場合に表示されます。
- CLSCRS_STAT_WRONG_ACTIVE_VERSION: Oracle Clusterwareをアップグレードする前にAPIを使用した場合に表示されます。
- CLSCRS_STAT_INTERNAL_ERROR: 予期しない非ユーザー・エラーが発生した場合に表示されます。
- CLSCRS_STAT_INVALID_ARGS: 正しくない引数が指定された場合に表示されます。

clscrs_whatif_register_resource

特定のリソースを追加または変更した場合にOracle Clusterwareが実行するアクションを決定します。

構文

```
clscrs_whatif_register_resource(const oratext *name, clscrs_splist *attrs,  
                                uword flags, clscrs_querylist *qlist,  
                                clscrs_crentitylist *status,  
                                clscrs_actionlist *alist);
```

入力

- name: リソースの名前。
- attrs: 指定されたリソースの属性。
- flags: CLSCRS_FLAG_NONE
- CLSCRS_FLAG_WHATIF_VERBOSE
- CLSCRS_FLAG_FORCE
- CLSCRS_FLAG_REG_UPDATE (リソースを変更する場合)

- qlist: Oracle Clusterwareエンティティ・ステータスのクライアント問合せを指定します。NULLは、問合せを指定しないことを示します。

出力

- status: 戻されるエンティティ情報が含まれるリスト。
- alist: Oracle Clusterwareによって実行されるアクションが含まれるアクション・リスト。

戻り値

- CLSCRS_STAT_SUCCESS: Oracle Clusterwareによってリクエストが正常に完了されたこと、またはエンティティがフィルタに一致しなかったことを示します。
- CLSCRS_STAT_AUTHORIZATION_FAILURE: 認可が失敗した場合に表示されます。
- CLSCRS_STAT_CONNECTION: 通信エラーが発生した場合に表示されます。
- CLSCRS_STAT_WRONG_ACTIVE_VERSION: Oracle Clusterwareをアップグレードする前にAPIを使用した場合に表示されます。
- CLSCRS_STAT_INTERNAL_ERROR: 予期しない非ユーザー・エラーが発生した場合に表示されます。
- CLSCRS_STAT_INVALID_ARGS: 正しくない引数が指定された場合に表示されます。

clscrs_whatif_relocate_resource

特定のリソースを再配置した場合にOracle Clusterwareが実行するアクションを決定します。

構文

```
clscrs_whatif_relocate_resource(clscrs_splist *name, const oratext *destnode,
                                uword flags, clscrs_querylist *qlist,
                                clscrs_crsentitylist *status,
                                clscrs_actionlist *alist);
```

入力

- name: リソースID、インスタンスIDまたはフィルタ。
- flags: CLSCRS_FLAG_NONE
- CLSCRS_FLAG_WHATIF_VERBOSE
- CLSCRS_FLAG_FORCE
- qlist: Oracle Clusterwareエンティティ・ステータスのクライアント問合せを指定します。NULLは、問合せを指定しないことを示します。

出力

- status: 戻されるエンティティ情報が含まれるリスト。
- alist: Oracle Clusterwareによって実行されるアクションが含まれるアクション・リスト。

戻り値

- CLSCRS_STAT_SUCCESS: Oracle Clusterwareによってリクエストが正常に完了されたこと、またはエンティティがフィルタに一致しなかったことを示します。
- CLSCRS_STAT_AUTHORIZATION_FAILURE: 認可が失敗した場合に表示されます。
- CLSCRS_STAT_CONNECTION: 通信エラーが発生した場合に表示されます。
- CLSCRS_STAT_WRONG_ACTIVE_VERSION: Oracle Clusterwareをアップグレードする前にAPIを使用した場合に表示されます。
- CLSCRS_STAT_INTERNAL_ERROR: 予期しない非ユーザー・エラーが発生した場合に表示されます。

- CLSCRS_STAT_INVALID_ARGS: 正しくない引数が指定された場合に表示されます。

clscrs_whatif_start_resource

特定のリソースを起動した場合にOracle Clusterwareが実行するアクションを決定します。

構文

```
clscrs_whatif_start_resource(clscrs_splist *name, const oratext *node,
                             uword flags, clscrs_querylist *qlist,
                             clscrs_crsentitylist *status,
                             clscrs_actionlist *alist);
```

入力

- name: リソースID、インスタンスIDまたはフィルタ。
- node: リソースを起動するノードの名前。NULLが許可されます。
- flags: CLSCRS_FLAG_NONE
- CLSCRS_FLAG_WHATIF_VERBOSE
- CLSCRS_FLAG_FORCE
- qlist: Oracle Clusterwareエンティティ・ステータスのクライアント問合せを指定します。NULLは、問合せを指定しないことを示します。

出力

- status: 戻されるエンティティ情報が含まれるリスト。
- alist: Oracle Clusterwareによって実行されるアクションが含まれるアクション・リスト。

戻り値

- CLSCRS_STAT_SUCCESS: Oracle Clusterwareによってリクエストが正常に完了されたこと、またはエンティティがフィルタに一致しなかったことを示します。
- CLSCRS_STAT_AUTHORIZATION_FAILURE: 認可が失敗した場合に表示されます。
- CLSCRS_STAT_CONNECTION: 通信エラーが発生した場合に表示されます。
- CLSCRS_STAT_WRONG_ACTIVE_VERSION: Oracle Clusterwareをアップグレードする前にAPIを使用した場合に表示されます。
- CLSCRS_STAT_INTERNAL_ERROR: 予期しない非ユーザー・エラーが発生した場合に表示されます。
- CLSCRS_STAT_INVALID_ARGS: 正しくない引数が指定された場合に表示されます。

clscrs_whatif_stop_resource

特定のリソースを停止した場合にOracle Clusterwareが実行するアクションを決定します。

構文

```
clscrs_whatif_stop_resource(clscrs_splist *name, uword flags,
                             clscrs_querylist *qlist,
                             clscrs_crsentitylist *status,
                             clscrs_actionlist *alist);
```

入力

- name: リソースID、インスタンスIDまたはフィルタ。
- flags: CLSCRS_FLAG_NONE
- CLSCRS_FLAG_WHATIF_VERBOSE

- CLSCRS_FLAG_FORCE
- qlist: Oracle Clusterwareエンティティ・ステータスのクライアント問合せを指定します。NULLは、問合せを指定しないことを示します。

出力

- status: 戻されるエンティティ情報が含まれるリスト。
- alist: Oracle Clusterwareによって実行されるアクションが含まれるアクション・リスト。

戻り値

- CLSCRS_STAT_SUCCESS: Oracle Clusterwareによってリクエストが正常に完了されたこと、またはエンティティがフィルタに一致しなかったことを示します。
- CLSCRS_STAT_AUTHORIZATION_FAILURE: 認可が失敗した場合に表示されます。
- CLSCRS_STAT_CONNECTION: 通信エラーが発生した場合に表示されます。
- CLSCRS_STAT_WRONG_ACTIVE_VERSION: Oracle Clusterwareをアップグレードする前にAPIを使用した場合に表示されます。
- CLSCRS_STAT_INTERNAL_ERROR: 予期しない非ユーザー・エラーが発生した場合に表示されます。
- CLSCRS_STAT_INVALID_ARGS: 正しくない引数が指定された場合に表示されます。

clscrs_whatif_register_serverpool

特定のサーバー・プールを登録した場合にOracle Clusterwareが実行するアクションを決定します。

構文

```
clscrs_whatif_register_serverpool(const oratext *pname, clscrs_splist *attrs,
                                   uword flags, clscrs_querylist *qlist,
                                   clscrs_crsentitylist *status,
                                   clscrs_actionlist *alist);
```

入力

- name: サーバー・プールの名前。
- attrs: 指定されたサーバー・プールの属性。
- flags: CLSCRS_FLAG_NONE
- CLSCRS_FLAG_WHATIF_VERBOSE
- CLSCRS_FLAG_WHYIF: 妥当なコマンド評価情報を表示します。
- CLSCRS_FLAG_FORCE
- CLSCRS_FLAG_REG_UPDATE (サーバー・プールを変更する場合)
- qlist: Oracle Clusterwareエンティティ・ステータスのクライアント問合せを指定します。NULLは、問合せを指定しないことを示します。

出力

- status: 戻されるエンティティ情報が含まれるリスト。
- alist: Oracle Clusterwareによって実行されるアクションが含まれるアクション・リスト。

戻り値

- CLSCRS_STAT_SUCCESS: Oracle Clusterwareによってリクエストが正常に完了されたこと、またはエンティティがフィルタに一致しなかったことを示します。
- CLSCRS_STAT_AUTHORIZATION_FAILURE: 認可が失敗した場合に表示されます。

- CLSCRS_STAT_CONNECTION: 通信エラーが発生した場合に表示されます。
- CLSCRS_STAT_WRONG_ACTIVE_VERSION: Oracle Clusterwareをアップグレードする前にAPIを使用した場合に表示されます。
- CLSCRS_STAT_INTERNAL_ERROR: 予期しない非ユーザー・エラーが発生した場合に表示されます。
- CLSCRS_STAT_INVALID_ARGS: 正しくない引数が指定された場合に表示されます。

clscrs_whatif_unregister_serverpool

特定のサーバー・プールを登録解除した場合にOracle Clusterwareが実行するアクションを決定します。

構文

```
clscrs_whatif_unregister_serverpool(const oratext *poolname, uword flags,
                                     clscrs_querylist *qlist,
                                     clscrs_crntitylist *status,
                                     clscrs_actionlist *alist);
```

入力

- name: サーバー・プールの名前。
- flags: CLSCRS_FLAG_NONE
- CLSCRS_FLAG_WHATIF_VERBOSE
- CLSCRS_FLAG_WHYYIF: 妥当なコマンド評価情報を表示します。
- CLSCRS_FLAG_FORCE
- qlist: Oracle Clusterwareエンティティ・ステータスのクライアント問合せを指定します。NULLは、問合せを指定しないことを示します。

出力

- status: 戻されるエンティティ情報が含まれるリスト。
- alist: Oracle Clusterwareによって実行されるアクションが含まれるアクション・リスト。

戻り値

- CLSCRS_STAT_SUCCESS: Oracle Clusterwareによってリクエストが正常に完了されたこと、またはエンティティがフィルタに一致しなかったことを示します。
- CLSCRS_STAT_AUTHORIZATION_FAILURE: 認可が失敗した場合に表示されます。
- CLSCRS_STAT_CONNECTION: 通信エラーが発生した場合に表示されます。
- CLSCRS_STAT_WRONG_ACTIVE_VERSION: Oracle Clusterwareをアップグレードする前にAPIを使用した場合に表示されます。
- CLSCRS_STAT_INTERNAL_ERROR: 予期しない非ユーザー・エラーが発生した場合に表示されます。
- CLSCRS_STAT_INVALID_ARGS: 正しくない引数が指定された場合に表示されます。

clscrs_whatif_add_server

サーバーを追加した場合にOracle Clusterwareが実行するアクションを決定します。

構文

```
clscrs_whatif_add_server(const oratext *name, clscrs_splist *attrs, uword flags,
                          clscrs_querylist *qlist, clscrs_crntitylist *status,
                          clscrs_actionlist *alist);
```

入力

- name: サーバーの名前。
- attrs: 指定されたサーバーの属性。
- flags: CLSCRS_FLAG_NONE
- CLSCRS_FLAG_WHATIF_VERBOSE
- CLSCRS_FLAG_WHYIF: 妥当なコマンド評価情報を表示します。
- CLSCRS_FLAG_FORCE
- qlist: Oracle Clusterwareエンティティ・ステータスのクライアント問合せを指定します。NULLは、問合せを指定しないことを示します。

出力

- status: 戻されるエンティティ情報が含まれるリスト。
- alist: Oracle Clusterwareによって実行されるアクションが含まれるアクション・リスト。

戻り値

- CLSCRS_STAT_SUCCESS: Oracle Clusterwareによってリクエストが正常に完了されたこと、またはエンティティがフィルタに一致しなかったことを示します。
- CLSCRS_STAT_AUTHORIZATION_FAILURE: 認可が失敗した場合に表示されます。
- CLSCRS_STAT_CONNECTION: 通信エラーが発生した場合に表示されます。
- CLSCRS_STAT_WRONG_ACTIVE_VERSION: Oracle Clusterwareをアップグレードする前にAPIを使用した場合に表示されます。
- CLSCRS_STAT_INTERNAL_ERROR: 予期しない非ユーザー・エラーが発生した場合に表示されます。
- CLSCRS_STAT_INVALID_ARGS: 正しくない引数が指定された場合に表示されます。

clscrs_whatif_delete_server

サーバーを削除した場合にOracle Clusterwareが実行するアクションを決定します。

構文

```
clscrs_whatif_delete_server(const oratext *name, uword flags,
                             clscrs_querylist *qlist,
                             clscrs_crsentitylist *status,
                             clscrs_actionlist *alist);
```

入力

- name: サーバーの名前。
- flags: CLSCRS_FLAG_NONE
- CLSCRS_FLAG_WHATIF_VERBOSE
- CLSCRS_FLAG_WHYIF: 妥当なコマンド評価情報を表示します。
- CLSCRS_FLAG_FORCE
- qlist: Oracle Clusterwareエンティティ・ステータスのクライアント問合せを指定します。NULLは、問合せを指定しないことを示します。

出力

- status: 戻されるエンティティ情報が含まれるリスト。
- alist: Oracle Clusterwareによって実行されるアクションが含まれるアクション・リスト。

戻り値

- CLSCRS_STAT_SUCCESS: Oracle Clusterwareによってリクエストが正常に完了されたこと、またはエンティティが

フィルタに一致しなかったことを示します。

- CLSCRS_STAT_AUTHORIZATION_FAILURE: 認可が失敗した場合に表示されます。
- CLSCRS_STAT_CONNECTION: 通信エラーが発生した場合に表示されます。
- CLSCRS_STAT_WRONG_ACTIVE_VERSION: Oracle Clusterwareをアップグレードする前にAPIを使用した場合に表示されます。
- CLSCRS_STAT_INTERNAL_ERROR: 予期しない非ユーザー・エラーが発生した場合に表示されます。
- CLSCRS_STAT_INVALID_ARGS: 正しくない引数が指定された場合に表示されます。

clscrs_whatif_relocate_server

サーバーを異なるサーバー・プールに再配置した場合にOracle Clusterwareが実行するアクションを決定します。

構文

```
clscrs_whatif_relocate_server(const oratext *name, const oratext *topool,  
                               uword flags, clscrs_querylist *qlist,  
                               clscrs_crentitylist *status,  
                               clscrs_actionlist *alist);
```

入力

- name: サーバーの名前。
- topool: サーバーの再配置先のサーバー・プールの名前。
- flags: CLSCRS_FLAG_NONE
- CLSCRS_FLAG_WHATIF_VERBOSE
- CLSCRS_FLAG_FORCE
- qlist: Oracle Clusterwareエンティティ・ステータスのクライアント問合せを指定します。NULLは、問合せを指定しないことを示します。

出力

- status: 戻されるエンティティ情報が含まれるリスト。
- alist: Oracle Clusterwareによって実行されるアクションが含まれるアクション・リスト。

戻り値

- CLSCRS_STAT_SUCCESS: Oracle Clusterwareによってリクエストが正常に完了されたこと、またはエンティティがフィルタに一致しなかったことを示します。
- CLSCRS_STAT_AUTHORIZATION_FAILURE: 認可が失敗した場合に表示されます。
- CLSCRS_STAT_CONNECTION: 通信エラーが発生した場合に表示されます。
- CLSCRS_STAT_WRONG_ACTIVE_VERSION: Oracle Clusterwareをアップグレードする前にAPIを使用した場合に表示されます。
- CLSCRS_STAT_INTERNAL_ERROR: 予期しない非ユーザー・エラーが発生した場合に表示されます。
- CLSCRS_STAT_INVALID_ARGS: 正しくない引数が指定された場合に表示されます。

サーバーのカテゴリ化API

Oracle Clusterwareは、clsrcs_entity_server_categoryというエンティティを備えています。

この項では、APIのパラメータを、入力を指定するものと、関数が正常に完了した際に情報を表示するものとに分けて説明します。

clscrs_servercategory_create

サーバー・カテゴリを作成します。

構文

```
clscrs_servercategory_create(clscrs_ctx *ctx, const oratext *sc_name,  
                             clscrs_crsentity **sc);
```

入力

- ctx: CLSCRSコンテキスト。
- sc_name: サーバー・カテゴリの名前。
-

出力

sc: 新しく作成したサーバー・カテゴリ。

戻り値

- clscrsretSUCC: Oracle Clusterwareがリクエストを正常に完了したことを示します。
- clscrsretNOMEM: メモリーを割り当てることができない場合に表示されます。
- clscrsretBADCTX: コンテキストがNULLの場合に表示されます。
- clscrsretBADARG: サーバー名がNULLの場合に表示されます。

clscrs_servercategory_destroy

サーバー・カテゴリのメモリーを解放します。

構文

```
clscrs_servercategory_destroy(clscrs_crsentity **sc);
```

入力

sc_name: メモリーを解放するために破棄するサーバー・カテゴリの名前。

戻り値

clscrsretSUCC: Oracle Clusterwareがリクエストを正常に完了したことを示します。

clscrs_register_servercategory

指定したサーバー・カテゴリを入力サーバー・カテゴリ・リストに登録します。

構文

```
clscrs_register_servercategory(clscrs_crsentitylist *in_entitylist, uword flags,  
                               clscrs_msgf2 msgf, void *msgarg,  
                               clscrs_crsentitylist *op_status);
```

入力

- in_entitylist: 登録するサーバー・カテゴリのリスト。
- flags: CLSCRS_FLAG_REG_UPDATE
- CLSCRS_FLAG_QUEUE
- CLSCRS_FLAG_FORCE

- CLSCRS_FLAG_NONE
- msgf: ユーザー・メッセージ・コールバック(NULLにできます)。
- msgarg: ユーザー・コールバック引数(NULLにできます)。

出力

op_status: 各サーバー・カテゴリの登録操作のステータスを保持するエンティティ・リスト。

戻り値

- CLSCRS_STAT_SUCCESS: すべての入力サーバー・カテゴリが正常に登録されていることを示します。
- CLSCRS_STAT_FAILURE: 1つ以上のサーバー・カテゴリに登録できない場合に表示されます。
- CLSCRS_STAT_CONNECTION: 通信エラーが発生した場合に表示されます。
- CLSCRS_STAT_WRONG_ACTIVE_VERSION: Oracle Clusterwareをアップグレードする前にAPIを使用した場合に表示されます。
- CLSCRS_STAT_INVALID_ARGS: 入力エンティティ・リストのいずれかのサーバー・カテゴリに属性がない場合に表示されます。

使用上のノート

- サーバー・カテゴリの属性は、入力サーバー・カテゴリ・リストに含まれます。
- op_statusリストには、各サーバー・カテゴリの登録操作の結果が含まれ、有効な属性は含まれません。
- コール元は、in_entitylistを作成して移入する必要があり、op_statusリストを作成する必要があります。これらのリストは両方とも、コール元によって破棄される必要があります。
- op_statusリストは別のAPIコールで再利用できません。これは、APIコールごとに作成して破棄する必要があります。
- CLSCRS_FLAG_REG_UPDATEフラグを渡すことで、すでに登録済のサーバー・カテゴリの1つ以上の属性を変更できます。
- フラグは、入力エンティティ・リストのすべてのサーバー・カテゴリに適用されます。

clscrs_unregister_servercategory

指定したサーバー・カテゴリを入力リストから登録解除します。

構文

```
clscrs_unregister_servercategory(clscrs_splist *sclist, uword flags,
                                clscrs_msgf2 msgf, void *msgarg,
                                clscrs_crntitylist *op_status);
```

入力

- sclist: 登録を解除するサーバー・カテゴリのリスト。
- flags: オプション・フラグを指定します。
- msgf: ユーザー・メッセージ・コールバック(NULLにできます)。
- msgarg: ユーザー・コールバック引数(NULLにできます)。

出力

op_status: 各サーバー・カテゴリの登録解除操作のステータスを保持するエンティティ・リスト。

戻り値

- CLSCRS_STAT_SUCCESS: すべての入力サーバー・カテゴリが正常に登録されていることを示します。
- CLSCRS_STAT_FAILURE: 1つ以上のサーバー・カテゴリを登録解除できない場合に 표시됩니다。
- CLSCRS_STAT_CONNECTION: 通信エラーが発生した場合に 표시됩니다。
- CLSCRS_STAT_WRONG_ACTIVE_VERSION: Oracle Clusterwareをアップグレードする前にAPIを使用した場合に 표시됩니다。

使用上のノート

- op_statusリストには、各サーバー・カテゴリの登録解除操作の結果が含まれます。
- コール元は、sc_listを作成して移入する必要があり、op_statusリストを作成する必要があります。これらのリストは両方とも、コール元によって破棄される必要があります。
- op_statusリストは別のAPIコールで再利用できず、APIコールごとに作成して破棄する必要があります。

clscrs_get_server_by_category

特定のサーバー・カテゴリに一致するサーバーのリストを取得します。

構文

```
clscrs_get_server_by_category(clscrs_splist *in_list,
                             clscrs_crsentitylist *out_entitylist);
```

入力

in_list: サーバー・カテゴリのリストまたはフィルタ。

出力

out_entitylist: サーバー・カテゴリに一致しているサーバーをリストします。

戻り値

- CLSCRS_STAT_SUCCESS: すべての入力サーバー・カテゴリが正常に登録されていることを示します。
- CLSCRS_STAT_FAILURE: 1つ以上のサーバー・カテゴリを登録解除できない場合に 표시됩니다。
- CLSCRS_STAT_CONNECTION: 通信エラーが発生した場合に 표시됩니다。

clscrs_register_server

サーバー属性を変更します。

構文

```
clscrs_register_server(clscrs_crsentitylist *in_entitylist, uword flags,
                      clscrs_msgf2 msgf, void *msgarg,
                      clscrs_crsentitylist *op_status);
```

入力

- in_entitylist: 登録するサーバー・カテゴリのリスト。
- flags: CLSCRS_FLAG_QUEUE
- CLSCRS_FLAG_FORCE
- CLSCRS_FLAG_REG_UPDATE
- CLSCRS_FLAG_NONE
- msgf: ユーザー・メッセージ・コールバック(NULLにできます)。
- msgarg: ユーザー・コールバック引数(NULLにできます)。

出力

op_status: 各サーバーの登録操作のステータスを保持するエンティティ・リスト。

戻り値

- CLSCRS_STAT_SUCCESS: すべての入力サーバー・カテゴリが正常に登録されていることを示します。
- CLSCRS_STAT_FAILURE: 1つ以上のサーバー・カテゴリを登録できない場合に表示されます。
- CLSCRS_STAT_CONNECTION: 通信エラーが発生した場合に表示されます。
- CLSCRS_STAT_WRONG_ACTIVE_VERSION: Oracle Clusterwareをアップグレードする前にAPIを使用した場合に表示されます。
- CLSCRS_STAT_INVALID_ARGS: 入力エンティティ・リストのいずれかのサーバー・カテゴリに属性がない場合に表示されます。

使用上のノート

- サーバーの属性は、入力サーバー・リストに含まれます。
- op_statusリストには、各サーバーの変更操作の結果が含まれますが、有効な属性は含まれません。
- コール元は、in_entitylistを作成して移入する必要があり、op_statusリストを作成する必要があります。これらのリストは両方とも、コール元によって破棄される必要があります。
- op_statusリストは別のAPIコールで再利用できず、APIコールごとに作成して破棄する必要があります。

ノート:



現在、Oracle ではサポートされているのみであるため、CLSCRS_FLAG_REG_UPDATE フラグは常に渡される必要があります。フラグは、入力エンティティ・リストのすべてのサーバーに適用されます。

(オプション)ここで、APIまたはサブプログラムの使用に関する詳細情報を提供します。

STAT3 API

Oracle Clusterwareでは、リソースやサーバー・プールなどのいくつかのエンティティが管理されます。ただし、clscrs_stat2 インタフェースでは、タイプによるエンティティの取得(読取り)のみが許可されます。つまり、1回の取得で1つのタイプのエンティティのみを返すことができます。そのため、異なるタイプのエンティティを取得する必要があったり、さらに決定を下すためにデータ構造の一貫したビューが必要なクライアントは、特殊なイベント順序IDを使用した回避策に頼ることになり、必要に応じて問合せリクエストを複数回にわたって再発行する必要があります。理論上、ソリューションなど、状態/構成が常に変化するシステムは、時間による制限を受けません。

Oracle Clusterware 12c以降のリリースでは、複数の種類のエンティティの一貫した読取りを実行するメカニズムを備えています。このメカニズムはエンティティに対して透過的に機能するため、新しい管理対象エンティティを追加してもメカニズムを変更する必要はありません。

このために使用されるのが、clscrs_stat3 APIです。

clscrs_stat3

qlistで特定されたOracle Clusterwareエンティティに関する情報を取得します。

構文

```
clscrs_stat3(clscrs_querylist *qlist,  
            uword flags,  
            clscrs_crsentitylist *out_entitylist);
```

入力

qlist: 問合せるOracle Clusterwareエンティティのリスト。

出力

out_entitylist: 戻されたエンティティ情報を保持するエンティティ・リスト。

戻り値

- CLSCRS_STAT_SUCCESS: APIがOracle Clusterwareに正常に問合せを実行したことを示します。
- CLSCRS_STAT_FAILURE: Oracle Clusterwareへの問合せでエラーが発生した場合に表示されます。
- CLSCRS_STAT_CONNECTION: 通信エラーが発生した場合に表示されます。

使用上のノート

- 情報はout_entitylistに戻されます。問合せは、その後の出力セットが一貫性を持つように実行されます。
- clscrs_queryオブジェクトを作成し、それらをclscrs_querylistオブジェクトに追加してから、そのオブジェクトをこの関数に渡します。各エンティティ・タイプに必要な属性は、clscrs_queryオブジェクトに含まれている必要があります。属性リストが空の場合、Oracle Clusterwareによってすべての属性が戻されます。
- out_entitylistは空のリストとして作成して渡す必要があります。エンティティのエラーはout_entitylistに戻されます。

- 出力はネストされたエンティティ・リストとして戻されます。個々の問合せの結果は、out_entitylistの各エンティティとして戻されます。個々の問合せの出力は、その特定のエンティティのエンティティ・リストとして戻されます。

out_entitylistに含まれるエンティティのOracle Clusterwareエンティティ・タイプを取得するためにclscrs_crsentity_get_typeをコールすることで、エンティティの結果のタイプを確認できます。

たとえば、一方はリソース、もう一方はリソース・タイプに対する2つの問合せがある場合、out_entitylistには前述のOracle Clusterwareエンティティ・タイプそれぞれに対して1つずつ、2つのエンティティ・オブジェクトが含まれます。それらの各エンティティ・オブジェクトのエンティティ・リストには、問合せの結果が含まれます。特定のオブジェクトがどのようなタイプのエンティティ・オブジェクトかを判断するには、そのエンティティ・オブジェクトに対して

clscrs_crsentity_get_type関数をコールする必要があります。問合せで0個の一致が戻された場合、そのエンティティ・オブジェクトのエンティティ・リストのサイズはゼロになります。

ノート:



Oracle では、エンティティ・タイプごとに 1 つの clscrs_query オブジェクトのみがサポートされています。同じエンティティ・タイプに対して複数の clscrs_query オブジェクトが作成された場合、Oracle では stat3 API の動作は保証されません。

その他のAPI

この項では、APIのパラメータを、入力を指定するものと、関数が正常に完了した際に情報を表示するものとに分けて説明します。

clscrs_get_error_details

他のCLSCRS APIの起動中に何らかの障害が発生した場合に、clsk例外スタックを戻します。

構文

```
clscrs_get_error_details(oratext* error_buf, size_t* buf_size);
```

入力

- error_buf: エラー・スタックが移入されるバッファ。
- buf_size: error_bufのバッファのサイズ。特定のバッファのサイズが必要なものよりも小さい場合、APIによって必要なバッファ・サイズの値が戻されます。

戻り値

- clscrsretSUCC: エラー・スタックが正常に出力されたことを示します。
- clscrsretEMPTY: エラー・スタックが空の場合に表示されます。
- clscrsretBADARG: error_bufかbuf_sizeのいずれかがNULLの場合に表示されます。
- clscrsretBUFFSMALL: 指定したバッファ・サイズが必要なサイズよりも小さい場合に表示されます。

使用上のノート

コール元がerror_bufのメモリーを割り当てます。

clscrs_request_action

名前が指定されたリソース・セットに対してアクションを実行するようにOracle Clusterwareに指定します。

構文

```
clscrs_request_action(oratext *action_name, clscrs_splist *ridlist,  
                     clscrs_env env, clscrs_msgf2 msgf,  
                     void *msgarg, uword flag, clscrs_reslist *op_status);
```

入力

- action_name: 実行されるアクションの名前。
- ridlist: 停止するリソースまたはリソース・インスタンスのIDのリスト、あるいはフィルタ。
- env: 停止操作に環境引数を指定します。
- msgf: ユーザー・メッセージ・コールバック(NULLにできます)。
- msgarg: ユーザー・コールバック引数(NULLにできます)。
- flag: asyncまたはqueueオプションのいずれか。

出力

op_status: 各リソースのアクション操作のステータスを保持するリソース・リスト。

戻り値

- CLSCRS_STAT_INVALID_RESNAME: ridlistが空の場合に表示されます。
- CLSCRS_STAT_AUTHORIZATION_FAILURE: 認可が失敗した場合に表示されます。
- CLSCRS_STAT_SUCCESS: リクエストしたすべてのエンティティに対してリクエストが正常に完了した場合またはフィルタに一致するエンティティがない場合に表示されます。
- CLSCRS_STAT_FAILURE: 1つ以上のリソースまたはリソースIDが正常に停止しない場合に表示されます。

- CLSCRS_STAT_CONNECTION: 通信エラーが発生した場合に表示されます。

使用上のノート

- 1つ目の引数には、リソースIDのリストまたはフィルタのいずれかを指定できます。
- フィルタを指定すると、1回のコールで問合せとアクションを実行できます。フィルタによってすべての登録済リソースが検索されます。

clscrs_restart_resource

名前が指定されたリソース・セットを再起動するようにOracle Clusterwareに指定します。

構文

```
clscrs_restart_resource(clscrs_splist *ridlist,
                        clscrs_env env, clscrs_msgf2 msgf, void *msgarg,
                        uword flag, clscrs_reslist *op_status);
```

入力

- ridlist: 再起動するリソースまたはリソース・インスタンスのIDのリスト、あるいはフィルタ。
- env: 再起動操作に環境引数を指定します。
- msgf: ユーザー・メッセージ・コールバック(NULLにできます)。
- msgarg: ユーザー・コールバック引数(NULLにできます)。
- flag: async、forceまたはeventオプションのいずれか。

出力

op_status: 各リソースの再起動操作のステータスを保持するリソース・リスト。

戻り値

- CLSCRS_STAT_INVALID_RESNAME: ridlistが空の場合に表示されます。
- CLSCRS_STAT_AUTHORIZATION_FAILURE: 認可が失敗した場合に表示されます。
- CLSCRS_STAT_SUCCESS: リクエストしたすべてのエンティティに対してリクエストが正常に完了した場合またはフィルタに一致するエンティティがない場合に表示されます。
- CLSCRS_STAT_FAILURE: 1つ以上のリソースまたはリソースIDが正常に起動しない場合に表示されます。
- CLSCRS_STAT_CONNECTION: 通信エラーが発生した場合に表示されます。
- CLSCRS_STAT_WRONG_ACTIVE_VERSION: Oracle Clusterwareをアップグレードする前にAPIを使用した場合に表示されます。

使用上のノート

- フラグがasyncの場合、指定したmsgfコールバック関数はコールされません。このAPIでは、Oracle Clusterwareへのコール開始後にOKステータスが返され、非同期に再起動が実行されます。
- フラグがasync以外で、msgfがNULL以外の場合、APIでは再起動プログラムから収集された出力によってmsgfが一度に1行ずつ処理されます。オプションのeventフラグを渡すことで、これがアクションを実行するリクエストではなく、アクションがすでに開始されていることを通知するものであると示すことができます。このフラグは、共通管理対象リソースの限定的なセットのみに対して使用することをお勧めします。
- 1つ目の引数には、リソースIDのリストまたはフィルタのいずれかを指定できます。フィルタを指定すると、1回のコールで問合せとアクションを実行できます。フィルタによってすべての登録済リソースが検索されます。

clscrs_start_resource_in_pools

サーバー・プール内の名前が指定されたリソース・セットを起動するようにOracle Clusterwareに指定します。

構文

```
clscrs_start_resource_in_pools(clscrs_splist *ridlist, clscrs_splist *spoollist,  
                               clscrs_env env, clscrs_msgf2 msgf2, void *msgarg,  
                               uword flags, clscrs_reslist *op_status);
```

入力

- ridlist: 起動するリソースまたはリソース・インスタンスのIDのリスト、あるいはフィルタ。
- spoollist: 起動するリソースが実行されているサーバー・プール名のリスト、またはフィルタ。
- env: 起動操作に環境引数を指定します。
- msgf: ユーザー・メッセージ・コールバック(NULLにできます)。
- msgarg: ユーザー・コールバック引数(NULLにできます)。
- flag: async、forceまたはeventオプションのいずれか。

出力

op_status: 各リソースの起動操作のステータスを保持するリソース・リスト。

戻り値

- CLSCRS_STAT_INVALID_RESNAME: ridlistが空の場合に表示されます。
- CLSCRS_STAT_AUTHORIZATION_FAILURE: 認可が失敗した場合に表示されます。
- CLSCRS_STAT_SUCCESS: リクエストしたすべてのエンティティに対してリクエストが正常に完了した場合またはフィルタに一致するエンティティがない場合に表示されます。
- CLSCRS_STAT_FAILURE: 1つ以上のリソースまたはリソースIDが正常に起動しない場合に表示されます。
- CLSCRS_STAT_CONNECTION: 通信エラーが発生した場合に表示されます。

使用上のノート

- clscrs_start_resource2と似た機能ですが、この関数は追加の引数spoollist (サーバー・プールのリストが生成される基になるサーバー・プール名のリストまたはフィルタ)を取る点が異なります。
- この関数は、引数としてnodeを取りません。

clscrs_stop_resource_in_pools

サーバー・プール内の名前が指定されたリソース・セットを停止するようにOracle Clusterwareに指定します。

構文

```
clscrs_stop_resource_in_pools(clscrs_splist *ridlist, clscrs_splist *spoollist,  
                              clscrs_env env, clscrs_msgf2 msgf2, void *msgarg,  
                              uword flags, clscrs_reslist *op_status);
```

入力

- ridlist: 停止するリソースまたはリソース・インスタンスのIDのリスト、あるいはフィルタ。
- spoollist: 停止するリソースが実行されているサーバー・プール名のリスト、またはフィルタ。
- env: 停止操作に環境引数を指定します。
- msgf: ユーザー・メッセージ・コールバック(NULLにできます)。

- msgarg: ユーザー・コールバック引数(NULLにできます)。
- flag: async、forceまたはeventオプションのいずれか。

出力

op_status: 各リソースの停止操作のステータスを保持するリソース・リスト。

戻り値

- CLSCRS_STAT_INVALID_RESNAME: ridlistが空の場合に表示されます。
- CLSCRS_STAT_AUTHORIZATION_FAILURE: 認可が失敗した場合に表示されます。
- CLSCRS_STAT_SUCCESS: リクエストしたすべてのエンティティに対してリクエストが正常に完了した場合またはフィルタに一致するエンティティがない場合に表示されます。
- CLSCRS_STAT_FAILURE: 1つ以上のリソースまたはリソースIDが正常に停止しない場合に表示されます。
- CLSCRS_STAT_CONNECTION: 通信エラーが発生した場合に表示されます。

使用上のノート

- clscrs_stop_resource2と似た機能ですが、この関数は追加の引数spoollist (サーバー・プールのリストが生成される基になるサーバー・プール名のリストまたはフィルタ)を取る点が異なります。
- この関数は、引数としてnodeを取りません。

I Oracle Cluster Registryユーティリティ・リファレンス

この付録はOracle Cluster Registry(OCR)構成ユーティリティOCRCONFIGの構文およびOCRのトラブルシューティングについて説明します。OCRCHECKとOCRDUMPに関する説明および使用情報も含まれています。

この付録の内容は次のとおりです。

- [OCRCONFIGについて](#)
- [OCRCONFIGコマンド・リファレンス](#)
- [Oracle Cluster Registryのトラブルシューティングおよび診断出力](#)

OCRCONFIGについて

ocrconfigコマンドを使用して、OCRを管理します。このユーティリティを使用すると、OCRのインポート、エクスポート、追加、削除、リストア、上書き、バックアップ、修復、置換、移動、アップグレードまたはダウングレードを行うことができます。

使用方法

- OCRCONFIG実行可能ファイルは、Grid_home/binディレクトリに格納されています。
- ocrconfig コマンドの構文は次のとおりです。

```
ocrconfig -option
```

ユーティリティ・ヘルプの使用

OCRCONFIGユーティリティのヘルプ出力を表示するには、次のコマンドを使用します。

```
ocrconfig -help
```

権限およびセキュリティ

OCRCONFIGユーティリティを使用するには、管理権限を持つユーザーとしてオペレーティング・システムにログインする必要があります。

ログ・ファイル

OCRCONFIGユーティリティでは、ログ・ファイルがORACLE_BASE/diag/crs/host_name/crs内に作成されます。

ロギングの量を変更するには、ORACLE_BASE/crsdata/host_name/crsdiag/ocrconfig.iniファイルでパスを編集します。

OCRCONFIGコマンド・リファレンス

この項では、次のOCRCONFIGコマンドについて説明します。

- [ocrconfig -add](#)
- [ocrconfig -backuploc](#)
- [ocrconfig -copy](#)
- [ocrconfig -delete](#)
- [ocrconfig -downgrade](#)

- [ocrconfig -export](#)
- [ocrconfig -import](#)
- [ocrconfig -manualbackup](#)
- [ocrconfig -overwrite](#)
- [ocrconfig -repair](#)
- [ocrconfig -replace](#)
- [ocrconfig -restore](#)
- [ocrconfig -showbackup](#)
- [ocrconfig -showbackuploc](#)
- [ocrconfig -upgrade](#)

ocrconfig -add

Oracle Automatic Storage Management (Oracle ASM)ディスク・グループにOCRの場所を追加します。

構文

```
ocrconfig -add +diskgroup_name
```

使用上のノート

- 追加するOCRの場所が存在し、十分な権限を所有している必要があります。また、追加する前に、この場所をマウントしておく必要があります。
- このコマンドは、rootで実行する必要があります。
- diskgroup_name変数は、Oracle ASMディスク・グループの名前です。たとえば：
 - +newdg: ディスク・グループが存在し、マウントされていることを確認します。
 ディスク・グループの名前の前に、プラス記号(+)を付ける必要があります。

関連項目:

- OCRの場所の作成、および適切な権限の設定の詳細は、『[Oracle Grid Infrastructureインストールおよびアップグレード・ガイドfor Linux](#)』を参照してください
- Oracle ASMディスク・グループ管理の詳細は、『[Oracle Automatic Storage Management管理者ガイド](#)』を参照してください。

例

dataというOracle ASMのデフォルトの場所にOCRの場所を追加するには、次のコマンドを使用します。

```
# ocrconfig -add +data
```

ocrconfig -backuploc

OCRバックアップ・ディレクトリの場所を指定します。

構文

```
ocrconfig -backuploc +disk_group_name
ocrconfig -local -backuploc dir_name
```

使用上のノート

- このコマンドは、rootで実行する必要があります。
- -localオプションを使用して、OLRバックアップ・ディレクトリの場所を指定します。
- OCRバックアップの場所のdisk_group_name変数は、Oracle ASMディスク・グループにする必要があります。たとえば:

+bkupdg: Oracle ASMディスク・グループが存在し、すべてのノードにマウントされていることを確認します

Oracle ASMディスク・グループの名前の前に、プラス記号(+)を付ける必要があります。

- OLRバックアップの場所のdir_name変数は、ディレクトリのフルパス名にすることができます。たとえば:
 - Grid_home/cdata/cluster3/: 場所が存在することを確認します
 - d:¥cdata¥cluster3: 場所が存在することを確認します

LinuxまたはUNIXシステムでOLRバックアップが生成されるデフォルトの場所は

Grid_base/crsdata/host_name/olr/で、host_nameは、バックアップ対象のOLRが存在するノードの名前です。WindowsでOLRバックアップが生成されるデフォルトの場所は、同じパス構造を使用します。

例

OCRバックアップの場所をOracle ASMディスク・グループに指定するには、次のようにします。

```
# ocrconfig -backuploc +bkupdg
```

OLRバックアップの場所を指定するには、次のようにします。

```
# ocrconfig -local -backuploc full_directory_path
```

ocrconfig -copy

OCRバックアップ・ファイルがある場所から別の場所にコピーします。

構文

```
ocrconfig -copy source_file_name destination_file_name
```

使用上のノート

- このコマンドは、rootで実行する必要があります。
- source_file_nameには有効なOCRバックアップ・ファイルを指定する必要があります。たとえば:
 - /oradbocfs/crs/BACKUP00.ocr
 - d:¥oracle¥BACKUP01.ocr
 - +bkupdg:/aimev1/ocrbackup/day.ocr.260.739756845

Oracle ASMディスク・グループを指定した場合は、そのディスク・グループが存在し、マウントされていることを確認します。

有効なdestination_file_name値の例を次に示します。

- /oradbocfs/crs/mybcp01.ocr
- d:¥oracle¥mybcp01.ocr
- +bkupdg:mybcp01

Oracle ASMディスク・グループの名前の前にプラス記号(+)を付ける必要があります。

例

OCRバックアップ・ファイルをファイル・システムからOracle ASMディスク・グループにコピーするには、次のコマンドを実行します。

```
# ocrconfig -copy /oradbocfs/crs/BACKUP00.ocr +bkupdg:mybcp01.ocr
```

OCRバックアップ・ファイルをOracle ASMディスク・グループからファイル・システムにコピーするには、次のコマンドを実行します。

```
# ocrconfig -copy +bkupdg:/aimev1/ocrbackup/day.ocr.260.739756845
/oradbocfs/crs/mybcp01.ocr
```

OCRバックアップ・ファイルのあるOracle ASMディスク・グループから別のOracle ASMディスク・グループにコピーするには、次のコマンドを実行します。

```
# ocrconfig -copy +bkupdg01:/aimev1/ocrbackup/day.ocr.260.739756845
+bkupdg02:mybcp01.ocr
```

ocrconfig -delete

OCRデバイスまたはOCRファイルを削除します。

構文

```
ocrconfig -delete +diskgroup_name
```

使用上のノート

- このコマンドは、rootで実行する必要があります。
- +diskgroup_name変数は、Oracle ASMディスク・グループの名前です。たとえば: +olddg
ディスク・グループの名前の前に、プラス記号(+)を付ける必要があります。

例

OCRの場所を削除するには、次のコマンドを使用します。

```
# ocrconfig -delete +olddg
```

ocrconfig -downgrade

このコマンドを使用してOCRを指定の旧バージョンにダウングレードするのは、rootスクリプトだけです。

構文

```
ocrconfig -downgrade
```

使用上のノート

このコマンドにはオプションはありません。

ocrconfig -export

OCRの内容をターゲット・ファイルにエクスポートします。

構文

```
ocrconfig [-local] -export file_name
```

使用上のノート

- このコマンドは、rootで実行する必要があります。
- -localオプションを使用して、OLRの内容をエクスポートします。
- file_name変数は、ファイル名またはOracle ASMディスク・グループの名前です(名前の前にプラス記号(+))を付ける必要があります)。たとえば:
 - /oradbocfs/crs/data.ocr
 - d:¥oracle¥data.ocr
 - +bkupdg:data.ocr

例

OCRの内容をファイルにエクスポートするには、次のコマンドを使用します。

```
# ocrconfig -export d:¥tmp¥a
```

ocrconfig -import

OCRの内容をエクスポートしたターゲット・ファイルの内容を、OCRにインポートし直します。

構文

```
ocrconfig [-local] -import file_name
```

使用上のノート

- このコマンドは、rootで実行する必要があります。
- -localオプションを使用して、OLRの内容をファイルからインポートします。
- file_name変数は、ファイル名またはOracle ASMディスク・グループの名前です(名前の前にプラス記号(+))を付ける必要があります)。たとえば:
 - /oradbocfs/crs/data.ocr
 - d:¥oracle¥data.ocr
 - +bkupdg:/aimev1/ocrbackup/data.ocr.260.739756845
- このコマンドを実行する前にOracle Clusterwareを停止する必要があります。

例

ファイルの内容をOCRにインポートするには、次のコマンドを使用します。

```
# ocrconfig -import d:¥tmp¥a
```

ocrconfig -manualbackup

必要に応じてOCRを特定の場所にバックアップします。

前提条件

[ocrconfig -backuploc](#)を使用して、オンデマンド・バックアップの場所を確認します。

構文

```
ocrconfig [-local] -manualbackup
```

使用上のノート

- このコマンドは、rootで実行する必要があります。
- -localオプションを使用して、OLRの手動バックアップを実行します。

ocrconfig -overwrite

OCRメタデータ内のOCR構成を、このコマンドの実行元ノードにあるOCR構成情報で上書きします。

構文

```
ocrconfig -overwrite
```

使用上のノート

このコマンドは、rootで実行する必要があります。

ocrconfig -repair

このコマンドの実行元ノードでOCR構成を修復します。

このコマンドを使用して、クラスタのOCR構成の変更中に停止していた可能性のあるノードのOCRの場所を追加、削除または置換します。追加するOCRの場所が存在し、十分な権限を所有している必要があり、また、追加する前に、この場所をマウントしておく必要があります。

前提条件

- 修復を正常に完了させるには、Oracle高可用性サービスを起動する必要があります。
- ocrconfig -repairを実行する前にクラスタ・レディ・サービス・デーモンを停止する必要があります。

構文

```
ocrconfig -repair -add +diskgroup_name | -delete +diskgroup_name | -replace  
+current_diskgroup_name -replacement +new_diskgroup_name
```

使用上のノート

- このコマンドは、rootで実行する必要があります。
- +diskgroup_name変数は、Oracle ASMディスク・グループの名前です。たとえば：
+newdgm/mycluster/OCRFILE/registry.255.842574125

Oracle ASMディスク・グループを指定する場合は、ディスク・グループの名前の前にプラス記号(+)を付ける必要があります。

Oracle ASMディスク・グループの名前を取得するには、OCR構成を修復する必要がないノードでocrcheck -config -detailsコマンドを実行します。

関連項目:

- OCRの場所の作成、および適切な権限の設定の詳細は、[『Oracle Grid Infrastructureインストレーション』](#)

[ンおよびアップグレード・ガイドfor Linux』](#)を参照してください

- Oracle ASMディスク・グループ管理の詳細は、[『Oracle Automatic Storage Management管理者ガイド』](#)を参照してください。
- ocrcheck -config -detailsコマンドの詳細は、「[OCRCHECKユーティリティの使用](#)」を参照してください。
- ocrconfig -repairと同時に使用できるオプションは1つのみです。
- このコマンドの実行によって変更されるのはローカル構成のみであり、影響を受けるのは現行のノードのみです。

例

OCR構成を修復するには、次の手順を実行する必要があります。

```
# ocrconfig -repair -delete +olddg
```

ocrconfig -replace

OCRの場所をOracle Automatic Storage Management (Oracle ASM)ディスク・グループに再配置します。

前提条件

- 追加するOCRの場所が存在し、十分な権限を所有している必要があり、また、追加する前に、この場所をマウントしておく必要があります。
- このコマンドを使用するには2つ以上のOCRデバイスが必要です。OCRデバイスが2つ未満の場合、[ocrconfig -add](#)コマンドを実行して新しいOCRデバイスを追加してから[ocrconfig -delete](#)コマンドを実行し、置換するOCRデバイスを削除します。

構文

```
ocrconfig -replace +current_diskgroup_name -replacement +new_diskgroup_name
```

使用上のノート

- このコマンドは、rootで実行する必要があります。
- +diskgroup_name変数は、Oracle ASMディスク・グループの名前です(名前の前にプラス記号(+))を付ける必要があります)。たとえば: +newdg。ディスク・グループは存在し、マウントされている必要があります。

関連項目:

- OCRの場所の作成、および適切な権限の設定の詳細は、[『Oracle Grid Infrastructureインストールおよびアップグレード・ガイドfor Linux』](#)を参照してください
- Oracle ASMディスク・グループ管理の詳細は、[『Oracle Automatic Storage Management管理者ガイド』](#)を参照してください。

例

OCRデバイスまたはファイルを置換するには、次のコマンドを使用します。

```
# ocrconfig -replace +olddg -replacement +newdg
```

ocrconfig -restore

自動的に作成されたOCRバックアップ・ファイルからOCRをリストアします。

前提条件

- このコマンドを実行する前に、元のOCRファイルまたはOLRファイルが存在することを確認します。元のファイルが存在しない場合、空のファイルを作成する必要があります。



ノート:

[ocrconfig -export](#) コマンドによって生成されたファイルを使用して OCR をリストアすることはできません。ファイルは手動バックアップまたは自動バックアップである必要があります。

- OCRがOracle ASMディスク・グループに存在する場合、ディスク・グループが存在し、マウントされていることを確認します。

構文

```
ocrconfig [-local] -restore file_name
```

使用上のノート

- このコマンドは、rootで実行する必要があります。
- OCRバックアップ・ファイルがOracle ASMディスク・グループにある場合、ディスク・グループが存在し、マウントされていることを確認します。
- Oracle ASMディスク・グループの名前の前に、プラス記号(+)を付ける必要があります。
- -localオプションを使用して、OLRのバックアップをリストアします。
- 次にファイル名の例を示します。
 - /oradbocfs/crs/BACKUP00.ocr
 - d:¥oracle¥BACKUP01.ocr
 - +bkupdg:/aimev1/ocrbackup/day.ocr.260.739756845

例

ファイルからOCRをリストアするには、次のコマンドを使用します。

```
# ocrconfig -restore /oradbocfs/crs/BACKUP00.ocr
```

ocrconfig -showbackup

利用可能なすべてのバックアップを表示します。

構文

```
ocrconfig [-local] -showbackup [auto | manual]
```

使用上のノート

- -localオプションを使用して、手動のOLRバックアップ情報を表示します。-localフラグは、manualオプションとあわせて指定した場合にのみ機能します。

- 必要に応じてautoまたはmanualを指定して、自動バックアップまたは手動バックアップのいずれかの情報のみを表示することもできます。
 - auto: 過去4時間、8時間、12時間、前日および先週に作成された自動バックアップの情報を表示します。
 - manual: [ocrconfig -manualbackup](#)コマンドを使用して、最後から最大5つの手動バックアップの情報を表示します。

例

このコマンドは、次のようにOLRの手動バックアップ情報を表示します。

```
$ ocrconfig -local -showbackup manual
cc40524 2020/05/08 15:06:57 /gridbase/crsdata/dglnx6/olr/backup_20200508_150657.olr 0
```

ocrconfig -showbackuploc

OCRおよびOLRバックアップ・ファイルのバックアップ場所を表示します。

構文

```
ocrconfig [-local] -showbackuploc
```

使用上のノート

- このコマンドは、rootで実行する必要があります
- -localオプションを使用して、OLRバックアップの場所を表示します。

例

このコマンドによって、次のような出力が返されます。

```
# ocrconfig -showbackuploc
The Oracle Cluster Registry backup location is [+bkupdg]
# ocrconfig -local -showbackuploc
The Oracle Local Registry backup location is [/gridbase/crsdata/dglnx6/olr]
```

ocrconfig -upgrade

OCRを以前のバージョンからアップグレードします。

構文

```
ocrconfig -upgrade
```

使用上のノート

このコマンドにはオプションはありません。

Oracle Cluster Registryのトラブルシューティングおよび診断出力

この項では、OCRに関する問題のトラブルシューティングを行う様々な方法およびOCRの管理に使用されるユーティリティから診断情報を取得する様々な方法について説明します。これらのユーティリティは、OLRのトラブルシューティングにも使用できます。

この項では、次の項目について説明します。

- [Oracle Cluster Registryのトラブルシューティング](#)

- [OCRCHECKユーティリティの使用](#)
- [OCRDUMPユーティリティを使用したOracle Cluster Registryの内容の表示](#)

Oracle Cluster Registryのトラブルシューティング

[表I-1](#)に、OCRの一般的な問題およびそれに対応する推奨の解決策を示します。

表I-1 Oracle Cluster Registryの一般的な問題および解決策

問題	解決策
現在使用されていない OCR のミラー化を有効にする必要があります。	-replace オプションを指定して ocrconfig コマンドを実行します。
OCR に障害が発生したため、置換する必要があります。 Oracle Enterprise Manager または OCR のログ・ファイルにエラー・メッセージが書き込まれます。	-replace オプションを指定して ocrconfig コマンドを実行します。
OCR の構成が不適切です。	説明されているように、-repair オプションを指定して ocrconfig コマンドを実行します。
OCR の処理によってパフォーマンスに重大な影響が及ぼされたか、または他の理由のために OCR を削除します。	説明されているように、-replace オプションを指定して ocrconfig コマンドを実行します。
OCR に障害が発生したため、修正する前に、1 つの OCR のみを使用してノードを再起動する必要があります。	-repair オプションを指定して ocrconfig を実行し、問題のある OCR の場所を削除します。定義済のすべての OCR が見つからない場合、Oracle Clusterware は起動されません。

OCRCHECKユーティリティの使用

OCRCHECKユーティリティを実行すると、構成したOCRのブロック形式のバージョン、使用可能な領域の合計、使用済の領域、OCRのIDおよび場所が表示されます。

OCRCHECKでは、構成したすべてのOCRにあるすべてのブロックに対して、ブロック単位のchecksum操作が実行されます。各ファイルごとのステータスと、OCR全体の整合性チェックの結果も返されます。

ノート:



Oracle では、最低でも Oracle クラスタ・レディ・サービス・スタックがクラスタ内のすべてのノードで OFFLINE の場合、ocrcheck の使用がサポートされています。これは、スタックが ONLINE の場合でもコマンドは実行されますが、OCR の更新が進行中のときにチェックが実行されると、OCR は破損していると誤って示されることがあるためです。

関連項目:

Oracleクラスタ・レディ・サービス・スタックの詳細は、「[Oracle Clusterware技術スタック](#)」を参照してください。

構文

```
ocrcheck [-local] [-config | -backupfile <backup_file_name>] [-details] [-help]
```

ocrcheck -helpコマンドを実行して、このユーティリティの使用方法を表示します。

-detailsオプションを指定すると、Oracle ASMディスク・グループの名前を含む、詳しいOCR構成情報が表示されます。

例

次の例に、OCRCHECKコマンドの出力とコマンドの使用方法のサンプルを示します。

例I-1 OCRCHECKユーティリティのサンプル出力

```
# ocrcheck
Status of Oracle Cluster Registry is as follows :
  Version                :                4
  Total space (kbytes)    :           409568
  Used space (kbytes)     :           17012
  Available space (kbytes):           392556
  ID                     :       467263828
  Device/File Name        : +ocrdg1
                           Device/File integrity check succeeded
  Device/File Name        : +ocrdg2
                           Device/File integrity check succeeded
                           Device/File not configured
                           Device/File not configured
                           Device/File not configured
  Cluster registry integrity check succeeded
  Logical corruption check succeeded
```

ノート:



論理破損チェックは、root として ocrcheck コマンドが実行される場合のみ実行されます。

[例I-2](#)では、ocrcheck -local -configコマンドを使用してOLR情報を取得します。

ノート:



Oracle クラスタ・レディ・サービス・スタックと Oracle 高可用性サービス・スタックの両方がローカル・ノードで OFFLINE の場合、Oracle では、ocrcheck -local コマンドの使用がサポートされています。

関連項目:

Oracleクラスタ・レディ・サービス・スタックおよびOracle高可用性サービス・スタックの詳細は、「[Oracle Clusterware技術スタック](#)」を参照してください。

例I-2 OLR情報を取得するOCRCHECKの使用

```
$ ocrcheck -local -config
Oracle Local Registry configuration is :
  Device/File Name        : /oraclebase/crsdata/dglnx6/olr/dglnx6_19.olr
```

[例I-3](#)では、`ocrcheck -config -details`コマンドを使用して、Oracle ASMディスク・グループでのOCRの場所を取得します。

例I-3 OCR情報を取得するOCRCHECKの使用

```
$ ocrcheck -config -details
Oracle Cluster Registry configuration is :
  Device/File Name      : +ocrvd11/mjk_1/OCRFILE/registry.255.842574113
  Device/File Name      : +ocrdv2/mjk_1/OCRFILE/registry.255.842574125
```

[例I-4](#)では、`ocrcheck -details`コマンドを使用して、さらに詳しいOCR構成を取得します。

例I-4 OCRの詳細情報を取得するOCRCHECKの使用

```
$ ocrcheck -details
Status of Oracle Cluster Registry is as follows :
  Version                : 4
  Total space (kbytes)   : 409568
  Used space (kbytes)    : 17012
  Available space (kbytes) : 392556
  ID                     : 467263828
  Device/File Name       : +OCRVD11/mycluster/OCRFILE/registry.255.842574113
                          Device/File integrity check succeeded
  Device/File Name       : +OCRVD2/mycluster/OCRFILE/registry.255.842574125
                          Device/File integrity check succeeded
                          Device/File not configured
                          Device/File not configured
                          Device/File not configured
  Cluster registry integrity check succeeded
  Logical corruption check succeeded
```

[例I-5](#)では、`ocrcheck -details`コマンドを使用してバックアップ・ファイル情報を取得します。

例I-5 バックアップ・ファイル情報を取得するOCRCHECKの使用

```
% ocrcheck -backupfile +bkupdg:/sbezawadhtest/OCRBKUP/backup00.ocr.258.919047805
Status of backup file is as follows:
Version : 4
Used space (kbytes) : 17012
Device/File Name : +bkupdg:/sbezawadhtest/OCRBKUP/backup00.ocr.258.919047805
Device/File integrity check succeeded
Backup file integrity check succeeded
Logical corruption check succeeded
Consistency check of entities managed by Cluster Ready Services succeeded.
```

OCRCHECKでは、ログ・ファイルがORACLE_BASE/diag/crs/host_name/crsディレクトリ内に作成されます。ログ・レベルを変更するには、ORACLE_BASE/crsdata/host_name/crsdiag/ocrcheck.iniファイルを編集します。この場合、ORACLE_BASEはOracle Grid Infrastructureソフトウェア・ホームに対応しています。

OCRDUMPユーティリティを使用したOracle Cluster Registryの内容の表示

OCRDUMPユーティリティを使用すると、OCRおよびOLRの内容を読みやすい形式でファイルまたはstdoutに書き込むことによって表示できます。

この項では、OCRDUMPユーティリティを使用してトラブルシューティングのためにOCRおよびOracle Local Registry(OLR)の内容を表示する方法について説明します。

OCRDUMPにはいくつかのオプションを使用できます。たとえば、出力をキーおよびその子に制限できます。ブラウザを使用して表示可能なXMLファイルに内容を書き込むこともできます。OCRDUMPは、ASCII文字列およびデータ型形式の値としてOCRのキーを書き込みます。OCRDUMPは、ベスト・エフォート原則に基づいてヘッダー情報を取得します。

OCR_DUMPでは、ログ・ファイルがORACLE_BASE/diag/crs/host_name/crs内に作成されます。ログ・レベルを変更するには、ORACLE_BASE/crsdata/host_name/crsdiag/ocrlog.iniファイルを編集します。

ロギング・コンポーネントを変更するには、comploglvl=エントリを含むエントリを編集します。たとえば、OCR_APIコンポーネントのログ・レベルを3に変更し、OCRRAWコンポーネントのログ・レベルを5に変更するには、ocrlog.iniファイルに次のエントリを作成します。

```
comploglvl="OCR_API:3;OCRRAW:5"
```

ノート:



OCR_DUMP ユーティリティを使用する前に、Grid_home ディレクトリにファイルを作成する権限を持っていることを確認してください。

この項には次のトピックが含まれます:

- [OCR_DUMPユーティリティの構文およびオプション](#)
- [OCR_DUMPユーティリティの例](#)
- [OCR_DUMPユーティリティの出力例](#)

OCR_DUMPユーティリティの構文およびオプション

この項では、OCR_DUMPユーティリティのコマンド構文および使用方法について説明します。次の構文を使用してocrdumpコマンドを実行します(この場合、file_nameにはOracle Cluster Registryの出力を書き込むターゲット・ファイルの名前、key_nameにはOracle Cluster Registryのサブツリーの内容を書き込むキーの名前を指定します)。

```
$ ocrdump [file_name | -stdout] [-local] [-backupfile backup_file_name  
[-keyname key_name] [-xml] [-noheader]  
]
```

[表I-2](#)に、OCR_DUMPユーティリティのオプションおよびオプションの説明を示します。

表I-2 OCR_DUMPのオプションおよびオプションの説明

オプション	説明
file_name	OCR_DUMP によって出力を書き込むファイルの名前です。 デフォルトでは、OCR_DUMP によって、事前定義された出力ファイル OCR_DUMPFILE に出力が書き込まれます。file_name オプションによって、OCR_DUMP 出力は指定したファイルにリダイレクトされます。
- stdout	このオプションを使用して、プログラムを起動したテキスト端末に OCR_DUMP 出力をリダイレクトできます。 出力をリダイレクトしない場合、OCR_DUMP によって、事前定義された出力ファイル OCR_DUMPFILE に出力が書き込まれます。

オプション	説明
- local	OLR の内容をダンプするには、このオプションを使用します。
- backupfile	OCR バックアップ・ファイルの内容を表示するには、このオプションを使用します。- local オプションをこのオプションとともに使用すると、OLR バックアップ・ファイルの内容が表示されます。
backup_file_name	表示する内容を含むバックアップ・ファイルの名前です。ocrconfig -showbackup コマンドを使用して、バックアップを問い合わせることができます。バックアップ・ファイルが Oracle ASM ディスク・グループにある場合、ディスク・グループが存在し、マウントされていることを確認します。
- keyname key_name	サブツリーがダンプされる Oracle Cluster Registry キーの名前です。
- xml	出力を XML 形式で書き込むにはこのオプションを使用します。
- noheader	コマンドを実行した時間および Oracle Cluster Registry 構成が作成された時間を出しません。

OCRDUMPユーティリティの例

次に示すocrdumpユーティリティの例では、様々なタイプのOCR情報を抽出し、様々なターゲットに書き込みます。

```
ocrdump
```

OCRの内容を現在のディレクトリのOCRDUMPFILというファイルに書き込みます。

```
ocrdump MYFILE
```

OCRの内容を現在のディレクトリのMYFILEというファイルに書き込みます。

```
ocrdump -stdout -keyname SYSTEM
```

OCRの内容をSYSTEMキーのサブツリーから端末ウィンドウに表示します。

```
ocrdump -stdout -xml
```

OCRの内容をXML形式で端末ウィンドウに表示します。

```
ocrdump -stdout -backupfile Grid_home/cdata/cluster_name/file_name
```

Grid_home/cdata/cluster_name/file_nameディレクトリのコンテンツを表示します。すべてのキーを表示できるようにするには、このコマンドをrootで実行する必要があります。OCRバックアップ・ファイルであることを認識できるように、ファイルにはBACKUP00.ocrなどの適切な名前を付けてください。

OCRDUMPユーティリティの出力例

次のOCRDUMPの例は、ocrdumpコマンドの2つの実行例について、KEYNAME、VALUE TYPE、VALUE、権限のセット (user、group、world)およびアクセス権を示します。テキスト値がAMERICAN_AMERICA.WE8ASCII37である

SYSTEM.languageキーの出力を次に示します。

```
[SYSTEM.language]
ORATEXT : AMERICAN_AMERICA.WE8ASCII37
SECURITY : {USER_PERMISSION : PROCR_ALL_ACCESS, GROUP_PERMISSION : PROCR_READ,
OTHER_PERMISSION : PROCR_READ, USER_NAME : user, GROUP_NAME : group}
```

3という整数値を持つSYSTEM.versionキーの出力を次に示します。

```
[SYSTEM.version]
UB4 (10) : 3
SECURITY : {USER_PERMISSION : PROCR_ALL_ACCESS, GROUP_PERMISSION : PROCR_READ,
OTHER_PERMISSION : PROCR_READ, USER_NAME : user, GROUP_NAME : group}
```

J Oracle Clusterwareのトラブルシューティング

この付録では、Oracle Clusterware環境の監視について説明し、動的デバッグを有効にして、Oracle Clusterwareの処理のトラブルシューティングを行う方法、また特定のコンポーネントおよび特定のOracle Clusterwareリソースのデバッグおよびトレースを有効にして、トラブルシューティングの対象を絞り込む方法について示します。

この付録の内容は次のとおりです。

- [Oracle Clusterwareの診断およびアラート・ログ・データ](#)
- [診断収集スクリプト](#)
- [ローリング・アップグレードおよびドライバのインストールに関する問題](#)
- [ゾーン委任のテスト](#)
- [Oracle Clusterwareアラート](#)

クラスタ・リソース・アクティビティ・ログを使用したクラスタ・リソースの障害の監視

クラスタ・リソース・アクティビティ・ログは、診断ログとは別に、リソースの障害に関する正確で詳細な情報を提供します。

Oracle Clusterware管理リソースで障害が発生した場合、Oracle Clusterwareは障害に関するメッセージのログを、グリッド・インフラストラクチャ管理リポジトリにある**クラスタ・リソース・アクティビティ・ログ**に書き込みます。障害は、リソース、ホスティング・ノードまたはネットワークの問題が原因で起こる場合があります。クラスタ・リソース・アクティビティ・ログでは、リソースの障害の原因を一元的に把握できます。

クラスタ・リソース・アクティビティ・ログへの書込みは、アクティビティIDでタグ付けされます。関連するデータは同じ親アクティビティIDを取得し、親データの下にネストされます。たとえば、Oracle Clusterwareが稼働しているときに、`crsctl stop clusterware -all`コマンドを実行した場合、すべてのアクティビティがアクティビティIDを取得し、関連するアクティビティが同じ親アクティビティIDでタグ付けされます。ノードごとに、親ID下にサブIDが作成され、個々のアクティビティがそれぞれ対応するアクティビティIDでタグ付けされます。さらに、個々のノード上のリソースごとに親IDに基づいてサブIDが作成され、アクティビティIDの階層が構築されます。アクティビティIDの階層を使用すると、データを分析して特定のアクティビティを確認できます。

たとえば、相互に複雑な依存関係を持ち、1つのデータベース・サービスを使用している多数のリソースがあるとします。金曜日には、すべてのリソースが1つのノードで実行されていますが、月曜日には各リソースが異なるノード上で実行されているため、その理由を確認する必要があります。`crsctl query calog`コマンドを使用すると、それらのリソースとデータベース・サービスを含むすべてのアクティビティをクラスタ・リソース・アクティビティ・ログに問い合わせることができます。出力には完全なフローが表示され、ユーザーは親サービス・フェイルオーバーID内の各サブIDを問い合わせ、発生した問題とその原因を具体的に確認できます。

フィルタを使用すると、クラスタ・リソース・アクティビティ・ログ内の任意の数のフィールドを問い合わせることができます。たとえば、`root`など、特定のオペレーティング・システム・ユーザーによって書き込まれたすべてのアクティビティを問い合わせることができます。`crsctl query calog`コマンドによって生成される出力は、表形式またはXML形式で表示できます。

クラスタ・リソース・アクティビティ・ログは、現在のOracle Clusterwareのロギングおよびアラート・ログ・メッセージを補助するものです。

ノート:



Oracle Clusterware では、セキュリティ関連の情報を含むメッセージ(ログイン資格証明など)はクラスタ・アクティビティ・ログに書き込まれません。

クラスタ・リソース・アクティビティ・ログの内容を管理および表示するには、次のコマンドを使用します。

crsctl query calog

特定の基準に一致するリソース・アクティビティ・ログを問い合わせます。

構文

```
crsctl query calog [-aftertime "timestamp"] [-beforetime "timestamp"]  
  [-duration "time_interval" | -follow] [-filter "filter_expression"]  
  [-fullfmt | -xmlfmt]
```

パラメータ

表J-1 crsctl query calogコマンドのパラメータ

パラメータ	説明
-aftertime "timestamp"	特定の時刻の後にログに記録されたアクティビティを表示します。 タイムスタンプを YYYY-MM-DD HH24:MI:SS[.FF][TZh:TzM]、YYYY-MM-DD または HH24:MI:SS[.FF][TZh:TzM]の形式で指定します。 TZh および TzM はタイムゾーンの時間と分を意味し、FF はマイクロ秒を意味します。 [TZh:TzM]を指定する場合、crsctl コマンドは、タイムゾーンが UTC であると仮定します。[TZh:TzM]を指定しない場合、crsctl コマンドは、crsctl コマンドの実行元であるクラスタ・ノードのローカル・タイムゾーンであると仮定します。 このパラメータを-beforetime とともに使用して、特定の時間間隔でログに記録されたアクティビティを問い合わせます。
-beforetime "timestamp"	特定の時刻の前にログに記録されたアクティビティを表示します。 タイムスタンプを YYYY-MM-DD HH24:MI:SS[.FF][TZh:TzM]、YYYY-MM-DD または HH24:MI:SS[.FF][TZh:TzM]の形式で指定します。 TZh および TzM はタイムゾーンの時間と分を意味し、FF はマイクロ秒を意味します。

パラメータ	説明	
	[TZH:TZM]を指定する場合、crsctl コマンドは、タイムゾーンが UTC であると仮定します。[TZH:TZM]を指定しない場合、crsctl コマンドは、crsctl コマンドの実行元であるクラスタ・ノードのローカル・タイムゾーンであると仮定します。 このパラメータを -aftertime とともに使用して、特定の時間間隔でログに記録されたアクティビティを問い合わせます。	
-duration "time_interval" -follow	-duration を使用して、-aftertime パラメータの使用時に問い合わせる時間間隔を指定します。 タイムスタンプを DD HH:MM:SS 形式で指定します。 -follow を使用して、発生したアクティビティの連続的なストリームを表示します。	
-filter "filter_expression"	-filter パラメータを使用して、クラスタ・リソース・アクティビティ・ログ内の任意の数のフィールドを問い合わせます。 複数のフィルタを指定するには、二重引用符("")で囲んだフィルタ式のカンマ区切りのリストを使用します。	
-fullfmt -xmlfmt	クラスタ・リソース・アクティビティ・ログのデータを表示するには、フル形式または XML 形式を選択します。	
クラスタ・リソース・アクティビティ・ログのフィールド		
-filterパラメータを使用して、クラスタ・リソース・アクティビティ・ログ内の任意の数のフィールドを問い合わせます。		
表J-2 クラスタ・リソース・アクティビティ・ログのフィールド		
フィールド	説明	ユース・ケース
timestamp	クラスタ・リソース・アクティビティがログに記録された時刻。	このフィルタを使用して、特定の時刻にログに記録されたすべてのアクティビティを問い合わせます。 これは -aftertime、-beforetime および -duration コマンド・パラメータのかわりです。
writer_process_id	クラスタ・リソース・アクティビティ・ログに書き込んでいるプロセスの ID。	特定のプロセスによって生成されたアクティビティのみを問い合わせます。

フィールド	説明	ユース・ケース
writer_process_name	クラスタ・リソース・アクティビティ・ログに書き込んでいるプロセスの名前。	特定のプロセスを問い合わせると、CRSCTL は特定のプロセスのすべてのアクティビティを戻します。
writer_user	クラスタ・リソース・アクティビティ・ログに書き込んでいるユーザーの名前。	特定のユーザーによって書き込まれたすべてのアクティビティを問い合わせます。
writer_group	クラスタ・リソース・アクティビティ・ログに書き込んでいるユーザーが属するグループの名前。	特定のユーザー・グループに属するユーザーによって書き込まれたすべてのアクティビティを問い合わせます。
writer_hostname	クラスタ・リソース・アクティビティ・ログが書き込まれるホストの名前。	特定のホストによって書き込まれたすべてのアクティビティを問い合わせます。
writer_clustername	クラスタ・リソース・アクティビティ・ログが書き込まれるクラスタの名前。	特定のクラスタによって書き込まれたすべてのアクティビティを問い合わせます。
nls_product	NLS メッセージの製品(CRS、ORA、srvm など)。	特定の製品名を持つすべてのアクティビティを問い合わせます。
nls_facility	NLS メッセージの機能(CRS、PROC など)。	特定の機能名を持つすべてのアクティビティを問い合わせます。
nls_id	NLS メッセージの ID (42008 など)。	特定のメッセージ ID を持つすべてのアクティビティを問い合わせます。
nls_field_count	NLS メッセージのフィールド数。	nls_field_count コマンド・パラメータを超えるか、これ未満またはこれと等しいフィールドを含む NLS メッセージに対応するすべてのアクティビティを問い合わせます。
nls_field1	NLS メッセージの最初のフィールド。	NLS メッセージの最初のパラメータに一致するすべてのアクティビティを問い合わせます。
nls_field1_type	NLS メッセージの最初のフィールドのタイプ。	NLS メッセージの最初のパラメータの特定のタイプに一致するすべてのアクティビティを問い合わせます。

フィールド	説明	ユース・ケース
nls_format	NLS メッセージの形式(Resource '%s' has been modified など)。	NLS メッセージの特定の形式に一致するすべてのアクティビティを問い合わせます。
nls_message	クラスタ・リソース・アクティビティ・ログに書き込まれた NLS メッセージ全体 (Resource 'ora.cvu' has been modified など)。	特定の NLS メッセージに一致するすべてのアクティビティを問い合わせます。
actid	各クラスタ・アクティビティ・ログの一意のアクティビティ ID。	特定の ID に一致するすべてのアクティビティを問い合わせます。 また、部分的な actid のみを指定し、actid がアクティビティ ID のサブセットであるすべてのアクティビティをリストします。
is_planned	アクティビティが計画済かどうかを確認します。 たとえば、ユーザーがノードでコマンド <code>crsctl stop crs</code> を発行すると、スタックが停止し、リソースがバウンスします。 <code>crsctl stop crs</code> コマンドを実行すると、アクティビティが生成され、<code>calog</code> に記録されます。これは計画済アクションであるため、<code>is_planned</code> フィールドは <code>true</code> (1) に設定されます。 その他の場合は <code>is_planned</code> フィールドが <code>false</code> (0) に設定されます。	計画済または計画外のすべてのアクティビティを問い合わせます。
onbehalfof_user	クラスタ・アクティビティ・ログの書込みを代理されているユーザーの名前。	特定のユーザーのために書き込まれたすべてのアクティビティを問い合わせます。
entity_isoraentity	<code>calog</code> アクティビティが記録されているエンティティが Oracle エンティティであるかどうかを確認します。 たとえば、リソース(<code>ora.***</code> など)が起動または停止された場合には、それらすべての	Oracle エンティティまたは Oracle 以外のエンティティによって記録されたすべてのアクティビティを問い合わせます。

フィールド	説明	ユース・ケース
	アクティビティがクラスタ・リソース・アクティビティ・ログに記録されます。 ora.***は Oracle エンティティであるため、entity_isoraentity フィールドは true (1) に設定されます。 その他の場合は entity_isoraentity フィールドが false (0) に設定されます。	
entity_type	エンティティのタイプ(クラスタ・アクティビティ・ログが書き込まれる server など)。	特定のエンティティに一致するすべてのアクティビティを問い合わせます。
entity_name	エンティティの名前(クラスタ・アクティビティ・ログが書き込まれる foo など)。	特定のエンティティ名に一致するすべてのクラスタ・アクティビティを問い合わせます。
entity_hostname	ホストの名前(クラスタ・アクティビティ・ログが書き込まれるエンティティに関連付けられた node1, など)。	特定のホスト名に一致するすべてのクラスタ・アクティビティを問い合わせます。
entity_clustername	クラスタの名前(クラスタ・アクティビティ・ログが書き込まれるエンティティに関連付けられた cluster1 など)。	特定のクラスタ名に一致するすべてのクラスタ・アクティビティを問い合わせます。

使用上のノート

簡易フィルタを、Boolean演算子を使用した式フィルタと呼ばれる式に組み込みます。

タイムスタンプおよび時間間隔を二重引用符("")で囲みます。

フィルタ式を二重引用符("")で囲みます。

カッコまたは空白を含む値を一重引用符(")で囲みます。

一致するレコードが見つからない場合は、Oracle Clusterware Control (CRSCTL)ユーティリティにより次のメッセージが表示されます。

```
CRS-40002: No activities match the query.
```

例

次にフィルタの例を示します。

- "writer_user==root": 表示をrootユーザーのみに限定します。
- "customer_data=='GEN_RESTART@SERVERNAME(rwsbi08)=StartCompleted~'": 表示を指定された値GEN_RESTART@SERVERNAME(node1)=StartCompleted~を含むcustomer_dataに限定

します。

すべてのリソース・アクティビティを問い合わせ、出力をフル形式で表示するには、次のようにします。

```
$ crsctl query calog -fullfmt
----ACTIVITY START----
timestamp                : 2016-09-27 17:55:43.152000
writer_process_id        : 6538
writer_process_name       : crsd.bin
writer_user               : root
writer_group              : root
writer_hostname           : node1
writer_clustername        : cluster1-mb1
customer_data             : CHECK_RESULTS=-408040060~
nls_product               : CRS
nls_facility              : CRS
nls_id                    : 2938
nls_field_count           : 1
nls_field1                : ora.cvu
nls_field1_type           : 25
nls_field1_len            : 0
nls_format                : Resource '%s' has been modified.
nls_message               : Resource 'ora.cvu' has been modified.
actid                     : 14732093665106538/1816699/1
is_planned                : 1
onbehalfof_user           : grid
onbehalfof_hostname       : node1
entity_isoraentity        : 1
entity_type               : resource
entity_name                : ora.cvu
entity_hostname           : node1
entity_clustername        : cluster1-mb1
----ACTIVITY END----
```

すべてのリソース・アクティビティを問い合わせ、出力をXML形式で表示するには、次のようにします。

```
$ crsctl query calog -xmlfmt
<?xml version="1.0" encoding="UTF-8"?>
<activities>
  <activity>
    <timestamp>2016-09-27 17:55:43.152000</timestamp>
    <writer_process_id>6538</writer_process_id>
    <writer_process_name>crsd.bin</writer_process_name>
    <writer_user>root</writer_user>
    <writer_group>root</writer_group>
    <writer_hostname>node1</writer_hostname>
    <writer_clustername>cluster1-mb1</writer_clustername>
    <customer_data>CHECK_RESULTS=-408040060~</customer_data>
    <nls_product>CRS</nls_product>
    <nls_facility>CRS</nls_facility>
    <nls_id>2938</nls_id>
    <nls_field_count>1</nls_field_count>
    <nls_field1>ora.cvu</nls_field1>
    <nls_field1_type>25</nls_field1_type>
    <nls_field1_len>0</nls_field1_len>
    <nls_format>Resource '%s' has been modified.</nls_format>
    <nls_message>Resource 'ora.cvu' has been modified.</nls_message>
    <actid>14732093665106538/1816699/1</actid>
    <is_planned>1</is_planned>
    <onbehalfof_user>grid</onbehalfof_user>
    <onbehalfof_hostname>node1</onbehalfof_hostname>
    <entity_isoraentity>1</entity_isoraentity>
    <entity_type>resource</entity_type>
    <entity_name>ora.cvu</entity_name>
    <entity_hostname>node1</entity_hostname>
    <entity_clustername>cluster1-mb1</entity_clustername>
```

```
</activity>
</activities>
```

特定の時刻後の2時間の間隔でのリソース・アクティビティを問い合わせ、出力をXML形式で表示するには、次のようにします。

```
$ crsctl query calog -aftertime "2016-09-28 17:55:43" -duration "0 02:00:00" -xmlfmt
<?xml version="1.0" encoding="UTF-8"?>
<activities>
  <activity>
    <timestamp>2016-09-28 17:55:45.992000</timestamp>
    <writer_process_id>6538</writer_process_id>
    <writer_process_name>crsd.bin</writer_process_name>
    <writer_user>root</writer_user>
    <writer_group>root</writer_group>
    <writer_hostname>node1</writer_hostname>
    <writer_clustername>cluster1-mb1</writer_clustername>
    <customer_data>CHECK_RESULTS=1718139884~</customer_data>
    <nls_product>CRS</nls_product>
    <nls_facility>CRS</nls_facility>
    <nls_id>2938</nls_id>
    <nls_field_count>1</nls_field_count>
    <nls_field1>ora.cvu</nls_field1>
    <nls_field1_type>25</nls_field1_type>
    <nls_field1_len>0</nls_field1_len>
    <nls_format>Resource '%s' has been modified.</nls_format>
    <nls_message>Resource 'ora.cvu' has been modified.</nls_message>
    <actid>14732093665106538/1942009/1</actid>
    <is_planned>1</is_planned>
    <onbehalfof_user>grid</onbehalfof_user>
    <onbehalfof_hostname>node1</onbehalfof_hostname>
    <entity_isoraentity>1</entity_isoraentity>
    <entity_type>resource</entity_type>
    <entity_name>ora.cvu</entity_name>
    <entity_hostname>node1</entity_hostname>
    <entity_clustername>cluster1-mb1</entity_clustername>
  </activity>
</activities>
```

特定の時刻のリソース・アクティビティを問い合わせるには、次のようにします。

```
$ crsctl query calog -filter "timestamp=='2016-09-28 17:55:45.992000'"
2016-09-28 17:55:45.992000 : Resource 'ora.cvu' has been modified. :
14732093665106538/1942009/1 :
```

フィルタwriter_userおよびcustomer_dataを使用してリソース・アクティビティを問い合わせるには、次のようにします。

```
$ crsctl query calog -filter "writer_user==root AND customer_data==
'GEN_RESTART@SERVERNAME(node1)=StartCompleted~'" -fullfmt
```

または

```
$ crsctl query calog -filter "(writer_user==root) AND (customer_data==
'GEN_RESTART@SERVERNAME(node1)=StartCompleted~')" -fullfmt

----ACTIVITY START----
timestamp           : 2016-09-15 17:42:57.517000
writer_process_id    : 6538
writer_process_name  : crsd.bin
writer_user          : root
writer_group         : root
writer_hostname      : node1
writer_clustername    : cluster1-mb1
customer_data        : GEN_RESTART@SERVERNAME(rwsbi08)=StartCompleted~
nls_product          : CRS
nls_facility         : CRS
```

```

nls_id                : 2938
nls_field_count       : 1
nls_field1            : ora.testdb.db
nls_field1_type       : 25
nls_field1_len        : 0
nls_format            : Resource '%s' has been modified.
nls_message           : Resource 'ora.devdb.db' has been modified.
actid                 : 14732093665106538/659678/1
is_planned            : 1
onbehalfof_user       : oracle
onbehalfof_hostname   : node1
entity_isoraentity    : 1
entity_type           : resource
entity_name           : ora.testdb.db
entity_hostname       : node1
entity_clustername    : cluster1-mb1
-----ACTIVITY END-----

```

UTC+08:00の時刻2016-11-15 22:53:08の後に生成されたすべてのcalogを問い合わせるには、次のようにします。

```
$ crsctl query calog -aftertime "2016-11-15 22:53:08+08:00"
```

UTC-08:00の時刻2016-11-15 22:53:08の後に生成されたすべてのcalogを問い合わせるには、次のようにします。

```
$ crsctl query calog -aftertime "2016-11-15 22:53:08-08:00"
```

マイクロ秒を使用したタイムスタンプを指定してすべてのcalogを問い合わせるには、次のようにします。

```

$ crsctl query calog -aftertime "2016-11-16 01:07:53.063000"
2016-11-16 01:07:53.558000 : Resource 'ora.cvu' has been modified. :
14792791129816600/2580/7 :
2016-11-16 01:07:53.562000 : Clean of 'ora.cvu' on 'rwsam02' succeeded :
14792791129816600/2580/8 :

```

crsctl get calog maxsize

Oracle Clusterware管理リソースのアクティビティ情報を格納するには、クラスタ・リソース・アクティビティ・ログに割り当てられている最大領域を問い合わせます。

構文

```
crsctl get calog maxsize
```

パラメータ

crsctl get calog maxsizeコマンドにはパラメータはありません。

例

次の例では、アクティビティを格納するためにクラスタ・リソース・アクティビティ・ログに割り当てられている最大領域が戻されます。

```

$ crsctl get calog maxsize
CRS-6760: The maximum size of the Oracle cluster activity log is 1024 MB.

```

crsctl get calog retentiontime

クラスタ・リソース・アクティビティ・ログの保存時間を問い合わせます。

構文

```
crsctl get calog retentiontime
```

パラメータ

crsctl get calog retentiontimeコマンドにはパラメータはありません。

例

次の例では、クラスタ・アクティビティ・ログの保存時間を時間数で戻します。

```
$ crsctl get calog retentiontime
CRS-6781: The retention time of the cluster activity log is 73 hours.
```

crsctl set calog maxsize

Oracle Clusterware管理リソースのアクティビティ情報を格納するために割り当てられる最大領域を構成します。

構文

```
crsctl set calog maxsize maximum_size
```

使用上のノート

クラスタ・リソース・アクティビティ・ログに割り当てる記憶領域の最大サイズの値をMB単位で指定します。



ノート:

記憶領域の容量を減らすと、記憶域のコンテンツが失われます。

例

次の例では、Oracle Clusterware管理リソースのアクティビティ情報を格納するための最大領域を1024 MBに設定します。

```
$ crsctl set calog maxsize 1024
```

crsctl set calog retentiontime

クラスタ・リソース・アクティビティ・ログの保存時間を構成します。

構文

```
crsctl set calog retentiontime hours
```

パラメータ

crsctl set calog retentiontimeコマンドは時間数をパラメータに取ります。

使用上のノート

クラスタ・リソース・アクティビティ・ログの保存時間を時間単位で指定します。

例

次の例では、クラスタ・リソース・アクティビティ・ログの保存時間を72時間に設定します。

```
$ crsctl set calog retentiontime 72
```

Oracle Clusterwareの診断およびアラート・ログ

次の内容を確認し、Oracle ClusterwareによるADRの使用方法におけるクラスタウェア固有の側面を理解します。

Oracle Clusterwareは、Oracle Database障害診断インフラストラクチャを使用して診断データとアラート・ログを管理しま

す。このため、ほとんどの診断データは自動診断リポジトリ(ADR)(インストール時に指定したベース・ディレクトリ内のディレクトリとファイルのコレクション)に含まれます。

ADRディレクトリ構造

Oracle Clusterware ADRデータは、ADRベースと呼ばれるルート・ディレクトリに書き込まれます。ADR以外のコンポーネントもこのディレクトリを使用するため、Oracleベースと呼ばれることもあります。ベースとして使用するファイル・システム・パスをOracle Grid Infrastructureのインストール時に指定します。これを変更するには、Oracle Grid Infrastructureを再インストールするしかありません。

ADRファイルはADRホーム・ディレクトリに存在します。任意のホストで実行するOracle ClusterwareでもADRホームの構造は常に次のようになります。

```
ORACLE_BASE/diag/crs/host_name/crs
```

この例で、ORACLE_BASEはOracle Grid Infrastructureのインストール時に指定したOracleベース・パス、host_nameはホストの名前です。Windowsプラットフォームでは、円記号(¥)でパスのディレクトリ名が区切られます。

ADRホーム内には、特定のタイプのADRデータのための様々なディレクトリがあります。最重要のディレクトリはincidentです。traceディレクトリには、Oracle Clusterwareデーモンとコマンドライン・プログラムによって書き込まれる正常なトレース・ファイルすべて(インシデント以外)と、Oracle Clusterwareアラート・ログの単純なテキスト・バージョンが含まれます。この構成は、Oracle Clusterware 12cリリース1 (12.1.0.2)より前のバージョンとは大幅に異なります。以前のバージョンでは診断ログ・ファイルはデーモンごとに個別のディレクトリに書き込まれていました。

ログ・レベルを変更するには、ORACLE_BASE/crsdata/host_name/crsdiag/ocrcheck.iniファイルを編集します。

traceディレクトリのファイル

Oracle Clusterware 12cリリース1 (12.1.0.2)以降、Oracle Clusterwareプログラムによって書き込まれる診断データ・ファイルは、トレース・ファイルと呼ばれ、.trcファイル拡張子が付けられて、ADRホームのtraceサブディレクトリにまとめられます。これらのファイルのネーミング規則では、通常、ファイル名として実行可能プログラム名が使用され、プログラムの種類によって他のデータが追加されることもあります。

Oracle Clusterwareコマンドライン・プログラムによって書き込まれるトレース・ファイルでは、同じコマンド・プログラムの複数の呼出しとデータを区別するために、オペレーティング・システム・プロセスID (PID)がトレース・ファイル名に含まれます。たとえば、CRSCTLによって書き込まれるトレース・データの名前は、crsctl_PID.trcのような構造になります。この例では、PIDは、10進数として表示されるオペレーティング・システム・プロセスIDです。

Oracle Clusterwareデーモン・プログラムによって書き込まれるトレース・ファイルはファイル名にPIDを含みません。また、ファイル・ローテーション・メカニズムの対象であり、名前に影響があります。ローテーションでは、現在のデーモン・トレース・ファイルが特定のサイズになると、そのファイルが閉じて名前が変更され、新しいトレース・ファイルが開きます。これが決められた回数行われます。その後、デーモンの一番古いトレース・ファイルが破棄され、ローテーション・セットが設定されたサイズに保たれます。

ほとんどのOracle Clusterwareデーモンで使用されるファイル・サイズの制限は25MB、ローテーション・セット・サイズは10ファイルです。この場合、合計250MBのトレース・データが保存されます。特定のデーモンの現在のトレース・ファイルでは、ファイル名としてプログラム名が使用されます。ローテーション内の古いファイルにはファイル名に番号が付けられます。たとえば、Oracle 高可用性サービス・デーモン(OHASD)が現在書き込んでいるトレース・ファイルの名前はohasd.trcです。直前にローテーションされたトレース・ファイルの名前はohasd_n.trcです(nは増分する10進数)。nが最も大きいファイルは、最も新しくアーカイブされたトレースです。nが最も小さいファイルは最も古いトレースです。

Oracle Clusterwareエージェントはデーモン・プログラムです。このトレース・ファイルには、エージェントの生成元(OHASDと

Cluster Ready Servicesデーモン(CRSD)のどちらによって生成されたか)と、エージェントを実行しているオペレーティング・システム・ユーザー名を示す特別なネーミング規則があります。つまり、エージェントでの名前の構造は次のようになります。

`origin_executable_user_name`

ノート:



この名前構造の最初の 2 つのアンダースコア(_)は、リテラルとしてトレース・ファイル名に含まれます。
`user_name` のアンダースコアは、ファイルのネーミング規則の一部ではありません。

この例で、`origin`は`ohasd`または`crsd`、`executable`は実行可能プログラム名、`user_name`はオペレーティング・システム・ユーザー名です。また、これらはデーモンであるため、エージェント・トレース・ファイルは前述のローテーション・メカニズムの対象になります。ローテーションが発生すると、ファイルに`_n`接尾辞が追加されます。

Oracle Clusterwareアラート・ログ

トレース・ファイルの他に、Oracle Clusterware ADRホームの`trace`サブディレクトリには単純なテキストのOracle Clusterwareアラート・ログが含まれます。名前は常に`alert.log`です。アラート・ログは、ADRホームの`alert`サブディレクトリにXMLファイルとしても書き込まれますが、テキスト・アラート・ログは簡単に読むことができます。

Oracle Clusterwareで問題が発生した場合に、まず確認するのがアラート・ログです。Oracle Databaseインスタンスのアラート・ログとは異なり、Oracle Clusterwareアラート・ログのメッセージは、識別番号が付き、ドキュメント化とローカライズ(翻訳)が行われています。Oracle Clusterwareアラート・メッセージは、発生した重要なイベントまたはエラーのほとんどに対して書き込まれます。

ノート:



Oracle Clusterwareトレース・ファイルに書き込まれるメッセージとデータは、通常はドキュメント化も翻訳も行われません。主に My Oracle Support が問題を診断するために使用します。

インシデント・トレース・ファイル

Oracle Clusterwareプログラムで特定のエラーが発生すると、ADRインシデントが発生します。ほとんどのケースで、このようなエラーは診断のためにMy Oracle Supportに報告する必要があります。通常、1つのインシデントが発生すると、1つ以上の説明メッセージがOracle Clusterwareアラート・ログに生成されます。

アラート・メッセージに加えて、インシデントの影響を受けたプログラムによって、エラーに関連する診断データを含む特別なトレース・ファイルが別に生成されます。このようなインシデント固有のトレース・ファイルは、ADRホームの`trace`サブディレクトリではなく`incident`サブディレクトリに収集されます。エラーを報告する際に、通常のトレース・ファイルとインシデント・トレース・ファイルが収集されてオラクル社に提出されます。

関連項目:

インシデントとデータ収集の詳細は、[『Oracle Database管理者ガイド』](#)を参照してください。

その他の診断データ

Oracle Clusterwareでは、ADRデータの他にも問題の診断に関連する他のデータを収集または使用します。Oracle

Clusterware 12cリリース1 (12.1.0.2)以降、このデータはADRによって使用される同じベース・パス内の別のディレクトリ構造(ORACLE_BASE/crsdata/host_nameという形式)に存在します。この例で、ORACLE_BASEはGrid Infrastructureのインストール時に指定したOracleベース・パス、host_nameはホストの名前です。

任意のホストで、このディレクトリにはいくつかのサブディレクトリが含まれます。問題の発生時に重要になる2つのサブディレクトリの名前は、coreとoutputです。coreディレクトリは、コア・ファイルのために使用する通常のADRの場所が使用できない場合(ADRサービスがプログラムで初期化される前など)に、Oracle Clusterwareデーモン・コア・ファイルが書き込まれるところです。outputディレクトリは、Oracle ClusterwareデーモンがC標準出力と標準エラー・ファイルをリダイレクトする場所です。通常、これらのファイルの名前は、実行可能ファイル名にOUTという語と.trcファイル拡張子(トレース・ファイルと同様)が付いて構成されます。たとえば、Cluster Time Synchronization Serviceデーモンからリダイレクトされる標準出力の名前はoctssdOUT.trcです。一般的に、デーモンによってこれらのファイルに書き込まれる内容はごくわずかですが、特定の障害シナリオでは重要なデータが書き込まれる場合があります。

関連トピック

- [Oracle Database管理者ガイド](#)
- [Oracle Databaseユーティリティ](#)

診断収集スクリプト

Oracle Clusterwareでエラーが発生する場合は、diagcollection.pl診断収集スクリプトを実行して、Oracle Clusterwareからトレース・ファイルに診断情報を収集します。診断によって、My Oracle Supportが問題を解決するための追加情報を取得できます。Grid_home/binディレクトリから、このスクリプトをrootとして実行します。

構文

diagcollection.plスクリプトは、次の構文で使使します。

```
diagcollection.pl {--collect [--crs | --acfs | -all] [--chmos [--incidenttime time [-incidentduration time]]]
  [--adr location [--aftertime time [--beforetime time]]]
  [--crshome path | --clean | --coreanalyze]}
```



ノート:

diagcollection.pl スクリプトのすべての引数の前には、2 本のダッシュ(--)を付けます。

パラメータ

[表J-3](#)に、diagcollection.plスクリプトで使使されるパラメータのリストおよび説明を示します。

表J-3 diagcollection.plスクリプトのパラメータ

パラメータ	説明
--collect	このパラメータは、次のいずれかの引数とともに使使します。 ● --crs: Oracle Clusterware の診断情報を収集する場合に、この引数を使使します。 ● --acfs: Oracle ACFS の診断情報を収集する場合に、この引数を使使

パラメータ	説明
	します。 ノート: この引数は、UNIX システムでのみ使用できます。 ● <code>--all</code>: CHM (OS)データを除くすべての診断情報を収集する場合に、この引数を使用します。 ノート: これはデフォルトです。 ● <code>--chmos</code>: 次の CHM の診断情報を収集する場合に、この引数を使用します。 <code>--incidenttime time</code>: 指定した時刻から CHM (OS)データを収集する場合に、この引数を使用します。 ノート: 時刻書式は MM/DD/YYYYHH24:MM:SS です。 <code>--incidentduration time</code>: 指定した時刻の後の期限内に CHM (OS)データを収集する場合に、この引数を <code>--incidenttime</code> とともに使用します。 ノート: 時刻書式は HH:MM です。 <code>--incidentduration</code> を使用しない場合は、 <code>--incidenttime</code> で指定した時刻の後にすべての CHM (OS)データが収集されます。 ● <code>--adr location</code>: 自動診断リポジトリ・コマンド・インタプリタ(ADRCI)はこの引数を使用して、ADR のために診断情報を収集する場所を指定します。 関連項目: ADRCI の詳細は、『Oracle Database ユーティリティ』を参照してください。 ● <code>--aftertime time</code>: 指定した時刻の後にアーカイブを収集する場合に、この引数を <code>--adr</code> 引数とともに使用します。 ノート: 時刻書式は YYYYMMDDHHMISS24 です。 ● <code>--beforetime time</code>: 指定した時刻の前にアーカイブを収集する場合に、この引数を <code>--adr</code> 引数とともに使用します。 ノート: 時刻書式は YYYYMMDDHHMISS24 です。 ● <code>--crshome path</code>: Oracle Clusterware ホームの場所を上書きする

パラメータ	説明
	場合に、この引数を使用します。
	ノート: diagcollection.pl スクリプトは、通常、システム構成 (olr.loc ファイルまたは Windows レジストリのいずれか) から Oracle Clusterware ホームの場所を導出するため、この引数は不要です。
--clean	diagcollection.pl スクリプトで収集された診断情報をクリーンアップする場合に、このパラメータを使用します。 ノート: このパラメータは、--collect とともに使用することはできません。
--coreanalyze	コア・ファイルから情報を抽出してテキスト・ファイルに格納する場合に、このパラメータを使用します。 ノート: このパラメータは、UNIX システムでのみ使用できます。

Oracle拡張クラスタの記憶域の分割

複数の異なるサイト間のプライベート・ネットワークがオンラインで使用可能になったが、その記憶域の分割が失敗した場合に記憶域の分割が発生します。

Oracle Automatic Storage Management (Oracle ASM)が、3つのサイト(2つのデータ・サイトと1つのクォーラム・サイト)を持つ一般的な拡張クラスタ構成で記憶域の分割を検出すると、いずれかのデータ・サイトが終了し、それ自体およびそこに含まれていたノードは他から隔離されます。Oracle ASMが隔離されたサイトで起動しようすると、エラーCRS-2971が発生します。

次のように問題を解決します。

1. 記憶域の分割の原因となったサイト間接続の問題を解決します。
2. 次のようにして、すべてのOracle ASMディスク・グループが、隔離されていないサイト上にマウントされていることを確認します。

```
SELECT group_number, name, state FROM v$asm_diskgroup_stat;
```

3. ディスク・グループごとに次のコマンドを実行して、これらのディスク・グループに属するオンライン・ディスクのリストを取得します。

```
SELECT path FROM v$asm_disk_stat WHERE group_number=group_number AND state = 'NORMAL' AND mode_status = 'ONLINE';
```

4. 前のステップで取得したパスごとに、次のように、隔離されたサイトからディスクにアクセスできることを確認します。

```
asmcmd lsdsk -I --member 'path'
```

5. 前述の検証が成功した場合は、次のように隔離されたサイトを再インスタンス化します。

```
crsctl modify cluster site site_name -s rejuvenate
```

ローリング・アップグレードおよびドライバのインストールに関する問題

アップグレード中に、Oracle Clusterware `root.sh`スクリプトを実行していると、次のメッセージが表示される場合があります。

- ACFS-9427 「ADVM/ACFSドライバのアンロードに失敗しました。」システムの再起動をお勧めします。
- ACFS-9428 「ADVM/ACFSドライバのロードに失敗しました。」システムの再起動をお勧めします。

初期(最初の)ノードのアップグレード中にこのようなエラー・メッセージが表示された場合、次の手順を行います。

1. クラスタ内の他のすべてのノードでアップグレードを完了します。
2. 初期ノードを再起動します。
3. 初期ノードで、`root.sh`スクリプトを再度実行します。
4. Oracle Grid Infrastructureをインストールしたユーザーとして`Grid_home/gridSetup - executeConfigTools`スクリプトを実行し、アップグレードを完了します。

初期ノード(インストールを開始したノード)以外のノード:

1. エラーが発生したノードを再起動します。
2. エラーが発生するノードで、`root`として`oraInstRoot.sh`スクリプトを実行します。
3. ディレクトリをGridホームに変更し、エラーが発生するノードで`root.sh`スクリプトを実行します。

ゾーン委任のテスト

関連項目:

この手順で言及されているCRSCTLコマンドの使用の詳細は、[\[Oracle Clusterware制御\(CRSCTL\)ユーティリティ・リファレンス\]](#)を参照してください。

次の手順を使用して、ゾーン委任をテストします。

1. `root`として次のコマンドを実行し、GNS VIPを起動します。

```
# crsctl start ip -A IP_name/netmask/interface_name
```

`interface_name`は、パブリック・ネットワークのパブリック・インタフェースおよびネットマスクである必要があります。

2. 次のコマンドを実行し、GNS VIPでテストDNSサーバーを起動します(ポート番号が1024より小さい場合は`root`としてこのコマンドを実行する必要があります)。

```
# crsctl start testdns -address address [-port port]
```

このコマンドによって、指定したIPおよびポートで、DNSが転送したパケットをリスニングするテストDNSサーバーが起動されます。

3. `root`として次のコマンドを実行し、GNS VIPが他のノードから到達可能であることを確認します。

```
crsctl status ip -A IP_name
```

4. 次のコマンドを実行し、直接DNSサーバーに問い合わせます。

```
crsctl query dns -name name -dnsserver DNS_server_address
```


このコマンドは、次のエラーが発生して失敗します。

CRS-10023: 名前asdf.foo.comのドメイン名参照に失敗しました。オペレーティング・システム・エラー: ホスト名参照エラー

Grid_home/log/host_name/client/odnsd_*.logを参照して、問合せがテストDNSサーバーで受信されたかどうかを確認します。これにより、DNS問合せがファイアウォールによってブロックされていないことが確認されます。

5. 次のコマンドを実行し、GNSドメイン問合せのDNS委任を問い合わせます。

```
crsctl query dns -name name
```

ノート:



このステップと前のステップの違いは、-dnsserver DNS_server_address オプションを指定しないことのみです。これにより、コマンドは/etc/resolv.conf で構成されているネーム・サーバーに問い合わせます。前のステップと同様に、コマンドは同じエラーが発生して失敗します。再度、odnsd*.log を参照して、odnsd で問合せが受信されたことを確認します。ステップ 5 が成功し、ステップ 6 が成功しない場合は、DNS 構成を確認する必要があります。

6. 次のコマンドを実行し、テストDNSサーバーを停止します。

```
crsctl stop testdns -address address
```

7. rootとして次のコマンドを実行し、GNS VIPを停止します。

```
crsctl stop ip -A IP_name/netmask/interface_name
```

Oracle Clusterwareアラート

すでに説明したように、Oracle Clusterwareによって様々な重要なイベントに関するメッセージがADRアラート・ログに書き込まれます。通常、アラート・ログのメッセージはローカライズ(翻訳)され、メッセージの詳細の検索に使用できるメッセージIDが付いています。Oracle Clusterwareに問題があるように見える場合には、まずアラート・ログを確認します。

次に、2つの異なるCRSデーモン・プロセスのアラート・ログ・メッセージの例を示します。

```
2014-07-16 00:27:22.074 [CTSSD(12817)]CRS-2403:The Cluster Time Synchronization Service on host stnsp014 is in observer mode.
2014-07-16 00:27:22.146 [CTSSD(12817)]CRS-2407:The new Cluster Time Synchronization Service reference node is host stnsp013.
2014-07-16 00:27:22.753 [CTSSD(12817)]CRS-2401:The Cluster Time Synchronization Service started on host stnsp014.
2014-07-16 00:27:43.754 [CRSD(12975)]CRS-1012:The OCR service started on node stnsp014.
2014-07-16 00:27:46.339 [CRSD(12975)]CRS-1201:CRSD started on node stnsp014.
```

次に、クラスタ再構成後のOracleクラスタ時刻同期化サービス(OCTSS)の起動例を示します。

```
2014-07-15 23:51:17.532 [CTSSD(12813)]CRS-2403:The Cluster Time Synchronization Service on host stnsp014 is in observer mode.
2014-07-15 23:51:18.292 [CTSSD(12813)]CRS-2407:The new Cluster Time Synchronization Service reference node is host stnsp013.
2014-07-15 23:51:18.961 [CTSSD(12813)]CRS-2401:The Cluster Time Synchronization Service started on host stnsp014.
```

Diagnostic Record Unique IDを使用したアラート・メッセージ

Oracle Database 11gリリース2(11.2)以降では、特定のOracle Clusterwareメッセージに、「(:」および「:)」で囲まれたテキスト識別子が含まれます。通常、この識別子は「Details in...」で始まるメッセージ・テキストの一部で、次の例のようなOracle Clusterware診断ログ・ファイルのパスおよび名前が含まれます。この識別子はDRUID (Diagnostic Record Unique ID)と呼ばれます。

```
2014-07-16 00:18:44.472 [ORAROOTAGENT(13098)]CRS-5822:Agent
'/scratch/12.1/grid/bin/orarootagent_root' disconnected from server. Details at
(:CRSAGF00117:) in /scratch/12.1/grid/log/stnsp014/agent/crsd/orarootagent_
root/orarootagent_root.log.
```

DRUIDは、外部製品のメッセージを、診断ログ・ファイルのエントリおよび内部Oracle Clusterwareプログラム・コードの場所に関連付けるために使用されます。これらはユーザーには直接関係ありませんが、主に、問題を診断するときにMy Oracle Supportによって使用されます。

用語集

アクション・スクリプト

リソースのstart、stop、stopおよびcheckアクションを定義するスクリプト。startアクションはリソースの起動中に起動され、stopアクションはリソースを停止するために起動され、checkアクションはリソースの動作ステータスのチェック中に起動されます。

管理者管理型

データベース・リソースを実行するサーバーをデータベース管理者が定義し、必要に応じてリソースを手動で配置します。これは、以前のリリースで使用されていた管理方法です。

エージェント

リソースを管理するためのエージェント・フレームワークおよびユーザー・コードを含むプログラム。

エージェント・フレームワーク

カスタマイズ・アプリケーションを管理するためにアプリケーション固有のコードをユーザーがプラグインできるようにするCライブラリ。

可用性ディレクティブ

クラスタに対してサーバーが削除または追加された際にシステムを再構成するためのOracle Clusterwareに対する指示。

カーディナリティ

[リソース](#)を同時に実行できるサーバーの数。

クライアント・クラスタ

マルチクラスタ環境では、クライアント・クラスタは、[サーバー・クラスタ](#)を使用して名前を通知します。

クラスタ

エンド・ユーザーおよびアプリケーションからは1つのサーバーとして認識される、相互に接続された複数のコンピュータまたはサーバー。

クラスタ対応

クラスタウェアを使用してデプロイされるように設計されたアプリケーション。

クラスタ管理者

設定されたポリシーおよび権限に基づいてクラスタの特定部分を管理できる管理者。

クラスタ構成ポリシー

システムで定義されているサーバー・プールごとに1つのみの定義を含む、ポリシー管理クラスタ用のドキュメント。

クラスタ構成ポリシー・セット

1つ以上の構成ポリシーを含むクラスタに構成されているすべてのサーバー・プールの名前を定義するドキュメント。一度に有効にできるのは1つのポリシーのみですが、管理者は、ビジネスのニーズおよびシステムの要求に従って、様々な日付または時刻で有効になるように様々なポリシーを設定できます。

クラスタ状態モニター

オペレーティング・システムおよびクラスタ・リソースに関連する状態の悪化および障害を検出および分析します。

クラスタ・リソース

クラスタ環境を認識し、ノード間のスイッチオーバーとフェイルオーバーの対象となるリソース。

クラスタ時刻同期化サービス

クラスタ内のすべてのノードの内部クロックがすべて同期化されるようにする時刻同期メカニズム。

クラスタ検証ユーティリティ(CVU)

共有ストレージ・デバイス、ネットワーク構成、システム要件、Oracle Clusterware、グループ、ユーザーなどのOracle RACの様々なコンポーネントを検証するツール。

データベース・クラウド

高パフォーマンス、可用性、およびリソースの最適使用率を確保しながら、1つ以上のグローバル・サービスを提供する単一仮想サーバーに対して、グローバル・サービスおよびロード管理フレームワークによって統合されている一連のデータベース。

依存性

2つ以上のリソース間の関係、およびそれらの間で示される相互作用。

ディスク・グループ

Oracle ASMディスクは、Oracle ASMで1つの単位として管理されるディスクの集合です。ディスク・グループ内では、Oracle Databaseファイルのファイル・システム・インタフェースがOracle ASMにより公開されます。ディスク・グループ内に保存されたファイルの内容は、均等に分散されるか、またはストライプ化されるため、ホット・スポットがなくなり、ディスク間のパフォーマンスが均一になります。必要に応じて、ディスク・グループ内でOracle ASMファイルをミラー化することもできます。ディスク・グループ内のディスクのパフォーマンスは、RAWデバイスのパフォーマンスと同程度です。

動的ホスト構成プロトコル(DHCP)

インターネット・プロトコル・ネットワークにおける操作用の構成情報を取得するために、デバイス(DHCPクライアント)によって使

用されるネットワーク・アプリケーション・プロトコル。このプロトコルによってシステム管理のワークロードが軽減され、手動操作をほとんどまたはまったく行わずに、デバイスをネットワークに追加できます。

fixupスクリプト

インストールの最小要件を満たしていない場合はOracle Universal Installerによって検知され、要件を満たしていないシステム構成の多くを解決するために、修正スクリプトと呼ばれるシェル・スクリプト・プログラムが作成されます。Oracle Universal Installerは不完全なタスクを検出すると、修正スクリプトを作成してから別のターミナル・セッションで修正スクリプトを実行するように求めます。特定のCVUコマンドで `-fixup` フラグを使用して、修正スクリプトを生成することもできます。

ゴールド・イメージ

ホーム固有の詳細情報が使用できないOracleホームにインストールされているソフトウェアのみのコピー。これにより、有効で使用可能なOracleホームとして機能させるために、Oracleホームのイメージを、新しいファイル・システム上の新しいホストに簡単にコピーできます。

一般にゴールド・イメージはアプリケーションに対応していますが、このドキュメントのコンテキストでは、ゴールド・イメージはOracle Databaseに対応しています。また、ゴールド・イメージは出荷ホームではなく、ゴールド・イメージをデプロイするときにOracle Universal Installerを実行する必要があることを意味しています。

Gridホーム・クライアント

Gridホーム・クライアントは、ゴールド・イメージをプロビジョニングするためにフリー・パッチ適用およびプロビジョニング・サーバーをサブスライブするリモートの[Oracle Grid Infrastructure](#) クラスタです。

Oracle Grid Infrastructure

エンタープライズ・グリッド・アーキテクチャ用のインフラストラクチャを提供するソフトウェア。クラスタの場合、このソフトウェアにはOracle Clusterwareと Oracle ASM が含まれます。スタンドアロン・サーバーの場合、このソフトウェアにはOracle RestartとOracle ASMが含まれます。Oracle Database 12cでは、これらのインフラストラクチャ製品が組み合されて、Oracle Grid Infrastructureホーム(Grid_home)という1つのソフトウェア・インストールになっています。

グリッド・ネーミング・サービス(GNS)

委任を受けた標準DNSゾーン内のホスト名をそのゾーンのIPアドレスにマッピングすることで、ホスト名を解決する汎用サービスです。GNSを使用すると、Oracle RACデータベース・ノードに対して動的ホスト構成プロトコル(DHCP)によるアドレスが使用されるため、デプロイメントが容易になります。また、GNSでは、SCANリスナーから戻されたホスト名の解決も行われます。

ハブ・アンカー

サポートされている1つ以上の他のノードのクラスタ・メンバーシップのための接続ポイントとして機能する、Oracle Flex Cluster内のハブ・ノード。これらのノードは、単一ハブ・アンカーとハートビートを交換します。

ハブ・ノード

他のサーバーと密接続して、共有記憶域に直接アクセスする、[Oracle Flex Cluster](#)内のノード。

イメージ

フリート・パッチ適用およびプロビジョニング・サーバーに存在する、Oracleデータベース・ホーム・ソフトウェアのマスター・コピー。

IPv4

インターネット・プロトコル・バージョン4。IPv4は、現在の標準のIPプロトコルです。IPv4は32ビット(4バイト)アドレスを使用します。通常、ドット区切りの10進数表記で表現されます。各オクテットの10進数の値は、192.168.2.22のようにピリオドで区切られます。

IPv6

インターネット・プロトコル・バージョン6。IPv4の後継として設計されたプロトコル。IPv6では、IPアドレスは通常、2001:0DB8:0000:0000:0000:0000:1428:57ABのように、コロンで区切られた8つのフィールドの16進数の値で表現されます。2001:DB8::1428:57ABのように、値が0のフィールドを圧縮することもできます。

ローカル・リソース

クラスタのノード上で実行され、ノードの範囲外のものは認識しないリソース。

OCRライター

各CRSDプロセスもOCRサーバーとして機能します。クラスタのCRSDプロセスの1つは、ディスク・グループまたはファイル、あるいはブロックまたはRAWデバイスに対してI/Oを実行するOCRサーバーです。

Oracle Automatic Storage Management (Oracle ASM)

Oracle ASMはOracle ClusterwareおよびOracle RACで使用される記憶域ディスクをディスク・グループ内で管理します。デフォルトでは、Oracle ClusterwareはOracle ASMを使用してOCRおよび投票ファイルを格納します。

Oracle Cluster Registry (OCR)

クラスタ・ノード・リストに関する情報およびインスタンスからノードへのマッピング情報を管理するOracle RAC構成情報のリポジトリ。OCRは、カスタマイズされたアプリケーション用のOracle Clusterwareリソース・プロファイルに関する情報も管理します。

Oracle Clusterware

接続されたサーバーのグループ(クラスタ)が動作すること、または1つの単位として制御されることを可能にするソフトウェア。

Oracle Clusterwareスタック

Oracle Clusterwareスタックには、Oracleクラスタ・レディ・サービス、イベント・マネージャ、クラスタ同期サービスおよびOracle ASM(使用する場合)が含まれます。

Oracle Flex Cluster

[ハブ・ノード](#)と、サポートされる他のノードで構成された大きなクラスタ。ハブ・ノードは、現在のメンバーシップ・アルゴリズムを使用

してクラスタを形成し、他のノードは、単一のハブ・ノードに対してメンバーシップを接続します。

ポリシー管理型

データベース管理者は、データベース・リソースを実行するサーバー・プール(汎用および空きを除く)を指定します。Oracle Clusterwareによって、データベース・リソースがサーバーに配置されます。

リーダー・ノード

リーダー・ノードは、Oracle RACデータベース・インスタンスが実行されているノードです。

リソース

Oracle Clusterwareによって管理されるデータベース、アプリケーションまたはプロセス。

リソース・グループ

Oracle Clusterwareが単一エンティティとして管理および監視する、論理的に関連した様々なリソースで構成されるアプリケーションのコンテナ。

リソースの状態

特定の時間の特定のリソースの状態。これによってクラスタの可用性を確認します。

リソース・タイプ

リソースが[クラスタ・リソース](#)または[ローカル・リソース](#)のいずれであるかを定義します。

サーバーのカテゴリ化

一連のサーバー属性を使用して、クラスタ内のサーバーを様々なカテゴリに分類する方法。管理者は、受け入れるカテゴリを制限するようにサーバー・プールを構成できます。サーバー・カテゴリは、SERVER_CATEGORYパラメータの属性を提供して作成します。

サーバー・クラスタ

マルチクラスタ環境では、サーバー・クラスタは、グリッド・ネーミング・サービス(GNS)・プロセスを実行するクラスタです。

サーバー・プール

アプリケーション、データベースまたはその両方をホスティングするグループに論理的に分割されているクラスタ内のサーバー。

単一クライアント・アクセス名(SCAN)

パブリック・ネットワークで3つのIPアドレスに解決される1つの名前。

起動試行の評価

Oracle Clusterwareがリソースを起動するときに実行する処理。この処理の間、Oracle Clusterwareはリソースのプロファイルに含まれるリソースの依存性を考慮します。

状態

特定のリソースの状態を示す一連の値。

作業用コピー

ソフトウェアのみがインストールされたOracleホーム、またはインスタンス化された構成済のOracleホームのいずれかにできる[ゴールド・イメージ](#)のOracle ACFSスナップショット。

ゾーン・データ

Oracle Cluster Registryに保存されたGNSデーモン・レコード。

索引

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#)

A

- アクセス制御リスト(ACL) [2.1.1](#)
- ACTION_FAILURE_EVENT_TEMPLATE
 - リソース属性 [B.1.2.1](#)
- ACTION_SCRIPT
 - リソース属性 [B.1.1.2](#)
- ACTION_TIMEOUT
 - リソース属性 [B.1.1.3](#)
- アクション・エントリ・ポイント
 - 定義 [8.2](#)
- ACTIONS
 - リソース属性 [B.1.1.4](#)
- アクション・スクリプト [8.1.4](#)
 - アクション [8.2](#)
- ACTIVE_PLACEMENT
 - リソース属性 [B.1.1.5](#)
- クラスタへのノードの追加 [6.2.1](#)
- アドレス, 手動構成 [1.2.4.2](#)
- 管理ツール
 - 概要および概念 [1.9](#)
- ADR [J.2](#)
 - ディレクトリ構造 [J.2](#)
- ADRホーム [J.2](#)
- AGENT_FILENAME
 - リソース属性 [B.1.1.6](#)
- エージェント・プログラム [8.1.3.1](#)
- エージェント [8.1.3.1](#)
 - appagent [1.3.1.2](#), [8.1.3.1](#)
 - appagent.exe [1.3.1.2](#), [8.1.3.1](#)
 - scriptagent [1.3.1.2](#), [8.1.3.1](#)
 - scriptagent.exe [1.3.1.2](#), [8.1.3.1](#)
- ALERT_TEMPLATE
 - リソース属性 [B.1.1.7](#)
- アラート・メッセージ
 - CRSD [J.7](#)
- API
 - エージェント・フレームワーク

- clsagfw_add_type() [G.8.1](#)
- clsagfw_check_resource() [G.8.2](#)
- clsagfw_create_attr_iterator() [G.8.3](#)
- clsagfw_delete_cookie() [G.8.4](#)
- clsagfw_exit2() [G.8.5](#)
- clsagfw_get_attr_from_iterator() [G.8.6](#)
- clsagfw_get_attrvalue() [G.8.7](#)
- clsagfw_get_check_type() [G.8.8](#)
- clsagfw_get_cmdid() [G.8.9](#)
- clsagfw_get_cookie() [G.8.10](#)
- clsagfw_get_request_action_name() [G.8.11](#)
- clsagfw_get_resource_id() [G.8.12](#)
- clsagfw_get_resource_name() [G.8.13](#)
- clsagfw_get_retry_count() [G.8.14](#)
- clsagfw_get_type_name() [G.8.15](#)
- clsagfw_init() [G.8.16](#)
- clsagfw_is_cmd_timedout() [G.8.17](#)
- clsagfw_log() [G.8.18](#)
- clsagfw_reset_attr_iterator() [G.8.20](#)
- clsagfw_send_status2() [G.8.21](#)
- clsagfw_set_cookie() [G.8.22](#)
- clsagfw_set_entrypoint() [G.8.23](#)
- clsagfw_set_exitcb() [G.8.24](#)
- clsagfw_set_resource_state_label() [G.8.25](#)
- clsagfw_startup() [G.8.26](#)
- clscrs_stat2 [H.1.4](#)
- コマンド評価
 - clscrs_whatif_add_server [H.4.9](#)
 - clscrs_whatif_delete_server [H.4.10](#)
 - clscrs_whatif_register_serverpool [H.4.7](#)
 - clscrs_whatif_set_activepolicy [H.4.1](#)
 - clscrs_whatif_unregister_serverpool [H.4.8](#)
- その他
 - clscrs_get_error_details [H.7.1](#)
 - clscrs_request_action [H.7.2](#)
 - clscrs_restart_resource [H.7.3](#)
 - clscrs_start_resource_in_pools [H.7.4](#)
 - clscrs_stat3 [H.6.1](#)
 - clscrs_stop_resource_in_pools [H.7.5](#)
- サーバーのカテゴリ化
 - clscrs_get_server_by_category [H.5.5](#)
 - clscrs_register_server [H.5.6](#)
 - clscrs_register_servercategory [H.5.3](#)

- clscrs_servercategory_create [H.5.1](#)
- clscrs_servercategory_destroy [H.5.2](#)
- clscrs_unregister_servercategory [H.5.4](#)
- What-If
 - clscrs_whatif_fail_resource [H.4.2](#)
 - clscrs_whatif_register_resource [H.4.3](#)
 - clscrs_whatif_relocate_resource [H.4.4](#)
 - clscrs_whatif_relocate_server [H.4.11](#)
 - clscrs_whatif_start_resource [H.4.5](#)
 - clscrs_whatif_stop_resource [H.4.6](#)
- アプリケーション・エージェント [1.3.1.2](#), [8.1.3.1](#)
- アプリケーション・メンバー・クラス [1.8](#)
- アプリケーション・リソース
 - 起動 [8.4.2](#)
 - 停止 [8.4.4](#)
- アプリケーション
 - VIPアドレスの定義 [1.12](#)
 - 高可用性 [1.12](#)
 - CLSCRSコマンドを使用した管理 [H.1](#)
 - 配置ポリシー [8.3.5](#)
 - リソースとして登録 [8.3](#)
 - 再配置 [8.4.3](#)
- アプリケーションVIP
 - 削除 [8.3.1](#)
- appvipcfg create [8.3.1](#)
- attraction起動依存性 [8.2.3.1.1](#)
 - 修飾子 [B.1.1.36](#)
- AUTO_START
 - リソース属性 [B.1.1.8](#)
- autoconfig [1.2.4](#), [2.5.2](#)
- 自動アドレス構成 [2.5.2.2](#)
- 自動クラスタ・アドレス構成 [2.5.2](#)
- 自動診断リポジット
 - 「ADR」を参照

B

- バックグラウンド・プロセス [1.3.1](#)
- Baseboard Management Controller(BMC)
 - 「Intelligent Platform Management Interface(IPMI)」を参照:
- ブラックボックス仮想マシン [8.1.1.1](#)
- ブロック単位のチェックサム操作 [1.3.2](#)
- 組込みエージェント [8.1.3.1](#)

C

- キャッシュ・フュージョン
 - 通信 [D.2.4](#)
- CARDINALITY
 - リソース属性 [B.1.1.9](#)
 - リソース・グループ属性 [B.1.1.9](#)
- CARDINALITY_ID
 - リソース属性 [B.1.1.10](#)
 - リソース・グループ属性 [B.1.1.10](#)
- インタフェース名の変更, 結果 [2.10.2.3](#)
- クラスタ・モードの変更 [4.2.1](#)
- VIPアドレスの変更 [2.10.1](#)
- CHECK_INTERVAL
 - リソース属性 [B.1.1.11](#)
- CHECK_TIMEOUT
 - リソース属性 [B.1.1.12](#)
- チェックサム操作
 - ブロック単位 [I.3.2](#)
- CHM
 - 「クラスタ状態モニター(CHM)」を参照:
- 暗号スイート [1.3.1.2.1](#)
- CLEAN_TIMEOUT
 - リソース属性 [B.1.1.13](#)
- クライアント・クラスタ [1.2.4](#)
- クライアント・データ・ファイル
 - 共有GNSを構成するための使用 [2.6.4.2](#)
- クライアント側の診断インフラストラクチャ(CRSD)
 - アラート [J.7](#)
- クローニング
 - Oracle Clusterware [7](#)
 - Oracle Grid Infrastructure [7](#)
- クローニング, クラスタウェアの概要 [1.11](#)
- CLSCRS API
 - 非推奨 [H.1.3](#)
 - フィルタ [E.1.2](#)
 - 対話型 [H.2](#)
 - 非対話型 [H.3](#)
 - 概要 [H.1.1](#)
- CLSCRSコマンド
 - 概要 [H.1](#)
- CLSD-1009メッセージ

- 解決 [5.1.2.5](#)
- CLSD-1011メッセージ
 - 解決 [5.1.2.5](#)
- クラスタ
 - GNSクライアント・クラスタへの変換 [2.6.4.4](#)
- CLUSTER_INTERCONNECTインタフェース
 - OIFCFGによる指定 [D.2.4](#)
- cluster_resource [8.1.2](#)
- クラスタ対応リソース [8.1.2](#)
- クラスタ構成ポリシー [3.1.7](#)
- クラスタ状態モニター(CHM) [1.9](#)
- クラスタ状態モニター(CRF)
 - デバッグ [E.5.2.1](#)
- クラスタ・インターコネクト
 - キャッシュ・フュージョン通信 [D.2.4](#)
 - プライベート・ネットワーク・アドレスの変更 [2.10.2.1](#)
- クラスタ・モード
 - 変更 [4.2.1](#)
- クラスタ・ノード
 - 追加と削除 [6](#)
 - Windowsシステムからの削除 [6.3.2](#)
- クラスタ・レディ・サービス(CRS)
 - デバッグ [E.5.2.1](#)
 - 定義 [1.3.1.1](#)
- クラスタ・リソース [8.1.2](#)
- クラスタ・リソース・アクティビティ・ログ [J.1](#)
- クラスタ・リソースの障害
 - 監視 [J.1](#)
- クラスタ
 - 管理者管理 [2](#)
 - 変換 [2.6.4](#)
 - Oracle Flex Clusterへの変換 [4.2.1](#)
 - ポリシー管理 [2](#)
- クラスタ記憶域
 - OCRによる記録 [1.2.3](#)
- クラスタ同期サービス(CSS) [1.3.1.1](#)
 - デバッグ [E.5.2.1](#)
 - 定義 [1.3.1.1](#)
- クラスタの時間管理 [1.13](#)
- クラスタ時刻同期化サービス(CTSS) [1.13](#)
 - デバッグ [E.5.2.1](#)
- クラスタ検証ユーティリティ [A.1.2.6](#)
 - 「CVU」、「cluvfy」も参照

- クラスタ検証ユーティリティ [A.1.2.6](#)
 - 「CVU」も参照
 - 非推奨およびサポート対象外のコマンド [A.1.2.6](#)
- クラスタ検証ユーティリティ(CVU)
 - 「CVU」を参照
- クラスタウェア, クローニングの概要 [1.11](#)
- cluvfy
 - 「CVU」を参照
- SRVCTLとカンマ区切りリスト [F.1](#)
- 互換性
 - Oracle Clusterware、Oracle ASMおよびデータベース [1.5.1](#)
- 構成
 - OCRの再初期化 [5.1.6](#)
- 構成ウィザード
 - 複数ノードの構成 [2.2.2](#)
- 構成
 - 投票ファイル [5](#)
- クラスタの変換 [2.6.4](#)
- CRITICAL_RESOURCES
 - リソース・グループ属性 [B.1.1.14](#)
- クラスタ間依存性プロキシ [2.11](#)
- CRS
 - 「クラスタ・レディ・サービス(CRS)」を参照:
- CRSCTL
 - Oracle Clusterwareのステータスのチェック [5.1.2.3](#)
 - コマンド評価 [1.10](#)
 - コマンド
 - add category [E.3.2](#)
 - add cluster site [E.3.18](#)
 - add crs administrator [E.3.34](#)
 - add policy [E.3.77](#)
 - add resource [E.2.4](#)
 - add resourcegroup [E.2.20](#)
 - add resourcegrouptype [E.2.35](#)
 - add serverpool [E.3.102](#)
 - add type [E.2.40](#)
 - add wallet [E.3.116](#)
 - check cluster [E.3.6](#)
 - check crs [E.3.27](#)
 - check css [E.2.1](#)
 - check ctss [E.3.60](#)
 - check evm [E.2.2](#)
 - check has [E.4.1](#)

- check resource [E.3.85](#)
- check resourcegroup [E.2.21](#)
- config crs [E.3.28](#)
- config has [E.4.2](#)
- create member_cluster_configuration [E.3.71](#)
- create policyset [E.3.82](#)
- delete category [E.3.3](#)
- delete cluster site [E.3.19](#)
- delete crs administrator [E.3.35](#)
- delete css votedisk [E.3.58](#)
- delete member_cluster_configuration [E.3.72](#)
- delete node [E.3.74](#)
- delete policy [E.3.78](#)
- delete resource [E.2.5](#)
- delete resourcegroup [E.2.22](#)
- delete resourcegrouptype [E.2.36](#)
- delete serverpool [E.3.103](#)
- delete type [E.2.41](#)
- delete wallet [E.3.117](#)
- disable crs [E.3.29](#)
- disable has [E.4.3](#)
- discover dhcp [E.3.61](#)
- enable crs [E.3.30](#)
- enable has [E.4.4](#)
- eval activate policy [E.3.79](#)
- eval add resource [E.2.6](#)
- eval add resourcegroup [E.2.23](#)
- eval add server [E.3.92](#)
- eval add serverpool [E.3.104](#)
- eval delete server [E.3.93](#)
- eval delete serverpool [E.3.105](#)
- eval fail resource [E.2.7](#)
- eval fail resourcegroup [E.2.24](#)
- eval modify resource [E.2.9](#)
- eval modify serverpool [E.3.106](#)
- eval relocate resource [E.2.8](#)
- eval relocate resourcegroup [E.2.25](#)
- eval relocate server [E.3.94](#)
- eval start resource [E.2.10](#)
- eval start resourcegroup [E.2.26](#)
- eval stop resource [E.2.11](#)
- eval stop resourcegroup [E.2.27](#)
- export resourcegroup [E.2.28](#)

- get calog maxsize [J.1.2](#)
- get calog retentiontime [J.1.3](#)
- get clientid dhcp [E.3.62](#)
- get cluster class [E.3.9](#), [E.3.55](#)
- get cluster configuration [E.3.10](#)
- get cluster extended [E.3.12](#)
- get cluster hubsize [E.3.13](#)
- get cluster mode [E.3.15](#)
- get cluster name [E.3.17](#)
- get cluster tlsciphersuite [E.3.22](#)
- get cluster type [E.3.23](#)
- get cpu equivalency [E.3.25](#)
- get css [E.3.44](#)
- get css ipmiaddr [E.3.49](#)
- get hostname [E.2.3](#)
- get nodename [E.3.75](#)
- get node role [E.3.76](#)
- getperm resource [E.2.12](#)
- getperm serverpool [E.3.107](#)
- getperm type [E.2.42](#)
- get resource use [E.3.87](#)
- get server css_critical [E.3.98](#)
- get server label [E.3.100](#)
- get tracefileopts [E.2.38](#)
- lsmodules [E.3.70](#)
- modify category [E.3.4](#)
- modify cluster site [E.3.20](#)
- modify policy [E.3.80](#)
- modify policyset [E.3.83](#)
- modify resource [E.2.13](#)
- modify resourcegroup [E.2.29](#)
- modify resourcegrouptype [E.2.37](#)
- modify server [E.3.95](#)
- modify serverpool [E.3.108](#)
- modify type [E.2.43](#)
- modify wallet [E.3.118](#)
- pin css [E.3.45](#)
- query calog [J.1.1](#)
- query cluster site [E.3.21](#)
- query crs activeversion [E.3.33](#)
- query crs administrator [E.3.36](#)
- query crs autostart [E.3.37](#)
- query crs releasepatch [E.3.39](#)

- query crs releaseversion [E.3.40](#)
- query crs site [E.3.41](#)
- query crs softwarepatch [E.3.42](#)
- query crs softwareversion [E.3.43](#)
- query css ipmiconfig [E.3.52](#)
- query css ipmidevice [E.3.54](#)
- query css votedisk [E.3.59](#)
- query dns [E.3.66](#)
- query has releaseversion [E.4.5](#)
- query has softwareversion [E.4.6](#)
- query member_cluster_configuration [E.3.73](#)
- query socket udp [E.3.111](#)
- query wallet [E.3.119](#)
- release dhcp [E.3.63](#)
- relocate resource [E.2.14](#), [E.3.86](#)
- relocate resourcegroup [E.2.30](#)
- relocate server [E.3.96](#)
- replace discoverystring [E.3.65](#)
- replace votedisk [E.3.115](#)
- request action [E.3.1](#)
- request dhcp [E.3.64](#)
- restart resource [E.2.15](#)
- restart resourcegroup [E.2.31](#)
- set calog maxsize [J.1.4](#)
- set calog retentiontime [J.1.5](#)
- set cluster disabledtlsciphersuite [E.3.11](#)
- set cluster hubsize [E.3.14](#)
- set cluster mode [E.3.16](#)
- set cluster type [E.3.24](#)
- set cpu equivalency [E.3.26](#)
- set crs autostart [E.3.38](#)
- set css [E.3.46](#)
- set css ipmiaddr [E.3.50](#)
- set css ipmiadmin [E.3.51](#)
- set css noautorestart [E.3.56](#)
- set log [E.5.1](#)
- setperm resource [E.2.16](#)
- setperm serverpool [E.3.109](#)
- setperm type [E.2.44](#)
- set resource use [E.3.88](#)
- set server css_critical [E.3.99](#)
- set server label [E.3.101](#)
- set tracefileopts [E.2.39](#)

- start cluster [E.3.7](#)
- start crs [E.3.31](#)
- start has [E.4.7](#)
- start ip [E.3.67](#)
- start resource [E.2.17](#)
- start resourcegroup [E.2.32](#)
- start rollingpatch [E.3.89](#)
- start rollingupgrade [E.3.91](#)
- start testdns [E.3.112](#)
- status category [E.3.5](#)
- status ip [E.3.68](#)
- status policy [E.3.81](#)
- status policyset [E.3.84](#)
- status resource [E.2.18](#)
- status resourcegroup [E.2.33](#)
- status server [E.3.97](#)
- status serverpool [E.3.110](#)
- status testdns [E.3.113](#)
- status type [E.2.45](#)
- stop cluster [E.3.8](#)
- stop crs [E.3.32](#)
- stop has [E.4.8](#)
- stop ip [E.3.69](#)
- stop resource [E.2.19](#)
- stop resourcegroup [E.2.34](#)
- stop rollingpatch [E.3.90](#)
- stop testdns [E.3.114](#)
- unpin css [E.3.47](#)
- unset css [E.3.48](#)
- unset css ipmiconfig [E.3.53](#)
- デュアル環境のコマンド [E.2](#)
- Oracle RAC環境のコマンド [E.3](#)
- Oracle Restart環境のコマンド [E.4](#)
- CRSTCTLコマンド
 - クラスタ対応 [1.9](#)
 - デバッグ・ログ [E.5.2.1](#)
 - lsmodules [E.5.2.1](#)
 - module_nameパラメータ [E.5.2.1](#)
 - 概要 [1.9](#)
 - resource_nameパラメータ [E.5.2.2](#)
 - set log [E.5.2.2](#)
- CRSDBackgroundプロセス
 - アラート・メッセージ [1.7](#)

- CSS
 - 「クラスタ同期サービス(CSS)」を参照:
- CVU
 - 概要 [A](#)
 - ベースラインの作成 [A.2.4](#)
 - コマンド
 - cluvfy comp acfs [A.2.1](#)
 - cluvfy comp admprv [A.2.2](#)
 - cluvfy comp asm [A.2.3](#)
 - cluvfy comp baseline [A.2.4](#)
 - cluvfy comp clocksync [A.2.5](#)
 - cluvfy comp clu [A.2.6](#)
 - cluvfy comp clumgr [A.2.7](#)
 - cluvfy comp clusterware [A.2.8](#)
 - cluvfy comp crs [A.2.9](#)
 - cluvfy comp DHCP [A.2.10](#)
 - cluvfy comp dns [A.2.11](#)
 - cluvfy comp freespace [A.2.12](#)
 - cluvfy comp gns [A.2.13](#)
 - cluvfy comp gnpn [A.2.14](#)
 - cluvfy comp ha [A.2.15](#)
 - cluvfy comp healthcheck [A.2.16](#)
 - cluvfy comp nodeapp [A.2.17](#)
 - cluvfy comp nodecon [2.10.1](#), [A.2.18](#)
 - cluvfy comp nodereach [A.2.19](#)
 - cluvfy comp ocr [A.2.20](#)
 - cluvfy comp ohasd [A.2.21](#)
 - cluvfy comp olr [A.2.22](#)
 - cluvfy comp peer [A.2.23](#)
 - cluvfy comp scan [A.2.24](#)
 - cluvfy comp software [A.2.25](#)
 - cluvfy comp space [A.2.26](#)
 - cluvfy comp ssa [A.2.27](#)
 - cluvfy comp sys [A.2.28](#)
 - cluvfy comp vdisk [A.2.29](#)
 - cluvfy stage -post acfscfg [A.2.31](#)
 - cluvfy stage -post appcluster [A.2.32](#)
 - cluvfy stage -post cfs [A.2.33](#)
 - cluvfy stage -post crsinst [A.2.34](#)
 - cluvfy stage -post hacfg [A.2.37](#)
 - cluvfy stage -post hwos [A.2.38](#)
 - cluvfy stage -post nodeadd [A.2.39](#)
 - cluvfy stage -post nodedel [A.2.40](#)

- cluvfy stage -post patch [A.2.42](#)
- cluvfy stage -pre acfscfg [A.2.31](#)
- cluvfy stage -pre appcluster [A.2.32](#)
- cluvfy stage -pre cfs [A.2.33](#)
- cluvfy stage -pre crsinst [A.2.34](#)
- cluvfy stage -pre dbcfg [A.2.35](#)
- cluvfy stage -pre dbinst [A.2.36](#)
- cluvfy stage -pre hacfg [A.2.37](#)
- cluvfy stage -pre nodeadd [A.2.39](#)
- cluvfy stage -pre patch [A.2.42](#)
- コンポーネント検証
 - Oracle ClusterwareおよびOracle Databaseインストールのチェック [A.2.16](#)
 - クラスタ・コンポーネント [A.2.6](#)
 - Cluster Managerサブコンポーネント [A.2.7](#)
 - クラスタ・ノード間の接続性 [A.2.18](#)
 - CTSSの整合性 [A.2.5](#)
 - ドメイン名サービス(DNS) [A.2.11](#)
 - DHCPの存在 [A.2.10](#)
 - 空き領域 [A.2.26](#)
 - グリッド・ネーミング・サービス(GNS) [A.2.13](#)
 - グリッド・プラグ・アンド・プレイ・サービスおよびプロファイル [A.2.14](#)
 - 高可用性の整合性 [A.2.15](#)
 - OCRの整合性 [A.2.20](#)
 - ohasdの整合性 [A.2.21](#)
 - OLRの整合性 [A.2.22](#)
 - Oracle ACFSの整合性 [A.2.1](#)
 - Oracle ASMの整合性 [A.2.3](#)
 - Oracle Clusterwareの整合性 [A.2.8](#)
 - 投票ファイルの整合性 [A.2.29](#)
 - ノード・アプリケーション [A.2.17](#)
 - ノードの比較 [A.2.23](#)
 - Oracle Clusterwareコンポーネント [A.2.9](#)
 - ノードの到達可能性 [A.2.19](#)
 - SCAN構成 [A.2.24](#)
 - ノードへのソフトウェア配布 [A.2.25](#)
 - 記憶域 [A.2.27](#)
 - システム要件 [A.2.28](#)
 - ユーザーおよび権限 [A.2.2](#)
- runcluvfy.shとcluvfy間の差異 [A.1.2.2](#)
- インストール要件 [A.1.2.1](#)
- インストール検証
 - アプリケーション・クラスタ [A.2.32](#)
- 既知の問題 [A.3.2](#)

- ノード・リストのショートカット [A.1.3.6](#)
 - オンライン・ヘルプ・システム [A.1.2.5](#)
 - 概要および概念 [1.9](#)
 - パッチの検証
 - パッチ適用前およびパッチ適用後 [A.2.42](#)
 - 検証の実行 [A.1.1](#)
 - runcluvfy.sh [A.1.2.2](#)
 - ステージ検証
 - データベース構成 [A.2.35](#)
 - 高可用性インストール [A.2.37](#)
 - すべてのノードのネットワークおよび記憶域 [A.2.38](#)
 - ノードの削除 [A.2.40](#)
 - ノードのインストール [A.2.39](#)
 - Oracle ACFS [A.2.33](#)
 - Oracle ACFS構成 [A.2.31](#)
 - Oracle Clusterwareのインストール [A.2.34](#)
 - Oracle RACのインストール [A.2.36](#)
-

D

- データベース管理リポジトリ [1.8](#)
- データベース・メンバー・クラスタ [1.8](#)
- デバッグ
 - CRS、CSSおよびEVMモジュール [E.5.2.1](#)
 - Oracle Clusterwareリソース [E.5.2.2](#)
- デフォルト・アプリケーションVIP
 - 作成
 - appvipcfg create [8.3.1](#)
- ネットワーク・インタフェースの定義
 - OIFCFGコマンドライン・インタフェース [D](#)
- DELETE_TIMEOUT
 - リソース属性 [B.1.1.15](#)
- デプロイメント・スキーム
 - 決定 [8.3.2.1](#)
- DESCRIPTION
 - リソース属性 [B.1.1.16](#)
- DHCP構成 [2.5.2.2](#)
- diagcollection.pl [J.3](#)
- 診断ディレクトリ
 - アラート [J.2](#)
 - コア [J.2](#)
 - インシデント [J.2](#)
 - 出力 [J.2](#)

- トレース [J.2](#)
 - 診断収集スクリプト [J.3](#)
 - ディスク・グループの冗長性 [5.2.1](#)
 - 外部 [5.2.1](#)
 - 高 [5.2.1](#)
 - 標準 [5.2.1](#)
 - dispersion起動依存性 [8.2.3.1.2](#)
 - 修飾子 [B.1.1.36](#)
 - DNS、GNSおよびSCANのエントリの例 [2.5.1](#)
 - ドメイン・サービス・クラスタ [1.8](#)
 - DRUID
 - Diagnostic Record Unique ID [J.7.1](#)
 - 動的ホスト構成プロトコル(DHCP) [1.2.4](#)
-

E

- ENABLED
 - リソース属性 [B.1.1.17](#)
 - Oracle Clusterwareリソースのデバッグの有効化 [E.5.2.2](#)
 - CRS、CSSおよびEVMモジュールのデバッグの有効化 [E.5.2.1](#)
 - イベント・マネージメント(EVM)
 - 定義 [1.3.1.1](#)
 - イベント・マネージャ(EVM)
 - デバッグ [E.5.2.1](#)
 - EVM
 - 「イベント・マネージャ(EVM)」を参照
 - 概要 [1.3.1.1](#)
 - exclusion起動依存性 [8.2.3.1.3](#)
 - 修飾子 [B.1.1.36](#)
 - Oracle Databaseホームの拡張
 - 非共有記憶域 [6.2.1](#)
 - 共有記憶域
 - ネットワーク接続ストレージ [6.2.1](#)
 - Oracle ACFS [6.2.1](#)
-

F

- FAILURE_INTERVAL
 - リソース属性 [B.1.1.18](#)
- FAILURE_THRESHOLD
 - リソース属性 [B.1.1.19](#)
- 障害グループ [5.2.1](#)

- クォーラム [5.2.1](#)
 - 障害分離
 - IPMIの構成 [2.7.2.1](#)
 - 高速アプリケーション通知(FAN) [1.3.1.1](#)
 - フリート・パッチ適用およびプロビジョニング
 - クラスタ・ノードの追加のための使用 [6.2.1](#)
 - 空きサーバー・プール [3.1.4](#)
 - 説明 [3.1.4.1](#)
-

G

- generic_application [8.1.2](#)
 - リソース・タイプ
 - リソースの作成 [8.3.2.4](#)
- 汎用サーバー・プール
 - 説明 [3.1.4.2](#)
- 汎用サーバー・プール [3.1.4](#)
- グローバル・インタフェース
 - 格納されるネットワーク・インタフェース [D.2.3](#)
- GNS [1.2.4.1](#)
 - 「グリッド・ネーミング・サービス(GNS)」を参照
 - 管理 [2.6](#)
 - サブドメインの変更 [2.6.6](#)
 - 非GNSクラスタのGNSクラスタへの変換 [2.6.4.1](#)
 - 高可用性
 - セカンダリGNSインスタンスの削除 [2.6.2](#)
 - 高可用性 [2.5.2.1](#)
 - 構成 [2.6.1](#)
 - プライマリGNSインスタンスの削除 [2.6.2](#)
 - プライマリGNSインスタンス [2.5.2.1](#)
 - セカンダリGNSインスタンス [2.5.2.1](#)
 - 起動 [2.6.3](#)
 - 停止 [2.6.3](#)
 - ゾーン・データ [2.5.2.1](#)
 - ゾーン転送 [2.5.2.1](#)
- GNS, GNS VIP [2.5.1](#)
- GNSデーモン
 - GNS VIP [2.5.1](#)
 - ポート番号 [2.5.1](#)
- グリッド・インフラストラクチャ管理リポジトリ [1.7](#)
 - 属性および要件 [1.7](#)
- グリッド・プロセス間通信(gipc)
 - デバッグ [E.5.2.1](#)

- グリッド・プロセス間通信(GIPC) [1.3.1.2](#)
 - グリッド・ネーミング・サービス(GNS)
 - 定義 [1.3.1.1](#)
 - 「GNS」を参照 [1.2.4](#)
 - グリッド・プラグ・アンド・プレイ(gpnp)
 - デバッグ [E.5.2.1](#)
 - グリッド・プラグ・アンド・プレイ(GPNPD) [1.3.1.2](#)
 - gridSetup.sh
 - ノードを追加するための使用 [6.2.1](#)
-

H

- HAIP
 - 高可用性IPアドレス [2.10.2.2](#)
 - hard起動依存性 [8.2.3.1.4](#)
 - 修飾子 [B.1.1.36](#)
 - hard停止依存性 [8.2.3.2.1](#)
 - 修飾子 [B.1.1.39](#)
 - ハードウェア要件 [1.2.1](#)
 - 高可用性
 - Oracle Clusterware [1.3.1.1](#)
 - Application Program Interface [1.12](#)
 - フレームワーク [1.12](#)
 - 高可用性GNS [2.5.2.1](#)
 - 構成 [2.6.1](#)
 - HOSTING_MEMBERS
 - リソース属性 [B.1.1.20](#)
 - ハブ・ノード [4.1](#)
-

I

- インポート
 - OCR [5.1.6](#)
- インシデント・トレース・ファイル [J.2](#)
- インストール
 - 概要 [1.5](#)
- インストール
 - 投票ファイルの構成 [5](#)
- INSTANCE_COUNT
 - リソース属性 [B.1.2.2](#)
- INSTANCE_FAILOVER
 - リソース属性 [B.1.1.21](#)

- Intelligent Management Platform Interface (IPMI) [2.7](#), [2.7.2.2](#)
 - 障害分離のための構成 [2.7.2.1](#)
 - IPMI構成の削除 [2.7.2.3](#)
- Intelligent Platform Management Interface(IPMI)
 - CRSTCTLコマンド
 - get css ipmiaddr [E.3.49](#)
 - query css ipmiconfig [E.3.52](#)
 - query css ipmidevice [E.3.54](#)
 - set css ipmiaddr [E.3.50](#)
 - set css ipmiadmin [E.3.51](#)
 - unset css ipmiconfig [E.3.53](#)
- インタフェース名, 変更の結果 [2.10.2.3](#)
- INTERMEDIATE_TIMEOUT
 - リソース属性 [B.1.1.22](#)
- IPMI
 - 「Intelligent Management Platform Interface (IPMI)」を参照
 - 管理者の変更 [E.3.51](#)
 - IPアドレスの取得 [E.3.49](#)
 - IPアドレスの格納 [E.3.50](#)
- IPv4 [2.8.1](#)
 - 静的IPv6アドレスへの変更 [2.10.5](#)
 - 動的IPv6アドレスへの変更 [2.10.6](#)
 - ネットワーク構成
 - IPv6ネットワークの追加 [2.10.7](#)
 - ネットワーク
 - IPv6ネットワークへの移行 [2.10.8](#)
- IPv6 [2.8.1](#)
 - 名前解決 [2.8.1.2](#)
- IPv6ステートレス・アドレス自動構成プロトコル [2.5.2.2](#)

L

- LAST_SERVER
 - リソース属性 [B.1.2.4](#)
- LAST_STATE_CHANGE
 - リソース属性 [B.1.2.5](#)
- リスナー
 - OCR [1.2.3](#)
- LOAD
 - リソース属性 [B.1.1.23](#)
- ロード対応のリソース配置 [2.3](#), [3.4](#)
- local_resource [8.1.2](#)
- local resource [8.1.2](#)

- ログ・レベル
 - Oracle Clusterwareの設定 [E.5.1](#)
 - lsmodulesパラメータ
 - CRSCTLコマンド [E.5.2.1](#)
-

M

- アプリケーションの管理
 - CLSCRSコマンド [H.1](#)
 - Oracle Clusterwareの管理
 - CRSCTLの使用 [1.9](#)
 - アドレスの手動構成 [1.2.4.2](#)
 - mDNS
 - 「マルチキャスト・ドメイン名サービス(mDNS)」を参照:
 - mDNSResponder
 - 用途 [1.3.2](#)
 - メモリー不足 [3.5.1](#)
 - ミラー化
 - OCR(Oracle Cluster Registry) [5.1.2](#)
 - MODIFY_TIMEOUT
 - リソース属性 [B.1.1.24](#)
 - IPMI構成の変更 [2.7.2.2](#)
 - module_nameパラメータ
 - CRSCTLコマンドへの指定 [E.5.2.1](#)
 - モジュール
 - デバッグ
 - CRF [E.5.2.1](#)
 - CRS [E.5.2.1](#)
 - CSS [E.5.2.1](#)
 - CTSS [E.5.2.1](#)
 - EVM [E.5.2.1](#)
 - gipc [E.5.2.1](#)
 - gpnp [E.5.2.1](#)
 - マルチキャスト・ドメイン名サービス(mDNS)
 - 定義 [1.3.1.2](#)
-

N

- NAME
 - リソース属性 [B.1.1.25](#)
- ネットワーク・インタフェース
 - 構成 [2.8.1](#)

- グローバル [D.2.3](#)
 - ノード固有 [D.2.3](#)
 - OIFCFG構文 [D.2.4](#)
 - ネットワーク・インタフェース・カード [2.10.2.1](#)
 - ネットワーク・インタフェース
 - OIFCFGによる定義 [D](#)
 - タイプ [D.2.4](#)
 - subnet分類の更新 [D.1](#)
 - ネットワーク
 - 作成 [2.10.3](#)
 - Oracle Flex Clusters [4.1](#)
 - ノード
 - クラスタへの追加
 - LinuxまたはUNIX [6.2.1](#)
 - Windows [6.3](#), [6.3.1](#)
 - 複数の構成 [2.2.2](#)
 - クラスタからの削除
 - LinuxまたはUNIX [6.2.2](#)
 - VIPアドレス [1.2.4](#)
 - ノード固有のインタフェース
 - 格納されるネットワーク・インタフェース [D.2.3](#)
 - 非デフォルト・アプリケーションVIP
 - 作成 [8.3.1](#)
-

O

- ocr.locファイル [5.1.2](#)
- OCR(Oracle Cluster Registry)
 - 追加 [5.1.2](#), [5.1.2.1](#)
 - 自動バックアップ [5.1.3](#)
 - バックアップ [5.1.3](#)
 - バックアップ・ファイルの場所の変更 [5.1.3.2](#)
 - 内容 [5.1](#)
 - OCRDUMPに関する問題の診断 [5.1.5](#)
 - エクスポート [5.1.6](#)
 - インポート
 - Windowsシステム [5.1.6.2](#)
 - 内容のインポート
 - LinuxおよびUNIXシステム [5.1.6.1](#)
 - バックアップ・ファイルのリスト [5.1.3.1](#)
 - 管理 [5.1](#)
 - 手動バックアップ [5.1.3](#)
 - Oracle ASMからの移行 [5.1.1.1](#)

- Oracle ASMへの移行 [5.1.1](#)
- ocr.locファイル [5.1.2](#)
- OCRDUMPユーティリティのコマンド例 [1.3.3.2](#)
- データ損失保護メカニズムの無効化 [5.1.2.5](#)
- クラスタ構成情報の記録 [1.1](#)
- クラスタ記憶域の記録 [1.2.3](#)
- 削除 [5.1.2](#), [5.1.2.2](#)
- 修復 [5.1.2](#), [5.1.2.4](#)
- 置換 [5.1.2](#), [5.1.2.3](#)
- リストア
 - Oracle Restart [5.1.4.4](#)
 - LinuxおよびUNIXシステム [5.1.4.2](#)
 - Windowsシステム [5.1.4.3](#)
- トラブルシューティング [5.1.5](#), [1.3.1](#)
- OCRDUMPによる内容の表示 [1.3.3](#)
- OCRCHECK
 - コマンド
 - -config [1.3.2.2](#)
 - -local [1.3.2.2](#)
- OCRCHECKユーティリティ
 - OLRのステータスのチェック [5.1.7](#)
 - OCRの問題の診断 [5.1.5](#)
 - ログ・ファイル [1.3.2.2](#)
 - サンプル出力 [1.3.2.2](#)
- OCR構成ツール
 - 「OCRCONFIGユーティリティ」を参照
- OCRCONFIGユーティリティ
 - OLRの管理 [5.1.7](#)
 - コマンド
 - -add [1.2.1](#)
 - -backuploc [1.2.2](#)
 - -copy [1.2.3](#)
 - -delete [1.2.4](#)
 - -downgrade [1.2.5](#)
 - -export [1.2.6](#)
 - -import [5.1.6.1](#), [1.2.7](#)
 - -manualbackup [1.2.8](#)
 - -overwrite [1.2.9](#)
 - -repair [1.2.10](#)
 - -replace [1.2.11](#)
 - -restore [1.2.12](#)
 - -showbackup [1.2.13](#)
 - -showbackuploc [1.2.14](#)

- -upgrade [I.2.15](#)
- ログ・ファイル [I.1](#)
- 概要および概念 [1.9](#)
- 構文 [I.1](#)
- OCRDUMPユーティリティ
 - ロギングの量の変更 [I.3.3](#)
 - コマンド例 [I.3.3.2](#)
 - コマンド [I.3.3.2](#)
 - バックアップ [I.3.3.2](#)
 - OCR問題の診断 [5.1.5](#), [I.3.3](#)
 - OLRの内容のダンプ [5.1.7](#)
 - ログ・ファイル [I.3.3](#)
 - サンプル出力 [I.3.3.3](#)
 - 構文およびオプション [I.3.3.1](#)
 - SYSTEM.languageキーの出力 [I.3.3.3](#)
 - SYSTEM.versionキーの出力 [I.3.3.3](#)
- ocrlog.iniファイル
 - 編集 [I.3.3](#)
- OFFLINE_CHECK_INTERVAL
 - リソース属性 [B.1.1.26](#)
- OIFCFGコマンドライン・インタフェース
 - コマンド [D.2.2](#)
 - インタフェース・タイプ [D.2.4](#)
 - 起動 [D.1](#)
 - 概要および概念 [1.9](#), [D](#)
 - 構文 [D.2.1](#)
- OLR(Oracle Local Registry)
 - 管理 [5.1.7](#)
 - バックアップ [5.1.7](#)
 - ステータスのチェック [5.1.7](#)
 - 定義 [5.1.7](#)
 - 内容のダンプ [5.1.7](#)
 - ファイルへのエクスポート [5.1.7](#)
 - ファイルのインポート [5.1.7](#)
 - リストア [5.1.7](#)
 - バックアップ・ファイルの表示 [5.1.7](#)
 - OCRDUMPによる内容の表示 [I.3.3](#)
- OLSNODESコマンド
 - 参照 [C](#)
- ONLINE_RELOCATION_TIMEOUT
 - リソース属性 [B.1.1.27](#)
- ONS
 - 「Oracle Notification Service (ONS)」を参照:

- オペレーティング・システム
 - Oracle Clusterwareの要件 [1.1](#)
- oraagent
 - 定義 [1.3.1.1](#), [1.3.1.2](#)
- Oracle Agent [1.3.1.1](#), [1.3.1.2](#)
- Oracle ASM
 - ディスク・グループ
 - 冗長性 [5.2.1](#)
 - 障害グループ [5.2.1](#)
 - OCRの場所の移行 [5.1.1](#)
- Oracle Cluster Registry
 - 「OCR (Oracle Cluster Registry)」を参照:
- Oracle Clusterware
 - 新規ノードへのホームの追加 [6.2.1](#)
 - アラート・ログ [J.2](#)
 - バックグラウンド・プロセス
 - Windows [1.3.2](#)
 - デバッグ
 - 動的 [E.5.1](#)
 - 定義 [1](#)
 - OCR
 - Oracle ASMからの移行 [5.1.1.1](#)
 - Oracle ASMへの移行 [5.1.1](#)
 - 管理の概要 [1](#)
 - プロセス
 - クラスタ・レディ・サービス(CRS) [1.3.1.1](#)
 - クラスタ同期サービス(CSS) [1.3.1.1](#)
 - イベント・マネージメント(EVM) [1.3.1.1](#)
 - グリッド・プロセス間通信(GIPC) [1.3.1.2](#)
 - グリッド・ネーミング・サービス(GNS) [1.3.1.1](#)
 - マルチキャスト・ドメイン名サービス(mDNS) [1.3.1.2](#)
 - oraagent [1.3.1.1](#), [1.3.1.2](#)
 - Oracle Notification Service(ONS) [1.3.1.1](#)
 - orarootagent [1.3.1.1](#), [1.3.1.2](#)
 - アップグレード
 - アウトオブブレース [1.6](#)
- Oracle Clusterware制御(CRSCTL)
 - 「CRSCTL」を参照
- Oracle Clusterwareホーム
 - 追加 [6.2.1](#), [6.3.1](#)
 - 手動で削除 [6.2.2](#)
- Oracle Database
 - 障害診断インフラストラクチャ [J.2](#)

- Oracle Database QoS Management [2.4](#)
- Oracle Databaseのサービスのクオリティ管理
 - 「Oracle Database QoS Management」を参照
- Oracle Enterprise Manager
 - リソースの追加のための使用 [8.3.3](#)
 - VIPの追加のための使用 [8.3.1.1](#)
 - 概要および概念 [1.9](#)
- Oracle拡張クラスタ [4.3](#)
 - 構成 [4.3.1](#)
 - 記憶域の分割 [J.4](#)
- Oracle Flex Cluster [1.1](#)
 - Oracle Flex ASMクラスタ [4.1](#)
 - Oracle Clusterware標準クラスタからの変更 [4.2.1.1](#)
 - CRSCTLコマンド
 - get cluster hubsize [E.3.13](#)
 - get cluster mode [E.3.15](#)
 - set cluster hubsize [E.3.14](#)
 - set cluster mode [E.3.16](#)
 - ハブ・ノード [4.1](#)
 - 管理 [4.2](#)
 - リーダー・ノード [4.1](#)
- Oracle Grid Infrastructure
 - クローニング [7](#)
 - Oracle Clusterwareのクローニング [7](#)
 - グリッド設定ウィザード [2.2](#)
 - インストーラを使用したノードの追加 [6.2.1](#)
- Oracle Interface Configurationツール
 - 「OIFCFG」を参照 [D](#)
- Oracle Local Registry
 - 「OLR (Oracle Local Registry)」を参照:
- Oracle Notification Service(ONS)
 - 定義 [1.3.1.1](#)
- Oracle Restart
 - OCRのリストア [5.1.4.4](#)
- Oracle Root Agent [1.3.1.1](#), [1.3.1.2](#)
- Microsoftトランザクション・サーバー用のOracleサービス
 - OraMTSサービスの作成 [6.3.1](#)
- Oracle Universal Installer
 - 共有GNSクライアントのクライアント・データ・ファイル [2.6.4.2](#)
 - Oracle Clusterwareのインストール [1.5](#)
- OraMTS
 - 「Microsoftトランザクション・サーバー用のOracleサービス」を参照:
- orarootagent

- 定義 [1.3.1.1](#), [1.3.1.2](#)
 - アウトオブプレイス・アップグレード [1.6](#)
-

P

- PLACEMENT
 - リソース属性 [B.1.1.28](#)
 - ポリシーベース管理 [3](#)
 - サーバー・プールへのリソースの追加 [8.3.2.2](#)
 - プライマリGNSインスタンス [2.5.2.1](#)
 - 構成 [2.6.1](#)
 - 削除 [2.6.2](#)
 - プライベート・ネットワーク・アドレス
 - 変更 [2.10.2.1](#)
 - PROFILE_CHANGE_EVENT_TEMPLATE
 - リソース属性 [B.1.2.6](#)
 - パブリック・インタフェース
 - OIFCFGによる指定 [D.2.4](#)
 - pullup起動依存性 [8.2.3.1.5](#)
 - 修飾子 [B.1.1.36](#)
-

Q

- クォーラム障害グループ [5.2.1](#)
-

R

- リーダー・ノード [4.1](#)
- 妥当なコマンド評価 [1.10](#)
- クラスタ構成情報の記録 [1.1](#)
- ノード・メンバーシップ情報の記録
 - 投票ファイル [1.1](#)
- 冗長性
 - 投票ファイル [1.2.3](#)
- 冗長インターコネクトの使用 [2.10.2.2](#)
- RELOCATE_BY_DEPENDENCY
 - リソース属性 [B.1.1.30](#)
- RELOCATE_KIND
 - リソース属性 [B.1.1.29](#)
- RESOURCE_LIST
 - リソース・グループ属性 [B.1.2.7](#)
- resource_nameパラメータ

- CRSCTLコマンドへの指定 [E.5.2.2](#)
- リソース属性
 - ACTION_FAILURE_EVENT_TEMPLATE [B.1.2.1](#)
 - ACTION_SCRIPT [B.1.1.2](#)
 - ACTION_TIMEOUT [B.1.1.3](#)
 - ACTIONS [B.1.1.4](#)
 - ACTIVE_PLACEMENT [B.1.1.5](#)
 - AGENT_FILENAME [B.1.1.6](#)
 - ALERT_TEMPLATE [B.1.1.7](#)
 - AUTO_START [B.1.1.8](#)
 - CARDINALITY [B.1.1.9](#)
 - CARDINALITY_ID [B.1.1.10](#)
 - CHECK_INTERVAL [B.1.1.11](#)
 - CHECK_TIMEOUT [B.1.1.12](#)
 - CLEAN_TIMEOUT [B.1.1.13](#)
 - DELETE_TIMEOUT [B.1.1.15](#)
 - DESCRIPTION [B.1.1.16](#)
 - ENABLED [B.1.1.17](#)
 - FAILURE_INTERVAL [B.1.1.18](#)
 - FAILURE_THRESHOLD [B.1.1.19](#)
 - HOSTING_MEMBERS [B.1.1.20](#)
 - INSTANCE_COUNT [B.1.2.2](#)
 - INSTANCE_FAILOVER [B.1.1.21](#)
 - INTERMEDIATE_TIMEOUT [B.1.1.22](#)
 - LAST_SERVER [B.1.2.4](#)
 - LAST_STATE_CHANGE [B.1.2.5](#)
 - LOAD [B.1.1.23](#)
 - MODIFY_TIMEOUT [B.1.1.24](#)
 - NAME [B.1.1.25](#)
 - OFFLINE_CHECK_INTERVAL [B.1.1.26](#)
 - ONLINE_RELOCATION_TIMEOUT [B.1.1.27](#)
 - PLACEMENT [B.1.1.28](#)
 - PROFILE_CHANGE_EVENT_TEMPLATE [B.1.2.6](#)
 - RELOCATE_BY_DEPENDENCY [B.1.1.30](#)
 - RELOCATE_KIND [B.1.1.29](#)
 - RESTART_ATTEMPTS [B.1.1.31](#)
 - RESTART_COUNT [B.1.2.8](#)
 - SCRIPT_TIMEOUT [B.1.1.32](#)
 - SERVER_CATEGORY [B.1.1.33](#)
 - SERVER_POOLS [B.1.1.34](#)
 - START_CONCURRENCY [B.1.1.35](#)
 - START_DEPENDENCIES [B.1.1.36](#)
 - START_TIMEOUT [B.1.1.37](#)

- STATE_CHANGE_EVENT_TEMPLATE [B.1.2.10](#)
- STATE_DETAILS [B.1.2.11](#)
- STOP_CONCURRENCY [B.1.1.38](#)
- STOP_DEPENDENCIES [B.1.1.39](#)
- STOP_TIMEOUT [B.1.1.40](#)
- TARGET [B.1.2.12](#)
- TARGET_SERVER [B.1.2.13](#)
- TYPE [B.1.2.14](#)
- UPTIME_THRESHOLD [B.1.1.41](#)
- USE_STICKINESS [B.1.1.43](#)
- USER_WORKLOAD [B.1.1.42](#)
- リソースの依存性
 - 定義 [8.2.3](#)
 - 起動依存性 [8.2.3.1](#)
 - attraction [8.2.3.1.1](#), [B.1.1.36](#)
 - dispersion [8.2.3.1.2](#), [B.1.1.36](#)
 - exclusion [8.2.3.1.3](#), [B.1.1.36](#)
 - hard [8.2.3.1.4](#), [B.1.1.36](#)
 - pullup [8.2.3.1.5](#), [B.1.1.36](#)
 - weak [8.2.3.1.6](#), [B.1.1.36](#)
 - 停止依存性 [8.2.3.2](#)
 - hard [8.2.3.2.1](#), [B.1.1.39](#)
- リソース・グループ[8.1.1](#)
- リソース・グループ属性
 - CARDINALITY [B.1.1.9](#)
 - CARDINALITY_ID [B.1.1.10](#)
 - CRITICAL_RESOURCES [B.1.1.14](#)
 - RESOURCE_LIST [B.1.2.7](#)
- リソース・グループ
 - 概要 [8.1.1.2](#)
 - リソースの追加 [8.4.1](#)
 - 自動リソース・グループ [8.1.1.2](#)
 - クラスタ・リソース・グループ・タイプ [8.1.1.2.1](#)
 - 作成 [8.1.1.2.2](#)
 - 依存性 [8.1.1.2](#)
 - ローカル・リソース・グループ・タイプ [8.1.1.2.1](#)
 - 権限 [8.1.1.2](#)
 - リソース・グループ・タイプ [8.1.1.2.1](#)
- リソース権限
 - 変更 [8.3.4](#)
- リソース
 - アクション・スクリプト [8.1.4](#)
 - 追加 [8.3.2](#)

- Oracle Enterprise Manager [8.3.3](#)
- サーバー・プールへの追加 [8.3.2.2](#)
- サーバー固有のデプロイメントを使用した追加 [8.3.2.3](#)
- generic_applicationリソース・タイプを使用した作成 [8.3.2.4](#)
- クリティカル [8.1.1.2](#)
- 定義 [8.1.1](#)
- クリティカルでない [8.1.1.2](#)
- 配置 [8.2.4](#)
- 登録 [8.1.6](#), [8.4.1](#)
- アプリケーションをリソースとして登録 [8.3](#)
- 再配置 [8.4.3](#)
- リソース・グループ [8.1.1](#)
- リソース・タイプ
 - cluster_resource [8.1.2](#)
 - 定義 [8.1.2](#)
 - generic_application [8.1.2](#)
 - local_resource [8.1.2](#)
- RESTART_ATTEMPTS
 - リソース属性 [B.1.1.31](#)
- RESTART_COUNT
 - リソース属性 [B.1.2.8](#)
- 再起動
 - 自動再起動の管理 [8.5.1](#)
- リストア
 - OCR [5.1.6](#)
- サービス登録の制限 [2.8.3](#)
- ロール別管理 [2.1](#)
 - 構成 [2.1.2](#)
 - 水平実装 [2.1](#)
 - 垂直実装 [2.1](#)
- runcluvfy.sh [A.1.2.2](#)

S

- スケーラビリティ
 - ノードおよびインスタンスの追加, クイック・スタート形式 [6.2](#)
- SCAN [1.2.4.1](#)
 - 「共有単一クライアント・アクセス名(SCAN)」を参照
 - 概要 [2.8.2](#)
- SCANリスナー [2.8.3](#)
- SCRIPT_TIMEOUT
 - リソース属性 [B.1.1.32](#)
- セカンダリGNSインスタンス [2.5.2.1](#)

- 構成 [2.6.1](#)
 - 削除 [2.6.2](#)
- SERVER_CATEGORY [3.1.7](#)
 - リソース属性 [B.1.1.33](#)
- SERVER_POOLS
 - リソース属性 [B.1.1.34](#)
- サーバーのカテゴリ化 [3.2](#)
 - 作成するための属性の使用 [3.1.7](#)
- サーバー・カテゴリ属性 [3.6](#)
- サーバー中心のリソース [8.1.2](#)
- サーバー・クラスタ [1.2.4](#)
- サーバー制御(SRVCTL)ユーティリティ(SRVCTL) [1.2.3](#)
 - 「SRVCTL」を参照
- サーバー・プール
 - 属性 [3.1.5](#)
 - 作成 [8.3.2.3](#)
 - 説明 [3.1.3](#)
 - 空き [3.1.4](#)
 - 一般 [3.1.4](#)
- サーバー
 - 説明 [3.5](#)
 - Oracle Clusterwareでの割当て方法 [3.1.6](#)
 - Oracle Clusterwareの要件 [1.1](#)
 - 状態 [3.5](#)
- サーバーの重みベースのノード削除 [2.3](#)
- サービス
 - リスナーへの登録の制限 [2.8.3](#)
- 共有GNS [1.2.4](#), [2.5.2.4](#)
 - 「GNS」も参照
 - 管理 [2.6](#)
 - 構成のためのクライアント・データ・ファイルの生成 [2.6.4.2](#)
 - 起動 [2.6.3](#)
 - 停止 [2.6.3](#)
- 共有単一クライアント・アクセス名(SCAN)
 - 概要 [2.9.1](#)
 - 専用クラスタでの構成 [2.9.2](#)
- 単一クライアント・アクセス名(SCAN)
 - 「SCAN」を参照
- 単一クライアント・アクセス名(SCAN), 共有
 - 概要 [2.9.1](#)
 - 専用クラスタでの構成 [2.9.2](#)
- シングルトン [3.1.3](#)
- 時刻の同期化(Slew) [1.13](#)

- ソフトウェア要件 [1.2.3](#)
- SRVCTL
 - コマンド・フィードバック [F.1](#)
 - コマンド [F.8.92](#), [F.8.180](#), [F.8.139](#), [F.8.140](#), [F.8.141](#), [F.8.182](#), [F.8.175](#), [F.8.176](#), [F.8.177](#), [F.8.178](#), [F.8.181](#), [F.8.179](#), [F.8.183](#)
 - add asm [F.8.2](#)
 - add asmnetwork [F.8.16](#)
 - add cdp [F.8.20](#)
 - add cdpproxy [F.8.29](#)
 - add cvu [F.8.39](#), [F.8.45](#)
 - add exportfs [F.8.49](#)
 - add filesystem [F.8.58](#)
 - add gns [F.8.68](#)
 - add havip [F.8.81](#)
 - add ioserver [F.8.91](#)
 - add mgmtadb [F.8.104](#)
 - add mgmtlsnr [F.8.117](#)
 - add mountfs [F.8.129](#)
 - add ovmm [F.8.138](#)
 - add qosmserver [F.8.142](#)
 - add rhpclient [F.8.153](#)
 - add rhpserver [F.8.163](#)
 - add vm [F.8.173](#)
 - check vm [F.8.174](#)
 - config asm [F.8.3](#)
 - config asmnetwork [F.8.17](#)
 - config cdpproxy [F.8.30](#)
 - config cvu [F.8.40](#)
 - config exportfs [F.8.50](#)
 - config filesystem [F.8.59](#)
 - config gns [F.8.69](#)
 - config havip [F.8.82](#)
 - config mgmtadb [F.8.105](#)
 - config mgmtlsnr [F.8.118](#)
 - config mountfs [F.8.130](#)
 - config rhpclient [F.8.154](#)
 - config rhpserver [F.8.164](#)
 - disable asm [F.8.4](#)
 - disable cdp [F.8.21](#)
 - disable cdpproxy [F.8.31](#)
 - disable cvu [F.8.41](#)
 - disable exportfs [F.8.51](#)
 - disable filesystem [F.8.60](#)

- disable gns [F.8.70](#)
- disable havip [F.8.83](#)
- disable ioserver [F.8.93](#)
- disable mgmtldb [F.8.106](#)
- disable mgmtlsnr [F.8.119](#)
- disable mountfs [F.8.131](#)
- disable qosmserver [F.8.144](#)
- disable rhpclient [F.8.155](#)
- disable rhpserver [F.8.165](#)
- enable asm [F.8.5](#)
- enable cdp [F.8.22](#)
- enable cdpproxy [F.8.32](#)
- enable cvu [F.8.42](#)
- enable exportfs [F.8.52](#)
- enable filesystem [F.8.61](#)
- enable gns [F.8.71](#)
- enable havip [F.8.84](#)
- enable ioserver [F.8.94](#)
- enable mgmtldb [F.8.107](#)
- enable mgmtlsnr [F.8.120](#)
- enable mountfs [F.8.132](#)
- enable qosmserver [F.8.145](#)
- enable rhpclient [F.8.156](#)
- enable rhpserver [F.8.166](#)
- -eval parameter [F.1](#)
- export gns [F.8.72](#)
- getenv asm [F.8.6](#)
- getenv ioserver [F.8.95](#)
- getenv mgmtldb [F.8.108](#)
- getenv mgmtlsnr [F.8.121](#)
- help [F.4](#)
- modify asm [F.8.7](#)
- modify asmnetwork [F.8.18](#)
- modify cdp [F.8.23](#)
- modify cdpproxy [F.8.33](#)
- modify cvu [F.8.43](#)
- modify exportfs [F.8.53](#)
- modify filesystem [F.8.62](#)
- modify gns [F.8.74](#)
- modify havip [F.8.85](#)
- modify ioserver [F.8.96](#)
- modify mgmtldb [F.8.109](#)
- modify mgmtlsnr [F.8.122](#)

- modify mountfs [F.8.133](#)
- modify qosmserver [F.8.146](#)
- modify rhpcient [F.8.157](#)
- modify rhpserver [F.8.167](#)
- predict asm [F.8.8](#)
- predict filesystem [F.8.63](#)
- predict qosmserver [F.8.147](#)
- relocate asm [F.8.9](#)
- relocate cdp [F.8.24](#)
- relocate cdpproxy [F.8.34](#)
- relocate cvu [F.8.44](#)
- relocate gns [F.8.75](#)
- relocate havip [F.8.86](#)
- relocate ioserver [F.8.97](#)
- relocate mgmtldb [F.8.110](#)
- relocate qosmserver [F.8.148](#)
- relocate rhpcient [F.8.158](#)
- relocate rhpserver [F.8.168](#)
- remove asm [F.8.10](#)
- remove asmnetwork [F.8.19](#)
- remove cdp [F.8.25](#)
- remove cdpproxy [F.8.35](#)
- remove exportfs [F.8.54](#)
- remove filesystem [F.8.64](#)
- remove gns [F.8.76](#)
- remove havip [F.8.87](#)
- remove ioserver [F.8.98](#)
- remove mgmtldb [F.8.111](#)
- remove mgmtlsnr [F.8.123](#)
- remove mountfs [F.8.134](#)
- remove qosmserver [F.8.149](#)
- remove rhpcient [F.8.159](#)
- remove rhpserver [F.8.169](#)
- setenv asm [F.8.11](#)
- setenv ioserver [F.8.99](#)
- setenv mgmtldb [F.8.112](#)
- setenv mgmtlsnr [F.8.124](#)
- start asm [F.8.12](#)
- start cdp [F.8.26](#)
- start cdpproxy [F.8.36](#), [F.8.38](#)
- start cvu [F.8.46](#)
- start exportfs [F.8.55](#)
- start filesystem [F.8.65](#)

- start gns [F.8.77](#)
- start havip [F.8.88](#)
- start ioserver [F.8.100](#)
- start mgmtldb [F.8.113](#), [F.8.125](#)
- start mountfs [F.8.135](#)
- start qosmserver [F.8.150](#)
- start rhpclient [F.8.160](#)
- start rhpserver [F.8.170](#)
- status asm [F.8.13](#)
- status cdp [F.8.27](#)
- status cdpproxy [F.8.37](#)
- status cvu [F.8.47](#)
- status exportfs [F.8.56](#)
- status filesystem [F.8.66](#)
- status gns [F.8.78](#)
- status havip [F.8.89](#)
- status ioserver [F.8.101](#)
- status mgmtldb [F.8.114](#)
- status mgmtlsnr [F.8.126](#)
- status mountfs [F.8.136](#)
- status qosmserver [F.8.151](#)
- status rhpclient [F.8.161](#)
- status rhpserver [F.8.171](#)
- stop asm [F.8.14](#)
- stop cdp [F.8.28](#)
- stop cvu [F.8.48](#)
- stop database [F.8.115](#)
- stop exportfs [F.8.57](#)
- stop filesystem [F.8.67](#)
- stop gns [F.8.79](#)
- stop havip [F.8.90](#)
- stop ioserver [F.8.102](#)
- stop mgmtlsnr [F.8.127](#)
- stop mountfs [F.8.137](#)
- stop qosmserver [F.8.152](#)
- stop rhpclient [F.8.162](#)
- stop rhpserver [F.8.172](#)
- unsetenv asm [F.8.15](#)
- unsetenv ioserver [F.8.103](#)
- unsetenv mgmtldb [F.8.116](#)
- unsetenv mgmtlsnr [F.8.128](#)
- update gns [F.8.80](#)
- 同時コマンド [F.1](#)

- config ioserver [F.8.92](#)
- config ovmm [F.8.139](#)
- config qosmserver [F.8.143](#)
- config vm [F.8.175](#)
- 非推奨になったコマンドおよびオプション [F.7](#)
- 非推奨になったコマンドおよびパラメータ [F.7.2](#)
- SRVCTLとCRSCTLの違い [F.6](#)
- disable vm [F.8.176](#)
- enable vm [F.8.177](#)
- modify ovmm [F.8.140](#)
- modify vm [F.8.178](#)
- 概要および概念 [1.9](#)
- relocate vm [F.8.179](#)
- remove ovmm [F.8.141](#)
- remove vm [F.8.180](#)
- 単一文字パラメータ [F.7.1](#)
- コマンドライン・エントリの継続指定 [F.1](#)
- start vm [F.8.181](#)
- status vm [F.8.182](#)
- アクティブなコマンドの停止 [F.6](#)
- stop vm [F.8.183](#)
- カンマ区切りリストの使用 [F.1](#)
- SRVCTLのコマンド・リファレンス [F](#)
- srvctl stop nodeappsコマンド [2.10.1](#)
- START_CONCURRENCY
 - リソース属性 [B.1.1.35](#)
- START_DEPENDENCIES
 - リソース属性 [B.1.1.36](#)
- START_TIMEOUT
 - リソース属性 [B.1.1.37](#)
- 起動試行の評価
 - 説明 [8.2.4](#)
- OIFCFGインタフェースの起動 [D.1](#)
- STATE_CHANGE_EVENT_TEMPLATE
 - リソース属性 [B.1.2.10](#)
- STATE_DETAILS
 - リソース属性 [B.1.2.11](#)
- ステートレス・アドレス自動構成プロトコル
 - 「autoconfig」を参照
- ステップ時間の同期 [1.13](#)
- STOP_CONCURRENCY
 - リソース属性 [B.1.1.38](#)
- STOP_DEPENDENCIES

- リソース属性 [B.1.1.39](#)
 - STOP_TIMEOUT
 - リソース属性 [B.1.1.40](#)
 - サブネット
 - VIPアドレスの構成 [1.2.4](#)
 - 構文
 - OCRDUMPユーティリティ [I.3.3.1](#)
 - SYSTEM.languageキー
 - 出力 [I.3.3.3](#)
 - SYSTEM.versionキー
 - 出力 [I.3.3.3](#)
-

T

- TARGET
 - リソース属性 [B.1.2.12](#)
 - TARGET_SERVER
 - リソース属性 [B.1.2.13](#)
 - スレッド・セーフティ [H.3](#)
 - トレース・ファイル [J.2](#)
 - トランスポート層セキュリティ [1.3.1.2.1](#)
 - トラブルシューティング
 - OCR [I.3.1](#)
 - TYPE
 - リソース属性 [B.1.2.14](#)
-

U

- 共通 [3.1.3](#)
 - アップグレード
 - 記憶域の移行 [5.1.1](#), [5.1.1.1](#)
 - アウトオブブレース [1.6](#)
 - アップグレード
 - SCAN [2.10.1](#)
 - UPTIME_THRESHOLD
 - リソース属性 [B.1.1.41](#)
 - USE_STICKINESS
 - リソース属性 [B.1.1.43](#)
 - USER_WORKLOAD
 - リソース属性 [B.1.1.42](#)
-

V

- 有効なノードの確認 [2.8.3](#)
- バージョン
 - Oracle Clusterware、Oracle ASMおよびOracle Databaseソフトウェアの互換性 [1.5.1](#)
- VIP [1.2.4](#), [8.3.1](#)
 - 追加
 - Oracle Enterprise Manager [8.3.1.1](#)
 - アドレス
 - 変更 [2.10.1](#)
 - アプリケーション用の定義 [1.12](#)
 - 要件 [1.2.4](#)
- 仮想インターネット・プロトコル・アドレス(VIP) [8.3.1](#)
- 仮想IP
 - 「VIP」を参照
- 仮想マシン [8.1.1.1](#)
 - ブラックボックス [8.1.1.1](#)
 - マネージャ [8.1.1.1](#)
 - サーバー [8.1.1.1](#)
- 投票ファイルの場所
 - 場所の特定 [5.2.4.4](#)
- 投票ファイル [1.2.3](#)
 - 追加 [5.2.4](#)
 - Oracle ASM以外の記憶域への追加 [5.2.4.2](#)
 - Oracle ASMへの追加 [5.2.4.1](#)
 - 管理 [5](#)
 - バックアップ [5.2.2](#)
 - 削除 [5.2.4](#), [5.2.4.2](#)
 - ファイル・ユニバーサル識別子(FUID)
 - 取得 [5.2.4.1](#), [5.2.4.2](#)
 - 管理 [5.2](#)
 - 移行 [5.2.4](#)
 - Oracle ASMへの移行 [5.2.4.1](#), [5.2.4.3](#)
 - Oracle ASMに格納されていない場合の変更 [5.2.4.2](#)
 - Oracle ASMに格納されている場合の変更 [5.2.4.1](#)
 - Oracle ASM以外の記憶域の置換 [5.2.4.2](#)
 - リストア [5.2.3](#)
 - Oracle ASMに格納 [5.2.1](#)

W

- weak起動依存性 [8.2.3.1.6](#)
 - 修飾子 [B.1.1.36](#)

- Windowsシステム
 - クラスタウェア用のサービス [1.3.2](#)