

Veritas™ High Availability Agent for DB2 インストール / 設定ガイド

AIX, Linux, Solaris

5.0

Veritas High Availability Agent for DB2 インストール / 設定ガイド

Copyright © 1998-2006 Symantec Corporation. All rights reserved.

Veritas Cluster Server 5.0

Symantec、Symantec ロゴ、Veritas は、Symantec Corporation または同社の米国およびその他の国における関連会社の商標または登録商標です。その他の会社名、製品名は各社の登録商標または商標です。

本書に記載する製品は、使用、コピー、頒布、逆コンパイルおよびリバース・エンジニアリングを制限するライセンスに基づいて頒布されています。Symantec Corporation からの書面による許可なく本書を複製することはできません。

Symantec Corporation が提供する技術文書は Symantec Corporation の著作物であり、Symantec Corporation が保有するものです。

保証の免責：技術文書は現状有姿で提供され、Symantec Corporation はその正確性や使用について何ら保証いたしません。技術文書またはこれに記載される情報はお客様の責任にてご使用ください。本書には、技術的な誤りやその他不正確な点を含んでいる可能性があります。Symantec は事前の通知なく本書を変更する権利を留保します。

使用を許諾されるソフトウェアおよび関連書類は、FAR section 12.212 および DFARS section 227.7202 に定義される「commercial computer software (商用コンピュータ・ソフトウェア)」および「commercial computer software documentation (商用コンピュータ・ソフトウェア説明書類)」であると見なされます。

サードパーティ（第三者）製ソフトウェアの権利に関する通知

本製品には、特定のサードパーティ製ソフトウェアが配布、組み込み、または同梱されている場合があります。また、本製品のインストールおよび使用にともない、サードパーティ製ソフトウェアの使用を推奨する場合があります。同サードパーティ製ソフトウェアのライセンスは、著作権の所有者により別途付与されます。サードパーティのソフトウェアの使用に必要なライセンスおよび著作権に関する情報については、本製品リリースノートのサードパーティに関する章を参照してください。

テクニカルサポート

製品のサポートを受けるには、<http://support.veritas.com> ページへアクセスし「Phone Support」または「E-mail Support」をクリックします。このページから TechNote、Software Alerts、ソフトウェアのダウンロード、ハードウェア互換性リスト、VERITAS Email Notifications サービスなどにアクセスすることもできます。「Knowledge Base Search」機能を使用し、製品ドキュメントのリリースなどの製品情報へアクセスすることができます。

目次

第 1 章

イントロダクション

新しい機能	9
バージョン番号とオペレーティングシステム	10
DB2 UDB 用のエージェントについて	10
DB2 処理用のエージェント	11
オンライン処理	11
オフライン処理	11
監視処理	11
詳細監視中の DB2 エラーコードの処理	11
ESE 設定 (バージョン 8 以上の場合)	12
クリーン処理	13
情報提供処理	13
アクション処理	14

第 2 章

DB2 UDB を VCS クラスタにインストールするための準備

DB2 UDB をインストールするための前提条件 (非 MPP 構成)	18
DB2 UDB をインストールするための前提条件 (MPP 構成)	19
DB2 インスタンス用のファイルシステムの作成	19
DB2 非 MPP インスタンス用のファイルシステムの作成	19
DB2 MPP インスタンス用の共有クラスタファイルシステムの作成	21
DB2 のユーザーアカウントとグループアカウントの定義	24
グループの作成	24
AIX システムでのユーザーアカウントの追加	25
Linux システムや Solaris システムでのユーザーアカウントの追加	25
VCS 環境での DB2 UDB のインストール	26
共有メモリパラメータの設定	26
バイナリのインストール	26
DB2 ライセンスのインストール	26
インスタンスのインストール	27
DB2 UDB 構成の設定	27
/etc/services の確認	27
\$DB2InstHome/.rhosts の作成	28
SUSE での ssh の設定	28
\$DB2InstHome/sql/lib/db2nodes.cfg ファイルの変更	29

	非 MPP 構成の場合	29
	MPP 構成の場合	29
	MPP 構成と非 MPP 構成の DB2 UDB のインストールの設定確認	30
第 3 章	DB2 用エージェントのインストール	
	DB2 UDB エージェントソフトウェアのインストール	33
第 4 章	DB2 用エージェントの設定	
	エージェントの設定	37
	Db2udb リソースタイプ属性	38
	DB2 UDB タイプ定義ファイル : Solaris 用の Db2udbTypes.cf	41
	非 MPP 構成の DB2 UDB サービスグループ	42
	MPP 構成の DB2 UDB サービスグループ	42
	Solaris ゾーンに設定された DB2 UDB インスタンス	44
	ローカルディスク上のゾーンルート	44
	Cluster Manager を使った DB2 UDB エージェントの設定	45
	Db2udbTypes.cf ファイルのインポート	45
	Cluster Manager を使った MPP 構成 DB2 UDB の サービスグループの追加	46
	Cluster Manager を使った非 MPP 構成 DB2 UDB の サービスグループの追加	49
	main.cf ファイルの編集による DB2 UDB エージェントの設定	52
	MPP 構成 DB2 UDB を使うためのエージェントの設定	52
	非 MPP 構成 DB2 UDB を使うためのエージェントの設定	53
	設定の確認	54
	エージェントの設定の変更	55
	DB2 UDB インスタンスの詳細監視の有効化	55
	コマンドラインからの詳細監視の有効化	56
	詳細監視の無効化	56
第 5 章	エージェントの無効化、削除、アップグレード	
	エージェントの無効化	59
	エージェントの削除	60
	エージェントのアップグレード	60
付録 A	設定ファイルの例	
	AIX 設定ファイルの例	63
	非 MPP の main.cf 設定 : DB2 UDB ESE シングルパーティションインスタンス	63
	非 MPP の main.cf 設定 : DB2 UDB ESE マルチパーティションインスタンス	66

MPP の main.cf 設定 : DB2 UDB ESE マルチパーティションインスタンス	68
Linux 設定ファイルの例	73
非 MPP の main.cf 設定 : DB2 UDB ESE シングルパーティションインスタンス	73
非 MPP の main.cf 設定 : DB2 UDB ESE マルチパーティションインスタンス	76
MPP の main.cf 設定 : DB2 UDB ESE マルチパーティションインスタンス	78
Solaris 設定ファイルの例	83
非 MPP の main.cf 設定 : DB2 UDB ESE シングルパーティションインスタンス	83
非 MPP の main.cf 設定 : DB2 UDB ESE マルチパーティションインスタンス	86
MPP の main.cf 設定 : DB2 UDB ESE マルチパーティションインスタンス	88
Solaris ゾーンで実行中の DB2 インスタンス	93

索引

イントロダクション

これは、Veritas High Availability Agent for DB2 UDB (バージョン 5.0) です。DB2 Universal Database は、リレーショナルデータベース管理システムです。このマニュアルでは、DB2 UDB エージェント、エージェントの処理モードとその動作モード、およびその属性について説明します。このエージェントのインストールと設定の方法についても説明します。

新しい機能

このリリースでは、DB2 用エージェントのインストールパスが新しくなっています。新しいパスは、`/opt/VRTSagents/ha/` です。パスの変更は、Symantec の製品ライン全体でパスに一貫性を持たせるために行われました。

`StartUpOpt` 属性と `ShutDownOpt` 属性は、起動とシャットダウンの新しいオプションを提供します。`StartUpOpt` 属性を使うと、インスタンスやパーティションの起動、プロセス開始後のデータベースコマンドのアクティブ化、カスタマイズした起動シーケンスの作成を行うことができます。`ShutDownOpt` 属性を使うと、通常の停止を実行したり、シャットダウンシーケンスをカスタマイズしたりできます。

38 ページの「[Db2udb リソースタイプ属性](#)」を参照してください。

以前のリリースでは、詳細監視を有効にする (`IndepthMonitor=1`) と、デフォルトの SQL クエリーが実行されました。詳細監視では、エラーの重大度に従って DB2 エラーに対する処理を分類できるようになりました。カスタマイズ可能な監視スクリプトを使って、それぞれのエラーコードに事前定義済みの処理を関連付けることができます。`/etc/VRTSagents/ha/conf/Db2udb/sample_db2udb` ディレクトリに、詳細監視スクリプトの例があります。カスタムスクリプトは、`/opt/VRTSagents/ha/bin/Db2udb` ディレクトリにインストールします。

11 ページの「[詳細監視中の DB2 エラーコードの処理](#)」と 38 ページの「[Db2udb リソースタイプ属性](#)」を参照してください。

`AgentDebug` 属性を有効にすると、エージェントやデータベースからより多くのデバッグ情報を取得できます。

38 ページの「[Db2udb リソースタイプ属性](#)」を参照してください。

バージョン番号とオペレーティングシステム

Veritas High Availability Agent for DB2 UDB (バージョン 5.0) は、シングルパーティションインスタンスとマルチパーティションインスタンス用の DB2 Universal Database Enterprise Server Edition (ESE) バージョン 8.1、8.2、9.1 をサポートします。ESE マルチパーティションインスタンスについては、対称マルチプロセッサ (SMP: Symmetric Multiprocessing) ハードウェア構成と、超並列プロセッサ (MPP: Massively Parallel Processing) ハードウェア構成の両方をサポートします。

DB2 ESE 用のエージェントは次のオペレーティングシステムで実行する必要があります。

- Solaris 2.8 以降
- AIX 5.2 以降
- RHEL 4.0
- SLES 9.0

Veritas Cluster Server (VCS) のバージョンは 5.0 以上である必要があります。メモリの必要条件は、使っている DB2 のバージョンによって異なります。メモリの必要条件について詳しくは『DB2 Universal Database 概説およびインストール』を参照してください。

DB2 UDB 用のエージェントについて

DB2 UDB 用のエージェントは、指定されたシステムで実行中の DB2 データベースインスタンスとデータベースを監視します。システムに障害が発生すると、エージェントがその障害を検出し、DB2 インスタンスをオフラインにします。VCS は、クラスタの別のシステムへのフェールオーバーを実行し、そのシステムでは、エージェントが DB2 インスタンスをオンラインにします。

ESE シングルパーティションインスタンスの場合、エージェントは、DB2 UDB データベースインスタンスをオンライン状態にし、データベースプロセスを監視し、それらをシャットダウンします。ESE マルチパーティションインスタンスの場合、エージェントは、DB2 UDB データベースパーティションをオンライン状態にし、パーティションレベルでデータベースプロセスを監視し、そのデータベースパーティションをシャットダウンします。

DB2 処理用のエージェント

DB2 用のエージェントは、次のいずれかに応じて、オンライン、オフライン、監視、クリーン、アクション、情報提供の各処理を実行します。

- DB2 UDB のバージョンが ESE シングルパーティションインスタンス
- ESE マルチパーティションインスタンスと ESE のバージョン

オンライン処理

ESE がバージョン 8 以上の場合、エージェントは `db2gcf` プログラムを使って、ESE シングルパーティションインスタンスか、SMP または MPP ハードウェア構成の ESE マルチパーティションインスタンスを起動します。コマンドは、次のようになります。

```
su $DB2InstOwner -c "$InstHome/sqlllib/bin/db2gcf -u -i
$DB2InstOwner -p $nodenum
```

オフライン処理

エージェントは `db2gcf` プログラムを使って、ESE シングルパーティションインスタンスか、SMP または MPP 構成の ESE マルチパーティションインスタンスを停止します。コマンドは、次のようになります。

```
su $DB2InstOwner -c "$InstHome/sqlllib/bin/db2gcf -d -i
$DB2InstOwner -p $nodenum
```

監視処理

DB2 インスタンスを監視するためにエージェントによって使われるコマンドは、DB2 バージョンとハードウェア構成によって異なります。

詳細監視中の DB2 エラーコードの処理

DB2 用のエージェントでは、詳細監視中に発生する DB2 エラーの処理が強化されています。エージェントは、DB2 のエラーを重大度別に分類し、定義済みのアクションを各エラーコードに関連付けます。

エージェントには `db2error.dat` という名前の参照ファイルが付属しており、このファイルをカスタマイズできます。このファイルには、DB2 のエラーと、そのエラーに関連付けられたアクション（エージェントがそのエラーを検出したときに実行する必要があるアクション）が記載されます。

このファイルには、次の形式で情報が格納されています。

```
SQL_error_string:action_to_be_taken
```

たとえば、次のようになります。

```
SQL1034N: IGNORE
SQL1039N: WARN
SQL1234N: FAILOVER
```

エージェントは、次のアクションをサポートします。

アクション 説明

IGNORE	エラーを無視します。
UNKNOWN	リソースの状態を UNKNOWN と判断し、 Notifier リソースが設定されている場合は通知を送信します。 VCS の通知について詳しくは、『 Veritas Cluster Server ユーザーズガイド 』を参照してください。 このアクションは、通常、設定エラーに関連付けられます。
WARN	リソースの状態を ONLINE と判断し、 Notifier リソースが設定されている場合は通知を送信します。 このアクションは、通常、重大度が低いエラーに関連付けられます。
FAILOVER (デフォルト)	リソースの状態を OFFLINE と判断します。これはサービスグループをエラーにし、次の使用可能なシステムにフェールオーバーされます。 これがエージェントのデフォルトの動作です。検出された DB2 エラーコードが db2error.dat ファイル内に存在しない場合、エージェントはデフォルトの動作を実行します。
NOFAILOVER	サービスグループを一時的にフリーズし、リソースの状態を OFFLINE と判断します。 Notifier リソースが設定されている場合は、通知も送信します。 このアクションは、通常、システム固有ではないエラーに関連付けられます。たとえば、データベースが壊れていた場合は、別のノードにフェールオーバーしても役に立ちません。

ESE 設定（バージョン 8 以上の場合）

バージョン 8.0 以上の DB2 ESE 設定の場合、SMP または MPP 構成のシングルパーティションインスタンスまたはマルチパーティションインスタンスに対して、エージェントは `db2gcf -s -i $DB2InstOwner -p $nodenum` コマンドを実行し、データベースパーティションの状態やノード番号を確認します。db2gcf コマンドの終了状態が 0 である場合、**monitor** は終了コード 110 を返します。0 でない場合は、終了コード 100 を返し、リソースはオフライン状態になります。エージェントは、他のタイプ独立型の属性（**RestartLimit** や **ToleranceLimit** など）により、リソースを再起動またはフェールオーバーします。db2gcf コマンドを利用可能なのは ESE バージョン 8 以降のみです。

詳細監視を行う場合は、**IndepthMonitor** 属性に 1 を設定します。このリリースより前の **IndepthMonitor** では、データベースに対してデフォルトの SQL クエリーが実行されていました。5.0 では、このデフォルトのクエリーは存在しなくなりました。

現在、エージェントは、`/opt/VRTSagents/ha/bin/Db2udb` ディレクトリで `custom_monitor_$db2instance_$nodenum` ファイルを検索します。ファイルが存在し、実行可能な場合、エージェントはカスタマイズされたこの詳細監視フ

イルを実行します。/etc/VRTSagents/ha/conf/sample_db2udb ディレクトリに、カスタム監視スクリプトの例があります。

メモ : DB2 インスタンスが Solaris 10 非グローバルゾーン内で稼動している場合は、非グローバルゾーン内に `custom_monitor_$db2instance_$nodenum` ファイルを作成する必要があります。つまり、`zlogin zonename` を実行してゾーンにログインしてからファイルを作成し、その後 /etc/VRTSagents/ha/bin/Db2udb ディレクトリ（ローカルゾーンへの相対パス）に移動する必要があります。

カスタム監視にエラーや問題があると、Db2udb エージェントの `WarnOnlyIfDBQueryFailed` 属性の値が確認されます。db2error.dat ファイルが /opt/VRTSagents/ha/bin/Db2udb ディレクトリ内にある場合、エージェントはこのファイルを調べ、エラー設定に従ってエラーを処理します。エラー処理情報は、11 ページの「[詳細監視中の DB2 エラーコードの処理](#)」を参照してください。

`WarnOnlyIfDBQueryFailed` 属性を 1（デフォルト）に設定し、`Notifier` リソースを設定した場合、エージェントは通知を送信し、終了コード 110 を返します。

`WarnOnlyIfDBQueryFailed` 属性を 0 に設定した場合、エージェントは db2error.dat ファイルのエラー処理を実行します。

クリーン処理

ESE がバージョン 8 以上の場合、エージェントは db2gc プログラムを使って、ESE シングルパーティションインスタンスか、SMP または MPP 構成の ESE マルチパーティションインスタンスを持つ DB2 データベースパーティションを強制終了します。コマンドは、次のようになります。

```
su $DB2InstOwner -c "$InstHome/sqlllib/bin/db2gc f -k -i
$DB2InstOwner -p $nodenum"
```

情報提供処理

DB2 エージェントは、データベースパーティションや重要なプロセスに関する静的な情報や動的な情報を提供する、情報提供処理をサポートしています。次に示す例では、エージェント機能が次のコマンドを実行することにより、表示されているデータベース情報を取得しています。

データベースの情報を取り込むには

- 1 設定を読み書き両用にします。


```
# haconf -makerw
```
- 2 情報エージェント機能を実行する一定の間隔を秒単位で指定します。このデフォルト値は 0 であり、情報 エージェント機能を実行しないことを示します。


```
# hatype -modify Db2udb InfoInterval 300
```

- 3 要求された **ResourceInfo** 値を表示します。次の例では、DB2 リソースのエージェントによって監視されているプロセスの名前と値のペアを出力します。

```
# hares -value db2udb1 ResourceInfo
State Valid
Msg
PARTITION: 0
      PID      TTY  TIME CMD
      413924    -   0:00 db2sysc
```

TS Fri Jan 14 18:11:52 2005

情報エージェント機能について詳しくは『Veritas Cluster Server ユーザーズガイド』と『Veritas Cluster Server エージェント開発者ガイド』を参照してください。

アクション処理

DB2 エージェントは、リソースに対して定義済みのアクションまたはカスタムアクションを実行できる、アクション処理をサポートしています。リソースでアクションを実行するには、次のコマンドを入力します。

```
# hares -action res token [-actionargs arg1 ...] [-sys
system] [-rclus cluster]
```

エージェントは、次に示す事前定義済みのアクションをサポートしています。

表 1-1 事前定義済みのアクショントークンとその説明

定義済みの action_token	説明
VRTS_GetInstanceName	設定された Db2udb リソースの DB2 インスタンス名を取得します。
VRTS_GetRunningServices	Db2udb リソースのエージェントによって監視されているプロセスの一覧を取得します。

たとえば、次のようにします。

```
# hares -action db2udb1 VRTS_GetInstanceName -sys systemName
VCS 通知 V-16-13323 リソース (db2udb0) : アクション
(VCS NOTICE V-16-13323 Resource (db2udb0): action)
(VRTS_GetInstanceName) が正常に完了しました。出力は次のとおりです。
((VRTS_GetInstanceName) completed successfully. Output is:)

db2inst1
```

```
# hares -action db2udb0 VRTS_GetRunningServices -sys systemName
```

```
VCS 通知 V-16-13323 リソース (db2udb0): アクション
(VCS NOTICE V-16-13323 Resource (db2udb0): action)
(VRTS_GetRunningServices) が正常に完了しました。
((VRTS_GetRunningServices) completed successfully.)
出力は次のとおりです。(Output is:)
```

```
PARTITION: 0
```

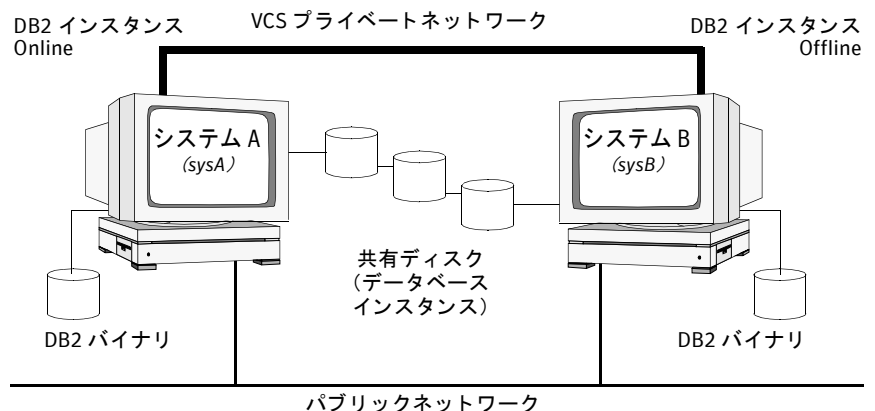
PID	TTY	TIME	CMD
9800	?	0:06	db2sysc

このエージェントのカスタムアクションを追加することもできます。カスタム `action_token` の定義については、『Veritas Cluster Server エージェント開発者ガイド』を参照してください。

DB2 UDB を VCS クラスタ にインストールするための 準備

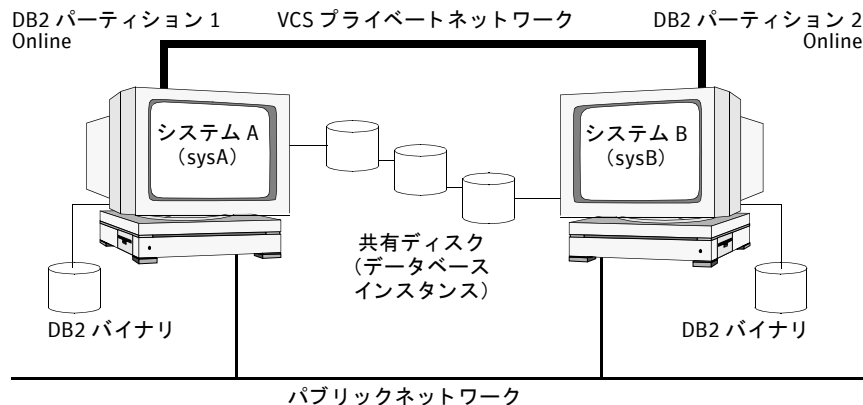
次の例は、2 ノードのクラスタで VCS が設定されています。DB2 UDB システムバイナリは、sysA と sysB 上のローカルファイルシステムでまったく同じになるように、インストールします。インスタンスホームディレクトリ、インスタンスバイナリおよびデータベースは、両方のノードから利用できるように共有ストレージにインストールします。非 MPP 構成の場合、インスタンスは一度に 1 つのシステム上でのみオンラインになり、他方のシステムはフェールオーバーシステムになります。

非 MPP 構成の DB2



MPP 構成の場合、データベースパーティションは各システム上で実行でき、各システムはフェールオーバーシステムになることができます。

MPP 構成の DB2



DB2 UDB をインストールするための前提条件 (非 MPP 構成)

- クラスタ内のすべてのシステムで DB2 UDB が動作するためのリソース（共有メモリなど）が十分にある必要があります。DB2 のメモリの必要条件は、DB2 のバージョンやハードウェア構成によって異なるため、確認してください。DB2 UDB システムバイナリはローカルにインストールされ、DB2 UDB データベースインスタンスは共有ストレージにインストールされている必要があります。
- VCS バージョン 5.0 をインストールし、設定します。インストーラや `installvcs` ユーティリティの実行について詳しくは『Veritas Cluster Server インストールガイド』を参照してください。
- DB2 UDB をインストールする前に、DB2 UDB のユーザーアカウントとグループアカウントを定義します。
次を参照してください。
 - 24 ページの「[DB2 のユーザーアカウントとグループアカウントの定義](#)」
 - 関連する『DB2 Universal Database 概説およびインストール』マニュアル

DB2 UDB をインストールするための前提条件 (MPP 構成)

- クラスタ内のすべてのシステムで DB2 UDB が動作するためのリソース（共有メモリなど）が十分にある必要があります。DB2 のメモリの必要条件は、DB2 のバージョンやハードウェア構成によって異なるため、確認してください。DB2 UDB システムバイナリは各システムのローカルファイルシステムにインストールされ、DB2 UDB データベースインスタンスは共有ストレージにインストールされている必要があります。
- MPP 構成には、Storage Foundation Cluster File System ソフトウェアが必要です。このソフトウェアには、DB2 用の Veritas エージェントに必要なクラスタファイルシステムコンポーネント、Veritas Cluster Server (VCS)、クラスタ機能が有効な Veritas Volume Manager (CVM)、クラスタ機能が有効な Veritas File System (CFS) が含まれています。
これらの製品の詳細、インストーラや `installsfcs` ユーティリティの実行について詳しくは『Veritas Storage Foundation Cluster File System インストールガイド』と『Veritas Storage Foundation Cluster File System 管理者ガイド』を参照してください。
- DB2 UDB をインストールする前に、DB2 UDB のユーザーアカウントとグループアカウントを定義します。
次を参照してください。
 - 関連する『DB2 Universal Database 概説およびインストール』マニュアル
 - 24 ページの「DB2 のユーザーアカウントとグループアカウントの定義」

DB2 インスタンス用のファイルシステムの作成

次の項では、DB2 データベースインスタンス用のディスクグループを作成する例について説明します。

DB2 非 MPP インスタンス用のファイルシステムの作成

ファイルシステムを作成するには、最初に、物理的に共有されているディスク上にディスクグループを作成し、そのディスクグループ内に十分なサイズのボリュームを作成します。

非 MPP インスタンス用のファイルシステムを AIX システム上に作成するには

- 1 共有ディスク上にディスクグループを作成します。lsdev -Cc disk コマンドを使って、ディスクを一覧表示します。この場合は、グループを hdisk5 の 1 つのディスクで構成します。たとえば、次のようにします。

```
# vxvg init db2db_dg hdisk5
```

ディスクグループをデポートし、インポートします。

```
# vxvg deport db2db_dg
```

```
# vxvg import db2db_dg
```

- 2 vxassist コマンドを使って、3 GB のボリュームを作成します。

```
# vxassist -g db2db_dg make db2db_vol 3g
```

- 3 ファイルシステムを作成します。

```
# mkfs -V vxfs -o largefiles /dev/vx/dsk/db2db_dg/db2db_vol
```

- 4 マウントポイントディレクトリを作成し、ファイルシステムをマウントします。

```
# mkdir /db2_mnt/db2inst1
```

```
# mount -V vxfs /dev/vx/dsk/db2db_dg/db2db_vol \  
/db2_mnt/db2inst1
```

非 MPP インスタンス用のファイルシステムを Linux システム上に作成するには

- 1 共有ディスク上にディスクグループを作成します。vxdisk list コマンドを使って、ディスクを一覧表示します。この場合は、グループを sdc の 1 つのディスクで構成します。たとえば、次のようにします。

```
# vxvg init db2db_dg /dev/sdc
```

ディスクグループをデポートし、インポートします。

```
# vxvg deport db2db_dg
```

```
# vxvg import db2db_dg
```

- 2 vxassist コマンドを使って、3 GB のボリュームを作成します。

```
# vxassist -g db2db_dg make db2db_vol 3g
```

- 3 ファイルシステムを作成します。

```
# mkfs -t vxfs /dev/vx/dsk/db2db_dg/db2db_vol
```

- 4 マウントポイントディレクトリを作成し、ファイルシステムをマウントします。

```
# mkdir /db2_mnt/db2inst1
```

```
# mount -t vxfs /dev/vx/dsk/db2db_dg/db2db_vol \  
/db2_mnt/db2inst1
```

- 5 マウントポイントディレクトリを作成し、ファイルシステムをマウントします。

```
# mkdir /db2_mnt/db2inst1
```

```
# mount -t vxfs /dev/vx/dsk/db2db_dg/db2db_vol \  
/db2_mnt/db2inst1
```

非 MPP インスタンス用のファイルシステムを Solaris システム上に作成するには

- 1 共有ディスク上にディスクグループを作成します。vxdisk list コマンドを使って、ディスクを一覧表示します。この場合は、グループを c4t0d0s2 の 1 つのディスクで構成します。たとえば、次のようにします。

```
# vxdg init db2db_dg c4t0d0s2
```

ディスクグループをデポートし、インポートします。

```
# vxdg deport db2db_dg
# vxdg import db2db_dg
```
- 2 vxassist コマンドを使って、3 GB のボリュームを作成します。

```
# vxassist -g db2db_dg make db2db_vol 3g
```
- 3 ファイルシステムを作成します。

```
# mkfs -F vxfs /dev/vx/rdsk/db2db_dg/db2db_vol
```
- 4 マウントポイントディレクトリを作成し、ファイルシステムをマウントします。

```
# mkdir /db2_mnt/db2inst1
# mount -F vxfs /dev/vx/dsk/db2db_dg/db2db_vol
/db2_mnt/db2inst1
```
- 5 マウントポイントディレクトリを作成し、ファイルシステムをマウントします。

```
# mkdir /db2_mnt/db2inst1
# mount -F vxfs /dev/vx/dsk/db2db_dg/db2db_vol \
/db2_mnt/db2inst1
```

DB2 MPP インスタンス用の共有クラスタファイルシステムの作成

共有ファイルシステムを作成するには、最初に、物理的に共有されているディスク上に共有ディスクグループを作成し、そのディスクグループ内に十分なサイズのボリュームを作成します。Storage Foundation Cluster File System ソフトウェアがインストールされている必要があります。

MPP インスタンス用の共有ファイルシステムを AIX 上に作成するには

- 1 共有ディスクグループを作成するためのコマンドは、CVM マスターノードから発行する必要があります。ノードがマスターまたはスレーブのどちらであるかを判別するには、次のコマンドを入力します。

```
# vxdctl -c mode
```

出力の中の、次の行を探します。

```
cluster active - MASTER
```

または

```
cluster active - SLAVE
```
- 2 マスターノードから、ディスクグループを作成します。vxdisklist コマンドを使って、ディスクを一覧表示します。

- 共有ディスクグループを作成します。この場合は、グループを 1 つのディスクで構成します。この例では、ディスクは `hdisk5` です。

```
# vxrdg -s init db2db_dg hdisk5
```

- ディスクグループをデポートし、インポートします。

```
# vxrdg deport db2db_dg
# vxrdg -s import db2db_dg
```

- `vxassist` コマンドを使って、7 GB のボリュームを作成します。

```
# vxassist -g db2db_dg make db2db_vol 7g
```

- ファイルシステムを作成します。

```
# mkfs -V vxfs -o largefiles /dev/vx/rdisk/db2db_dg/db2db_vol
```

- マウントポイントディレクトリを作成し、ファイルシステムをマウントします。

```
# mkdir /db2_mnt/db2inst1
# mount -V vxfs -o cluster /dev/vx/dsk/db2db_dg/db2db_vol \
/db2_mnt/db2inst1
```

MPP インスタンス用の共有ファイルシステムを Linux 上に作成するには

- 共有ディスクグループを作成するためのコマンドは、CVM マスターノードから発行する必要があります。ノードがマスターまたはスレーブのどちらであるかを判別するには、次のコマンドを入力します。

```
# vxrdctl -c mode
```

出力の中の、次の行を探します。

```
cluster active - MASTER
```

または

```
cluster active - SLAVE
```

- マスターノードから、ディスクグループを作成します。`vxrdlist` コマンドを使って、ディスクを一覧表示します。

- 共有ディスクグループを作成します。この場合は、グループを 1 つのディスクで構成します。この例では、ディスクは `sdcc` です。

```
# vxrdg -s init db2db_dg hdisk5
```

- ディスクグループをデポートし、インポートします。

```
# vxrdg deport db2db_dg
# vxrdg -s import db2db_dg
```

- `vxassist` コマンドを使って、7 GB のボリュームを作成します。

```
# vxassist -g db2db_dg make db2db_vol 7g
```

- ファイルシステムを作成します。

```
# mkfs -t vxfs -o largefiles /dev/vx/rdisk/db2db_dg/db2db_vol
```

- 7 マウントポイントディレクトリを作成し、ファイルシステムをマウントします。

```
# mkdir /db2_mnt/db2inst1
# mount -t vxfs -o cluster /dev/vx/dsk/db2db_dg/dbq2db_vol \
/db2_mnt/db2inst1
```

MPP インスタンス用の共有ファイルシステムを Solaris 上に作成するには

- 1 共有ディスクグループを作成するためのコマンドは、CVM マスターノードから発行する必要があります。ノードがマスターまたはスレーブのどちらであるかを判別するには、次のコマンドを入力します。

```
# vxdctl -c mode
```

出力の中の、次の行を探します。

```
cluster active - MASTER
```

または

```
cluster active - SLAVE
```

- 2 マスターノードから、ディスクグループを作成します。vxdisklist コマンドを使って、ディスクを一覧表示します。
- 3 共有ディスクグループを作成します。この場合は、グループを 1 つのディスクで構成します。この例では、ディスクは c5t0d0s2 です。

```
# vxdg -s init db2db_dg c5t0d0s2
```

- 4 ディスクグループをデポートし、インポートします。

```
# vxdg deport db2db_dg
# vxdg -s import db2db_dg
```

- 5 vxassist コマンドを使って、7 GB のボリュームを作成します。

```
# vxassist -g db2db_dg make db2db_vol 7g
```

- 6 ファイルシステムを作成します。

```
# mkfs -F vxfs -o largefiles /dev/vx/rdisk/db2db_dg/db2db_vol
```

- 7 マウントポイントディレクトリを作成し、ファイルシステムをマウントします。

```
mkdir /db2_mnt/db2inst1
mount -F vxfs -o cluster /dev/vx/dsk/db2db_dg/db2db_vol \
/db2_mnt/db2inst1
```

DB2 のユーザーアカウントとグループアカウントの定義

DB2 UDB のバイナリをインストールしてインスタンスを作成する前に、各システム上の各インスタンスに対して DB2 UDB のユーザーアカウントとグループアカウントを定義する必要があります。次の必要条件に注意してください。

- DB2 のユーザーとグループの ID は、クラスタ内のすべてのシステムで同一になるようにしてください。
- DB2 インスタンスオーナーのホームディレクトリは、共有ストレージ上に作成された、DB2 インスタンスによって使われるマウントポイントで、各ノードに存在する必要があります。存在しない場合は、各ノードにマウントポイントディレクトリを作成します。
- すべての DB2 ユーザーアカウントがローカルシステム上に存在している必要があります。ユーザーをローカルに設定せず、NIS または NIS+ で定義している場合は、ネットワーク障害などでそのサービスが使えないことがあるため、クラスタでの使用を控えてください。またこれらのサービスが中断されると、VCS が正しく動作できない場合があります。

グループの作成

クラスタ内のノードにはそれぞれ 3 つのユーザーグループアカウントが必要です。

AIX システムのクラスタ内の各ノードでグループアカウントを作成するには

- 1 DB2 インスタンスオーナーのグループを作成します。たとえば次のように入力します。

```
# mkgroup id=999 db2iadm1
```
- 2 ブロックされたユーザ定義関数 (UDF: User-Defined Function) またはストアドプロシージャを実行するユーザのためのグループを作成します。たとえば次のように入力します。

```
# mkgroup id=998 db2fadm1
```
- 3 DB2 管理サーバー (DAS) のグループを作成します。たとえば次のように入力します。

```
# mkgroup id=997 db2asgrp
```

Linux システムや Solaris システムのクラスタ内の各ノードでグループアカウントを作成するには

- 1 DB2 インスタンスオーナーのグループを作成します。たとえば次のように入力します。

```
# groupadd -g 999 db2iadm1
```


- 2 ブロックされたユーザ定義関数 (UDF: User-Defined Function) またはストアドプロシージャを実行するユーザのためのグループを作成します。たとえば次のように入力します。

```
# groupadd -g 998 db2fadm1
```

- 3 DB2 管理サーバー (DAS) のグループを作成します。たとえば次のように入力します。

```
# groupadd -g 997 db2asgrp
```

AIX システムでのユーザーアカウントの追加

クラスタ内の各ノードでユーザーアカウントを作成します。

AIX システムのクラスタ内の各ノードでユーザーアカウントを作成するには

- 最初の例は、DB2 UDB インスタンスオーナーであるユーザー `db2inst1` の作成を示しています。マウントポイント `/db2_mnt/db2inst1` は、各ノードからアクセス可能な、共有ストレージ上の DB2 UDB インスタンスのホームディレクトリを置くファイルシステムに使われます。たとえば、次のようになります。

```
# mkuser id=1004 pgrp=db2iadm1 groups=db2iadm1 home=/  
db2_mnt/db2inst1 db2inst1
```

- 次の例は、`db2fenc1` と `db2as` のユーザーアカウントの作成を示しています。これらのユーザーのホームディレクトリは、各ノードのローカルファイルシステムの `/home` の下に置きます。

```
# mkuser id=1003 pgrp=db2fadm1 groups=db2fadm1 home=/home/  
db2fenc1 db2fenc1  
# mkuser id=1002 pgrp=db2asgrp groups=db2asgrp home=/home/  
db2as db2as
```

Linux システムや Solaris システムでのユーザーアカウントの追加

ユーザーアカウントの作成方法を示す次の例では、`-g` オプションがグループを、`-u` がユーザー ID を、`-d` がホームディレクトリをそれぞれ指定しています。また、`-m` は、ホームディレクトリが存在しない場合にそのディレクトリを作成することを、`-s` は、ユーザーのログインシェルを指定しており、最後にユーザーのログイン名を指定します。

クラスタ内の各ノードでユーザーアカウントを作成します。

Solaris システムや Solaris システムのクラスタ内の各ノードでユーザーアカウントを作成するには

- 最初の例は、DB2 UDB インスタンスオーナーであるユーザー `db2inst1` の作成を示しています。マウントポイント `/db2_mnt/db2inst1` は、各ノードからアクセス可能な、共有ストレージ上の DB2 UDB インスタンスのホーム

ディレクトリを置くファイルシステムに使われます。たとえば、次のようにします。

```
# useradd -g db2iadm1 -u 1004 -d /db2_mnt/db2inst1 -m -s  
/bin/ksh/ db2inst1
```

- 次の例は、db2fenc1 と db2as のユーザーアカウントの作成を示しています。これらのユーザーのホームディレクトリは、各ノードのローカルファイルシステムの /home の下に置きます。

```
# useradd -g db2fadm1 -u 1003 -d /home/db2fenc1 -m -s /bin/  
ksh db2fenc1  
# useradd -g db2asgrp -u 1002 -d /home/db2as -m -s /bin/ksh  
db2as
```

VCS 環境での DB2 UDB のインストール

VCS 環境で DB2 UDB バージョン 8 以上の ESE をインストールする場合は、関連する『DB2 Universal Database 概説およびインストール』マニュアルに記載されているインストール手順に従うことを推奨します。各ノードのローカルディスク上にバイナリをインストールし、各クラスタノードからアクセス可能な共有ストレージ上にデータベースインスタンスをインストールします。

共有メモリパラメータの設定

各ノード上の /etc/system ファイルに共有メモリパラメータを設定する作業について詳しくは、関連マニュアル『DB2 Universal Database 概説およびインストール』マニュアルも参照してください。

バイナリのインストール

DB2 UDB システムのバイナリは、共有ストレージではなく、各ノードのローカルディスクにインストールします（ミラー化したディスクへのインストールをお勧めします）。db2setup ツールを使えます。

DB2 ライセンスのインストール

各ノードに DB2 ライセンスをインストールします。たとえば次のように入力します。

```
# /opt/IBM/db2/V8.1/adm/db2licm -a db2ese.lic
```

インスタンスのインストール

インスタンスのホームディレクトリがマウントされているいずれか 1 つのノードで、共有ストレージにデータベース インスタンスをインストールします。シングルパーティションインスタンスをインストールすることもマルチパーティションインスタンスをインストールすることもできます。db2setup ツールを使えます。

- db2setup を使う場合は、[DB2 インスタンスのプロパティ (DB2 Instance Properties)] ウィンドウで、[システム・ブート時に DB2 インスタンスを自動開始してください。(Auto start DB2 instance at system boot)] オプションを選択しないでください (このオプションが、ご使用の DB2 バージョンに存在する場合)。VCS は、DB2 インスタンスを起動する前に、DB2 が属しているサービスグループのリソースを特定の順番でオンラインにする必要があります。
- インスタンスのホームディレクトリは、共有ストレージ上のマウントポイントです。

DB2 UDB 構成の設定

次の手順を使って、DB2 UDB ESE マルチパーティションインスタンス (非 MPP) と DB2 UDB ESE マルチパーティションインスタンス (MPP) を VCS 環境で設定します。

/etc/services の確認

クラスタ内の各システムで、more コマンドを使って /etc/services ファイルを確認します。

- 各パーティションにポート番号が設定されていることを確認します。予約されているポートの数は、パーティションの数によって異なります。
- ポートが他のサービスで使われないか確認します。必要に応じて手動で新たに番号を設定します。
- クラスタ内のすべてのシステムの /etc/services ファイルに同じエントリが用意されていることを確認してください。

次は、2 つの DB2 UDB インスタンス (db2inst1 と db2inst2) の例です。どちらのインスタンスにも 2 つのパーティションがあります。各インスタンスには、パーティションごとに 2 つのポートと 1 つの予備ポートが必要であり、したがってインスタンスごとに 4 つの回線が必要です。

```
# more /etc/services
DB2_db2inst1      60000/tcp
DB2_db2inst1_1    60001/tcp
DB2_db2inst1_2    60002/tcp
DB2_db2inst1_END  60003/tcp
```

```
DB2_db2inst2      60004/tcp
DB2_db2inst2_1    60005/tcp
DB2_db2inst2_2    60006/tcp
DB2_db2inst2_END  60007/tcp
```

このファイルで、重複して設定されているポート番号がないことを確認してください。

\$DB2InstHome/.rhosts の作成

各システムで、`$DB2InstHome/.rhosts` という名前のファイルを作成し、このファイルの中に + 文字を記述します。このファイルにより、パスワード入力を求められることなくシステムがデータベースにアクセスできるようになります。セキュリティが問題である場合は、次の例のように、`hostname` と `userid` を `.rhosts` ファイルの中に設定します。

```
dbmach01  db2inst1
dbmach02  db2inst1
dbmach03  db2inst1
dbmach04  db2inst1
```

または

```
+ db2inst1
```

コマンド `rsh system_name` を使って、DB2 インスタンス (`db2inst1` など) で、クラスタのあるシステムから別のシステムにパスワード入力を求められずにリモートログインできることをテストしてください。このテストをクラスタの各システムから他のすべてのシステムに対して実行します。

SUSE での ssh の設定

SUSE で `ssh` を設定するには

- 1 VCS のインストールを実行するシステムにログオンします。
- 2 このシステムで、次のコマンドを実行して DSA キーペアを生成します。

```
# ssh-keygen -t dsa
```
- 3 デフォルトの場所 `~/.ssh/id_dsa` を使います
- 4 メッセージが表示されたら、パスフレーズを入力し、確認します。
- 5 次のように入力して `.ssh` ディレクトリの権限を変更します。

```
# chmod 755 ~/.ssh
```
- 6 ファイル `~/.ssh/id_dsa.pub` には、`ssh_dss` で始まり、ファイルが作成されたシステムの名前で終わる行が含まれています。VCS をインストールする予定のすべてのシステムで、この行を `/root/.ssh/authorized_keys2` ファイルにコピーします。

メモ: ローカルシステムがクラスタの一部である場合は、必ずそのシステムの `authorized_keys2` ファイルを編集してください。

- 7 インストールを実行するシステムで次のコマンドを実行します。

```
# exec /usr/bin/ssh-agent $SHELL
# ssh-add
```

メモ: このステップはシェルに固有なので、そのシェルが存続している間は有効です。

- 8 メッセージが表示されたら、DSA パスフレーズを入力します。
これで、インストールを行う予定の任意のシステム、またはクラスタ外部の独立したコンピュータで `installvcs` スクリプトを実行することによって、複数のシステムに VCS をインストールする準備ができました。
各シェルで `ssh-agent` が実行されることがないように、X-Window システムを実行し、パスフレーズの入力を求められないように設定します。詳しくは Red Hat のマニュアルを参照してください。

\$DB2InstHome/sqllib/db2nodes.cfg ファイルの変更

DB2 は、ノード間のフェールオーバー中に `$DB2InstHome/sqllib/db2nodes.cfg` ファイルを使います。

非 MPP 構成の場合

各 DB2 UDB ESE マルチパーティションインスタンス (非 MPP) データベースパーティションに、エントリを作成するファイル `$DB2InstHome/sqllib/db2nodes.cfg` に仮想 IP アドレスをホスト名として設定して変更します。たとえば、次のようにします。

```
0 virtualhostname 0
1 virtualhostname 1
```

`virtualhostname` は、`/etc/hosts` ファイルで仮想 IP アドレスと関連付けを行います。この仮想 IP アドレスが動作していることを確認してください。

MPP 構成の場合

MPP 構成の場合、各データベースパーティションを起動するホスト名を使って `$DB2InstHome/sqllib/db2nodes.cfg` ファイルを変更します。DB2 は、データベースパーティションをノード間でフェールオーバーできるように、`db2nodes.cfg` ファイルを自動的に変更し、更新します。DB2 は、4 番目の列に `netname` 値 (デフォルトではホスト名) を追加します。MPP 構成の場合、`db2nodes.cfg` ファイルでは仮想 IP は使われません。

たとえば、次のようにします。

```
0 sysA 0
1 sysB 0
2 sysC 0
3 sysD 0
```

3 列目で指定する相対ポート番号が、ホスト上の各パーティションで一意であるようにします。たとえば、次のようにします。

```
0 sysA 0
1 sysA 1
2 sysB 0
3 sysC 0
4 sysD 0
```

MPP 構成と非 MPP 構成の DB2 UDB のインストールの設定確認

共有ファイルシステムがマウントされているホストで、各インスタンスを起動および停止できるかどうかを確認し、DB2 が正しくインストールされていることを確認します。

DB2 インスタンスの起動と停止が可能かどうかを確認するには

- 1 インスタンスのオーナーとしてログインします。
`# su - db2inst1`
- 2 インスタンスを起動します。
`$ db2start`
アプリケーションは、db2nodes.cfg ファイルに指定されているノードで起動する必要があります。
- 3 前のコマンドが成功した場合、インスタンスを停止します。
`$ db2stop`
- 4 各ノードでアプリケーションが正常に起動しない場合や、正しく停止しない場合は、設定を確認してください。DB2 UDB のマニュアル (『DB2 Universal Database 概説およびインストール』など) を参照してください。
- 5 データベースを作成します。
`$ db2 create database dbname`
- 6 データベースディレクトリを表示します。
`$ db2 list database directory`

クラスタ内のその他の DB2 設定を確認するには

- 1 VCS クラスタの各ノードごとにディスクグループをインポートし、ディスクグループ内のすべてのボリュームを起動します。
- 2 DB2 インスタンスとデータベースを含むボリュームのファイルシステムをマウントします。

- 3 前に説明した手順を実行します。
30 ページの「[DB2 インスタンスの起動と停止が可能かどうかを確認するには](#)」を参照してください。
- 4 ディスクグループをマウント解除し、デポートします。
- 5 クラスタ内の各ノードでこの手順を繰り返します。

DB2 用エージェントのインストール

次の手順で、DB2 UDB エージェントソフトウェアをインストールします。

DB2 UDB エージェントソフトウェアのインストール

AIX システムにエージェントをインストールするには

- 1 スーパーユーザーとしてログインします。
- 2 ディスクドライブのデバイスアクセス名を特定します。たとえば次のように入力します。

```
# cd /dev
# lsdev -C -c cdrom
```

出力は次のようになります。

```
cd0 Available 10-60-00-4,0 16 Bit SCSI Multimedia CD-ROM Drive
```

この例では、`cd0` がドライブのデバイスアクセス名です。

- 3 DB2 UDB エージェントソフトウェアが収録されているソフトウェアディスクを、システムのディスクドライブに挿入します。
- 4 [手順 2](#) で判明したデバイスアクセス名を使ってソフトウェアディスクをマウントします。

```
# mkdir -p /cdrom
# mount -V cdrfs -o ro /dev/cd0 /cdrom
# cd /cdrom
```

- 5 次のように入力して DB2 UDB エージェントソフトウェアをインストールします。

```
# installp -ac -d /cdrom/VRTSvcsdb.rte.bff VRTSvcsdb.rte
```

- 6 DB2 サービスグループの一部になる各システムで、この手順を繰り返します。

Linux システムにエージェントをインストールするには

- 1 スーパーユーザーとしてログインします。
- 2 システムのドライブにソフトウェアディスクを挿入します。ディスクは自動的にマウントされます。
ディスクが自動的にマウントされない場合は、次のように入力します。

```
# mount -o ro /dev/cdrom /mnt/cdrom
```
- 3 rpm があるディレクトリに移動します。
 - RHEL 4 の場合：

```
# cd /mnt/cdrom/rhel4_i686/cluster_server_agents/db2_agent/rpms
```
 - SLES 9 の場合：

```
# cd /mnt/cdrom/sles9_i586/cluster_server_agents/db2_agent/rpms
```
- 4 DB2 UDB エージェントソフトウェアをインストールします。

```
# rpm -i VRTSvcsdb-5.0.00.10-GA_GENERIC.noarch.rpm
```
- 5

```
# rpm -q VRTSvcsdb
```

 と入力して、パッケージがインストールされたことを確認します。
出力は次のようになります。

```
VRTSvcsdb-5.0.00.10-GA_GENERIC
```

Solaris システムにエージェントをインストールするには

- 1 スーパーユーザーとしてログインします。
- 2 インストール用の一時ディレクトリを作成します。

```
# mkdir /tmp/install
```
- 3 システムに接続されているドライブに、適切なディスクを挿入します。
 - Solaris ボリューム管理ソフトウェアを実行している場合は、このソフトウェアがディスクを `/cdrom/cdrom0` として自動的にマウントします。次のコマンドを入力してディスクの場所に移動します。

```
# cd /cdrom/cdrom0
```
 - Solaris ボリューム管理ソフトウェアを実行していない場合、ディスクを手動でマウントする必要があります。たとえば、次のようになります。

```
# mount -F hsfs -o ro /dev/dsk/c0t6d0s2 /cdrom
```


この例では、`/dev/dsk/c0t6d0s2` がデフォルトの CD ドライブです。
ディスクをマウントしたら次のコマンドを入力して、ディスクの場所に移動します。

```
# cd /cdrom
```
- 4 圧縮されたパッケージファイルを、ソフトウェアディスクから一時ディレクトリへコピーします。

```
# cp -r db2_agent/pkgs/* /tmp/install
```

- 5 一時ディレクトリに移動し、圧縮されたパッケージファイルを解凍します。

メモ: システムに `gunzip` ユーティリティがインストールされていない場合は、次のコマンドでディスクからコピーできます。

```
# cp /cdrom_path/gnu/gunzip /tmp/install
```

```
# cd /tmp/install  
# gunzip VRTS*.gz
```

- 6 `tar` ファイルで圧縮されたファイルを解凍します。

```
# tar -xvf VRTSvcsdb.tar
```

- 7 パッケージをインストールします。

```
# pkgadd -d . VRTSvcsdb
```

Solaris システムに日本語の言語パックをインストールするには

- 1 エージェントをインストールしたら、システムのドライブに言語ディスクを挿入します。次のコマンドを入力します。

```
# cd /cdrom/cdrom0
```

- 2 圧縮されたパッケージファイルを、ソフトウェアディスクから一時ディレクトリへコピーします。

```
# cp -r ja/db2_agent/pkgs/* /tmp/install
```

- 3 一時ディレクトリに移動し、圧縮されたパッケージファイルを解凍します。

```
# cd /tmp/install  
# gunzip VRTSjacsb.tar.gz
```

- 4 `tar` ファイルで圧縮されたファイルを解凍します。

```
# tar -xvf VRTSjacsb.tar
```

- 5 日本語版パッケージをインストールします。

```
# pkgadd -d . VRTSjacsb
```


DB2 用エージェントの設定

この章では、DB2 UDB エージェントの設定方法について説明します。エージェントは、次の 3 通りの方法で設定できます。

- VCS Cluster Manager (Java コンソール) を使って DB2 UDB エージェントのサービスグループテンプレートを編集します。
45 ページの「[Cluster Manager を使った DB2 UDB エージェントの設定](#)」を参照してください。
- VCS コマンドを使います。コマンドラインから VCS を設定する方法については、『Veritas Cluster Server ユーザーズガイド』を参照してください。
- タイプ設定ファイルを使い、DB2 UDB エージェントに付属の `main.cf` ファイルの例を参照して、`main.cf` ファイルを直接編集します。この方法では、編集前に VCS を停止し、編集後に VCS を再起動する必要があります。
52 ページの「[main.cf ファイルの編集による DB2 UDB エージェントの設定](#)」を参照してください。

エージェントの設定

DB2 UDB エージェントの設定には、DB2 UDB リソースタイプ属性への値の設定が含まれます。これについては、次の表で説明しますので参照してください。リソースタイプの定義ファイル `Db2udbTypes.cf` も参考として紹介します。`main.cf` 設定ファイルの例については、[付録 A 「設定ファイルの例」](#) を参照してください。

Db2udb リソースタイプ属性

DB2 リソースにはいくつかの必須属性と省略可能な属性があります。

表 4-1 DB2 用エージェントの必須属性

必須属性	説明
DB2InstHome	DB2 インスタンスの設定ファイルが格納されている DB2 UDB インスタンスホームディレクトリへのパス。 データ形式と値の種類: 文字列 - スカラー
DB2InstOwner	DB2 UDB インスタンスを起動するインスタンスのオーナーのユーザー ID。各インスタンスには、一意のユーザー ID が必要です。 データ形式と値の種類: 文字列 - スカラー 注意: この属性に不正な変更を加えると、DB2 が一貫性のない状態になる可能性があります。

表 4-2 DB2 用エージェントの省略可能な属性

省略可能な属性	説明
DatabaseName	詳細監視用のデータベースの名前。詳細監視が有効な場合 (InDepthMonitor = 1) に必要です。 メモ: MPP 環境では、この属性が変わるとすべてのパーティションでその変更が反映されます。 データ形式と値の種類: 文字列 - スカラー
NodeNumber	データベースのノード番号またはパーティション番号。ESE マルチパーティションインスタンス環境で特定のデータベースパーティションを監視するときに使われます。ESE シングルパーティションインスタンスでも ESE マルチパーティションインスタンスでもデフォルト値は 0 です。 データ形式と値の種類: 整数 - スカラー

表 4-2 DB2 用エージェントの省略可能な属性

省略可能な属性	説明
StartUpOpt	<p>起動オプションを提供します。許可される値は START、ACTIVATEDB、CUSTOM です。</p> <ul style="list-style-type: none">■ START (デフォルト) DB2 インスタンスまたはパーティションを起動します。■ ACTIVATEDB db2 プロセスの起動後にデータベースをアクティブにするコマンドを実行します。■ CUSTOM エージェントは、StartUpOpt 属性が CUSTOM に設定されている場合、すべてのオンライン処理をユーザーに完全に任せます。エージェントは <code>/opt/VRTSagents/ha/bin/Db2udb</code> ディレクトリで、<code>start_custom_\$db2instance_\$nodenum</code> という名前のファイルを検索します。このファイルが存在し、実行可能である場合、エージェントは代わりに、カスタマイズされたこのオンラインファイルを実行します。 例: <code>db2inst1</code> という名前の <code>db2</code> インスタンスに関してパーティション / ノード番号 <code>1</code> のオンライン処理をカスタマイズするため、DB2 用のエージェントは、<code>/opt/VRTSagents/ha/bin/Db2udb</code> ディレクトリにあるカスタマイズされたこのファイル (<code>start_custom_db2inst1_1</code>) を実行します。<p>データ形式と値の種類: 文字列 - スカラー</p>
ShutDownOpt	<p>この属性で許可される値は STOP と CUSTOM です。</p> <ul style="list-style-type: none">■ STOP DB2 インスタンスまたはパーティションを通常の方法でシャットダウンします。■ CUSTOM ShutDownOpt 属性が CUSTOM に設定されている場合、すべてのオフライン処理をユーザーに完全に任せます。エージェントは、<code>/opt/VRTSagents/ha/bin/Db2udb</code> ディレクトリで、<code>stop_custom_\$db2instance_\$nodenum</code> という名前のファイルを検索します。このファイルが存在し、実行可能である場合、エージェントは代わりに、カスタマイズされたこのオフラインファイルを実行します。 例: <code>db2inst1</code> という名前の <code>db2</code> インスタンスに関してパーティション / ノード番号 <code>0</code> のオフライン処理をカスタマイズする場合、DB2 用のエージェントは、<code>/opt/VRTSagents/ha/bin/Db2udb</code> ディレクトリにあるカスタマイズされたこのファイル (<code>stop_custom_db2inst1_0</code>) を実行します。<p>データ形式と値の種類: 文字列 - スカラー</p>

表 4-2 DB2 用エージェントの省略可能な属性

省略可能な属性	説明
IndepthMonitor	<p>詳細監視を有効にする場合は、IndepthMonitor 属性の値を 1 に設定します。このリリースより前の IndepthMonitor では、データベースに対してデフォルトの SQL クエリーが実行されていました。5.0 では、このデフォルトのクエリーは存在しなくなりました。現在、エージェントは、/opt/VRTSagents/ha/bin/Db2udb ディレクトリで custom_monitor_\$db2instance_\$nodenum ファイルを検索します。</p> <p>ファイルが存在し、実行可能な場合、エージェントはカスタマイズされたこの詳細監視ファイルを実行します。sample_db2udb ディレクトリに、カスタム監視スクリプトの例があります。</p> <p>データ形式と値の種類：文字列 - 整数</p>
Encoding	<p>DB2 UDB の出力表示で使うエンコードに対応する、オペレーティングシステムのエンコードを指定します。たとえば、環境変数 LANG が「ja」に設定されている場合は、「eucJP」が Solaris の Encoding 属性値になります。各エンコードの値については、DB2 UDB および Solaris のマニュアルを参照してください。デフォルトは「」です。</p> <p>データ形式と値の種類：文字列 - スカラー</p>
AgentDebug	<p>1 に設定すると、エージェントは追加のデバッグメッセージをログに記録します。</p> <p>データ形式と値の種類：ブール - スカラー</p>
WarnOnlyIfDBQueryFailed	<p>この属性では、SQL エラーをログに記録するか、特別な処理が必要なエラーをチェックするかを指定します。</p> <p>WarnOnlyIfDBQueryFailed 属性の値を 1 に設定すると有効になります。有効にすると、すべての SQL エラーは無視され、1 日に 1 回警告メッセージがエージェントログに記録されます。</p> <p>WarnOnlyIfDBQueryFailed 属性の値を 0 に設定すると無効になります。無効にすると、db2error.dat ファイルでエラーコードに特別な処理が指定されているかどうかチェックされます。db2error.dat ファイル内にそのエラーコードが存在しない場合は、監視のために OFFLINE が返されます。そのエラーコードが存在する場合は、db2error.dat ファイルに記載されたそのエラーコードのアクションが実行されます。</p> <p>データ形式と値の種類：ブール - スカラー</p>
ContainerName	<p>Solaris ゾーンの名前 (Solaris 10 のみ)</p> <p>データ形式と値の種類：文字列 - スカラー</p>

表 4-3 DB2 用エージェントの内部属性

必須属性	説明
AgentDirectory	エージェントに関連するその他のファイルとスクリプトの場所を示します。 使用禁止。内部使用。

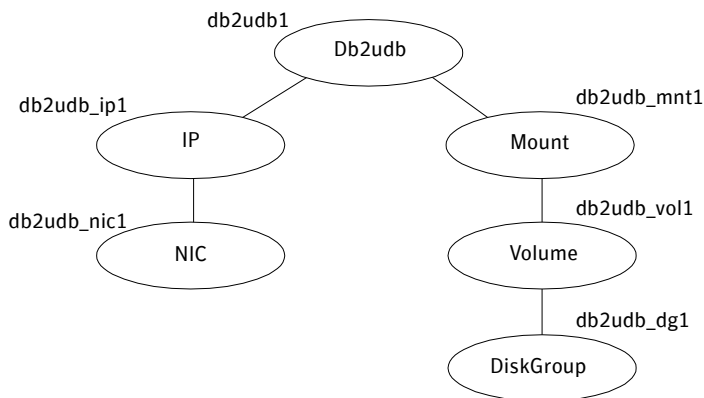
DB2 UDB タイプ定義ファイル : Solaris 用の Db2udbTypes.cf

```
type Db2udb (  
    static str ContainerType = Zone  
    static str AgentDirectory = "/opt/VRTSagents/ha/bin/Db2udb"  
    static str AgentFile = "/opt/VRTSagents/ha/bin/Db2udb/  
    Db2udbAgent"  
    static keylist SupportedActions = { VRTS_GetInstanceName,  
    VRTS_GetRunningServices }  
    static int CleanTimeout = 240  
    static int MonitorTimeout = 240  
    static int OfflineTimeout = 240  
    static int OnlineRetryLimit = 2  
    static int OnlineTimeout = 180  
    static int OnlineWaitLimit = 1  
    static int RestartLimit = 3  
    static int ToleranceLimit = 1  
    static str ArgList[] = { DB2InstOwner, DB2InstHome,  
    IndepthMonitor, DatabaseName, NodeNumber, StartUpOpt,  
    ShutDownOpt, AgentDebug, Encoding, WarnOnlyIfDBQueryFailed,  
    LastWarningDay, ContainerName }  
    str DB2InstOwner  
    str DB2InstHome  
    int IndepthMonitor  
    str DatabaseName  
    int NodeNumber  
    str StartUpOpt = START  
    str ShutDownOpt = STOP  
    boolean AgentDebug = 0  
    str Encoding  
    boolean WarnOnlyIfDBQueryFailed = 1  
    temp str LastWarningDay  
    str ContainerName  
)
```

非 MPP 構成の DB2 UDB サービスグループ

図 4-1 は、非 MPP 構成 DB2 UDB インスタンスリソースグループ用に設定されたリソース間の依存関係を示しています。

図 4-1 Db2udb インスタンスの依存関係ツリー



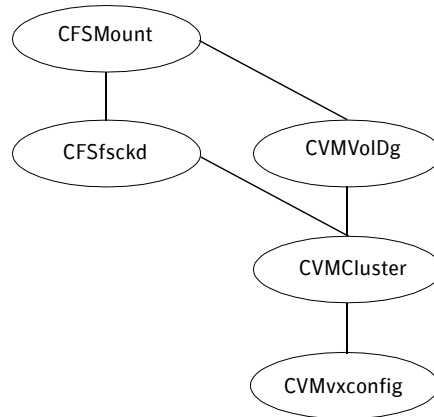
この設定は、DB2 UDB インスタンス用のサービスリソースグループを示しています。db2udb1 リソース（データベース）には、IP リソースと Mount リソースが必要です。DB2 UDB サーバーのサービスグループの IP アドレスは、IP リソース（db2udb_ip1）と NIC リソース（db2udb_nic1）を使って設定されます。Mount リソース（db2udb_mnt1）は Volume リソース（db2udb_vol1）に依存し、Volume リソースは DiskGroup リソース（db2udb_dg1）に依存しています。これらのリソースがオンラインになった後に、DB2 UDB インスタンスを起動できます。

MPP 構成の DB2 UDB サービスグループ

DB2 UDB エージェントは、2つのサービスグループを使って DB2 バージョン 8.1 以上の MPP 構成をサポートします。

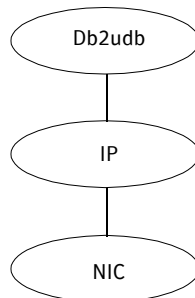
最初はベースとなる CVM サービスグループで、これはパラレルグループです。このサービスグループをクラスタの各ノード上に設定します。このサービスグループには、CVM リソースと、CFS のサポートに必要なリソースがあります。また、このグループには、すべてのクラスタノードで共有されるインスタンスのホームディレクトリなど、DB2 に必要なすべての共通コンポーネントも含まれます。

図 4-2 CVM パラレルサービスグループ



次は DB2 サービスグループで、これはフェールオーバーグループです。このサービスグループは、MPP 構成の DB2 バージョン 8 において 1 つのデータベースパーティションを監視します。DB2 サービスグループ（フェールオーバー）と CVM サービスグループ（パラレル）は、online local firm の依存関係を持ちます。

図 4-3 DB2 フェールオーバーサービスグループ



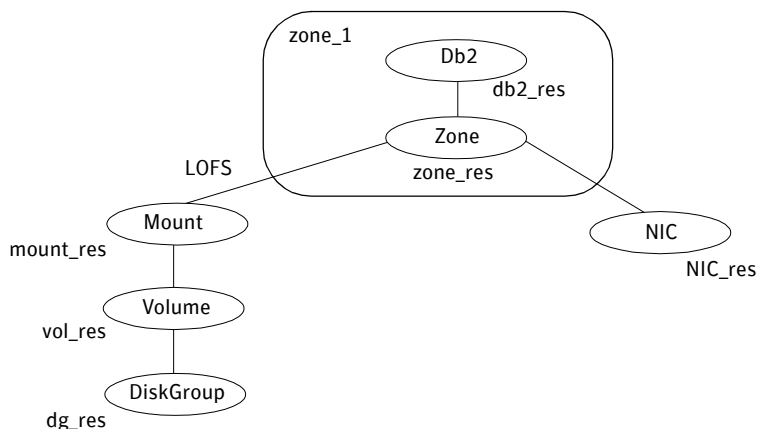
Solaris ゾーンに設定された DB2 UDB インスタンス

次の例は、Solaris ゾーンに設定された DB2 インスタンスの状態を監視するように設定された、標準的なサービスグループを示しています。

ローカルディスク上のゾーンルート

ゾーンのルートファイルシステムが各ノードのローカルディスク上にある場合、システムを起動するとこのファイルシステムがマウントされます。したがって、サービスグループは、ゾーン用に個別の **DiskGroup** リソースおよび **Volume** リソースを必要としません。

図 4-4 Solaris ゾーンに設定された DB2 UDB インスタンス



クラスタ内の共有ディスクグループとボリュームは、それぞれ **DiskGroup** タイプおよび **Volume** タイプのリソースとして設定します。これらのボリュームは、**Mount** エージェントによりマウントされます。**Solaris** ゾーンは、ゾーンリソースにより監視されます。このゾーンリソースは、**Mount** リソースおよび **NIC** リソースに依存します。これらの各リソースがオンラインにされた後で、**DB2** サーバーを起動できます。

DB2 インスタンスのホームディレクトリは、グローバルゾーンにマウントされています。非グローバルゾーンでファイルシステムを使用可能にするには、グローバルゾーンで次のコマンドを実行する必要があります。

次の例に示されている行では、グローバルゾーンの `/zones/db2data` を `/db2inst1` として、`zone1` という名前の非グローバルゾーンにマウントすることを指定します。使うファイルシステムタイプは **LOFS** です。この例にある `/db2inst1` ディレクトリは、**DB2** インスタンスのホームディレクトリです。

```
# zonecfg -z zone1
zonecfg:zone1> add fs
```

```
zonecfg:zone1:fs> set dir=/db2inst1
zonecfg:zone1:fs> set type=lofs
zonecfg:zone1:fs> set special=/zones/db2data
zonecfg:zone1:fs> end
```

VCS を Solaris 10 ローカルゾーンで使う場合の設定について詳しくは『Veritas Cluster Server ユーザーズガイド』の付録で非グローバルゾーンでの VCS の設定についての項を参照してください。

Cluster Manager を使った DB2 UDB エージェントの設定

DB2 UDB リソースグループのテンプレートは、DB2 用のエージェントをインストールしたときに自動的にインストールされます。VCS Cluster Manager (Java コンソール) でテンプレート (/etc/VRTSagents/ha/Templates/Db2udbGroup.tf) を使って、DB2 UDB サービスグループ、サービスグループのリソース、その属性を設定できます。また、Java コンソールを使って、設定の属性値を動的に変更することもできます。

Db2udbTypes.cf ファイルのインポート

DB2 UDB テンプレートを使う前に、Cluster Manager (Java コンソール) を使って Db2udbTypes.cf ファイルを VCS エンジンにインポートしてください。

Db2udbTypes.cf ファイルをインポートするには

- 1 クラスタのいずれかのシステムで、Cluster Manager を起動します。
hagui&
- 2 クラスタにログインし、Cluster Explorer が起動するのを待ちます。
- 3 [Cluster Explorer] ウィンドウで、[ファイル (File)] をクリックし、ドロップダウンメニューから [タイプのインポート (Import Types)] を選択します。要求があった場合は、クラスタを読み書き両用モードに切り替えます。
- 4 [タイプのインポート (Import Type)] ダイアログボックスで、[ファイル名 (File Name)] ボックスに Db2udbTypes.cf ファイルのパス名を入力します。
/etc/VRTSagents/ha/conf/Db2udb/Db2udbTypes.cf
- 5 [インポート (Import)] をクリックし、ファイルがインポートされるのを待ちます。
- 6 [Cluster Explorer] ウィンドウで、[設定の保存 (Save Configuration)] アイコンをクリックします。

DB2 UDB のタイプが VCS エンジンにインポートされたら、DB2 UDB エージェントを設定できます。

- MPP 構成 DB2 UDB を使っている場合：
46 ページの「[Cluster Manager を使った MPP 構成 DB2 UDB のサービスグループの追加](#)」を参照してください。
- 非 MPP 構成 DB2 UDB を使っている場合：
49 ページの「[Cluster Manager を使った非 MPP 構成 DB2 UDB のサービスグループの追加](#)」を参照してください。

Cluster Manager を使った MPP 構成 DB2 UDB のサービスグループの追加

Db2udbTypes.cf ファイルがインポートされている場合、テンプレート (/etc/VRTSagents/ha/Templates/Db2udbGroup.tf) を使ってサービスグループを設定できます。

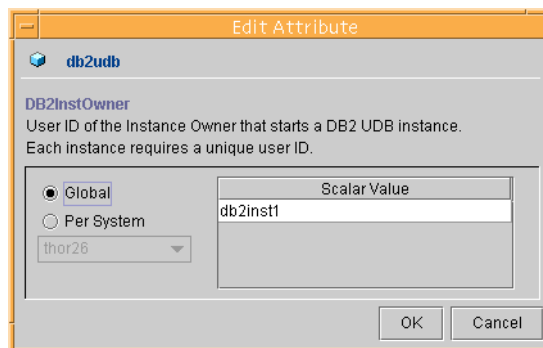
45 ページの「[Db2udbTypes.cf ファイルのインポート](#)」を参照してください。

Cluster Manager にログインすると、[状態 (Status)] タブには、クラスタの各システムで CVM サービスグループがオンラインであることが示されます。CVM サービスグループは、VERITAS Storage Foundation Cluster File System (SFCFS) ソフトウェアのインストールが完了すると自動的に設定されます。

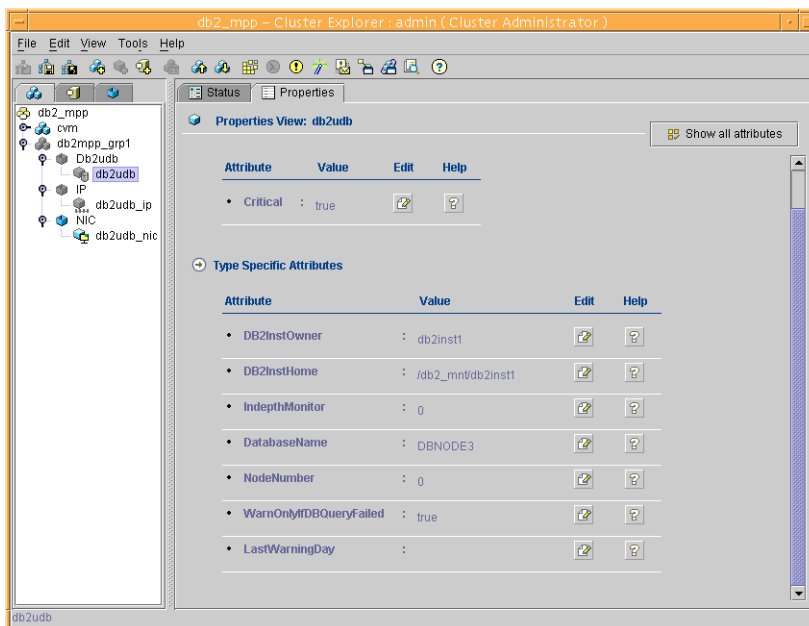
MPP 構成 DB2 UDB データベースのサービスグループを追加するには

- 1 [Cluster Explorer] ウィンドウで、ツールバーの [サービスグループの追加 (Add Service Group)] アイコンをクリックします。
- 2 [サービスグループの追加 (Add Service Group)] ウィンドウで、追加作成するサービスグループの名前を入力します。たとえば、db2mpp_grp1 と入力します。Return キーまたは Enter キーは押さないでください。
- 3 [使用可能なシステム (Available Systems)] ボックスで、設定に含めるシステムをダブルクリックします。
- 4 設定に追加するシステムが表示されたウィンドウで、サービスグループを自動的に起動するシステムのチェックボックスをオンにします。
- 5 [フェールオーバー (Failover)] ラジオボタンをクリックし、サービスグループタイプを指定します。
- 6 [テンプレート (Templates)] ボタンをクリックします。

- 7 [テンプレートの選択 (Select Templates)] ウィンドウで、[テンプレート (Templates)] ボックスに表示されているリストから `db2udb_mpp_grp` を選択します。依存関係のグラフ情報とタイプ情報が、選択したテンプレートに応じて変更されます。[OK] をクリックします。[サービスグループの追加 (Add Service Group)] ウィンドウに、選択に応じたテンプレートの名前が表示されます。
- 8 [サービスグループの追加 (Add Service Group)] ウィンドウの下部にある [OK] をクリックします。グループが追加されます。[Cluster Manager] ウィンドウの左のペインに、追加したサービスグループが CVM サービスグループの下に表示されます。[状態 (Status)] タブには、各システムのグループがオフラインで表示されます。
- 9 左のペインで、`db2mpp_grp1` サービスグループをダブルクリックします。グループに設定できるリソースのタイプ (Db2udb、IP、NIC) が表示されます。
- 10 Db2udb リソースタイプをダブルクリックします。Db2udb タイプの下のリソース `db2udb` を選択し、[プロパティ (Properties)] タブをクリックします。
- 11 db2udb リソースの [プロパティ (Properties)] タブに [タイプに固有の属性 (Type Specific Attributes)] のリストが表示されます。設定する各属性の [編集 (Edit)] アイコンをクリックします。[属性の編集 (Edit Attribute)] ウィンドウで、必要な属性値情報を入力します。たとえば、DB2InstOwner の値として `db2inst1` を入力します。



属性値を設定すると、[タイプに固有の属性 (Type Specific Attributes)] のリストは次のようになります。



- 12 db2udb リソースに値を設定したときと同じ方法で IP および NIC リソースの値を設定します。この場合、タイプをダブルクリックしてリソースを表示し、選択します。[プロパティ (Properties)] タブが表示された状態で、各リソースの [タイプに固有の属性 (Type Specific Attributes)] を編集できます。
- 13 左のペインで db2mpp_grp1 サービスグループを右クリックします。ドロップダウンメニューの [リンク (Link)] をクリックします。[サービスグループのリンク設定 (Link Service Groups)] ウィンドウの [親グループ (Parent Group)] に db2mpp_grp1 が、[子グループ (Child group)] に CVM が、[関係 (relationship)] に「online local」が、[依存関係の種類 (Dependency Type)] に「firm」が、それぞれ表示されます。
- 14 [OK] をクリックし、依存関係のリンクを作成します。
- 15 [設定の保存 (Save Configuration)] アイコンをクリックします。
- 16 db2udb および IP リソースを有効にします。リソースを右クリックし、ドロップダウンメニューの [有効 (Enabled)] をクリックします。必要に応じて、設定を読み書き両用モードにします。
- 17 [サービスグループをオンラインにします (Online Service Group)] アイコンをクリックします。

- 18 ウィンドウで、**Online** 状態にするサービスグループとシステムを選択します。[OK] をクリックします。

Cluster Manager を使った非 MPP 構成 DB2 UDB のサービスグループの追加

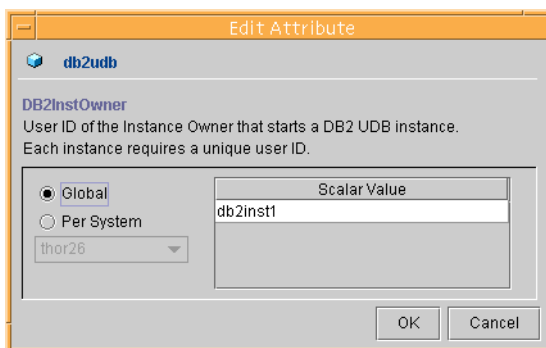
Db2udbTypes.cf ファイルがインポートされている場合、Db2udb_Group テンプレートを使ってサービスグループを設定できます。

- 45 ページの「[Db2udbTypes.cf ファイルのインポート](#)」を参照してください。

Db2udbTypes.cf ファイルがインポートされている場合にサービスグループを設定するには

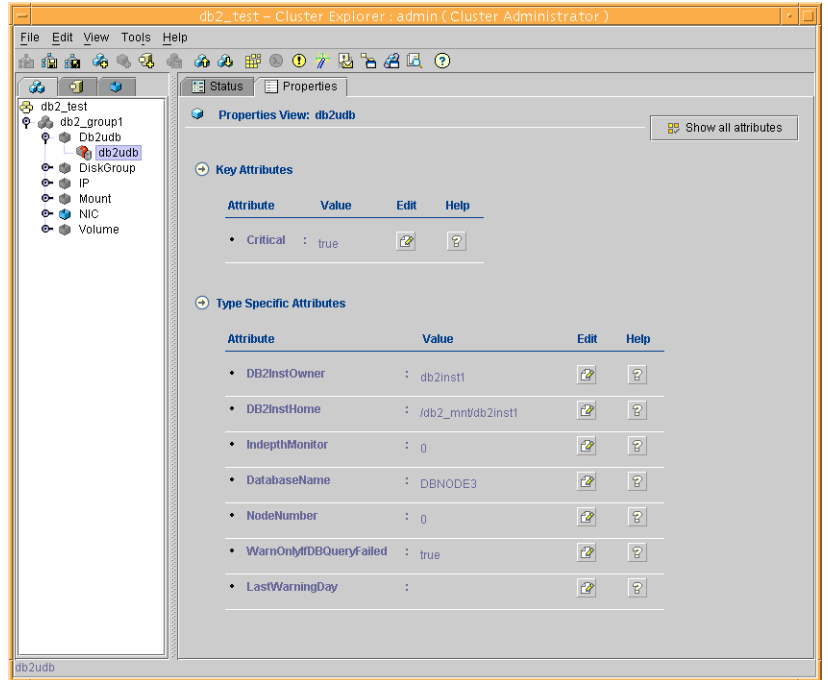
- 1 [Cluster Explorer] ウィンドウで設定ウィザードを使うかを確認するプロンプトが表示されたら、[いいえ (No)] を選択します。ウィザードの使用を選択した場合も、後続の手順は同じであることを注意してください。
- 2 [Cluster Explorer] ウィンドウで、ツールバーの [サービスグループの追加 (Add Service Group)] アイコンをクリックします。
- 3 [サービスグループの追加 (Add Service Group)] ウィンドウで、追加作成するサービスグループの名前を入力します。たとえば、**db2_group1** と入力します。Return キーまたは Enter キーは押さないでください。
- 4 [使用可能なシステム (Available Systems)] ボックスに表示されているシステムのうち、設定に含めるシステムをダブルクリックします。
- 5 設定に追加するシステムが表示されたウィンドウで、サービスグループを自動的に起動するシステムのチェックボックスをオンにします。
- 6 [フェールオーバー (Failover)] ラジオボタンをクリックし、サービスグループタイプを指定します。
- 7 [テンプレート (Templates)] ボタンをクリックします。
- 8 [テンプレートの選択 (Select Templates)] ウィンドウで、[テンプレート (Templates)] ボックスに表示されているリストから **db2udb_grp** を選択します。依存関係のグラフ情報とタイプ情報が、選択したテンプレートに応じて変更されます。[OK] をクリックします。[サービスグループの追加 (Add Service Group)] ウィンドウに、選択に応じたテンプレートの名前が表示されます。
- 9 [サービスグループの追加 (Add Service Group)] ウィンドウの下部にある [OK] をクリックします。グループが追加されます。[Cluster Manager] ウィンドウの左のペインに、追加したサービスグループがクラスタ名の下に表示されます。[状態 (Status)] タブには、各システムのグループがオフラインで表示されます。

- 10 左のパインで、**db2_group1** サービスグループをダブルクリックします。グループに対して設定できるリソースのタイプ (**Db2udb**、**DiskGroup**、**IP**、**Mount**、**NIC**、**Volume**) が表示されます。
- 11 **Db2udb** リソースタイプをダブルクリックします。**Db2udb** タイプの下のリソース **db2udb** を選択し、[プロパティ (**Properties**)] タブをクリックします。
- 12 **db2udb** リソースの [プロパティ (**Properties**)] タブに [タイプに固有の属性 (**Type Specific Attributes**)] のリストが表示されます。設定する各属性の [編集 (**Edit**)] アイコンをクリックします。[属性の編集 (**Edit Attribute**)] ウィンドウで、必要な属性値情報を入力します。たとえば、**DB2InstOwner** の値として **db2inst1** を入力します。



メモ : **DB2InstOwner** と **DB2InstHome** は必須の属性です。これらの属性を編集する必要があります。

属性値を設定すると、[タイプに固有の属性 (Type Specific Attributes)] のリストは次のようになります。



- 13 db2udb リソースに値を設定したときと同じ方法で DiskGroup、IP、Mount、NIC および Volume リソースの値を設定します。この場合、タイプをダブルクリックしてリソースを表示し、選択します。[プロパティ (Properties)] タブが表示された状態で、[タイプに固有の属性 (Type Specific Attributes)] を編集できます。
必須属性の一覧とそれらの属性の説明について詳しくは『Veritas Cluster Server 付属エージェントリファレンスガイド』を参照してください。
- 14 db2_group1 のリソースを有効にします。各リソースを右クリックし、ドロップダウンメニューの [有効 (Enabled)] をクリックします。必要に応じて、設定を読み書き両用モードにします。
- 15 [設定を保存して閉じます (Save and Close Configuration)] アイコンをクリックします。
- 16 [サービスグループをオンラインにします (Online Service Group)] アイコンをクリックします。
- 17 ウィンドウで、Online 状態にするサービスグループとシステムを選択します。オンライン状態にするシステムをクリックします。[OK] をクリックします。確認メッセージで [はい (Yes)] をクリックします。

main.cf ファイルの編集による DB2 UDB エージェントの設定

DB2 用の VCS エージェントでは、3 つの VCS 設定ファイルの例が /etc/VRTSagents/ha/conf/Db2udb/sample_db2udb ディレクトリにインストールされます。これらのサンプルはそれぞれ、ESE シングルパーティションインスタンス設定用、SMP 構成 ESE マルチパーティションインスタンス設定用、MPP 構成 ESE マルチパーティションインスタンス設定用です。適切なファイルを参考にして、現在の main.cf 設定ファイルを直接変更できます。この方式を利用した場合は、設定を変更する前に VCS を停止し、変更終了後に VCS を再起動する必要があります。

main.cf ファイルの編集準備を行うには

- 1 root ユーザーとしてシステム A にログインします。
- 2 main.cf ファイルを編集する前に、現行の各システムの状態、アプリケーションやシステムの設定を main.cf ファイルに保存し、クラスタを読み取り専用モードに切り替えます。

```
# haconf -dump -makero
```
- 3 main.cf の編集中は VCS を停止してください。hastop コマンドを発行してすべてのシステムで VCS エンジンを停止し、リソースを VCS 以外からコントロールすることが可能な状態にします。

```
# hastop -all -force
```
- 4 main.cf ファイルのバックアップコピーを作成します。

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

設定内容に応じて、次に記載した DB2 用のエージェントの設定方法について、適切な項目に進んでください。

MPP 構成 DB2 UDB を使うためのエージェントの設定

main.cf ファイルを編集します。/etc/VRTSagents/ha/conf/Db2udb/sample_db2udb/main.cf.MPP を参考に使います。CVM サービスグループは設定ファイルにあります。

MPP 構成 DB2 UDB を使うためにエージェントを設定するには

- 1 Db2udbTypes.cf ファイルを指す完全修飾パスを追加します。

```
include "/etc/VRTSagents/ha/conf/Db2udb/Db2udbTypes.cf"
```
- 2 DB2 UDB リソースのサービスグループを作成します。設定ファイルの例を参照してください。この例は、4 つの DB2 MPP サービスグループと CVM サービスグループを示します。

88 ページの「[MPP の main.cf 設定 : DB2 UDB ESE マルチパーティションインスタンス](#)」を参照してください。

- 3 DB2 MPP サービスグループで、Db2udb、IP および NIC の各リソースの定義を含めるとともに、リソース属性の設定パラメータに適切な値を設定します。

次を参照してください。

- 38 ページの「[Db2udb リソースタイプ属性](#)」
- 設定ファイルの例
- IP リソースと NIC リソースについては『Veritas Cluster Server 付属エージェントリファレンスガイド』

- 4 各 DB2 MPP サービスグループと cvm サービスグループを、online localfirm の依存関係に設定します。たとえば、次のようにします。

```
requires group cvm online local firm
```

- 5 サービスグループの依存関係を設定したあと、新しく作成したリソースの依存関係を設定します。該当する設定ファイルの例を参照してください。依存関係の設定について詳しくは『Veritas Cluster Server ユーザーズガイド』を参照してください。グループ db2mpp_grp0 について詳しくは 88 ページの「[MPP の main.cf 設定 : DB2 UDB ESE マルチパーティションインスタンス](#)」を参照してください。次に例を示します。

```
db2udb0 requires Db2_IP0  
Db2_IP0 requires Db2_NIC0
```

- 6 ファイルを保存して閉じます。

非 MPP 構成 DB2 UDB を使うためのエージェントの設定

main.cf ファイルを編集します。/etc/VRTSagents/ha/conf/Db2udb/sample_db2udb/main.cf.EE または /etc/VRTSagents/ha/conf/Db2udb/sample_db2/main.cf.EEE を参考に使います。

非 MPP 構成 DB2 UDB を使うためにエージェントを設定するには

- 1 Db2udbTypes.cf ファイルを指す完全修飾パスを追加します。

```
include "/etc/VRTSagents/ha/conf/Db2udb/Db2udbTypes.cf"
```

- 2 DB2 UDB リソースのサービスグループを作成します。

DB2 UDB ESE シングルパーティションインスタンスを使っている場合は、2 つのグループ db2udb_grp1 と db2udb_grp2 を示している例、73 ページの「[非 MPP の main.cf 設定 : DB2 UDB ESE シングルパーティションインスタンス](#)」を参照してください。

DB2 UDB ESE マルチパーティションインスタンスを使っている場合は、2 つのパーティションが定義された db2_grp1 というグループを示している例、66 ページの「[非 MPP の main.cf 設定 : DB2 UDB ESE マルチパーティションインスタンス](#)」を参照してください。

- 3 Db2udb、DiskGroup、IP、Mount、NIC、Volume の各リソースをはじめとする、サービスグループのすべてのリソースを追加し、リソース属性の設定パラメータに適切な値を設定します。

38 ページの「Db2udb リソースタイプ属性」および設定ファイルの例を参照してください。また、DiskGroup、IP、Mount、NIC、Volume の各リソースについて詳しくは『Veritas Cluster Server 付属エージェントリファレンスガイド』も参照してください。

- 4 新しく作成されたリソースの依存関係を設定します。該当する設定ファイルの例を参照してください（依存関係の設定について詳しくは『VCS ユーザーズガイド』を参照してください）。グループ db2udb_grp1 については、73 ページの「非 MPP の main.cf 設定 : DB2 UDB ESE シングルパーティションインスタンス」を参照してください。次に例を示します。

```
db2udb1 requires db2udb_ip1
db2udb1 requires db2udb_mnt1
db2udb_ip1 requires db2udb_nic1
db2udb_mnt1 requires db2udb_vol1
db2udb_vol1 requires db2udb_dg1
```

また、グループ db2udb_grp3 については、次のように入力します。

```
db2udb3 requires db2udb_ip3
db2udb3 requires db2udb_mnt3
db2udb_ip3 requires db2udb_nic3
db2udb_mnt3 requires db2udb_vol3
db2udb_vol3 requires db2udb_dg3
```

- 5 ファイルを保存して閉じます。

設定の確認

main.cf ファイルを編集したら、設定を確認してください。

設定を確認するには

- 1 /etc/VRTSvcs/conf/config/main.cf ファイルの構文を検証します。

```
# cd /etc/VRTSvcs/conf/config
# hacf -verify .
```
- 2 システム A で VCS エンジンを起動します。

```
# hstart
```
- 3 hastatus コマンドを入力します。

```
# hastatus
```
- 4 LOCAL_BUILD がメッセージ列に表示されている場合は、システム B でも VCS を起動します。

```
# hstart
```

- 5 すべての DB2 UDB サービスグループリソースが、システム A 上で Online 状態であることを確認します。

```
# hagr -display
```
- 6 システム A でサービスグループをオフラインにして、すべてのリソースが停止していることを確認します。

```
# hagr -offline db2udb_grp1 -sys sysa  
# hagr -offline db2udb_grp3 -sys sysa  
# hagr -display
```
- 7 システム A でサービスグループを再度オンラインにして、すべてのリソースが利用可能であることを確認します。

```
# hagr -online db2udb_grp1 -sys sysa  
# hagr -online db2udb_grp3 -sys sysa  
# hagr -display
```
- 8 DB2 UDB サービスグループをシステム B に切り替えます。

```
# hagr -switch db2udb_grp1 -to sysb  
# hagr -switch db2udb_grp3 -to sysb
```
- 9 すべての DB2 UDB サービスグループリソースが、システム B 上で Online 状態であることを確認します。

```
# hagr -display
```
- 10 すべてのシステムについて、次のログファイルでエラーや状態について調べてください。

```
/var/VRTSvcs/log/engine_A.log  
/var/VRTSvcs/log/Db2udb_A.log
```

エージェントの設定の変更

DB2 用の Veritas エージェントの設定を動的に変更するには、Cluster Manager を使うか、VCS コマンドラインを使います。以降の説明では、詳細監視に関する設定変更について、コマンドラインから VCS コマンドを使う方法を示します。『Veritas Cluster Server ユーザーズガイド』の、コマンドラインからの VCS の再設定に関する章を参照してください。

DB2 UDB インスタンスの詳細監視の有効化

DB2 UDB インスタンスの通常監視では、インスタンスまたはデータベースパーティションのアクティブプロセスを表示する db2nps の出力の確認か、または db2gcf コマンドの終了状態の確認が行われます。それに対して、詳細監視では、さらにデータベースに対するクエリーを実行してデータベースが利用可能かどうかを検証することによって、インスタンスやパーティション、およびそのデータベースの可用性においてより高いレベルの信頼性が得られます。

コマンドラインからの詳細監視の有効化

詳細監視は、動的に設定できます。詳細監視を開始する前に、エージェントのデフォルトの監視（通常監視）を使って DB2 UDB を正常に実行することを推奨します。MPP 構成の場合、データベースパーティションごとにデータベースにローカルにアクセスできることを確認してください。

英語以外のロケールの場合は、`$INSTHOME/sqlib/userprofile` ファイルに次の行を追加する必要があります。次の例では、日本語サポートを追加します。

```
LANG=ja
export LANG
```

特定のインスタンスの詳細監視を開始するには

- 1 VCS の設定を読み書き両用にします。
`# haconf -makerw`
- 2 設定が不完全なために、VCS が自動的にフェールオーバーを行わないよう、サービスグループをフリーズさせます。

```
# hagrps -freeze db2udb_grp1
```

- 3 次のコマンドを使って詳細監視を有効にします。

```
hares -modify resource DatabaseName name
hares -modify resource IndepthMonitor 1
```

たとえば、次のようにします。

```
# hares -modify db2udb DatabaseName SAMPLE
# hares -modify db2udb IndepthMonitor 1
# haconf -dump -makero
# hagrps -unfreeze db2udb_grp1
```

メモ: カスタム監視スクリプトが必要です。

40 ページの「[IndepthMonitor](#)」を参照してください。

詳細監視の無効化

詳細監視は、動的に無効にできます。

詳細監視を動的に無効にするには

- 1 VCS の設定を読み書き両用にします。
`# haconf -makerw`
- 2 設定が不完全なために、VCS が自動的にフェールオーバーを行わないよう、サービスグループをフリーズさせます。

```
# hagrps -freeze db2udb_grp1
```


- 3 **MonScript** 属性に NULL 値を設定して詳細監視を無効にします。次のコマンドを使います。

```
hares -modify resource IndepthMonitor 0
```

たとえば、次のようにします。

```
# hares -modify db2udb IndepthMonitor 0  
# haconf -dump -makero  
# hagrps -unfreeze db2udb_grp1
```


エージェントの無効化、 削除、アップグレード

この章では、DB2 UDB エージェントの無効化および削除方法について説明します。

エージェントの無効化

システム上のエージェントを無効にするには、最初にシステム上で DB2 UDB サービスグループを OFFLINE 状態に変更する必要があります。アプリケーションを完全に停止し、サービスグループを他のシステムに切り替えることが可能です。

エージェントを無効にするには

- 1 次のコマンドを使って、サービスグループがオンラインかどうかを確認します。

```
# hagr -state service_group -sys system_name
```

- 2 サービスグループがオンラインの場合は、次のとおり入力してオフラインにします。

```
# hagr -switch service_group -to system_name
```

または

```
# hagr -offline service_group -sys system_name
```

- 3 システム上でエージェントを停止するには、次のように入力します。

```
# haagent -stop service_group -sys system_name
```

[ログファイルの中からメッセージを検索してください。(Please look for messages in the log file)] というメッセージが表示された場合は、`/var/VRTSvcs/log/engine_A.log` ファイルを調べ、エージェントが停止したことを示すメッセージがあるかどうかを確認してください。

また、`ps` コマンドを使って、エージェントが停止していることを確認することもできます。

エージェントが停止したら、システム、サービスグループまたはリソースタイプを VCS 設定から削除できます。詳しくは『Veritas Cluster Server ユーザーズガイド』でコマンドラインからの VCS の再設定に関する章を参照してください。

エージェントの削除

エージェントを削除する前に、無効にする必要があります。エージェントを削除するには、次の指示を実行します。

AIX システムからエージェントを削除するには

- ◆ エージェントがある各システムで、次のように入力します。
installp -u VRTSvcsdb.rte

Linux システムからエージェントを削除するには

- ◆ エージェントがある各システムで、次のように入力します。
rpm -e VRTSvcsdb

Solaris システムからエージェントを削除するには

- ◆ エージェントがある各システムで、次のように入力します。
pkgrm VRTSvcsdb

エージェントのアップグレード

手動でアップグレードできるのは、DB2 用の HA エージェントのみです。installvcs プログラムでは、VRTSvcsdb パッケージは自動的にアップグレードされません。

以前の設定情報を保存しないでアップグレードするには、エージェントを無効にし、削除してから再インストールします。

アップグレードして以前の設定情報を再利用する手順を次に示します。

DB2 エージェント 4.0 または 4.1 からアップグレードするには

- 1 エージェントを無効にして削除します。
 - 59 ページの「[エージェントの無効化](#)」を参照してください。
 - 60 ページの「[エージェントの削除](#)」を参照してください。
- 2 DB2 用の HA エージェントが収録されているディスクから、新しいパッケージを追加します。
33 ページの「[DB2 UDB エージェントソフトウェアのインストール](#)」を参照してください。

- 3 新しい Db2udbTypes.cf を /etc/VRTSagents/ha/conf ディレクトリから /etc/VRTSagents/ha/conf/config ディレクトリにコピーします。
- 4 main.cf ファイルの include 文の Db2udbTypes.cf ファイルの場所を更新します。たとえば、次のようにこの文を変更します。

```
include "Db2udbTypes.cf"
```

読み込むには、次のように指定します。

```
include "/etc/VRTSagents/ha/conf/config/DB2udbTypes.cf"
```
- 5 詳細監視の使用を続行するには、カスタム監視スクリプトの例を使います。
11 ページの「[詳細監視中の DB2 エラーコードの処理](#)」を参照してください。

62 | エージェントの無効化、削除、アップグレード
| エージェントのアップグレード

設定ファイルの例

この章では、VCS 設定ファイルの DB2 UDB 設定の例について説明します。

AIX 設定ファイルの例

非 MPP の main.cf 設定 : DB2 UDB ESE シングルパーティションインスタンス

次の設定は、ESE 単一パーティションインスタンス環境での、DB2 UDB における 2 インスタンスの設定例を示しています。

```
include "types.cf"
include "/etc/VRTSagents/ha/conf/Db2udb/Db2udbTypes.cf"

cluster db2_clus (
    UserNames = { admin = "cDRpdxPmHpzS." }
    Administrators = { admin }
    CounterInterval = 5
)

system sysA (
    CPUUsageMonitoring = { Enabled = 0, ActionThreshold = 0,
        ActionTimeLimit = 0, Action = NONE,
        NotifyThreshold = 0, NotifyTimeLimit = 0 }
)

system sysB (
    CPUUsageMonitoring = { Enabled = 0, ActionThreshold = 0,
        ActionTimeLimit = 0, Action = NONE,
        NotifyThreshold = 0, NotifyTimeLimit = 0 }
)
```

```
group db2udb_grp1 (
  SystemList = { sysA = 0, sysB = 1 }
  AutoStartList = { sysA }
)
Db2udb db2udb1 (
  DB2InstOwner = db2inst1
  DB2InstHome = "/db2inst1"
  InDepthMonitor = 1
  DatabaseName = SAMPLE
)

DiskGroup db2udb_dg1 (
  DiskGroup = db2_dg1
)

IP db2udb_ip1 (
  Device = en0
  Address = "166.98.9.163"
  NetMask = "255.255.220.0"
)

Mount db2udb_mnt1 (
  MountPoint = "/db2inst1"
  BlockDevice = "/dev/vx/dsk/db2_dg1/inst1_vol1"
  FSType = vxfs
  MountOpt = rw
)

NIC db2udb_nic1 (
  Device = en0
  NetworkHosts = { "166.98.128.180" }
)

Volume db2udb_vol1 (
  Volume = inst1_vol
  DiskGroup = db2_dg1
)

db2udb1 requires db2udb_ip1
db2udb1 requires db2udb_mnt1
db2udb_ip1 requires db2udb_nic1
db2udb_mnt1 requires db2udb_vol1
db2udb_vol1 requires db2udb_dg1
```



```

// resource dependency tree
//
//      group db2udb_grp1
//      {
//      Db2udb db2udb1
//      {
//      IP db2udb_ip1
//      {
//      NIC db2udb_nic1
//      }
//      Mount db2udb_mnt1
//      {
//      Volume db2udb_vol1
//      {
//      DiskGroup db2udb_dg1
//      }
//      }
//      }
//      }
//      }

group db2udb_grp2 (
    SystemList = { sysA = 0, sysB = 1 }
    AutoStartList = { sysA }
)

Db2udb db2udb2 (
    DB2InstOwner = db2inst2
    DB2InstHome = "/db2inst2"
    IndepthMonitor = 1
    DatabaseName = MYDB
)

IP db2udb_ip2 (
    Device = en0
    Address = "166.98.9.168"
    NetMask = "255.255.220.0"
)

LVMVG db2udb_vg00 (
    VolumeGroup = vg00
    MajorNumber = 50
    Disks = { hdisk2, hdisk3 }
    VaryonvgOpt = p
)

Mount db2udb_mnt2 (
    MountPoint = "/db2inst2"
    BlockDevice = "/dev/vg00"
    FSType = jfs
    MountOpt = rw
)

```

```
NIC db2udb_nic2 (
    Device = en0
    NetworkHosts = { "166.98.128.181" }
)

db2udb2 requires db2udb_ip2
db2udb2 requires db2udb_mnt2
db2udb_ip2 requires db2udb_nic2
db2udb_mnt2 requires db2udb_vg00

// resource dependency tree
//
//          group db2udb_grp2
//          {
//          Db2udb db2udb2
//          {
//          IP db2udb_ip2
//          {
//          NIC db2udb_nic2
//          }
//          Mount db2udb_mnt2
//          {
//          LVMVG db2udb_vg00
//          }
//          }
//          }
//          }
```

非 MPP の main.cf 設定 : DB2 UDB ESE マルチパーティションインスタンス

次の main.cf 設定ファイルは、ESE マルチパーティションインスタンス SMP 環境での DB2 UDB を示します。2つのデータベースパーティションが設定されています。

```
include "types.cf"
include "/etc/VRTSagents/ha/conf/Db2udb/Db2udbTypes.cf"

cluster db2_clus (
    UserNames = { admin = "cDRpdxPmHpzS." }
    Administrators = { admin }
    CounterInterval = 5
)

system sysA (
    CPUUsageMonitoring = { Enabled = 0, ActionThreshold = 0,
    ActionTimeLimit = 0, Action = NONE,
    NotifyThreshold = 0, NotifyTimeLimit = 0 }
)

system sysB (
```

```

CPUUsageMonitoring = { Enabled = 0, ActionThreshold = 0,
ActionTimeLimit = 0, Action = NONE,
NotifyThreshold = 0, NotifyTimeLimit = 0 }
)
group db2_grp1 (
SystemList = { sysA = 0, sysB = 1 }
AutoStartList = { sysA }
)

Db2udb db2udb1 (
DB2InstOwner = db2inst1
DB2InstHome = "/db2_mnt/db2inst1"
IndepthMonitor = 1
DatabaseName = DWCNTL
NodeNumber = 0
)

Db2udb db2udb2 (
DB2InstOwner = db2inst1
DB2InstHome = "/db2_mnt/db2inst1"
IndepthMonitor = 0
NodeNumber = 1
)

DiskGroup db2dg1 (
DiskGroup = db2dg1
)

IP db2ip1 (
Device = en0
Address = "166.98.9.188"
NetMask = "255.255.220.0"
)

Mount db2mnt1 (
MountPoint = "/db2_mnt/db2inst1"
BlockDevice = "/dev/vx/dsk/db2dg1/db2dg1home"
FSType = vxfs
MountOpt = rw
)

NIC db2nic1 (
Device = en0
NetworkHosts = { "166.98.128.180" }
)

Volume db2vol1 (
Volume = db2dg1home
DiskGroup = db2dg1
)

db2ip1 requires db2nic1

```

```
db2mnt1 requires db2vol1
db2udb1 requires db2ipl
db2udb1 requires db2mnt1
db2vol1 requires db2dgl
db2udb2 requires db2ipl
db2udb2 requires db2mnt1
```

MPP の main.cf 設定 : DB2 UDB ESE マルチパーティションインスタンス

次の設定ファイルは、ESE マルチパーティションインスタンス MPP 環境での DB2 UDB を示します。4 つのデータベースパーティションが設定されています。各クラスタノードで 1 つのパーティションが設定されています。各データベース サービスグループは、クラスタの共有ストレージを管理する CVM サービスグループに依存しています。

```
include "types.cf"
include "CFSTypes.cf"
include "CVMTTypes.cf"
include "/etc/VRTSagents/ha/conf/Db2udb/Db2udbTypes.cf"

cluster db2_aix_mpp (
    UserNames = { admin = gpqIpkPmqLqqOyqKpn }
    Administrators = { admin }
    HacliUserLevel = COMMANDROOT
    CounterInterval = 5
)

system sysA (
)

system sysB (
)

group cvm (
    SystemList = { sysA = 0, sysB = 1, sysC = 2, sysD = 3 }
    AutoFailOver = 0
    Parallel = 1
    AutoStartList = { sysA, sysB, sysC, sysD }
)

CFSMount db2cfsmnt (
    MountPoint = "/db2_mnt/db2inst1"
    BlockDevice = "/dev/vx/dsk/cdb2dgl1/cdb2dgl1home"
    MountOpt = "cluster"
    NodeList = { sysA, sysB, sysC, sysD }
)

CFSfsckd vxfsckd (
)
```

```

CVMCluster cvm_clus (
    CVMClustName = db2_aix_mpp
    CVMNodeId = { sysA = 0, sysB = 1, sysC = 2, sysD = 3 }
    CVMTransport = gab
    CVMTimeout = 200
)

CVMVolDg db2dg (
    CVMDiskGroup = cdb2dg1
    CVMActivation = sw
)

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

cvm_clus requires cvm_vxconfigd
db2cfsmnt requires db2dg
db2cfsmnt requires vxfsckd
db2dg requires cvm_clus
vxfsckd requires cvm_clus

// resource dependency tree
//
//     group cvm
//     {
//     CFSMount db2cfsmnt
//     {
//     CVMVolDg db2dg
//     {
//     CVMCluster cvm_clus
//     {
//     CVMVxconfigd cvm_vxconfigd
//     }
//     }
//     CFSfsckd vxfsckd
//     {
//     CVMCluster cvm_clus
//     {
//     CVMVxconfigd cvm_vxconfigd
//     }
//     }
//     }
//     }
//     }

group db2mpp_grp0 (
    SystemList = { sysA = 0, sysB = 1, sysC = 2, sysD = 3 }
    AutoStartList = { sysA }
)

```

```
Db2udb db2udb0 (  
    DB2InstOwner = db2inst1  
    DB2InstHome = "/db2_mnt/db2inst1"  
    IndepthMonitor = 1  
    DatabaseName = sample  
)  
  
IP Db2_IP0 (  
    Device = en0  
    Address = "11.192.10.32"  
    NetMask = "255.255.244.0"  
)  
  
NIC mynic0 (  
    Device = en0  
    NetworkHosts = { "11.192.11.90" }  
)  
  
requires group cvm online local firm  
Db2_IP0 requires mynic0  
db2udb0 requires Db2_IP0  
  
// resource dependency tree  
//  
//          group db2mpp_grp0  
//          {  
//          Db2udb db2udb0  
//          {  
//          IP Db2_IP0  
//          {  
//          NIC mynic0  
//          }  
//          }  
//          }  
//          }  
  
group db2mpp_grp1 (  
    SystemList = { sysA = 0, sysB = 1, sysC = 2, sysD = 3 }  
    AutoStartList = { sysB }  
)  
  
Db2udb db2udb1 (  
    DB2InstOwner = db2inst1  
    DB2InstHome = "/db2_mnt/db2inst1"  
    IndepthMonitor = 1  
    DatabaseName = test1  
    NodeNumber = 1  
)  
  
IP Db2_IP1 (  
    Device = en0  
    Address = "11.192.10.33"  
    NetMask = "255.255.244.0"  
)
```

```

NIC mynic1 (
    Device = en0
    NetworkHosts = { "11.192.11.90" }
)

requires group cvm online local firm
Db2_IP1 requires mynic1
db2udb1 requires Db2_IP1

// resource dependency tree
//
//          group db2mpp_grp1
//          {
//          Db2udb db2udb1
//          {
//          IP Db2_IP1
//          {
//          NIC mynic1
//          }
//          }
//          }
//          }

group db2mpp_grp2 (
    SystemList = { sysA = 0, sysB = 1, sysC = 2, sysD = 3 }
    AutoStartList = { sysC }
)

Db2udb db2udb2 (
    DB2InstOwner = db2inst1
    DB2InstHome = "/db2_mnt/db2inst1"
    IndepthMonitor = 1
    DatabaseName = test2
    NodeNumber = 2
)

IP Db2_IP2 (
    Device = en0
    Address = "11.192.10.33"
    NetMask = "255.255.244.0"
)

NIC mynic2 (
    Device = en0
    NetworkHosts = { "11.192.11.90" }
)

requires group cvm online local firm
Db2_IP2 requires mynic2
db2udb2 requires Db2_IP2

```

```

// resource dependency tree
//
//      group db2mpp_grp2
//      {
//      Db2udb db2udb2
//      {
//      IP Db2_IP2
//      {
//      NIC mynic2
//      }
//      }
//      }
//      }

group db2mpp_grp3 (
  SystemList = { sysA = 0, sysB = 1, sysC = 2, sysD = 3 }
  AutoStartList = { sysD }
)

Db2udb db2udb3 (
  DB2InstOwner = db2inst1
  DB2InstHome = "/db2_mnt/db2inst1"
  NodeNumber = 3
)

IP Db2_IP3 (
  Device = en0
  Address = "11.192.10.33"
  NetMask = "255.255.244.0"
)

NIC mynic3 (
  Device = en0
  NetworkHosts = { "11.192.11.90" }
)

requires group cvm online local firm
Db2_IP3 requires mynic3
db2udb3 requires Db2_IP3

// resource dependency tree
//
//      group db2mpp_grp3
//      {
//      Db2udb db2udb3
//      {
//      IP Db2_IP3
//      {
//      NIC mynic3
//      }
//      }
//      }
//      }

```


Linux 設定ファイルの例

非 MPP の main.cf 設定 : DB2 UDB ESE シングルパーティションインスタンス

次の設定は、ESE 単一パーティションインスタンス環境での、DB2 UDB における 2 インスタンスの設定例を示しています。

```
include "types.cf"
include "/etc/VRTSagents/ha/conf/Db2udb/Db2udbTypes.cf"

cluster vcs (
    CounterInterval = 5
)

system vcstc1 (
    CPUUsageMonitoring = { Enabled = 0, ActionThreshold = 0,
        ActionTimeLimit = 0, Action = NONE, NotifyThreshold = 0,
        NotifyTimeLimit = 0 }
)

system vcstc2 (
    CPUUsageMonitoring = { Enabled = 0, ActionThreshold = 0,
        ActionTimeLimit = 0, Action = NONE, NotifyThreshold = 0,
        NotifyTimeLimit = 0 }
)

group db2udb_grp1 (
    SystemList = { vcstc1 = 0, vcstc2 = 1 }
    AutoStartList = { vcstc1 }
)

Db2udb db2udb1 (
    DB2InstOwner = db2inst1
    DB2InstHome = "/db2inst1"
    IndepthMonitor = 1
    DatabaseName = SAMPLE
)

DiskGroup db2udb_dg1 (
    DiskGroup = db2_dg1
)

IP db2udb_ip1 (
    Device = eth0
    Address = "166.98.9.163"
    NetMask = "255.255.252.0"
)

Mount db2udb_mnt1 (
```

```
MountPoint = "/db2inst1"
BlockDevice = "/dev/vx/dsk/db2_dg1/inst1_vol"
FSType = vxfs
MountOpt = rw
FsckOpt = "-n"
)

NIC db2udb_nic1 (
    Device = eth0
)

Volume db2udb_vol1 (
    Volume = inst1_vol
    DiskGroup = db2_dg1
)

db2udb1 requires db2udb_ip1
db2udb1 requires db2udb_mnt1
db2udb_ip1 requires db2udb_nic1
db2udb_mnt1 requires db2udb_vol1
db2udb_vol1 requires db2udb_dg1

// resource dependency tree
//
//          group db2udb_grp1
//          {
//          Db2udb db2udb1
//          {
//          IP db2udb_ip1
//          {
//          NIC db2udb_nic1
//          }
//          Mount db2udb_mnt1
//          {
//          Volume db2udb_vol1
//          {
//          DiskGroup db2udb_dg1
//          }
//          }
//          }
//          }
//          }

group db2udb_grp3 (
    SystemList = { vcstc1 = 0, vcstc2 = 1 }
    AutoStartList = { vcstc1 }
)

Db2udb db2udb3 (
    DB2InstOwner = db2inst3
    DB2InstHome = "/db2inst3"
    IndepthMonitor = 1
)
```

```

        DatabaseName = MYDB
    )

DiskGroup db2udb_dg3 (
    DiskGroup = db2_dg3
)

IP db2udb_ip3 (
    Device = eth0
    Address = "192.2.40.21"
    NetMask = "255.255.252.0"
)

Mount db2udb_mnt3 (
    MountPoint = "/db2inst3"
    BlockDevice = "/dev/vx/dsk/db2_dg3/inst3_vol1"
    FSType = vxfs
    MountOpt = rw
    FsckOpt = "-n"
)

NIC db2udb_nic3 (
    Device = eth0
)

Volume db2udb_vol3 (
    Volume = inst3_vol
    DiskGroup = db2_dg3
)

db2udb3 requires db2udb_ip3
db2udb3 requires db2udb_mnt3
db2udb_ip3 requires db2udb_nic3
db2udb_mnt3 requires db2udb_vol3
db2udb_vol3 requires db2udb_dg3

// resource dependency tree
//
//         group db2udb_grp3
//         {
//         Db2udb db2udb3
//         {
//             IP db2udb_ip3
//             {
//                 NIC db2udb_nic3
//             }
//         }
//         Mount db2udb_mnt3
//         {
//             Volume db2udb_vol3
//             {
//                 DiskGroup db2udb_dg3
//             }
//         }
//         }
//         }

```

```
//          }  
//          }  
//      }
```

非 MPP の main.cf 設定 : DB2 UDB ESE マルチパーティションインスタンス

次の main.cf 設定ファイルは、ESE マルチパーティションインスタンス SMP 環境での DB2 UDB を示します。2つのデータベースパーティションが設定されています。

```
include "types.cf"  
include "/etc/VRTSagents/ha/conf/Db2udb/Db2udbTypes.cf"  
  
cluster vcs (  
    CounterInterval = 5  
)  
  
system vcstc1 (  
    CPUUsageMonitoring = { Enabled = 0, ActionThreshold = 0,  
        ActionTimeLimit = 0, Action = NONE, NotifyThreshold = 0,  
        NotifyTimeLimit = 0 }  
)  
  
system vcstc2 (  
    CPUUsageMonitoring = { Enabled = 0, ActionThreshold = 0,  
        ActionTimeLimit = 0, Action = NONE, NotifyThreshold = 0,  
        NotifyTimeLimit = 0 }  
)  
  
group db2udb_grp1 (  
    SystemList = { vcstc1 = 0, vcstc2 = 1 }  
    AutoStartList = { vcstc1 }  
)  
  
    Db2udb db2udb1 (  
        DB2InstOwner = db2inst1  
        DB2InstHome = "/db2inst1"  
        IndepthMonitor = 1  
        DatabaseName = SAMPLE  
        NodeNumber = 0  
    )  
  
    Db2udb db2udb2 (  
        DB2InstOwner = db2inst1  
        DB2InstHome = "/db2inst1"  
        IndepthMonitor = 0  
        NodeNumber = 1  
    )
```

```

DiskGroup db2udb_dg1 (
    DiskGroup = db2_dg1
)

IP db2udb_ip1 (
    Device = eth0
    Address = "166.98.9.163"
    NetMask = "255.255.252.0"
)

IP db2udb_ip2 (
    Device = eth0
    Address = "166.98.9.165"
    NetMask = "255.255.252.0"
)

Mount db2udb_mnt1 (
    MountPoint = "/db2inst1"
    BlockDevice = "/dev/vx/dsk/db2_dg1/inst1_vol1"
    FSType = vxfs
    MountOpt = rw
    FsckOpt = "-n"
)

NIC db2udb_nic1 (
    Device = eth0
)

Volume db2udb_vol1 (
    Volume = inst1_vol
    DiskGroup = db2_dg1
)

db2udb1 requires db2udb_ip1
db2udb1 requires db2udb_mnt1
db2udb2 requires db2udb_ip2
db2udb2 requires db2udb_mnt1
db2udb_ip1 requires db2udb_nic1
db2udb_ip2 requires db2udb_nic1
db2udb_mnt1 requires db2udb_vol1
db2udb_vol1 requires db2udb_dg1

// resource dependency tree
//
//         group db2udb_grp1
//         {
//         Db2udb db2udb1
//         {
//             IP db2udb_ip1
//             {
//                 NIC db2udb_nic1
//             }
//         }
//     }

```

```
//          Mount db2udb_mnt1
//          {
//              Volume db2udb_vol1
//              {
//                  DiskGroup db2udb_dg1
//              }
//          }
//      }
//      Db2udb db2udb2
//      {
//          IP db2udb_ip2
//          {
//              NIC db2udb_nic1
//          }
//          Mount db2udb_mnt1
//          {
//              Volume db2udb_vol1
//              {
//                  DiskGroup db2udb_dg1
//              }
//          }
//      }
//  }
```

MPP の main.cf 設定 : DB2 UDB ESE マルチパーティションインスタンス

次の設定ファイルは、ESE マルチパーティションインスタンス MPP 環境での DB2 UDB を示します。4つのデータベースパーティションが設定されています。各クラスタノードで1つのパーティションが設定されています。各データベースサービスグループは、クラスタの共有ストレージを管理する CVM サービスグループに依存しています。

```
include "types.cf"
include "CFSTypes.cf"
include "CVMTypes.cf"
include "/etc/VRTSagents/ha/conf/Db2udb/Db2udbTypes.cf"

cluster db2_mpp (
    CounterInterval = 5
)

system vcstc1 (
)

system vcstc2 (
)

system vcstc3 (
)
```

```

system vcstc4 (
)

group cvm (
  SystemList = { vcstc1 = 0, vcstc2 = 1, vcstc3 = 2, vcstc4 = 3 }
  AutoFailOver = 0
  Parallel = 1
  AutoStartList = { vcstc1, vcstc2, vcstc3, vcstc4 }
)

CFSSMount db2cfsmnt (
  MountPoint = "/db2_mnt/db2inst1"
  BlockDevice = "/dev/vx/dsk/cdb2dg1/cdb2dg1home"
)

CFSfsckd vxfsckd (
)

CVMCluster cvm_clus (
  CVMClustName = db2_mpp
  CVMNodeId = { vcstc1 = 0, vcstc2 = 1, vcstc3 = 2, vcstc4 = 3 }
)
  CVMTransport = gab
  CVMTimeout = 200
)

CVMVolDg db2dg (
  CVMDiskGroup = cdb2dg1
  CVMVolume = { cdb2dg1home }
  CVMActivation = sw
)

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

cvm_clus requires cvm_vxconfigd
db2cfsmnt requires db2dg
db2cfsmnt requires vxfsckd
db2dg requires cvm_clus
vxfsckd requires cvm_clus

// resource dependency tree
//
//      group cvm
//      {
//      CFSSMount db2cfsmnt
//      {
//      CVMVolDg db2dg
//      {

```

```

//                                CVMCluster cvm_clus
//                                {
//                                CVMVxconfigd cvm_vxconfigd
//                                }
//                                }
//                                CFSfsckd vxfscd
//                                {
//                                CVMCluster cvm_clus
//                                {
//                                CVMVxconfigd cvm_vxconfigd
//                                }
//                                }
//                                }
//                                }
//                                }

group db2mpp_grp0 (
    SystemList = { vcstc1 = 0, vcstc2 = 1, vcstc3 = 2, vcstc4 = 3 }
    AutoStartList = { vcstc1 }
    AutoStart = 1
)

Db2udb db2udb0 (
    DB2InstOwner = db2inst1
    DB2InstHome = "/db2_mnt/db2inst1"
    IndepthMonitor = 1
    DatabaseName = SAMPLE
    NodeNumber = 0
)

IP Db2_IP0 (
    Device = eth0
    Address = "10.118.2.144"
    NetMask = "255.255.248.0"
)

NIC Db2_NIC0 (
    Device = eth0
)

requires group cvm online local firm
Db2_IP0 requires Db2_NIC0
db2udb0 requires Db2_IP0

// resource dependency tree
//
//     group db2mpp_grp0
//     {
//     Db2udb db2udb0
//     {
//         IP Db2_IP0
//         {
//             NIC Db2_NIC0

```



```

//          }
//          }
//      }

group db2mpp_grp1 (
    SystemList = { vcstc1 = 0, vcstc2 = 1, vcstc3 = 2, vcstc4 = 3 }
    AutoStartList = { vcstc2 }
    AutoStart = 1
)
Db2udb db2udb1 (
    DB2InstOwner = db2inst1
    DB2InstHome = "/db2_mnt/db2inst1"
    IndepthMonitor = 1
    DatabaseName = TEST1
    NodeNumber = 1
)

IP Db2_IP1 (
    Device = eth0
    Address = "10.118.2.145"
    NetMask = "255.255.248.0"
)

NIC Db2_NIC1 (
    Device = eth0
)

requires group cvm online local firm
Db2_IP1 requires Db2_NIC1
db2udb1 requires Db2_IP1

// resource dependency tree
//
//      group db2mpp_grp1
//      {
//          Db2udb db2udb1
//          {
//              IP Db2_IP1
//              {
//                  NIC Db2_NIC1
//              }
//          }
//      }
//
//      }

group db2mpp_grp2 (
    SystemList = { vcstc1 = 0, vcstc2 = 1, vcstc3 = 2, vcstc4 = 3 }
    AutoStartList = { vcstc3 }
    AutoStart = 1
)

Db2udb db2udb2 (
    DB2InstOwner = db2inst1

```

```
DB2InstHome = "/db2_mnt/db2inst1"
InDepthMonitor = 1
DatabaseName = TEST2
NodeNumber = 2
)

IP Db2_IP2 (
  Device = eth0
  Address = "10.118.2.146"
  NetMask = "255.255.248.0"
)

NIC Db2_NIC2 (
  Device = eth0
)

requires group cvm online local firm
Db2_IP2 requires Db2_NIC2
db2udb2 requires Db2_IP2

// resource dependency tree
//
//      group db2mpp_grp2
//      {
//      Db2udb db2udb2
//          {
//          IP Db2_IP2
//              {
//              NIC Db2_NIC2
//              }
//          }
//      }
//
//      }

group db2mpp_grp3 (
  SystemList = { vcstc1 = 0, vcstc2 = 1, vcstc3 = 2, vcstc4 = 3 }
  AutoStartList = { vcstc4 }
  AutoStart = 1
)

Db2udb db2udb3 (
  DB2InstOwner = db2inst1
  DB2InstHome = "/db2_mnt/db2inst1"
  NodeNumber = 3
)

IP Db2_IP3 (
  Device = eth0
  Address = "10.118.2.147"
  NetMask = "255.255.248.0"
)
```

```

NIC Db2_NIC3 (
    Device = eth0
)

requires group cvm online local firm
Db2_IP3 requires Db2_NIC3
db2udb3 requires Db2_IP3

// resource dependency tree
//
//     group db2mpp_grp3
//     {
//     Db2udb db2udb3
//         {
//             IP Db2_IP3
//                 {
//                     NIC Db2_NIC3
//                 }
//             }
//         }
//     }

```

Solaris 設定ファイルの例

非 MPP の main.cf 設定 : DB2 UDB ESE シングルパーティションインスタンス

次の設定は、ESE 単一パーティションインスタンス環境での、DB2 UDB における 2 インスタンスの設定例を示しています。

```

include "types.cf"
include "/etc/VRTSagents/ha/conf/Db2udb/Db2udbTypes.cf"

cluster db2_clus (
    UserNames = { admin = "cDRpdxPmHpzS." }
    Administrators = { admin }
    CounterInterval = 5
)

system sysA (
    CPUUsageMonitoring = { Enabled = 0, ActionThreshold = 0,
        ActionTimeLimit = 0, Action = NONE, NotifyThreshold = 0,
        NotifyTimeLimit = 0 }
)

system sysB (
    CPUUsageMonitoring = { Enabled = 0, ActionThreshold = 0,

```

```
ActionTimeLimit = 0, Action = NONE, NotifyThreshold = 0,
NotifyTimeLimit = 0 }
)

group db2udb_grp1 (
  SystemList = { sysA = 0, sysB = 1 }
  AutoStartList = { sysA }
)

Db2udb db2udb1 (
  DB2InstOwner = db2inst1
  DB2InstHome = "/db2inst1"
  IndepthMonitor = 1
  DatabaseName = SAMPLE
  StartUpOpt = ACTIVATEDDB
)

DiskGroup db2udb_dg1 (
  DiskGroup = db2_dg1
)

IP db2udb_ip1 (
  Device = hme0
  Address = "166.98.9.163"
)

Mount db2udb_mnt1 (
  MountPoint = "/db2inst1"
  BlockDevice = "/dev/vx/dsk/db2_dg1/inst1_vol1"
  FSType = vxfs
  MountOpt = rw
  FsckOpt = "-y"
)

NIC db2udb_nic1 (
  Device = hme0
  NetworkType = ether
)

Volume db2udb_vol1 (
  Volume = inst1_vol1
  DiskGroup = db2_dg1
)

db2udb1 requires db2udb_ip1
db2udb1 requires db2udb_mnt1
db2udb_ip1 requires db2udb_nic1
db2udb_mnt1 requires db2udb_vol1
db2udb_vol1 requires db2udb_dg1

// resource dependency tree
//
//          group db2udb_grp1
```

```

//      {
//      Db2udb db2udb1
//      {
//      IP db2udb_ip1
//      {
//      NIC db2udb_nic1
//      }
//      Mount db2udb_mnt1
//      {
//      Volume db2udb_vol1
//      {
//      DiskGroup db2udb_dg1
//      }
//      }
//      }
//      }

group db2udb_grp2 (
  SystemList = { sysA = 0, sysB = 1 }
  AutoStartList = { sysA }
)

Db2udb db2udb2 (
  DB2InstOwner = db2inst2
  DB2InstHome = "/db2inst2"
  InDepthMonitor = 1
  DatabaseName = MYDB
)

DiskGroup db2udb_dg2 (
  DiskGroup = db2_dg2
)

IP db2udb_ip2 (
  Device = hme0
  Address = "192.2.40.21"
)

Mount db2udb_mnt2 (
  MountPoint = "/db2inst2"
  BlockDevice = "/dev/vx/dsk/db2_dg2/inst2_vol1"
  FSType = vxfs
  MountOpt = rw
  FsckOpt = "-y"
)

NIC db2udb_nic2 (
  Device = hme0
  NetworkType = ether
)

```

```
Volume db2udb_vol2 (
    Volume = inst2_vol
    DiskGroup = db2_dg2
)

db2udb2 requires db2udb_ip2
db2udb2 requires db2udb_mnt2
db2udb_ip2 requires db2udb_nic2
db2udb_mnt2 requires db2udb_vol2
db2udb_vol2 requires db2udb_dg2

// resource dependency tree
//
//          group db2udb_grp2
//          {
//          Db2udb db2udb2
//          {
//          IP db2udb_ip2
//          {
//          NIC db2udb_nic2
//          }
//          Mount db2udb_mnt2
//          {
//          Volume db2udb_vol2
//          {
//          DiskGroup db2udb_dg2
//          }
//          }
//          }
//          }
//          }
```

非 MPP の main.cf 設定 : DB2 UDB ESE マルチパーティションインスタンス

次の main.cf 設定ファイルは、ESE マルチパーティションインスタンス SMP 環境での DB2 UDB を示します。2つのデータベースパーティションが設定されています。

```
include "types.cf"
include "/etc/VRTSagents/ha/conf/Db2udb/Db2udbTypes.cf"

cluster db2_clus (
    UserNames = { admin = "cDRpdxPmHpzS." }
    Administrators = { admin }
    CounterInterval = 5
)

system sysA (
    CPUUsageMonitoring = { Enabled = 0, ActionThreshold = 0,
        ActionTimeLimit = 0, Action = NONE,
        NotifyThreshold = 0, NotifyTimeLimit = 0 }
```

```
)

system sysB (
    CPUUsageMonitoring = { Enabled = 0, ActionThreshold = 0,
        ActionTimeLimit = 0, Action = NONE,
        NotifyThreshold = 0, NotifyTimeLimit = 0 }
)

group db2_grp1 (
    SystemList = { sysA = 0, sysB = 1 }
    AutoStartList = { sysA }
)

Db2udb db2udb1 (
    DB2InstOwner = db2inst1
    DB2InstHome = "/db2_mnt/db2inst1"
    IndepthMonitor = 1
    DatabaseName = DWCNTRL
    NodeNumber = 0
)

Db2udb db2udb2 (
    DB2InstOwner = db2inst1
    DB2InstHome = "/db2_mnt/db2inst1"
    IndepthMonitor = 0
    NodeNumber = 1
)

DiskGroup db2dgl (
    DiskGroup = db2dgl
)

IP db2ip1 (
    Device = hme0
    Address = "192.2.40.21"
)

Mount db2mnt1 (
    MountPoint = "/db2_mnt/db2inst1"
    BlockDevice = "/dev/vx/dsk/db2dgl1/db2dgl1home"
    FSType = vxfs
    MountOpt = rw
    FsckOpt = "-y"
)

NIC db2nic1 (
    Device = hme0
    NetworkType = ether
)

Volume db2vol1 (
    Volume = db2dgl1home
    DiskGroup = db2dgl
)
```

```
)  
  
db2ip1 requires db2nic1  
db2mnt1 requires db2vol1  
db2udb1 requires db2ip1  
db2udb1 requires db2mnt1  
db2vol1 requires db2dg1  
db2udb2 requires db2ip1  
db2udb2 requires db2mnt1
```

MPP の main.cf 設定 : DB2 UDB ESE マルチパーティションインスタンス

次の設定ファイルは、ESE マルチパーティションインスタンス MPP 環境での DB2 UDB を示します。4 つのデータベースパーティションが設定されています。各クラスターノードで 1 つのパーティションが設定されています。各データベースサービスグループは、クラスターの共有ストレージを管理する CVM サービスグループに依存しています。

```
include "types.cf"  
include "CFSTypes.cf"  
include "CVMTypes.cf"  
include "/etc/VRTSagents/ha/conf/Db2udb/Db2udbTypes.cf"  
  
cluster db2_mpp (  
    CounterInterval = 5  
)  
  
system sysA (  
)  
  
system sysB (  
)  
  
system sysC (  
)  
  
system sysD (  
)  
  
group cvm (  
    SystemList = { sysA = 0, sysB = 1, sysC = 2, sysD = 3 }  
    AutoFailOver = 0  
    Parallel = 1  
    AutoStartList = { sysA, sysB, sysC, sysD }  
)  
  
CFSMount db2cfsmnt (  
    MountPoint = "/db2_mnt/db2inst1"  
    BlockDevice = "/dev/vx/dsk/cdb2dg1/cdb2dg1home"  
    Primary = sysD
```



```

)

CFSfsckd vxfsckd (
)

CVMcluster cvm_clus (
  Critical = 0
  CVMClustName = db2_mpp
  CVMNodeId = { sysA = 0, sysB = 1, sysC = 2,
               sysD = 3 }
  CVMTransport = gab
  CVMTimeout = 200
)

CVMVolDg db2dg (
  CVMDiskGroup = cdb2dg1
  CVMVolume = { cdb2dg1home }
  CVMActivation = sw
)

db2cfsmnt requires db2dg
db2cfsmnt requires vxfsckd
db2dg requires cvm_clus
vxfsckd requires qlogckd

// resource dependency tree
//
//      group cvm
//      {
//      CFSMount db2cfsmnt
//      {
//      CVMVolDg db2dg
//      {
//      CVMcluster cvm_clus
//      }
//      CFSfsckd vxfsckd
//      {
//      CFSQlogckd qlogckd
//      }
//      }
//      }
//      }

group db2mpp_grp0 (
  SystemList = { sysA = 0, sysB = 1, sysC = 2, sysD = 3 }
  AutoStartList = { sysA }
)

Db2udb db2udb0 (
  DB2InstOwner = db2inst1
  DB2InstHome = "/db2_mnt/db2inst1"
  IndepthMonitor = 1
  DatabaseName = SAMPLE
)

```

```
    )

IP Db2_IP0 (
    Device = hme0
    Address = "10.118.2.144"
    NetMask = "255.255.248.0"
)

NIC Db2_NIC0 (
    Device = hme0
    NetworkHosts = { "10.118.11.90" }
)

requires group cvm online local firm
Db2_IP0 requires Db2_NIC0
db2udb0 requires Db2_IP0

// resource dependency tree
//
//      group db2mpp_grp0
//      {
//      Db2udb db2udb0
//      {
//          IP Db2_IP0
//          {
//              NIC Db2_NIC0
//          }
//      }
//      }
//      }

group db2mpp_grp1 (
    SystemList = { sysA = 0, sysB = 1, sysC = 2, sysD = 3 }
    AutoStartList = { sysB }
)

Db2udb db2udb1 (
    DB2InstOwner = db2inst1
    DB2InstHome = "/db2_mnt/db2inst1"
    IndepthMonitor = 1
    DatabaseName = TEST1
    NodeNumber = 1
)

IP Db2_IP1 (
    Device = hme0
    Address = "10.118.2.145"
    NetMask = "255.255.248.0"
)
```

```

NIC Db2_NIC1 (
    Device = hme0
    NetworkHosts = { "10.118.11.90" }
)

requires group cvm online local firm
Db2_IP1 requires Db2_NIC1
db2udb1 requires Db2_IP1

// resource dependency tree
//
//          group db2mpp_grp1
//          {
//          Db2udb db2udb1
//          {
//          IP Db2_IP1
//          {
//          NIC Db2_NIC1
//          }
//          }
//          }
//          }

group db2mpp_grp2 (
    SystemList = { sysA = 0, sysB = 1, sysC = 2, sysD = 3 }
    AutoStartList = { sysC }
)

Db2udb db2udb2 (
    DB2InstOwner = db2inst1
    DB2InstHome = "/db2_mnt/db2inst1"
    IndepthMonitor = 1
    DatabaseName = TEST2
    NodeNumber = 2
)

IP Db2_IP2 (
    Device = hme0
    Address = "10.118.2.146"
    NetMask = "255.255.248.0"
)

NIC Db2_NIC2 (
    Device = hme0
    NetworkHosts = { "10.118.11.90" }
)

requires group cvm online local firm
Db2_IP2 requires Db2_NIC2
db2udb2 requires Db2_IP2

```

```
// resource dependency tree
//
//      group db2mpp_grp2
//      {
//      Db2uadb db2uadb2
//      {
//      IP Db2_IP2
//      {
//      NIC Db2_NIC2
//      }
//      }
//      }
//      }

group db2mpp_grp3 (
  SystemList = { sysA = 0, sysB = 1, sysC = 2, sysD = 3 }
  AutoStartList = { sysD }
)

Db2uadb db2uadb3 (
  DB2InstOwner = db2inst1
  DB2InstHome = "/db2_mnt/db2inst1"
  NodeNumber = 3
)

IP Db2_IP3 (
  Device = hme0
  Address = "10.118.2.147"
  NetMask = "255.255.248.0"
)

NIC Db2_NIC3 (
  Device = hme0
  NetworkHosts = { "10.118.11.90" }
)

requires group cvm online local firm
Db2_IP3 requires Db2_NIC3
db2uadb3 requires Db2_IP3

// resource dependency tree
//
//      group db2mpp_grp3
//      {
//      Db2uadb db2uadb3
//      {
//      IP Db2_IP3
//      {
//      NIC Db2_NIC3
//      }
//      }
//      }
//      }
```

Solaris ゾーンで実行中の DB2 インスタンス

この Solaris ゾーンで稼動している DB2 インスタンスの例では、次の設定を行い、Solaris 10 ゾーン環境で稼動している DB2 UDB インスタンスを反映します。

```
include "types.cf"
include "/etc/VRTSagents/ha/conf/Db2udb/Db2udbTypes.cf"

cluster db2zone (
    UserNames = { "z_zoneres@vcs_lzs@sysA.engba.veritas.com" = Gn,
                  "z_z1@vcs_lzs@sysA.engba.veritas.com" = aH }
    ClusterAddress = "10.178.6.32"
    SecureClus = 1
    CredRenewFrequency = 0
    CounterInterval = 5
)

system sysA (
)

system sysB (
)

group Db2grp (
    SystemList = { sysA = 0, sysB = 1 }
    AutoStartList = { sysA, sysB }
    Administrators = { "z_zoneres@vcs_lzs@sysA.engba.veritas.com" }
)

DiskGroup z-dg (
    DiskGroup = db2dg1
)

IP ipres (
    Device = bge0
    Address = "10.178.6.28"
    ContainerName = "zone1"
)

Mount z-mnt (
    MountPoint = "/zones/db2data"
    BlockDevice = "/dev/vx/dsk/db2dg1/db2dg1data"
    FSType = vxfs
    FsckOpt = "-y"
)

NIC z-nic (
    Device = bge0
    NetworkType = ether
    NetworkHosts = { "10.178.2.4" }
)
```

```
Volume z-vol (  
    Volume = db2dg1data  
    DiskGroup = db2dg1  
)  
  
Zone zoneres (  
    ZoneName = zone1  
)  
  
Db2udb db2udb1 (  
    ContainerName = "zone1"  
    DB2InstOwner = "db2inst1"  
    DB2InstHome = "/db2inst1"  
)  
  
ipres requires z-nic  
ipres requires zoneres  
z-mnt requires z-vol  
z-vol requires z-dg  
zoneres requires z-nic  
zoneres requires z-mnt  
db2udb1 requires zoneres  
  
// resource dependency tree  
//  
//     group Db2grp  
//     {  
//     IP ipres  
//     {  
//         NIC z-nic  
//     }  
//     Mount z-mnt  
//     {  
//         Volume z-vol  
//         {  
//             DiskGroup z-dg  
//         }  
//     }  
//     Zone zoneres  
//     {  
//         NIC z-nic  
//     }  
// }
```

索引

記号

.rhosts、データベースアクセスのための編集 28
/etc/services、設定されたポートの確認 27
/etc/system ファイル (共有メモリパラメータ) 26

C

Cluster Manager

Db2udbTypes.cf のインポートに使用 45
DB2 サービスグループの設定 45

CVM (インフラストラクチャ) サービスグループ 42, 43

CVM サービスグループの平行モード 42, 43

D

DatabaseName 属性 38

db2gcf -d コマンド 11

db2gcf -k コマンド (クリーン処理) 13

db2gcf -u コマンド 11

DB2InstHome 属性 38

DB2InstOwner 属性 38

db2nodes.cfg、パーティションの仮想ホスト 29

db2setup ツール、バイナリ用、インスタンス 26

db2start プログラム 30

db2stop プログラム 30

DB2 UDB EE (Enterprise Edition)

設定例 83

DB2 UDB EE (Enterprise Edition)、設定例 63, 73

DB2 UDB ESE (Enterprise Server Edition)

設定ファイルの例 86, 88

DB2 UDB ESE (Enterprise Server Edition)、設定
ファイルの例 66, 68, 76, 78

Db2udbTypes.cf

main.cf ファイルに追加 52, 53

タイプ定義ファイル 41

Db2udbTypes.cf ファイルのインポート 45

Db2udbTypes.cf ファイル、インポート 45

DB2 UDB インスタンス

2 インスタンスの設定、例 63, 73, 83

共有ストレージにインストール 27

DB2 UDB エージェントの設定

Cluster Manager の使用 45

設定ファイルの編集 52

DB2 UDB エンタープライズエージェントのイン
ストール 33

DB2 UDB エンタープライズエージェント、イン
ストール 33

DB2 UDB タイプの定義ファイル 41

DB2 UDB のインストール

共有ストレージのインスタンス 27

システムバイナリ 26

設定の確認 30

必要条件 26

Db2udb リソース

IP に依存、Mount リソース 42

main.cf ファイルでの例 67, 87

Db2udb リソースタイプの属性 38

DB2 インスタンスの起動 30

DB2 インスタンスの停止 11

DB2 コマンド

create 30

list 30

DB2 サービスグループ 43

DiskGroup リソース

main.cf での例 67, 87

Volume リソースに必要 42

E

Encoding 属性 40

G

groupadd コマンド 24

H

hagrp コマンド 59

hagui コマンド 45

I

IP アドレス、仮想 29

I

IP リソース
 main.cf ファイルでの例 67, 87
 NIC リソースが必要 42

J

Java コンソール、サービスグループの設定 46, 49

L

list コマンド (DB2) 30

M

main.cf ファイル

DB2 UDB EEE の例 66, 76
 DB2 UDB ESE マルチパーティションの例 68,
 78, 88
 設定の確認 54
 編集 52
 例 52

mkfs コマンド 20, 21, 22, 23

mkgroup コマンド 24

mkuser コマンド 25

Mount リソース

main.cf ファイルでの例 67, 87
 Volume リソースが必要 42

MPP サービスグループ設定 43

MPP (超並列プロセッサ)、サポート 10

MPP マルチパーティション、監視 12

N

NIC リソース

IP リソースに必要 42
 main.cf ファイルでの例 67, 87

NodeNumber 属性 38

O

Oracle、エラー処理 11

P

pkgmgr コマンド 60

R

ResourceInfo 属性 14

rsh コマンド 28

S

SMP (対称マルチプロセッサ)、サポート 10

U

useradd コマンド 25, 26

V

VCS

再設定用コマンド 55
 サポートバージョン 10
 設定ファイルの例 52, 63

Volume リソース

DiskGroup リソースが必要 42
 main.cf ファイルでの例 64, 84

vxassist コマンド 20, 21, 22, 23

vxctl コマンド 21, 22, 23

vxvg コマンド 20, 21, 22, 23

W

WarnOnlyIfDBQueryFailed 属性 13

あ

アカウント、DB2 UDB ユーザーに必要 25
 アクションエージェント機能の処理 14

い

依存関係

DB2 と CVM のサービスグループ間 48
 main.cf での定義 54

インスタンス

起動と停止 30
 共有ディスクへのインストール 27

え

エージェント

インストール 33
 設定、概要 37

お

オフライン処理、使われるコマンド 11
 オンライン処理、使われるコマンド 11

か

仮想 IP アドレス 29

監視

ESE マルチパーティションインスタンス、
MPP 12

詳細監視の設定 56

通常と詳細 55

監視処理 11

き

共有メモリパラメータ 26

く

クリーン処理 13

グループ (ユーザー)、追加 24

こ

コマンド

groupadd 24

hagrp 59

hagui 45

mkfs 20, 21, 22, 23

mkgroup 24

mkuser 25

pkgrm 60

rsh 28

useradd 25, 26

vxassist 20, 21, 22, 23

vxctl 21, 22, 23

vxdg 20, 21, 22, 23

コマンド、DB2

create 30

list 30

さ

作成コマンド (DB2) 30

サービスグループ

Cluster Manager を使った追加 46

CVM 42, 43

DB2 43

DB2 と CVM の依存関係リンク 48

main.cf ファイルに作成 52, 53

MPP 構成、依存関係 42

ウィザードを使った追加 49

非 MPP 構成、依存関係 42

し

詳細監視

無効化 56

有効化 56

情報エージェント機能の処理 13

せ

設定ファイル

db2nodes.cfg、パーティションのホスト 29

Db2udbTypes.cf 41

main.cf、例 63, 66, 68, 73, 76, 78, 83, 86, 88

つ

通常監視、説明 55

て

テスト

サービスグループの設定 55

リモートログイン 28

テンプレート

DB2 MPP サービスグループ 47

非 MPP サービスグループに使用 49

は

バイナリ、DB2 UDB、ローカルにインストール 26

パーティション番号 (NodeNumber) 38

ふ

ファイル

.rhosts 28

/etc/system 26

db2nodes.cfg 29

Db2udbTypes.cf 41

main.cf (VCS 設定) 52, 63

ファイルシステム

MPP 構成の作成 21

非 MPP インスタンス用の作成 19

ゆ

ユーザーアカウント、追加 25

