CA Spectrum®

データベース管理ガイド

リリース 9.4

ca technologies

CA Technologies 製品リファレンス

このガイドでは、CA Spectrum® について説明します。

CA への連絡先

テクニカル サポートの詳細については、弊社テクニカル サポートの Web サイト（http://www.ca.com/jp/support/）をご覧ください。

このドキュメント（組み込みヘルプ システムおよび電子的に配布される資料を含む、以下「本ドキュメント」）は、お客様への情報提供のみを目的としたもので、日本 CA 株式会社（以下「CA」）により随時、変更または撤回されることがあります。

CA の事前の書面による承諾を受けずに本ドキュメントの全部または一部を複写、譲渡、開示、変更、複本することはできません。 本ドキュメントは、CA が知的財産権を有する機密情報です。ユーザは本ドキュメントを開示したり、（i）本ドキュメントが関係する CA ソフトウェアの使用について CA とユーザとの間で別途締結される契約または (ii) CA とユーザとの間で別途締結される機密保持契約により許可された目的以外に、本ドキュメントを使用することはできません。

上記にかかわらず、本ドキュメントで言及されている CA ソフトウェア製品のライセンスを受けたユーザは、社内でユーザおよび従業員が使用する場合に限り、当該ソフトウェアに関連する本ドキュメントのコピーを妥当な部数だけ作成できます。ただし CA のすべての著作権表示およびその説明を当該複製に添付することを条件とします。

本ドキュメントを印刷するまたはコピーを作成する上記の権利は、当該ソフトウェアのライセンスが完全に有効となっている期間内に限定されます。 いかなる理由であれ、上記のライセンスが終了した場合には、お客様は本ドキュメントの全部または一部と、それらを複製したコピーのすべてを破棄したことを、CA に文書で証明する責任を負います。

準拠法により認められる限り、CA は本ドキュメントを現状有姿のまま提供し、商品性、特定の使用目的に対する適合性、他者の権利に対して侵害のないことについて、黙示の保証も含めいかなる保証もしません。 また、本ドキュメントの使用に起因して、逸失利益、投資損失、業務の中断、営業権の喪失、情報の喪失等、いかなる損害（直接損害か間接損害かを問いません）が発生しても、CA はお客様または第三者に対し責任を負いません。CA がかかる損害の発生の可能性について事前に明示に通告されていた場合も同様とします。

本ドキュメントで参照されているすべてのソフトウェア製品の使用には、該当するライセンス契約が適用され、当該ライセンス契約はこの通知の条件によっていかなる変更も行われません。

本ドキュメントの制作者は CA です。

「制限された権利」のもとでの提供：アメリカ合衆国政府が使用、複製、開示する場合は、FAR Sections 12.212、52.227-14 及び 52.227-19(c)(1)及び(2)、ならびに DFARS Section252.227-7014(b)(3) または、これらの後継の条項に規定される該当する制限に従うものとします。

Copyright © 2014 CA. All rights reserved. 本書に記載された全ての製品名、サービス名、商号およびロゴは各社のそれぞれの商標またはサービスマークです。

目次

第 1 章: 概要 7

SpectroSERVER データベース 7
分散データ マネージャ（DDM）データベース 8

第 2 章: SpectroSERVER データベースの保守 11

データベースのバックアップ 11
オンライン バックアップ 12
手動バックアップ 18
データベースのリストア 20
データベースのロード 21
データベースの初期化とロード 22
開発者情報のロード 24
データベース ロックの削除 24
モデル タイプのインポートとエクスポート 26
データベース破損からのリカバリ 27
SpectroSERVER データベースのトラブルシューティング 27
インストール/アップグレードの後の SSdbload を使用した古いオブジェクトのロード 28
SSdbload の -i オプションの不適切な使用 28
オペレーティング システム/プラットフォーム間のデータベース ファイルのコピー 29
モデリング カタログおよびモデルのロード順序 29
同じ開発者 ID を持つ複数のデータベースへのモデリング カタログ オブジェクトの追加 29
データベース ファイルとディレクトリでの UNIX ファイル アクセス権限 30
CA Spectrum データベース ツール 30
db_remove 31
dbtool 32
MapUpdate 40
HostUpdate 41
reports 42
SSdbload 51
SSdbsave 56
データベース モデル変更ツール（DBconv） 59

第 3 章: DDM データベースの保守 71

データベース セキュリティ 71

データベース サイズの管理 72
データベース バックアップ 73
post_olb_script 74
ddm_save 75
ddm_backup 76
バックアップに関する注意事項 77
データベースのリストア 77
ddm_load 77
リストアに関する注意事項 78
データベース保守と最適化 79
db_maintenance.pl 79
db_optimize.pl 80
DDM データベース クエリ 81
その他のデータベース ユーティリティ 83
アーカイブ マネージャ リソース 83
DDM およびアーカイブ マネージャのトラブルシューティング 85
ARCHMGR.OUT のエラー メッセージ 86
イベントがリストされない 88

第 1 章: 概要

この章では、CA Spectrum データベース保守の概要、および SpectroSERVER（SS）と分散データ マネージャ（DDM）のデータベースについて説明します。

このガイドでは、複数の SpectroSERVER がある分散ネットワーク環境での CA Spectrum データベースの管理について説明します。 分散コンピューティングは業界標準ですが、このガイドの情報は非分散環境に適用できる可能性があります。 CA Spectrum での分散データベース管理には、複数のレベルの CA Spectrum データベース保守が含まれます。 さまざまなモデリング ルールおよび手順に従う必要があり、多数のアプリケーションを使用して分散 CA Spectrum ネットワークおよびその関連するデータベースを管理できます。

このガイドでは、各 SpectroSERVER の一部であるローカル データベースと複数の SpectroSERVER を表す履歴データベースの両方の保守について説明します。

このセクションには、以下のトピックが含まれています。

SpectroSERVER データベース (P. 7)

SpectroSERVER データベース

SpectroSERVER データベースは *$SPECROOT*/SS ディレクトリにあり、次の情報が含まれます。

- すべてのネットワーク情報に対する構造であるモデリング カタログ（モデル タイプおよび関係）
- この SpectroSERVER に属する、作成されたモデル
- ネットワーク イベントと統計用のストレージ領域

OneClick でネットワーク デバイスを表すアイコンは、実際のデバイスの *モデル*（ソフトウェア シミュレーション）から取得される情報を伝えています。 モデルは SpectroSERVER データベースで保持され、複数のソースからの情報で更新されます。 SpectroSERVER の 1 つの実行から次の実行に保持される情報も、SpectroSERVER データベースに格納されます。

分散データ マネージャ（DDM）データベース

各 SpectroSERVER には分散データ マネージャ（DDM）データベースがあり、複数のランドスケープで使用される CA Spectrum イベントおよび統計データが格納されます。 CA Spectrum Report Manager や CA Spectrum AlarmNotifier などのクライアント アプリケーションは、ランドスケープによってデータを要求し、受信し、照合できます。 DDM データベースは、SpectroSERVER *$SPECROOT*/mysql/data/ddmdb ディレクトリにあります。

注: 詳細については、「分散 SpectroSERVER 管理者ガイド」を参照してください。

SpectroSERVER データベースと同様に、CA 製プログラムの複数のユーザが DDM データベースに同時にアクセスすることはできません。 CA Spectrum はソフト ロック ファイル（*$SPECROOT*/SS/DDM/.DDMDB.LOCK）を適用して、複数の CA 製アプリケーションによる同時アクセスを防ぎます。 特定の状況（たとえば、異常シャットダウンからリカバリするとき）では、ソフト ロック ファイルを削除できます。

重要: CA 製アプリケーションがデータベース ロック ファイルを検出すると、エラー メッセージで警告されます。 ただし、非 CA ツールおよびアプリケーションはこのロックを確認しないため、メッセージを生成しない場合があります。 このような非 CA エンティティがロックを回避すると、データベースが破損する可能性があります。 ベスト プラクティスとしては、すべての非 CA ツールまたはアプリケーションが CA Spectrum データベースにアクセスする前に、すべての CA Spectrum プロセスがシャットダウンされていることを確認します。

アーカイブ マネージャは、DDM と SpectroSERVER データベースの間、および DDM データベースとクライアント アプリケーションの間のすべての通信を制御します。

詳細情報:

データベース ロックの削除 (P. 24)

アーカイブ マネージャ

各ランドスケープにはアーカイブ マネージャ サーバが存在し、SpectroSERVER からイベントと統計データを取得して、圧縮し、DDM データベースに格納します。 データ圧縮により、より多くのパフォーマンス データを格納でき、アプリケーションと DDM データベースの間のネットワーク トラフィックが減ります。

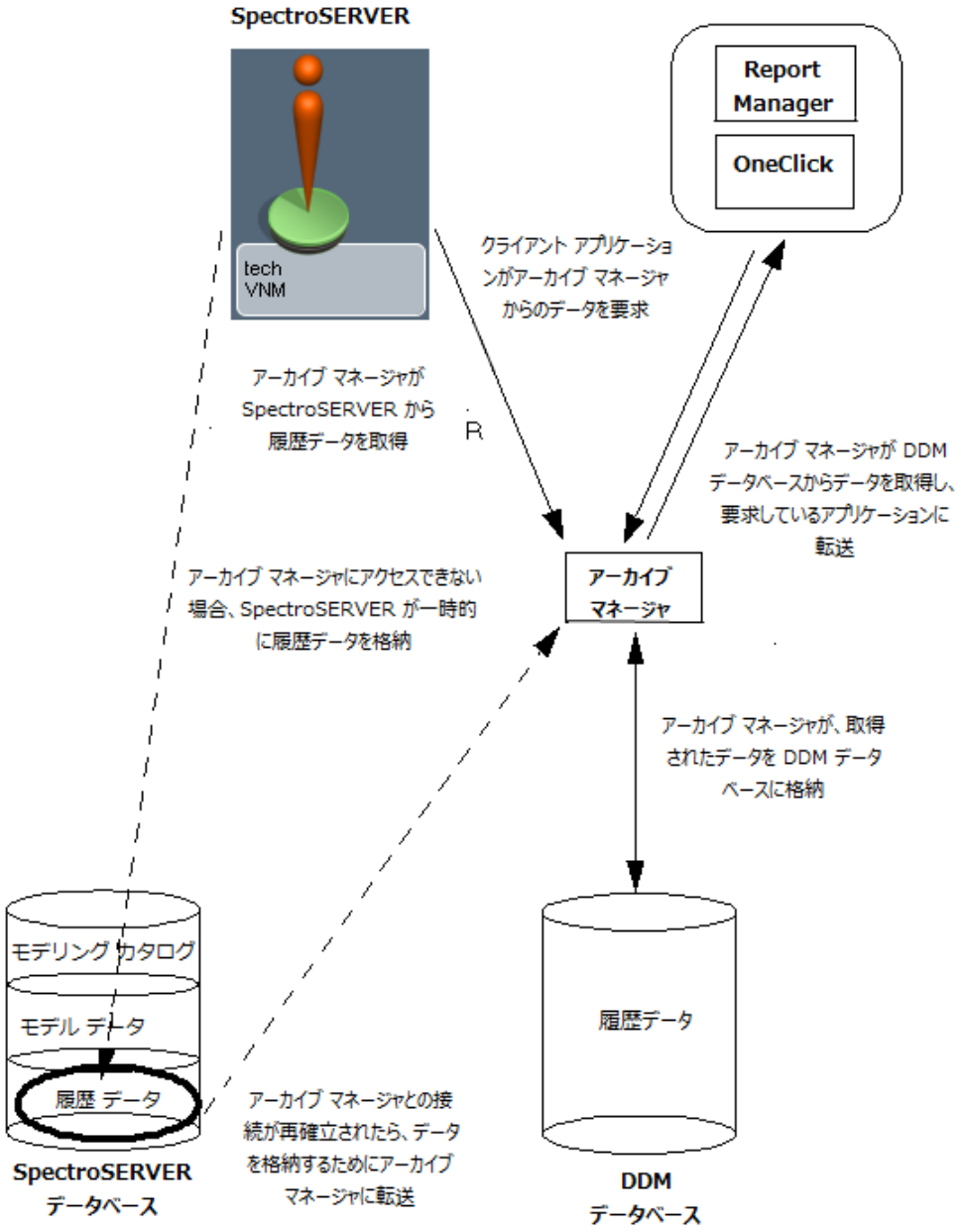
図で示すように、SpectroSERVER は、アーカイブ マネージャに接続できない場合は、接続が再確立されるまでイベントと統計データを格納します。その後、SpectroSERVER は保存用にデータをアーカイブ マネージャに送信します。 .vnmrc ファイルの Events および Statistics Archive オプションにより、SpectroSERVER が格納するデータの量が確定します。 .configrc ファイルのオプションにより、履歴データが DDM データベースに保存される期間が確定します。

注: 詳細については、「分散 SpectroSERVER 管理者ガイド」を参照してください。

また、アーカイブ マネージャは、クライアント アプリケーションからの要求に応えて以下の情報を提供することもできます。

- 情報が使用可能なランドスケープのリスト
- 各ランドスケープについて、使用可能な情報の時間範囲、および情報が使用可能なモデル タイプのリスト
- 各モデル タイプについて、情報が使用可能なモデルのリスト
- 各モデルについて、情報が使用可能な属性のリスト
- 指定された時間範囲内の統計データ
- 指定された時間範囲内のイベント データ

以下の図では、SpectroSERVER と DDM データベースの間の相互作用を示します。



第 2 章: SpectroSERVER データベースの保守

この章では、SpectroSERVER データベースのデータベース保守手順について説明します。 分散データ マネージャ（DDM）データベースの保守手順については、「DDM データベースの保守 (P. 71)」で別に説明します。

このセクションには、以下のトピックが含まれています。

データベースのバックアップ (P. 11)
データベースのリストア (P. 20)
SpectroSERVER データベースのトラブルシューティング (P. 27)
CA Spectrum データベース ツール (P. 30)

データベースのバックアップ

データベースのバックアップ コピーを規則正しく作成することは、データベース保守の基礎です。 データベースの信頼できるバックアップ コピーがあれば、停電または他のシステム中断の後で、データベースをリストアできます。

CA Spectrum には、SpectroSERVER データベースのバックアップを実行する 2 つの方法が用意されています。

- SpectroSERVER を実行しながら自動的に。
- SpectroSERVER をシャットダウンして手動で。

重要: このセクションで説明するオンライン バックアップおよび SSdbsave ユーティリティによるデータベース バックアップ方法以外では、SpectroSERVER データベースをバックアップできません。 サードパーティのバックアップ ソフトウェアを使用すると、データベースが破損する可能性があります。

詳細情報:

オンライン バックアップ (P. 12)
手動バックアップ (P. 18)

オンライン バックアップ

オンライン バックアップを使用すると、SpectroSERVER をシャットダウンすることなく、SpectroSERVER データベースのバックアップ コピーを作成できます。 要件に応じて、オンライン バックアップをオンデマンドを実行できます。または、標準バックアップを自動的に実行するようにスケジュールできます。 ディスク容量が制限されている場合は、CA Spectrum gzip ユーティリティを使用して自動的にバックアップ ファイルを圧縮するようにオンライン バックアップを設定できます。

オンライン バックアップは、モデリング カタログおよびモデルを含むデータベース全体を保存します。 ただし、［オンライン バックアップ］でも手動のバックアップ処理でも以下の情報は保存されません。

- キャッシュされたイベント情報
- アラーム
- キャッシュされた統計情報
- DDM データベースの履歴レコード（後述の注を参照）
- SpectroSERVER リソース ファイル（.vnmrc）

オンライン バックアップ アクティビティは、VNM モデルと関連付けられたイベントとして記録されます。 CA Spectrum は、バックアップ処理中に遭遇したエラーをアラームとしてレポートします。

オンライン バックアップでの保存は、2 つの主要な手順で実行されます。

1. コピーを作成して、データベース ファイルの「スナップショット」を保持します。

   この最初の手順の間、ポーリング、トラップ処理、ネットワーク管理アクティビティは一時停止されます。 処理は比較的短い時間で終わりますが、ベスト プラクティスとして、自動保存操作をスケジュールする頻度およびタイミングを検討します。 コピー操作の実行に要する時間は、ワークステーション ハードウェアおよびデータベースのサイズに依存します。

2. 手動データベース保存ユーティリティ SSdbsave と同じ形式を使用して、コピーを保存します（必要な場合は圧縮します）。

オンライン バックアップ自体では、DDM データベースは保存されません。ただし、post_olb_script を自動的に実行するように設定し、SpectroSERVER データベースのオンライン バックアップが実行されるときに常に DDM データベースを保存できます。

注: フォールト トレラント環境では、オンライン バックアップを実行する前に、プライマリとセカンダリの両方の SpectroSERVER に同じユーザとしてログオンしていることを確認します。

詳細情報:

オンライン バックアップのイベントとアラーム (P. 17)

バックアップ ファイルのメンテナンス

自動的なバックアップが有効になっている場合、バックアップ ファイルがバックアップ ディレクトリに蓄積し、使用可能なディスク領域がなくなる可能性があります。 バックアップ障害を回避するには、ファイルをときどき削除するか、より永続的なストレージ媒体にバックアップ ファイルを移動します。

オンライン バックアップの設定

オンライン バックアップを設定するときは、バックアップ間隔および最初のスケジュールされたバックアップの日時を指定できます。 さらに高度なスケジュール オプションの場合は、*$SPECROOT*/SS-Tools ディレクトリの StartOnlineBackup アプリケーションを使用してオンライン バックアップを開始します。 StartOnlineBackup アプリケーションは、ホスト システムのタスク スケジューラまたは crontab から起動できます。 StartOnlineBackup アプリケーション内の高度なスケジュール オプションでは、夏時間のためにスケジュールされたバックアップがずれる問題を回避できます。

次の手順に従ってください:

1. OneClick でユニバース トポロジ ビューを開き、VNM モデルを選択します。
2. ［情報］タブをクリックし、［オンライン データベース バックアップ］サブビューを展開します。

3. 必要に応じて以下の設定を設定します。

自動バックアップ

有効にすると、［バックアップ間隔］設定で指定されている時間間隔に従って、オンライン バックアップがオンライン バックアップ機能によって自動的に実行されます。

バックアップ間隔

自動的なバックアップの間隔を時間と分で指定します。 データベースが毎日同時刻にバックアップされるように、24 時間 0 （ゼロ）分のデフォルト間隔を使用することをお勧めします。 任意の値を入力します。 たとえば、168 時間 0 分（1 週間の間隔の場合）と 10,080 分（やはり 1 週間の間隔）は同じです。

次のバックアップの日時

次のスケジュールされたバックアップの日時が表示されます。 最初のバックアップの日時を指定できます。 ただし、それより後のバックアップは、［バックアップ間隔］設定で指定されている間隔で実行されます。

バックアップ圧縮

有効にすると、バックアップ ファイルは圧縮ユーティリティを使用して圧縮されてから、ディスクに書き込まれます。 圧縮ファイルは、ファイル名に .gz サフィックスが追加されて保存されます。 無効にすると、ファイルは圧縮さずに保存されます。 デフォルト値は「有効」です。

バックアップ ファイル名のプレフィックス

バックアップ ファイル名のユーザ定義の部分を指定します。 デフォルトのプレフィックスは "db_" です。 ただし、SpectroSERVER を実行しているシステムで有効なファイル名を作成する任意の文字列を指定できます。 設定しないと、プレフィックスはファイル名に追加されません。

ファイル名のサフィックスはバックアップが実行された日時を示し、*yyyymmdd_hhmm.SSdb* [*.gz*] という形式が使用されます。

形式	説明
yyyy	4 桁の年数
MM	月

形式	説明
dd	1 ～ 31 の日
hh	1 ～ 24 の時刻
mm	00 ～ 59 の分
gz	圧縮されたバックアップを示します（圧縮ファイルのみ）
SSdb	すべてのデータベース保存ファイルのデフォルト サフィックス

バックアップ ディレクトリ

バックアップ ファイルが格納されるディレクトリを指定します。ローカル ディレクトリを使用することをお勧めします。 デフォルトのディレクトリは、*$SPECROOT*/SS-DB-Backup です。

必要な最小ディスク容量（MB）

オンライン バックアップを開始できる最小ディスク空き容量を指定します。 デフォルトしきい値は 20 MB です。 使用可能なディスク容量がしきい値以下に低下したときに自動バックアップが開始した場合は、イエロー アラームが生成されます。 想定される原因としては、ディスク空き容量不足によりバックアップ失敗が示されます。

4. 現在の設定を使用して、必要に応じてバックアップを開始するには、［今すぐバックアップを開始］ボタンをクリックします。

   オンライン データベース バックアップ ユーティリティは、最初に SpectroSERVER の操作を一時停止します。 その後、データベースをコピーして保存します。

注: 進行中のオンライン バックアップ操作のステータスが、［今すぐバックアップを開始］ボタンの隣に表示されます。エラーが発生した場合は、イベントおよび関連するアラームが［イベント］タブおよび［アラーム］タブに表示されます。 オンライン バックアップ中に生成される可能性のあるイベントおよびアラームのリストについては、「オンライン バックアップのイベントとアラーム (P. 17)」を参照してください。

詳細情報:

オンライン バックアップのイベントとアラーム (P. 17)
StartOnlineBackup アプリケーション (P. 16)

StartOnlineBackup アプリケーション

StartOnlineBackup アプリケーションでは、オペレーティング システムのスケジュール アプリケーションを使用して、定期的なバックアップをスケジュールできます。たとえば、Windows 環境ではタスク スケジューラを、Solaris 環境では crontab を使用できます。 これらのアプリケーションを使用すると、スケジュールされたバックアップがずれる夏時間に関する潜在的な問題を回避できます。 このアプリケーションは、自動バックアップ セットアップを使用するより高度な代わりです。

StartOnlineBackup アプリケーションは、タスク スケジューラまたは crontab から起動できます。*$SPECROOT*/SS-Tools ディレクトリにあります。StartOnlineBackup の構文は次のとおりです。

```
StartOnlineBackup -lh <ランドスケープ ハンドル>
```

VNM モデルの［情報］タブの［オンライン データベース バックアップ］サブビューで、StartOnlineBackup のバックアップ パラメータを設定できます。 次のパラメータは例外です： 自動バックアップ、バックアップ間隔、次のバックアップの日時。

詳細情報:

オンライン バックアップの設定 (P. 13)

オンライン バックアップ ファイルでのデータベースのリストア

オンライン バックアップが生成する圧縮されていないバックアップ ファイルは、-cm オプションを指定した SSdbsave ユーティリティを使用して保存されるファイルと同じ形式で格納されます。

注: サフィックスが .gz の圧縮ファイルをリストアするには、CA Spectrum に付属する gzip ユーティリティを使用します。 このユーティリティは、SSdbsave ユーティリティが使用する形式にファイルをリストアします。

詳細情報:

データベースのリストア (P. 20)

オンライン バックアップのイベントとアラーム

オンライン バックアップに関係のあるイベントおよびアラームは次のとおりです。

イベント コード	イベント	イベントの原因	作成されるアラーム
0x00010903	OB_EVENT_GEN_FAILURE	データベースを開くことができない	レッド
0x00010903	OB_EVENT_GEN_FAILURE	データベースを閉じることができない	レッド
0x00010903	OB_EVENT_GEN_FAILURE	バックアップ失敗の原因となる他の理由	オレンジ
0x00010904	OB_EVENT_BACKUP_ON	自動バックアップがユーザによって有効にされた	なし
0x00010906	OB_EVENT_BACKUP_STARTED	オンライン バックアップが開始した	なし
0x00010907	OB_EVENT_BACKUP_COMPLETED	オンライン バックアップが正常に完了した	なし
0x00010908	OB_EVENT_NO_FILE_OR_DIRECTORY	ファイル/ディレクトリが存在しない、または読み取り/実行権限がない	イエロー オレンジ
0x0001090a	OB_EVENT_NO_CREATE_BACKUP_DIR	バックアップ ディレクトリを作成できない	オレンジ
0x0001090b	OB_EVENT_LOW_DISK_SPACE	ディスク空き容量不足、バックアップ失敗	イエロー
0x0001090c	OB_EVENT_COPY_FAILED	データベースをコピーできない	レッド
0x0001090e	OB_EVENT_DB_INCONSISTENT	データベース不整合	オレンジ
0x00010920	OB_EVENT_OFFLINE_SAVE_FAILED	保存プロセスのオフライン部分が失敗した	レッド

イベント コード	イベント	イベントの原因	作成されるアラーム
0x00010921	OB_EVENT_VNM_RESUMED	通常動作再開中、SpectroSERVER が一時停止された	なし
0x00010922	OB_EVENT_BAD_FILENAME	指定されたファイル名が有効な Posix ディレクトリ名ではない	オレンジ
0x00010924	OB_EVENT_DBSYNC_FAILED	バックアップ サーバ同期の試行が失敗した	オレンジ

手動バックアップ

CA Spectrum に含まれる SSdbsave データベース ユーティリティを使用すると、データベースのバックアップを手動で作成できます。 データベースには、モデル（モデル タイプ、関係、ルール）およびモデル自体を作成するために使用されるテンプレート情報のカタログが含まれます。 SSdbsave を使用して、これらのコンポーネントの一方または両方を含むデータベースのバックアップ コピーを作成できます。 モデリング カタログとモデルの両方を含む完全な保存を実行することをお勧めします。

詳細情報:

SSdbsave (P. 56)

完全なバックアップの作成

データベースの完全なバックアップ コピーを作成できます。 完全なバックアップには、モデリング カタログ（モデル タイプ階層、関係、ルール）およびモデル自体の両方が含まれます。

次の手順に従ってください:

1. *$SPECROOT*/SS ディレクトリに移動します。
2. SpectroSERVER および SpectroSERVER データベースにアクセスする他のプログラムが実行していないことを確認します。

注: 初めてインストールしたときには、このユーティリティは *$SPECROOT*/SS-Tools ディレクトリにあります。場所がシステム パス ステートメントで設定されない場合は、フル パスを使用する、相対パスを使用する、ファイルを移動する、または SpectroSERVER データベースと同じディレクトリにリンクする必要があります。

3. モデリング カタログとモデルの両方を保存するには、次のコマンドを入力します。

```
../SS-Tools/SSdbsave -cm <save_file>
```

保存ファイルの推奨される命名規則は、日付スタンプと保存コマンドのフラグを組み合わせたものです。 この場合は、「c」（カタログ）フラグと「m」（モデル）フラグの両方を使用しているので、2013 年 10 月 14 日とすると、保存ファイル名は db_20131014_cm のように入力します。

「db_20001014_cm.SSdb」という名前のファイルが作成されます。「.SSdb」というサフィックスは自動的に追加されます。

分割したバックアップの作成

SpectroSERVER データベース内のモデルは毎日変わる可能性がありますが、モデリング カタログは、アップグレード インストールを実行したとき、新しい管理モジュールを追加したとき、または Model Type Editor を使用して直接変更したときにのみ変化します。 そのため、データベースが非常に大きい場合は、バックアップを分割することによって時間を節約できます。 分割バックアップでは、モデルは定期的に保存しますが、モデリング カタログは変更した後でのみ保存します。 その後、モデリング カタログのバックアップを安全な場所に格納します。 モデル情報の最新の保存と併せて、このバックアップ ファイルは完全なデータベース バックアップを提供します。

以下の手順では、2013 年 10 月 14 日にデータベースの分割バックアップを実行し、モデリング カタログとモデルに異なる保存ファイルを作成する方法について説明します。

次の手順に従ってください:

1. *$SPECROOT*/SS ディレクトリに移動します。
2. SpectroSERVER および SpectroSERVER データベースにアクセスする他のプログラムが実行していないことを確認します。

   注: 初めてインストールしたときには、このユーティリティは *$SPECROOT*/SS-Tools ディレクトリにあります。場所がシステム パス ステートメントで設定されない場合は、フル パスを使用する、相対パスを使用する、ファイルを移動する、または SpectroSERVER データベースと同じディレクトリにリンクする必要があります。
3. モデリング カタログのみを保存するには、次のコマンドを入力します。

   ```
   ../SS-Tools/SSdbsave -c db_20131014_c
   ```
4. モデルのみを保存するには、次のコマンドを入力します。

   ```
   ../SS-Tools/SSdbsave -m db_20131014_m
   ```

   モデリング カタログは db_20131014_c.SSdb というファイル名で保存され、モデルは db_20131014_m.SSdb という名前で保存されます。

   注: モデルは、定期的に、またはネットワーク モデルを大きく変更したときに常に、保存してください。 モデリング カタログは、CA Spectrum の新バージョンをインストールしたとき、新しい管理モジュールを追加したとき、または Model Type Editor でカタログを変更したときに保存します。

データベースのリストア

CA Spectrum データベースは、モデル タイプ階層および関係が含まれるモデリング カタログ、ネットワーク内のエンティティを表すモデル、および開発者情報から構成されます。 SSdbload データベース ユーティリティ プログラムを使用すると、次のタスクを実行できます。

- 以前にバックアップしたデータベース ファイルをロードします。
- データベースを初期化し、モデリング カタログが含まれる以前にバックアップしたデータベース ファイルをロードします。
- 開発者情報をロードします。

データベースのロード

データベースが破損していない場合は、次のコマンドを使用して、データベースを初期化せずにバックアップ ファイルをロードできます。

```
SSdbload -l
```

SSdbsave ユーティリティを使用すると、SSdbload ユーティリティが作成したファイル、および CA Spectrum オンライン バックアップ ツールを使用して保存された圧縮されていないファイルを、データベースにロードできます。

注: サフィックスが .gz の圧縮ファイルをリストアするには、CA Spectrum に付属する gzip ユーティリティを使用します。 このユーティリティは、SSdbsave ユーティリティが使用する形式にファイルをリストアします。

SSdbload は保存ファイルを読み取って、保存された情報の種類（モデリング カタログおよびモデル）を判別します。 その後、データベースから対応する情報をすべて削除し、データベースにバックアップ情報をロードします。 たとえば、モデルのみが含まれる保存ファイル（-m オプションのみを使用して保存されたもの）をロードする場合、SSdbload は最初にデータベースからモデル情報をすべて削除し、次に保存ファイルからモデル情報をロードします。

同様に、カタログのみが含まれる保存ファイルをロードする場合は、データベースからカタログが削除された後、保存ファイルからカタログがロードされます。 モデリング カタログとモデル情報に個別のバックアップ ファイルを作成した場合は、カタログ（モデル タイプと関係）情報を最初にロードし、次にモデル情報をロードします。

重要: -l オプションを使用して、既存のデータベースに追加のカタログまたはモデルをロードすることは できません。 -l オプションは、データベースのコンテンツを追加しません。 代わりに、ロードされた保存ファイルのコンテンツを特定し、ロードの前に同じタイプの既存情報をすべて削除します。

次の手順に従ってください:

1. *$SPECROOT*/SS ディレクトリに移動します。
2. SpectroSERVER および SpectroSERVER データベースにアクセスする他のプログラムが実行していないことを確認します。

3. データベース構造にロードするファイルの名前を指定して、SSdbload コマンドを実行します。

   ```
   ../SS-Tools/SSdbload -l <filename>
   ```

   注: 初めてインストールしたときには、このユーティリティは *$SPECROOT*/SS-Tools ディレクトリにあります。場所がシステム パス ステートメントで設定されない場合は、フル パスを使用する、相対パスを使用する、ファイルを移動する、または SpectroSERVER データベースと同じディレクトリにリンクする必要があります。

4. ステップ 3 でロードしたバックアップ ファイルにカタログ情報だけが含まれた場合は、モデル情報が含まれるバックアップ ファイルの名前を置き換えて、ステップ 3 を繰り返します。

   データベースが使用できる状態になります。

データベースの初期化とロード

1 つの操作でデータベースを初期化してロードできます。データベースを初期化すると、モデリング カタログ、モデル情報、ベンダー情報、およびアーカイブされていないイベントと統計ログ レコードが削除され、すべてのデータベース ファイルが更新されます （データベースを同期しておくには、SpectroSERVER データベースを初期化するときに、ddm_load を使用して履歴データベースも初期化します）。

破損の可能性がある場合は、常にデータベースを初期化します。 データベースを初期化すると、-I オプションは、バックアップ ファイルの作成時に使用された SSdbsave オプションに従って、新しいデータベース情報（モデリング カタログおよびモデル）をロードします。

重要: -i オプションは、モデル、モデル タイプ、関係、ベンダー情報、およびアーカイブされていないイベントと統計ログ レコードをすべてデータベースから削除します。-i オプションを指定して SSdbload を実行すると、カタログをロードし、モデルなしで SpectroSERVER を開始できます。 または、データベースを完全にリストアできます。 SSdbsave の -c オプションと -m オプションの両方を使用して保存された 1 つのバックアップ ファイル、または分割されたバックアップ（-c および -m オプションを別々に使用して保存されたファイル）をロードすることにより、データベースをリストアします。

分割されたバックアップを使用する場合は、ロードするバックアップファイルごとに SSdbload を 1 回実行します。 -i オプションは、最初の SSdbload のときに 1 回だけ使用します。カタログをモデルの前にロードする必要があります。

1 つの操作で初期化しロードするには、-l オプションと一緒に -i オプションを使用します。 この方法は、SSdbsave で -c および -m オプションを使用して保存したファイルから新しいデータベース情報をロードする場合にお勧めします。

次の手順に従ってください:

1. データベースが含まれるディレクトリに移動します。
2. SpectroSERVER および SpectroSERVER データベースにアクセスする他のプログラムが実行していないことを確認します。
3. データベースを初期化する場合、現在の実行可能ファイルより古い .after ファイルを使用する場合は注意する必要があります。 .after ファイルが実行可能ファイルより古い場合、期待する情報が存在しない可能性があります。 .after ファイルの説明については、SSdbload を参照してください。
4. 以下の SSdbload を入力します。

   ```
   ../SS-Tools/SSdbload -i -l <filename>
   ```

   データベースが初期化され、指定したバックアップ ファイルからの情報がロードされます。 データベースが使用できる状態になります。

詳細情報:

SSdbload (P. 51)

開発者情報のロード

登録済みの開発者は、SSdbload と -d オプションを使用して、開発者情報（開発者 ID が含まれるファイル）を SpectroSERVER データベースにロードできます。 データベースの初期化の後には開発者情報を再ロードする必要があります。しかし、次の初期までにこの情報をロードできるのは 1 回だけです。 -d オプションを使用して 2 回目の SSdbload を実行すると、エラー メッセージが表示され、開発者情報は変更されません。

重要: ただ 1 つのデータベースに開発者情報をロードし、そのデータベースでモデリング カタログのすべての編集を実行します。 その後、必要であれば、SSdbload を使用して変更を他のデータベースに伝達できます。ただし、同じ開発者情報を複数のデータベースにロードすると、モデル タイプ、属性、または関係ハンドルが重複する可能性があります。 重複すると、ランドスケープ間で履歴データを相関できなくなります。

次の手順に従ってください:

1. *$SPECROOT*/SS ディレクトリに移動します。
2. SpectroSERVER および SpectroSERVER データベースにアクセスする他のプログラムが実行していないことを確認します。

   注: 初めてインストールしたときには、SSdbsave は *$SPECROOT*/SS-Tools ディレクトリにあります。 その場所がシステム パス ステートメントで設定されない場合は、フル パスを使用する、相対パスを使用する、ファイルを移動する、または SpectroSERVER データベースと同じディレクトリにリンクする必要があります。
3. 以下のコマンドを入力します。

   ```
   ../SS-Tools/SSdbload -d <開発者ファイル名>
   ```

データベース ロックの削除

CA 製プログラムが CA Spectrum データベースにアクセスする場合は常に、そのデータベース ディレクトリにロック ファイルが作成されます。 SpectroSERVER データベース内のロック ファイルの名前は .VNMDB.LOCK です。 DDM データベース内のロックファイルの名前は .DDMDB.LOCK です。 ロック ファイルは、データベースが現在使用されていることを示すインジケータとして機能し、別の CA 製プログラムからはアクセスできません。

以下の CA 製プログラムがロック ファイルを作成します。

- converter
- dbtool
- lh_set
- mte （Model Type Editor）
- reports
- SpectroSERVER
- SSdbdelete
- SSdbload
- SSdbsave

CA 製プログラムが正常終了すると、ロック ファイルは削除されます。 ただし、CA 製プログラムが異常終了し、ロック ファイルが後に残る場合があります。 ロック ファイルが削除されなければ、データベースにアクセスする他の CA 製プログラムの実行を妨げます。ロック ファイルが存在している状態で、別の CA 製プログラムが開始すると、以下のメッセージが表示されます。

```
Database already locked by:<user id>,
by process:<process name>,
with process ID:<process id>,
on network node:<node name>,
which started at:<date/time stamp>
```

異常終了したプログラムによって後に残されたと信じる .VNMDB.LOCK または .DDMDB.LOCK ファイルは手動で削除できますが、最初にプログラムが実行されていないことを確認する必要があります。 プログラムがまだ実行している（そして、高い優先度のプログラムがデータベースにアクセスするのを妨げている）場合は、最初にプログラムの通常終了を試行します。 それが不可能な場合は、UNIX の kill コマンド（オプションなし）または Windows タスク マネージャの［プロセスの終了］ボタンを使用してプログラムを停止できます。 これらのどの方法でもロック ファイルを削除できます。

プログラムが異常終了した場合は、*$SPECROOT*/SS または *$SPECROOT*/SS/DDM ディレクトリに移動し、次のいずれかのコマンドを入力して、ロック ファイルを手動で削除できます。

- rm .VNMDB.LOCK （SS ディレクトリからロック ファイルを削除する場合）
- rm .DDMDB.LOCK （DDM ディレクトリからロック ファイルを削除する場合）

重要: データベースの整合性を維持するには、*$SPECROOT/SS* または *$SPECROOT*/SS/DDM ディレクトリのいずれかからロック ファイルを削除する必要があるときはいつでも、SpectroSERVER および DDM データベースの両方をリストアする必要があります。 データベース リストア処理を実行するまで、SpectroSERVER は再起動しません。

また、.VNMDB.LOCK および .DDMDB.LOCK ファイルは非 CA 製プログラムによるデータベース アクセスは妨げず、そのようなプログラムがロックに遭遇してもエラー メッセージは生成されないことに注意してください。そのため、CA 製ではないツールまたはアプリケーションが CA Spectrum データベースにアクセスする前に、そのデータベースにアクセスするすべての CA Spectrum プロセスがシャットダウンされていることを確認する必要があります。

詳細情報:

データベースのリストア (P. 20)

モデル タイプのインポートとエクスポート

dbtool ユーティリティを使用すると、モデル タイプおよび関連するオブジェクト（属性、関係、メタルール）を SpectroSERVER データベースからインポートおよびエクスポートできます。

詳細情報:

dbtool (P. 32)

データベース破損からのリカバリ

ハードウェア障害や停電などにより、データベースが壊れ、以下のようなエラー メッセージが発生することがあります。

```
Sep 18 15:42:39 ERROR at CsSSDbRp.cc(642):
table open failed @ TableImpl.cc:674(0x2)
Db::open: No such file or directory
Could not open the database. VNM exiting.
Landscape not initialized. VNM exiting.
```

また、SpectroSERVER または他のアプリケーションが、ユーザ モデルの不足やデータベース ロックの存在などの特定原因に容易に起因しないエラー メッセージを生成する場合にも、データベース破損が発生した可能性があります。

定期的にバックアップを行っている場合は、SSdbload ユーティリティを使用してデータベースを初期化し、データベースの最後の既知の「正常な保存」つまりモデリング カタログとモデルの両方を含み、破損の前に作成された保存を再ロードすることにより、最小限の情報損失でデータベースの破損からリカバリできます。 データベースの既知の正常な保存がない場合、または正常な保存であると考えるものを再ロードした後でもアプリケーション エラーが解決しない場合は、CA サポート担当者にご連絡ください。

詳細情報:

SSdbload (P. 51)

SpectroSERVER データベースのトラブルシューティング

このセクションでは、カスタマによって CA サポートによく報告される問題について説明します。

インストール/アップグレードの後の SSdbload を使用した古いオブジェクトのロード

インストールまたはアップグレードの後に SSdbload を使用して古いモデル、モデル タイプ、または関係をロードすると、深刻なデータベースの問題を引き起こす可能性があります。 これらの問題を回避する最適な方法は、SSdbload がどのように作動するか理解することです。 -l オプションを使用してファイルのロードを試みると、SSdbload はファイルを検索して含まれるオブジェクトの種類を特定します（ファイルを作成するために SSdbsave で使用されたオプション フラグに依存します）。 ファイル内の各オブジェクト タイプについて（モデリング カタログおよび/またはモデル）、SSdbload は最初にデータベースからそのタイプのすべての既存オブジェクトを削除します。 たとえば、次のコマンドを実行します。

```
SSdbload -l somefile_c.SSdb
```

新しくアップグレードされたデータベースを古いモデリング カタログで更新します。データベースのモデリング カタログを保存ファイルからのものに置き換えます。 アップグレードされたモデリング カタログは新しいバージョンの CA Spectrum で作動する必要がある可能性があるので、これはデータベースを使用できなくなる可能性があります。 また、Model Type Editor で作成された変更も失われます。

SSdbload の -i オプションの不適切な使用

SSdbload ツールの -i オプションの使用はいくつかの潜在的に深刻な影響を含んでいます。 このオプションが使用される場合、データベース内の既存のオブジェクトがすべて削除されます。 削除されるオブジェクトには、モデル タイプ、モデル、関係、属性、ルール、アーカイブされていないイベント、アーカイブされていない統計が含まれます。 したがって、以下のコマンドを実行する場合、

```
SSdbload -i -l somefile_m.SSdb
```

そして somefile_mSSdb ファイルにモデルのみが含まれる場合、ロードは失敗します。 モデリング カタログのみが含まれるファイル（つまり、-c オプションのみで保存されたもの）を同じコマンドで実行する場合、結果的なデータベースにはファイルからロードされるモデル タイプおよび関係のみが含まれます。 以前にデータベースに存在した他のすべてのオブジェクトは、ユーザ作成のすべてのモデルを含めて失われます。

オペレーティング システム/プラットフォーム間のデータベース ファイルのコピー

データベース ファイル（*.db、*.ix、log.*）はオペレーティング システムおよびプラットフォームに固有です。 オペレーティング システム間またはプラットフォーム間のデータベースのコピーはサポートされていません。 これらのファイルに含まれているデータを移動する、サポートされる唯一の方法は、移動元コンピュータで SSdbsave を実行し、移動先コンピュータで SSdbload を実行する方法です。 これらのツールが各コンピュータに存在し、それらがそれぞれのサーバ用にコンパイルされていれば、データの転送は成功します。

モデリング カタログおよびモデルのロード順序

CA Spectrum モデルに関する 2 つの事実を理解することが重要です。

- モデルはモデル タイプのインスタンスであり、モデリング カタログ内で定義されています。
- モデルとモデル タイプには依存関係があります。

特定のモデルのモデル タイプが存在しない場合、または正しいバージョンではない場合、そのモデルが含まれるファイルをロードする試行は失敗します。 常に、モデルをロードする前に、そのモデルを作成する基になっている前提条件のモデリング カタログをロードします。

同じ開発者 ID を持つ複数のデータベースへのモデリング カタログ オブジェクトの追加

Model Type Editor を使用してモデリング カタログ オブジェクト（モデル タイプ、関係、属性）をデータベースに追加すると、それらには一意のハンドルが割り当てられます。 ハンドルは、アクティブな開発者 ID と、そのオブジェクト タイプに対して次に使用可能なシーケンシャル番号で構成されます。 ただし、これらのシーケンシャル番号は各データベースで個別に割り当てられ、保持されます。 したがって、新しいオブジェクトが同じ開発者 ID を使用して別のデータベースに追加される場合、同じハンドルが各データベースで別のオブジェクトを表すことがあります。 分散 SpectroSERVER 環境では、この種の競合により、ランドスケープ間で履歴データを相関させることが不可能になります。

重複したハンドルに関する問題を回避するには、モデリング カタログ情報は 1 つのデータベースでのみ変更するようにします。 SSdbsave と -c オプションを使用して変更を保存し、次に必要に応じて、SSdbload と -l オプションを使用して他のデータベースに変更を伝達します。

データベース ファイルとディレクトリでの UNIX ファイル アクセス権限

SpectroSERVER およびデータベース ツールが正しく動作するには、すべてのデータベース ファイル（*.db、*.ix、 log.*）およびファイルが含まれるディレクトリに対する書き込み権限が必要です。 データベース ツールまたは SpectroSERVER からプログラムがロック ファイルを開くことができなかったことを示すメッセージを受け取った場合は、データベース ファイルとディレクトリに対する権限を確認してください。

CA Spectrum データベース ツール

このセクションでは、以下のデータベース ツール、ユーティリティ プログラム、スクリプトについて説明します。

- db_remove
- dbtool
- HostUpdate
- MapUpdate
- reports
- SSdbload
- SSdbsave
- データベース モデル変更ツール（DBconv）

注: 前記のツールはすべて、*$SPECROOT*/SS-Tools ディレクトリにインストールされます。 ツールの場所がシステム パス ステートメントで設定されていない場合は、フル パスを使用するか、ツールを SpectroSERVER データベースと同じディレクトリに移動する必要があります。

db_remove

このユーティリティは、廃止されたモデル タイプ（およびそれらの関連する親属性およびメタルール）を SpectroSERVER データベースのモデリング カタログから削除します。

注: このツールは、一般的な使用は意図されていません。 CA によって発行された具体的な指示または技術告示がない場合は、db_remove を実行しないでください。 詳細については、CA サポート担当者にお問い合わせください。

このユーティリティの形式は以下のとおりです。

```
db_remove [-debug] [<MTH_FILE>]
```

-debug

冗長な情報メッセージ出力を有効にします。

<MTH_FILE>

データベースから削除するべきモデル タイプ ハンドルの空白区切りリストが含まれるテキスト ファイルを指定します。

エラー

Warning: Model Type 0x???????? - “*model type name*” could not be removed.

このメッセージは、ユーザ（または別のベンダーまたはパートナー）が削除しているモデル タイプから新しいモデル タイプを派生している珍しい状況で発生します。その場合、このモデル タイプから派生しているものを確認するには Model Type Editor を実行します。 その後、別の廃止されていない派生ポイントに派生しているモデル タイプを移動します。 最後に、元の MTH_FILE で db_remove ツールを再実行します。

Warning: Model Type 0x???????? - "*model type name*" is currently referenced in a default Attribute value. The reference occurs in Attribute 0x???????? - "*attribute name*", at Model Type 0x???????? - "*model type name*".

このメッセージは、ユーザ（または別のベンダーまたはパートナー）が（MODELTYPE_HANDLE または MODELTYPE_HANDLE リストの）属性のデフォルト値からの参照を削除しているモデル タイプ ハンドルの値に追加している場合に表示されます。 このメッセージは、モデル タイプの削除を妨げません。 ただし、属性のデフォルト値を調べてください。それがどのように使用されているかに応じて、参照を削除するか、または適切で廃止されていないモデル タイプ ハンドルに置換します。

dbtool

ネットワークが成長すると、通常モデル タイプをデータベースに追加します。 場合によっては、新しいデータベースをインストールせずに、新しいモデル タイプを追加することもできます。 dbtool ユーティリティを使用すると、以下の関連するタスクを実行できます。

- SpectroSERVER データベースの永続カタログからモデル タイプをエクスポートします。
- SpectroSERVER データベースの永続カタログにモデル タイプをインポートします。
- エクスポート ファイルの内容を表示します（ダンプ）。

  ダンプ機能は、ワークステーションの標準出力に出力を送信します。 通常は画面に表示されます。 ただし、出力をファイルまたはプリンタに送信することもできます。

Model Type Editor には、モデル タイプをエクスポートおよびインポートする同様の機能が含まれます。ただし、dbtool ユーティリティではコマンド ライン引数として複数のファイルを指定できます。 このため、ファイルのセットを一括処理する方が便利です。

注: 詳細については、「Model Type Editor ユーザ ガイド」を参照してください。

機能（エクスポート、インポート、ダンプ）に応じて、dbtool ユーティリティは以下のいずれかのタイプのファイルを使用します。

- ファイル拡張子が .m のモデル タイプ リスト ファイル。このファイルには、モデル タイプ ID コードのリストが含まれます。 各 ID は、個別の行（リターン文字によって分離された）に記述するか、またはスペースで区切って同じ行に記述します。 適当なシェル テキスト エディタでこれらのファイルを作成できます。
- ファイル拡張子が .e. の抽出ファイル。これらのバイナリ ファイルは、カタログ ファイルとも呼ばれ、dbtool のエキスポート機能または Model Type Editor を使用して作成できます。 関連する *.m ファイル（dbtool によって作成される場合）でリストされているモデル タイプに関連するデータベースの情報がすべてを含まれます。または、Model Type Editor がカタログ ファイルを作成する場合は、選択されたモデル タイプのすべてに関連する情報が含まれます。

  カタログ ファイルは、電子メール ファイル、DVD などのメディアを使用して、あるシステムから別のシステムにモデル タイプを転送する手段です。

重要: dbtool を実行する前に、SpectroSERVER と、サードパーティ プログラムを含む SpectroSERVER データベースにアクセスする他のプログラムをシャットダウンしてください。 dbtool は常に、SpectroSERVER データベースが含まれるディレクトリから実行します。

dbtool ユーティリティは、可能な場合は記号名を使用できるように、データベースをロードします。

このユーティリティの形式は以下のとおりです。

```
dbtool          [dump <file>.e [ <file>.e ...] ]
dbtool          [dump_mt <file>.e [ <file>.e ...] ]
dbtool          [import <file>.e [ <file>.e ...] .xml [ <file>.xml ...] ]
dbtool          [export <file>.m [ <file>.m ...] ]
```

dump

指定された .e ファイル（カタログ ファイル）の内容を読み取り可能な形式で表示します。 dump_mt 引数とは異なり、この引数には、出力のモデル タイプ情報だけでなく、属性、関係、およびメタルールの情報も含まれます。 必要に応じて、ダンプする複数のカタログ ファイルを指定できます。

注: このオプションは、エクスポートされているモデル タイプが元になっている属性の属性名だけをレポートします。 拡張されたフラグ、OID プレフィックス値、または OID 参照値への変更に関する出力は提供しません。

この引数はデータベースでは作動しません。

dump_mt

指定された .e ファイルでリストされているモデル タイプに関する情報を読み取り可能な形式で表示します。 必要に応じて、ダンプする複数のカタログ ファイルを指定できます。 dump 引数とは異なり、この引数の出力には属性、関係、メタルールの情報は含まれません。

この引数はデータベースでは作動しません。

import

指定された .e ファイルまたは .xml ファイル内のモデル タイプ、属性、関係、およびメタルールをデータベースにインポートします。 必要に応じて、インポートする複数のカタログ ファイルを指定できます。

データベースにすでにあるモデル タイプをインポートする場合は、モデル タイプの再定義に関する警告メッセージが表示されます。この警告メッセージは無視できます。

export

指定された .m ファイルを使用して、.m ファイルのモデル ハンドルで指定されているモデル タイプを含むカタログ ファイルを作成します。 複数の .m ファイルを指定して、複数のカタログ ファイルを作成できます。

.m ファイル内のモデル タイプ ハンドル エントリは、「0x」（ゼロと小文字の「x」）が前に付いた 16 進の整数である必要があります。 さらに、エントリは少なくとも 1 つのスペース文字または改行文字によって分離されている必要があります。

help

コマンドの使用方法に関する情報を表示します。

詳細情報:

カタログ ファイルの内容の出力デバイスの指示 (P. 38)

dbtool を使用したモデル タイプのエクスポート

dbtool ユーティリティを使用して、モデル タイプをエクスポートできます。

注: SpectroSERVER データベースに現在ロードされている開発者 ID を使用して作成されたモデル タイプ、属性、関係（および関連するメタルール）のみをエクスポートできます。 その開発者 ID はこれらのオブジェクトの「所有者」です。 ユーザがエクスポート操作に含まれるモデル タイプまたは他のオブジェクトの所有者でない場合、操作はエラー メッセージで終了します。

エクスポート プロセスが作成するカタログ ファイル（.e ファイル）には以下の情報が含まれます。

- エクスポートされているモデル タイプが起源の属性記述子。
- （たとえば、デフォルト値を指定して継承値をオーバーライドすることによって）特殊化された属性記述子。
- モデル タイプおよびすべての祖先モデル タイプが先行詞または述語として参加している関係および関連するメタルール。

「周辺」モデル タイプには、同じ *.e ファイルにエクスポートされないベース モデル タイプが少なくとも 1 つあります。 これらのタイプは、含まれないベース モデル タイプから継承される属性値および拡張が含まれる点で異なります。 その包含は、それらの値と拡張の可用性を保証します。

次の手順に従ってください:

1. データベースが含まれるディレクトリに作業ディレクトリを変更します。
2. dbtool ユーティリティが現在のディレクトリまたはシステム検索パスにあることを確認します。 または、後で dbtool コマンドを呼び出すときに（ステップ 4）、適切な相対パス名または絶対パス名を使用します。

3. エクスポートするモデル タイプを指定する .m ファイルを作成 (P. 36) します。
4. 以下のコマンドを使用して、モデル タイプをエクスポートします。

   ```
   ../SS-Tools/dbtool [export<filename_11>.m [<filename_2>.m … ] ]
   ```

   次に例を示します。

   ```
   ../SS-Tools/dbtool export smart_hub.m smart_router.m
   ```

   この例のコマンドでは、smart_hub.m で定義されているモデル タイプをエクスポートし、smart_hub.e という名前のファイルに結果の出力を書き込みます。 その後、同じ方法で smart_router.m を処理します。

詳細情報:

dbtool (P. 32)
.m ファイルの作成 (P. 36)

.m ファイルの作成

dbtool ユーティリティのエクスポート機能は、1 つ以上の *.m ファイルを使用して、モデル タイプ ハンドルによってエクスポートされるモデル タイプを指定します。 dbtool を使用してエクスポートを実行する前に、適当なシェル テキスト エディタを使用してこれらのファイルを 1 つ以上作成します。

通常、エクスポートするモデル タイプのリストには、エクスポートするモデル タイプによって必要とされ、送信先データベースに存在しないすべてのベース モデル タイプが含まれます。 ただし、依存関係は、通常、1 つ以上の「コア」カタログに含まれている一般的に使用される特定のベース モデル タイプに制限されます。 これらのカタログは基本的な CA Spectrum システムの一部として含まれています。

次の手順に従ってください:

1. モデル タイプ ハンドル エントリは、「0x」（ゼロと小文字の「x」）が前に付いた 16 進の整数として入力します。 0x の後に、4 桁の開発者 ID と、4 桁のシーケンス番号を入力する必要があります。
2. モデル ハンドル エントリは、少なくとも 1 つのスペース文字または改行文字によって分離します。

3. ハンドルのリストのファイル名には .m ファイル拡張子を付けます。

   たとえば、次のリストでは、エクスポートする 5 つのモデル タイプ（デフォルト開発者 ID を使用して作成された）が指定されています。

   ```
   0xffff0003 0xffff0008
   0xffff0017 0xffff0023
   0xffff0045
   ```

   この例では、現在ロードされている開発者 ID（この場合は、デフォルトの開発者 ID）で作成された 3 番目、8 番目、17 番目、23 番目、45 番目のデータベース情報を含む *.e ファイルが作成されます。

重要: データベース アクセスは一度に 1 つのアプリケーションに制限される必要があります。 dbtool が使用中の場合、他のすべてのアプリケーション（OneClick および Model Type Editor を含めて）はアクセスを拒否されます。CA 製プログラムは自動的に他の CA 製品をロックしますが、この注意がサードパーティ アプリケーションによってバイパスされた場合、データベースが破損する可能性があります。

dbtool を使用したモデル タイプのインポート

dbtool インポート機能を使用すると、dbtool エクスポート機能または Model Type Editor を使用して作成したカタログ ファイル（.e ファイル）から SpectroSERVER データベースにモデル タイプをインポートできます。

データベース アクセスは一度に 1 つのアプリケーションに制限される必要があります。 dbtool が使用中の場合、他のすべてのアプリケーション（OneClick および Model Type Editor を含めて）はアクセスを拒否される必要があります。CA 製プログラムは自動的に他の CA 製品をロックしますが、この注意がサードパーティ アプリケーションに関して無視またはバイパスされた場合、データベースが破損する可能性があります。

次の手順に従ってください:

1. SpectroSERVER データベースをバックアップします。
2. データベースが含まれるディレクトリに現在の作業ディレクトリを変更します。
3. インポートするモデル タイプ ファイルが現在のディレクトリまたはシステム検索パスにあることを確認します。

4. データベースが初期化される場合は（.d および .k 拡張子のファイル）、次のステップに進みます。 それ以外の場合は、SSdbload -i -l ユーティリティ コマンドを使用して、データベースを初期化し、コア モデル タイプ派生をロードします。
5. 以下のコマンドを使用して、モデル タイプをインポートします。

   ```
   ../SS-Tools/dbtool import [<filename_1.e> … <filename_n.e>]
   ```

   次に例を示します。

   ```
   ../SS-Tools/dbtool import rmon1.e rmon2.e
   ```

   最後のソース ファイルの内容がインポートされた後、操作完了のメッセージが表示されます。

詳細情報:

データベースのバックアップ (P. 11)
SSdbload (P. 51)

カタログ ファイルの内容の出力デバイスの指示

dump 引数または dump_mt 引数を指定して dbtool ユーティリティを実行すると、ユーティリティは出力をワークステーションの標準出力デバイスに送信します。通常は表示画面です。 ファイルまたはプリンタに出力を直接リダイレクトする場合は、UNIX のコマンド ラインで標準パイプ コマンドを使用して目的の出力にリダイレクトします。

例 1

次のコマンドは、rmon.e という名前のカタログ ファイルの出力をワークステーションの標準出力デバイスにダンプします。

```
dbtool dump rmon.e
```

出力ワークステーションに 1 回だけ表示され、長すぎる場合は出力の最後だけが表示されます（つまり、ディスプレイに複数行表示できても）。

例 2

次のコマンドは前の例と同じカタログ ファイルの出力をダンプしますが、出力はワークステーション画面に増分表示ファイルとして送信され、最初の画面が表示されます。

```
dbtool dump rmon.e | more
```

画面を表示しながら連続する行をインクリメントするには Return キーを押し、次の画面にインクリメントするにはスペース キーを押します。

例 3

次のコマンドは前の例と同じカタログ ファイルの出力をダンプしますが、出力は filesave.out という名前の ASCII ファイルに書き込まれます。

```
dbtool dump rmon.e > dumpouts/filesave.txt
```

ファイルは、作成済みの *$SPECROOT*/SS/dumpouts ディレクトリに作成されます。

例 4

次のコマンドは前の例と同じカタログ ファイルの出力をダンプしますが、出力は <ptr> によって指定されているプリンタに印刷ファイルとして送信されます。

```
dbtool dump rmon.e > lpr -P<ptr>
```

dbtool のトラブルシューティング

dbtool を実行すると、次のエラーが発生する可能性があります。

database open failed

SpectroSERVER データベースは現在のディレクトリに存在しません。

database files are missing are missing read and/or write permissions

データベースに開発者 ID がロードされていないか、ユーザは指定されたデータベースに関して有効なユーザではありません。

Database already locked by:<user id>
by process:<process name>
with process ID:<process id>
on network node:<node name>
which started at:<date/time stamp>

データベースは別のプロセスによってロックされています。

詳細情報:

開発者情報のロード (P. 24)
データベース ロックの削除 (P. 24)

MapUpdate

MapUpdate はランドスケープ マップを変更し表示するために使用されるユーティリティです。 このプログラムは *$SPECROOT*/SS-Tools ディレクトリにあり、次のタスクを実行します。

- ランドスケープ マップからランドスケープ エントリを削除します
- 現在のランドスケープ マップを表示します

セカンダリ SpectroSERVER を削除する場合は、MapUpdate を実行して、プライマリ SpectroSERVER 上のロードされたランドスケープのリストからセカンダリ SpectroSERVER を削除してください。ランドスケープ エントリに対してタイムアウト値を使用している場合は、セカンダリ SpectroSERVER でランドスケープ エントリがタイムアウトする前に、MapUpdate -remove を実行します。 そうしないと、プライマリ SpectroSERVER のロードされたランドスケープのリストからセカンダリ SpectroSERVER を正しく削除できない可能性があります。

注: 以前の CA Spectrum リリースでは、ランドスケープ エントリはデフォルトで 1 時間後にタイムアウトになり、ランドスケープ マップから自動削除されました。CA Spectrum 9.2.2 以降、ランドスケープ エントリはデフォルトではタイムアウトになりません。 MapUpdate を使用して、ランドスケープ マップからエントリを手動で削除する必要があります。 ランドスケープ エントリに対してタイムアウト値を使用できます。 詳細については、「分散 SpectroSERVER 管理者ガイド」を参照してください。

コマンドの形式は以下のとおりです。

```
MapUpdate [-remove LANDSCAPE_HANDLE] [-precedence PRECEDENCE] [-view]
```

-remove *LANDSCAPE_HANDLE*

削除するランドスケープのハンドルを指定します。 削除する SpectroSERVER をまずシャットダウンする必要があります。 さらに、セカンダリ SpectroSERVER を削除する場合は、プライマリ SpectroSERVER が実行している必要があります。

デフォルト： 0x400000

-precedence *PRECEDENCE*

削除するランドスケープの優先順位値。

-view

指定すると、現在のランドスケープ マップが表示されます。

HostUpdate

SS-Tools ディレクトリにある HostUpdate ユーティリティを使用すると、指定するホストに対してホストによって分割されているすべてのランドスケープ マップ エントリを削除できます。

```
HostUpdate [-remove HOSTNAME] [-view]
```

-remove

ホスト名の入力により指定するホストのエントリをすべて削除します。

注: ホスト エントリは、自動的にランドスケープ マップからタイムアウトします。 そのため、このオプションは恐らく不要です。 ただし、自動タイムアウト メカニズムが失敗した場合、またはタイムアウト間隔の前にエントリを削除する場合は、このオプションを使用できます。

-view

ホストによって分割されるエントリをすべて表示します。

reports

SS-Tools ディレクトリにあるこのコマンド ライン ユーティリティを使用すると、現在のモデリング カタログ内の選択したオブジェクトのリストを表示できます。 reports は、SpectroSERVER データベースがインストールされているディレクトリから実行します。

このコマンドは、レポート生成シーケンス中にデータベースをロックします。 その結果、一度に 1 つのレポートのみを実行できます。

注: reports を実行する前に、SpectroSERVER と、VNM、Model Type Editor、サードパーティ ユーティリティなどの SpectroSERVER データベースにアクセスする他のプログラムをシャットダウンしてください。 そうしないと、データベースが破損する可能性があります。

このユーティリティの形式は以下のとおりです。

```
reports [-mtype <name_pattern>] [-relation <name_pattern>] [-handle <handle>]
[-attrflags <defglmprsvw>] [-fields <cdefgimnotvGE>] [-types <bierdtcgmMRlaoIAOTU>]
[-recursive]
[-invisible] [-lists] [-nolists] [-groups] [-help]
```

-mtype

レポートに含めるモデル タイプを指定します。指定したテキスト文字列が名前に含まれるモデル タイプがすべて含まれます。たとえば、モデル タイプが IRM の場合、Hub_CSI_CIRM、Hub_CSI_IRM2、Hub_CSI_SIRM、および名前に「IRM」が含まれる他のすべてのモデル タイプのセクションがリストされます。

注: 「該当するすべての」エントリに対するワイルドカードとして、ピリオド（.）を使用できます。レポートの表示に長時間を必要とするので、ワイルドカード オプションは控え目に使用します。属性情報のないすべてのモデル タイプについてのレポートを表示するには、*$SPECROOT*/SS ディレクトリから以下のコマンドを実行します。

```
../SS-Tools/reports -mtype . -fields e
```

-relation

レポートに含める関係を指定します。出力にはモデル タイプ間の関係がリストされます。

注: 「該当するすべての」エントリに対するワイルドカードとして、ピリオド（.）を使用できます。すべての関係についてのレポートを表示するには、*$SPECROOT*/SS ディレクトリから以下のコマンドを実行します。

```
../SS-Tools/reports -rel .
```

-handle

レポートに含めるモデル タイプの 16 進のハンドルを指定します。「0x」プレフィックスを前に付けても付けなくてもかまいません。指定したテキスト文字列がハンドルに含まれるモデル タイプがすべて含まれます。この引数は mtype 引数と似ていますが、名前の文字列ではなくモデル タイプ ハンドルを受け付けることが異なります。

たとえば、0x180027 のハンドルを指定した場合、レポートには 0x180000 の開発者 ID で作成された 27 番目のモデル タイプのセクションが含まれます。逆に、ハンドルが 180 の場合は、レポートには、その開発者 ID で作成されたすべてのモデル タイプだけでなく、同じ 3 桁がハンドルのどこかに含まれる他のモデル タイプも含まれます。

-attrflags defglmprsvw

指定したフラグ セットのいずれかを持つモデル タイプの属性だけがレポートに含まれるように、レポートをフィルタします。

d

データベース

e

外部

f

グローバル

g

保証済み

l

ログ記録

m

メモリ

p

ポーリング

r

読み取り可能

s

共有

v

保持

w

書き込み可能

注: フラグのリストの代わりに連続する 2 個の一重引用符または二重引用符を入力することは、すべてのフラグを入力することと同じです。値を指定しないでこの引数を使用すると、レポート ユーティリティはエラーで失敗します。 ヘルプ ファイルが表示されます。 これらの属性コンポーネントの詳細については、「Model Type Editor ユーザ ガイド」を参照してください。

-fields cdefgimnotvGE

指定した属性コンポーネント（フィールド）のみを含むように、レポートをフィルタします。

c

作成者モデル タイプ

d

OID 参照

e

属性情報を印刷しない

f

フラグ

g

ポーリング グループ

i

属性 ID

m

マニフェスト定数名

n

開発者 ID を含む名前

o

OID プレフィックス

t

タイプ

v

デフォルト値

G

グループ ID

E

列挙された値リスト

注: フラグのリストの代わりに連続する 2 個の一重引用符または二重引用符を入力することは、すべてのフラグを入力することと同じです。値を指定しないでこの引数を使用すると、レポート ユーティリティはエラーで失敗します。 ヘルプ ファイルが表示されます。 これらの属性コンポーネントの詳細については、「Model Type Editor ユーザ ガイド」を参照してください。

-types bierdtcgmMRlazoIAGOTU

指定した型の属性のみを含むように、レポートをフィルタします。

b

ブール値

i

整数

e

列挙値

r

実数

d

日付

t

時刻

c

カウンタ

g

ゲージ

m

モデル ハンドル

M

モデル タイプ ハンドル

R

関係ハンドル

l

ランドスケープ ハンドル

a

属性 ID

z

テキスト文字列

o

オブジェクト ID

I

IP アドレス

A

エージェント ID

O

オクテット文字列

T

タグ付きオクテット

U

64 ビット符号なし整数

-recursive

レポートに子孫（子）モデル タイプを含めます。

-invisible

レポートに以下のモデル タイプを含めます。

- 検索パターンに一致するすべての表示モデル タイプ（表示モデル タイプ フラグが true に設定されている）
- 現在ロードされている開発者 ID によって作成され、検索パターンに一致した、すべての非表示モデル タイプ（表示モデル タイプ フラグが false に設定されている）

-lists

複数の値を許可する属性のみをレポートに含めます。

-nolists

複数の値を許可する属性をレポートに含めません。

-groups

モデル タイプ グループをレポートに含めます。

-help

コマンドの使用方法に関する情報を表示します。

詳細情報:

モデル タイプ レポートの実行 (P. 48)

モデル タイプ レポートの実行

モデル タイプ レポートは、特定のモデル タイプの属性情報をリストします。 モデル タイプ レポートを実行するには、以下の構文を使用します。

```
../SS-Tools/ ¥
reports [-mtype <model type>][-attrflags w][-fields][-invisible]
```

または

```
../SS-Tools/ ¥
reports [-handle <model handle>][-attrflags “”][-fields][-invisible]
```

例として、次のコマンドは、HUB_CSI_IRM2 モデル タイプについてのレポートを生成します。

```
reports -mtype HUB_CSI_IRM2 -attrflags e -fields don -invisible
```

レポートには、外部フラグが設定されているすべての属性がリストされます。 含まれる各属性について、OID 参照、OID プレフィックス、および名前（開発者 ID 付き）が示されます。 また、レポートには、指定したモデル タイプのベース モデル タイプ（親モデル タイプ）の全リストが含まれます。ベース モデル タイプは継承の順にリスト表示されます。

レポートは、ワークステーションの標準出力デバイスに送信されます。

詳細情報:

レポートの出力デバイスの指示 (P. 49)

レポートの出力デバイスの指示

reports ユーティリティは、ワークステーションの標準出力デバイスに出力を送信します。通常は画面に表示されます。 ファイルまたはプリンタに出力を直接リダイレクトするには、UNIX のコマンド ラインで標準パイプ コマンドを使用します。

例 1

次のコマンドは、HUB_CSI_IRBM モデル タイプについてのレポートを生成します。

```
reports -mtype HUB_CSI_IRBM -attrflags e -fields n
```

このレポートには、モデル タイプの開発者、その名前およびハンドル、6 つの属性フラグの状態、および直接のベース モデル タイプの ID を示す、標準的なヘッダ セクションが含まれます。 レポートの残りの部分は外部フラグが設定された属性に制限され、レポートの細目では属性名のみがリストされます。

レポートは単に一度だけの表示としてワークステーション画面に表示されます。 レポートが長すぎる場合、レポートの最後だけが表示されます（ディスプレイが表示できるテキストの最大行）。

例 2

次のコマンドは同じレポートを生成しますが、レポートを増分表示ファイルとしてワークステーション画面に送信し、最初の画面が表示されます。

```
reports -mtype HUB_CSI_IRBM -attrflags e -fields n | more
```

Return キーを押すとその画面を表示しながら連続する行をインクリメントでき、スペース キーを押すと次の画面を表示できます。

例 3

次のコマンドは同じレポートを生成し、現在のディレクトリの REPORT_1 という名前の ASCII ファイルにそれを書き込みます。

```
reports -mtype HUB_CSI_IRBM -attrflags e -fields n > REPORT_1
```

指定した名前のファイルが現在のディレクトリに存在する場合は、新しいレポートによって上書きされます。

前にディレクトリ パスを付けてレポート ファイル名を指定できます。

例 4

次のコマンドは同じレポートを生成し、<ptr> で指定するプリンタに印刷ファイルとして送信します。

```
reports -mtype HUB_CSI_IRBM -attrflags e -fields n > lpr -P<ptr>
```

関係レポートの実行

関係レポートには、選択した関係に従ってモデル タイプ間の関係がリストされます。 関係レポートを実行するには、以下の構文を使用します。

```
../SS-Tools/reports -rel <relation>
```

<relation> は、 レポートに含める関係の名前です。 一度に 1 つの関係を指定します。

例として、次のコマンドは、関係 Connects_to についてのレポートを生成します。

```
../SS-Tools/reports -rel Connects_to
```

レポートは、Connects_to 関係に含まれる各シーケンシャル ルールをリストし、ワークステーションの標準出力デバイスに送信されます。 レポートをファイルに書き込む方法、またはプリンタに送信する方法については、「レポートの出力デバイスの指示 (P. 49)」を参照してください。

詳細情報:

レポートの出力デバイスの指示 (P. 49)

SSdbload

$SPECROOT/SS-Tools ディレクトリにあるこのユーティリティ プログラムを使用すると、以前に作成したバックアップ ファイルでの SpectroSERVER データベースのリストア、間髪者 ID 情報のロード、フォールト トレラント環境での SpectroSERVER の優先順位値の設定を行うことができます。

注: フォールト トレランスの設定の詳細については、「分散 SpectroSERVER 管理者ガイド」を参照してください。

SSdbload を実行する前に、SpectroSERVER と、SpectroSERVER データベースにアクセスする他のプログラムをすべて、シャットダウンする必要があります。

このユーティリティの形式は以下のとおりです。

```
SSdbload [-quiet] [-initialize] [-developer <DEV_INFO_FILE>] [-load] [-models]
[-catalog] [-replace <PRECEDENCE>] [-add <PRECEDENCE>] [-port <PORT_NO>] [-showmap]
[-version]
[-extension] [-new_primary <NEWHOSTNAME>]
[<SAVE_FILE>]
```

引数名の最初の文字が太字で示されている場合は、文字列全体を入力するのではなく、その文字だけを使用してもかまいません。

-quiet

プロンプト（対話型モード）を無効にします。 スクリプトの内部からロード コマンドを実行する場合に役立ちます。

-initialize

モデリング カタログ、すべてのモデル、およびアーカイブされていないすべてのイベントと統計ログ レコードを削除することにより、データベースを初期化します。

重要: -i （initialize）オプションを使用する場合は、少なくともカタログをリストアする必要があります。 分割バックアップを使用する場合は、ロードするバックアップ ファイルごとに SSdbload を 1 回実行する必要があります。 -i オプションを、SSdbload の最初の実行と一緒に、1 回のみ特に使用する必要があります。モデルの前にモデリング カタログをロードします。

-developer

<DEV_INFO_FILE> 変数を使用して指定する開発者情報ファイルをロードします。

注: ただ 1 つのデータベースに開発者情報をロードし、そのデータベースでモデリング カタログのすべての編集を実行します。同じ開発者情報を複数のデータベースで使用すると、モデル タイプ、属性、または関係ハンドルが重複する可能性があります。

-d オプションは開発者情報をロードするために使用できますが、1 回だけです。 -d オプションを使用して 2 回目の SSdbload を実行すると、エラー メッセージが表示され、開発者情報は変更されません。

-load

<SAVE_FILE> 変数を使用して指定した保存ファイルのオブジェクトでデータベースをロードします。

重要: -l （load）オプションは、既存のデータベースにモデルまたはモデリング カタログ コンポーネントを追加するようには設計されていません。 データベースの内容を追加するのではなく、保存ファイルに含まれるものと同じタイプの既存情報をすべて削除した後、保存ファイルの内容をロードします。たとえば、保存ファイルが SSdbsave -m オプションを使用して作成され、モデルのみを含む（モデリング カタログを含まない）場合は、SSdbload -l オプションは既存のモデリング カタログに影響を与えません。 ただし、既存のモデルはすべて削除し、保存ファイルに含まれるモデルに置換します。

-models

<SAVE_FILE> 変数を使用して指定した保存ファイルからのモデルをロードします。

-catalog

指定した保存ファイルからモデリング カタログ（モデル タイプ、関係、ルール）をロードします。

-replace

特定の SpectroSERVER に現在割り当てられている優先順位値を <PRECEDENCE> 変数で指定する新しい値に置き換えるために、フォールト トレラント環境でのみ使用されます。

-add

<PRECEDENCE> 変数を使用して特定の SpectroSERVER に優先順位値を割り当てるために、フォールト トレラント環境でのみ使用されます。

-port

SpectroSERVER のポート番号を指定するために、-add または -replace 引数で使用します。<PORT_NO> 変数を使用して特定のポートを指定しないと、.vnmrc ファイルで comm.port リソースに対して指定されているポートがデフォルトで使用されます。

-showmap

ランドスケープがロードされているサーバーおよび優先順位レベルを示すランドスケープ マップ情報を表示します。

-version

SSdbload のバージョン、およびロードしようとしている保存ファイルのバージョンを表示します。 これらのバージョンに互換性がない場合、エラー メッセージが表示され（クワイエット モードで作動していても）、ファイルはロードされません。

-extension

ファイル拡張子の適用を無効にします。

-new_primary

別のランドスケープで保存されたデータベースをロードするときに使用する必要があります。 データベースをロードしているホスト名を使用し、<NEWHOSTNAME> 変数を使用してランドスケープの名前を指定します。 そうしないと、もともと保存された SpectroSERVER のホスト名がランドスケープに与えられます。

<SAVE_FILE>

ロードするバックアップ ファイルの名前。 バックアップは、SSdbsave ユーティリティまたはオンライン バックアップ機能のいずれかを使用して保存されたファイルです。 また、CA Spectrum インストール プログラムによって作成された保存ファイルの 1 つでもかまいません。

このプログラムの実行が成功するたびに、2 つの保存ファイルが *$SPECROOT*/SS ディレクトリに作成され、それぞれにはインストールされているモデリング カタログのコピーが含まれます。 1 番目は .after 拡張子の日付がスタンプされたファイルです。 その後、.after ファイルのコピーが「legacy.SSdb」という名前で作成され、既存の legacy.SSdb ファイルは上書きされます。

最も新しくインストールされたモデリング カタログでデータベースを再初期化するには、legacy.SSdb ファイルを使用します。 特定のインストールと関連付けられたカタログをリストアするには、必要に応じて .after ファイルを使用します。

.after ファイル名の日付スタンプの後にあるシーケンシャル カウンタにより、たとえば、同じ日に作成された複数のファイルを区別できます。

```
db_20001014,1.after.ssdb
db_20001014,2.after.SSdb
db_20001014,3.after.SSdb
```

データベースの初期化に使用する .after ファイルが現在の実行可能ファイルより古い場合、予期される情報がデータベース内にない可能性があり、OneClick によってアクセスできません。

例

データベースを初期化し、「db_950318_cm」という名前の以前に保存したファイルからモデリング カタログ（モデル タイプおよび関係）およびモデル情報をロードするには、以下のコマンドを使用します。

```
SSdbload -il db_950318_cm
```

エラー

Can't open database.

SpectroSERVER データベースが現在のディレクトリに存在しません。

Database already locked by: *<user id>*
by process: *<process name>*
with process ID: *<process id>*
on network node:*<node name>*
which started at: *<date/time stamp>*

データベースは別のプロセスによってロックされています。

保存ファイルのバージョン: <version_number>.
SSdbload **のバージョン**: <version_number>。**この保存ファイルは** SSdbload **の現在のバージョンではロードできません。**.
この保存ファイルは、CA Spectrum **のバージョン** <version_number> **ではロードできません。バージョン** <version_number> **として保存されました。**

バージョンの非互換性が検出された場合、SSdbload では保存ファイルをロードさせません。 このエラー メッセージは、［-version］ 引数を指定した場合に発生します。 -quiet オプションを指定しても、適切な場合は表示されます。

この保存ファイルは、CA Spectrum **のバージョン** <version_number> **ではロードできません。バージョン** <version_number> **として保存されました。**

バージョンの非互換性が検出された場合、SSdbload では保存ファイルをロードさせません。 このエラー メッセージは、［-version］ 引数を指定した場合に発生します。 -quiet オプションを指定しても、適切な場合は表示されます。

注: 致命的ではない属性記述子エラーがロード中に検出された場合、SSdbload はデータベースが含まれるディレクトリに SSdbload.log という名前のログ ファイルを作成します。 このファイルの致命的ではない属性エントリの例を以下に示します。

重要: <SSdbload path>/SSdbload は、モデル 400000 の 属性 10004 を読み取れません

詳細情報:

データベース ロックの削除 (P. 24)

ホスト名の変更

1 つの SpectroSERVER ホストのホスト名を変更する場合、変更する前に SSdbsave を使用し、後で SSdbload を使用する必要はありません。 データベースは新しいホスト名を反映するように自動的に変更され、SpectroSERVER を再起動すると以下のメッセージがコントロール パネルに表示されます。

```
This database was previously loaded on <old hostname> port <old port number>, but is
now being loaded on <new hostnames> port <new port number>.
```

ただし、1 つ以上のバックアップ SpectroSERVER が含まれるフォールト トレラント環境では、サーバは、ホスト名と関連付けられている*優先順位*値によって相互の関係を認識します。 そのため、フォールト トレラント関係を維持するには、SSdbsave と SSdbload を次の順序で使用して、プライマリ（またはセカンダリ） SpectroSERVER のホスト名を変更する必要があります。

1. -cm オプションを付けて SSdbsave を使用し、データベースを保存します。
2. ホスト名を変更します。
3. SSdbload と -il および -replace オプションを実行することにより、ステップ 1 で作成した保存ファイルでデータベースを再ロードします。 再ロードを行うと、データベースは新しいホスト名を既存の優先順位値と関連付けることができます。

   ```
   SSdbload -il -replace <precedence> <save file>
   ```

詳細情報:

SSdbsave (P. 56)
SSdbload (P. 51)

SSdbsave

$SPECROOT/SS-Tools ディレクトリにあるこのユーティリティ プログラムを使用すると、既存の SpectroSERVER データベースのモデリング カタログ（モデル タイプおよび関係）および（または）実際のモデルとそれには含まれる関連データの、バックアップ コピーを作成できます。

SSdbsave を実行する前に、SpectroSERVER と、SpectroSERVER データベースにアクセスする他のプログラムをすべて、シャットダウンする必要があります。

このユーティリティの形式は以下のとおりです。

```
SSdbsave [-quiet] [-extension] [-version]
[-catalog] [-models] <SAVE_FILE>
```

引数名の最初の文字が太字で示されている場合は、名前全体を入力するのではなく、その文字だけを使用してもかまいません。

-quiet

対話型/冗長モードを無効にします。

-extension

ファイル拡張子の適用を無効にします。

-version

SSdbsave のバージョンを表示します。バージョン番号は保存ファイルに含まれます。 ファイルを保存するときにこの引数を使用した場合、保存されているファイルをロードするために使用できる SSdbload のバージョンを示すメッセージが表示されます。

-models

保存ファイル内のモデルを含めます。

-catalog

保存ファイルにモデリング カタログ（モデル タイプ、関係、ルール）を含めます。

<SAVE_FILE>

保存されるデータベースの保存先ファイルの名前を指定します。 保存ファイルの推奨される命名規則は、日付スタンプと保存に使用されるオプション フラグを組み合わせたものです。 たとえば、db_20121014_cm というファイル名は、2012 年 10 月 14 日に「c」（カタログ）オプションと「m」（モデル）オプションの両方を使用して実行されたバックアップを示します。

サンプル出力

次に示すのは、-models および -catalogs 引数を使用して SSdbsave を実行すると生成される出力の例です（../SS-Tools/SSdbsave -mc Mar16_2013DB）。保存されるデータベース ファイルの名前は Mar16_2013DB です。 この出力の最後の行は、この保存ファイルをロードするために使用できる SSdbload のバージョン番号を示します。

```
Number of Model Types saved: 3493
Number of Relations saved: 92
Number of Models saved: 103
SSdbsave has successfully saved the database model and catalog information as
'Mar16_2013DB.SSdb'.
This file can be loaded with version 7.0.0.000 of SSdbload.
```

エラー

Can't open database

SpectroSERVER データベースが現在のディレクトリに存在しません。

Database already locked by: *<user id>*
by process: *<process name>*
with process ID: *<process id>*
on network node: *<node name>*
which started at: *<date/time stamp>*

データベースは別のプロセスによってロックされています。

SSdbsave: Warning. Expected attr: 0x999999 not found, for model: 0x999999, of Model Type: 0x999999 Cs Whatever MT. Processing continues

対応する属性記述子がないモデル属性値を保存しようとすると、このメッセージが表示されます。ユーザまたは開発者が属性を削除した場合、この情報メッセージが表示されます。

詳細情報:

データベース ロックの削除 (P. 24)
SSdbload (P. 51)

データベース モデル変更ツール（DBconv）

$SPECROOT/SS-Tools ディレクトリにあるこのユーティリティ プログラムを使用すると、CA Spectrum データベース内の一連のモデルのモデル タイプを変換できます。 DBconv は、アプリケーションを再発見し、モデルのセット内のインターフェースを再設定するためにも使用できます。

このユーティリティの形式は以下のとおりです。

```
DBconv [-file=]<Input File Name>
[-src_mth=]<Source Model Type Handle>
[-dest_mth=]<Destination Model Type Handle>
[-landscape=]<Landscape Handle>
[-rediscover=]原因：<d><i>
[-all_landscapes] [-test=]<Test Level> [-quiet] [-debug]
```

DBconv と設定

DBconv は、入力ファイル（コマンド ラインで指定された）またはコマンド ライン自体からその設定を受け取ることができます。 コマンド ライン オプションのリストを取得するには、オプションなしで DBconv を実行します。

-file

使用する入力ファイルを指定します。

-src_mth

変換前のモデル タイプ ハンドルを指定します。-rediscover を指定する場合は、再発見するモデルのタイプを指定します。

-dest_mth

変換後のモデル タイプ ハンドルを指定します。 入力ファイル内の設定を無視します。 -rediscover を指定すると、このオプションは無視されます。

-landscape

モデルを検索するランドスケープ ハンドルを指定します。

デフォルト： 0x400000

-all_landscapes

すべてのランドスケープでモデルを検索することを指定します。

-test

テスト レベル 0 （デフォルト）は、変換を実際に行うことを意味します。 テスト レベル 1 は、DBconv がコマンド ラインおよび入力ファイルを検証した直後に処理を停止することを意味します。テスト レベル 2 は、古いモデルを削除せず、新しいモデルを作成しないことを意味します。

-quiet

エラー メッセージ以外の出力を生成しないことを指定します。

-debug

余分な出力があることを指定します。

-rediscover

このオプションを指定すると、検出されたモデルは変換されません。代わりに、以下で指定するアクションの 1 つ以上が、検出された各モデルに対して実行されます。

d

検出された各デバイス モデルのアプリケーション モデルをすべて破壊します。

r

検出された各デバイス モデルにアプリケーション再発見アクションを送信します。

i

検出された各デバイス モデルにインターフェース再設定アクションを送信します。

注: コマンド ラインでは、上記のアクションを任意の順序で指定できます。ただし、常にここで示した順序で実行されます。

詳細情報:

DBconv コマンド ライン使用の例 (P. 61)
入力ファイルでの DBconv の使用 (P. 61)

DBconv コマンド ライン使用の例

次に示すのは、コマンド ラインでの DBconv ツールの標準的な使用法の例です。

- 指定した入力ファイルのみを使用して、変換を実行します。

  ```
  $ DBconv -file=config.dbc
  ```

- すべての既知のランドスケープで、すべての GnSNMPDev モデルを Smart Switch Routers に変換します。

  ```
  $ DBconv -src_mth=0x3d0002 -dest_mth=0x2c60000 -all_landscapes
  ```

- ランドスケープ 0x400000 で、すべての GnSNMPDev モデルを Smart Switch Router に変換します。

  ```
  $ DBconv -src_mth=0x3d0002 -dest_mth=0x2c60000 -landscape=0x400000
  ```

- すべてのランドスケープのすべての SmartSwitch Router について、アプリケーションをすべて破壊し、アプリケーションを再発見し、インターフェースを再設定します。

  ```
  $ DBconv -src_mth=0x2c60000 -rediscover=rdi -all_landscapes
  ```

- ランドスケープ 0x400000 内のすべての SmartSwitch Router について、インターフェースを再設定します。

  ```
  $ DBconv -src_mth=0x2c60000 -rediscover=i -landscape=0x400000
  ```

注: DBconv の実行には、短い初期フェーズとそれより長い第 2 フェーズがあり、どのフェーズが有効かがウィンドウに表示されます。 接続できないモデルは変換されません。いずれの場合も、エラー メッセージが表示されます。 これらのエラー メッセージのいくつかの考えられる原因は次のとおりです。

- 指定されたモデル タイプによって表されるデバイスへの接続が失われています
- 指定されたモデル タイプがデータベースに存在しません
- SpectroSERVER が応答していません
- SpectroSERVER にはユーザのユーザ ID に対するモデルがありません

入力ファイルでの DBconv の使用

DBconv は入力ファイルがあってもなくても使用できます。 入力ファイルを使用する場合は、出発点として *<$SPECROOT>*/SS-Tools ディレクトリにある template.dbc ファイルを使用できます。

入力ファイルを構成する複数のセクションを使用すると、コマンド ライン引数より正確に変換をカスタマイズできます。 入力ファイルは複数のセクションで構成され、Configuration セクション以外はすべてオプションです。 ここで指定するセクションの形式および大文字と小文字の使い分けを使用します。 空白行および「#」文字（シャープ記号）で始まる行は無視されます。

DBconv 入力ファイルの要素

以下のセクションでは、DBconv で使用される入力ファイルの必要およびオプションの内容について説明します。

Configuration（必須）

このセクションには、以下で説明する単一の構成アイテムが含まれます。

注: 入力ファイルでは、Configuration セクションの Source_Model_Type、Destination_Model_Type、Landscape_Handle フィールドのみが必須です。その他のすべてのフィールドはオプションです。

例:

```
Configuration {
Source_Model_Type = 0x3d0002
Destination_Model_Type = 0x2c60000
Reconnect_Sleep_Time = 90
Dont_Change_Discovery_Attributes = true
Models_To_Convert = 0
Landscape_Handle = all
Is_Obsolete_Model_Type = false
Relation_Section_Ignores = false
}
```

- Source_Model_Type = <古いモデル タイプ ハンドル>

  変換前のモデル タイプ ハンドル。 16 進で記述し、「0x」を前に付ける必要があります。

- Destination_Model_Type = <新しいモデル タイプ ハンドル>

  変換後のモデル タイプ ハンドル。 16 進で記述し、「0x」を前に付ける必要があります。

- Landscape_Handle = <ランドスケープ ハンドル>

  変換を行うランドスケープを指定します。 「all」を指定すると、DBconv はエンタープライズ モードに切り替わり、すべてのランドスケープのモデル タイプを変換します。 「selection」を指定すると、Landscapes { } セクションで指定されているすべてのランドスケープが使用されます。

- Models_To_Convert = <変換する最大モデル数>

  この行に 0 より大きい値を指定すると、指定した数の古いタイプのモデルだけが変換されます。 値が 0 の場合は、すべてのモデルが変換されます。

- Reconnect_Sleep_Time = <スリープ時間>

  Model_State がアクティブに変化してからポートの再接続が試行されるまでの、スリープ期間を秒単位で設定できます。 デフォルト値は 60 秒です。

- Reconnect_Interval = <秒数>

  Reconnect_Sleep_Time が経過した後、DBconv はインターフェースとポートの再接続を試行します。 この時間の後でモデルがまだアクティブにならない場合（一部のデバイス モデルは、すべてのモジュールとアプリケーションが作成されるまでアクティブになりません）、DBconv は指定された時間だけ待ってから再度試行します。 デフォルト値は 30 秒です。

- Reconnect_Interval_Count = <回数>

  このオプションは、DBConv がインターフェースとポートへの再接続を試行する回数を制御します。 デフォルト値は 30 回です。

- Dont_Change_Discovery_Attributes = <true/false>

  これが false である場合、変換プログラムは古いモデル タイプの検出関連の属性を変更します。 これは、検出に古いモデル タイプではなく新しいモデル タイプを使用させるためです。 (a) 古いモデル タイプが GnSNMPDev である場合、または (b) 古いモデル タイプと新しいモデル タイプが同じ場合、このデフォルト値は true になります。 デフォルト値は False です。

- Is_Obsolete_Model_Type = <true/false>

  このスイッチは、モデル タイプの廃止フラグが設定されているモデルを変換します（これを設定すると、どの属性も読み取れません）。 この場合、変換の間に属性の読み取り/検証/書き込みは行われません。 インターフェースまたはポートはアクセスされません。 ただし、モデルの関係は処理されます。 これは、デバイス モデルがこのフラグで変換された場合、名前や IP アドレスなどがない新しいモデルが作成されることを意味します。 デフォルト値は False です。

- Relation_Section_Ignores = <true/false>

  このスイッチは、Left_Relationships および Right_Relationship セクションを解釈します。 値が false の場合、Left/Right セクションで指定されている関連付けのみが、デバイス モデルにリストアされます（Left/Right セクションがどちらも空の場合はなし）。 値が true の場合、指定されたものを除くすべての関連付けがリストアされます。 これはデバイス モデルの左の CONNECTS_TO には影響しません。 DBconv は、デバイスがモデル化されるビューを維持するため、このリストアを常に試みます。

- Convert_Scm_Configs = <true/false>

  このスイッチは、SCM の設定がモデル タイプと共に変換されるかどうかを決定します。 これを true に設定すると、変換対象のモデル/モデル タイプの代わりであった SCM 設定も変換されます。 共有 SCM 設定は変換後のモデル タイプに変換され、非共有 SCM 設定は変換後のモデル ハンドルおよびモデル タイプに変換されます。

  注: ホスト SCM 設定は有効に転送されますが、属性設定は変換中に失われる可能性があります。

- Convert_Sanm_Policies = <true/false>

  このスイッチは、モデル タイプと共に SANM ポリシーを変換するかどうかを決定します。 true に設定すると、ポリシーにおける変換前モデル タイプに対するすべての参照は、変換後モデル タイプ名に変換されます。

- ランドスケープ

  変換するランドスケープのリスト。 DBconv がこのセクションに注目するためには、Configuration セクションの Landscape_Handle エントリを「all」に設定する必要があります。

  例：

  ```
  Landscapes {
  0x400000
  0x80c00000
  }
  ```

- Model_Handles

  変換するモデルのリスト。選択したモデルのみを変換する場合は、そのモデル ハンドルをこのセクションに入力します。 ほかのモデルはすべて無視されます。

  例：

  ```
  Model_Handles {
  0x80c00be6
  0x80c00be8
  }
  ```

- Transfer_Attributes

  古いモデルから新しいモデルに転送する属性のリスト。 これらの属性は古いモデル タイプと同様に新しいモデル タイプでも有効であることを確認してください。 Network Address および Community Name 属性は常に転送されるので、このセクションで指定する必要はありません。

  例：

  ```
  Transfer_Attributes {
  0x001006e
  0x00010024
  }
  ```

- Transpose_Attributes

  古いモデルから新しいモデルに転換する属性のリスト。 これらの属性は古いモデル タイプと同様に新しいモデル タイプでも有効であることを確認してください。 古い属性と新しい属性は同じタイプである*必要があります*。

  例：

  ```
  Transpose_Attributes {
  0x001006e = 0x001884
  0x00010024 = 0x777533
  }
  ```

- Set_Attributes

  データを強要する属性のリスト。これらの属性が有効であり、データが属性のタイプに対して有効であることを確認する必要があります。

  例：

  ```
  Set_Attributes {
  0x001006e = Router 42
  0x00010024 = public
  }
  ```

- Transfer_Port_Attributes

  古いモデルのポートから新しいモデルのポートに保存する属性のリスト。

  例：

  ```
  Transfer_Port_Attributes {
  0x00011564
  }
  ```

- Right_Relationships

  右側の古いモデルとの関連付けが無視されるか含まれる関係のリスト（詳細についてはこのセクションの「**Relation_Section_Ignores**」スイッチを参照）。

  例：

  ```
  Right_Relationships {
  0x10004
  0x230000
  }
  ```

- Left_Relationships

  左側の古いモデルとの関連付けが無視されるか含まれる関係のリスト（詳細についてはこのセクションの「**Relation_Section_Ignores**」スイッチを参照）。

  例：

  ```
  Left_Relationships {
  0x10004
  0x230000
  }
  ```

- Object_ID_Exists

  各ターゲット デバイスに存在する必要がある MIB オブジェクトのリスト。 すべての MIB オブジェクトが存在し、読み取り可能な場合、ターゲット デバイスは変換できます。

  例：

  ```
  Object_ID_Exists {
  1.3.6.1.4.1.52.2501.1.270.4.1.1.5
  1.3.6.1.4.1.52.2501.1.1.5
  }
  ```

- Filter_Attributes

  モデルの選択に使用する属性のリスト。 検出された古いモデルの attribute_id の attribute_value が Filter_Attributes 行で指定されている値と等しい場合、そのモデルは選択されます。 text_string 属性の場合、attribute_value は古いモデルから返される文字列の部分文字列であることだけが必要です。

  例：

  ```
  Filter_Attributes {
  0x10053 = 1.3.6.1.4.1.49.2.3.5
  0x10b5a = Network Administrator
  0x1154f = false
  }
  ```

- Advanced_Transfer_Attributes

  既知の関係パスによってデバイス モデルに関連付けられている任意の数のモデルからの属性の転送を有効にします。 任意の数のこれらのセクションを、入力ファイルに含めることができます。

  構文：

  ```
  Advanced_Transfer_Attributes {
  Path = <Relation Path>
  Identifier = <Identifier Attribute>
  If <Attribute ID> = <Value>
  If ModelType = <Model Type ID>
  Transfer = <Attribute to transfer>
  Transpose <Dest Attribute ID> = <Src Attribute ID>
  }
  ```

  説明：

  Path = *<Relation Path> [Mandatory]*

  転送するモデルの検索方法を指定します。 関係パスでは、「.」で区切った関係名のセットを指定します。 関係パスの要素数に制限はありません。

例：

Path = HASPART

デバイス モデルに対する HASPART 関係の右側のモデルをすべて検索します（つまり、すべてのインターフェース）。

Path = Contains.HASPART

Contains 関係の右側のモデルをすべてスキャンします。 これらの各モデルについて、HASPART 関係の右側でモデルをスキャンし、これらの発見されたモデルに対して転送を行います。

- Identifier = <Identifier Attribute> [Mandatory]

  関係パスによって指定されたモデルのセットが見つかったときは、モデル ハンドルが変わるので、これらのモデルを変換されたデータベースでのモデルに一致させる方法が必要です。 DBconv は、Identifier によって指定された属性の値を読み取ります。 その後、変換されたモデルとこの値を一致させようとします。 一致が見つかる場合、DBconv はこのモデルが変換前のモデルに相当すると考えます。 このエントリの共通の属性は Component_OID です。

  例：

  `Identifier = 0x1006a`

  If <Attribute ID> = <Value>

  これはフィルタ エントリです。 <Attribute ID> の値が <Value> と等しい場合、指定された属性だけが転送または転換されます。

  例：

  以下の 3 つの行は、ifType が 6 と等しいときに限り、Component_OID によって識別されるすべてのインターフェースと一致します。

  Path = HASPART

  Identifier = 0x1006a

  If 0x1134c = 6

  If ModelType = <Model Type ID>

  モデル タイプが <Model Type ID> と一致する場合に、指定された属性だけを転送または転換するフィルタ エントリ。

例：

以下は、RS-8000 の SSR_PortIf インターフェースすべてに一致します。

Path = HASPART

Identifier = 0x1006a

If ModelType = 0x2c60006

- Transfer = <Attribute to transfer>

  転送される属性を指定します。エントリの数に制限はありません。1 つの Transfer エントリまたは 1 つの Transpose エントリのいずれかが必要です。

  例：

  Transfer = 0x11564

- Transpose <Dest Attribute ID> = <Src Attribute ID>

  転換される属性を指定します。<Src Attribute ID> で指定されている値を取得し、<Dest Attribute ID> に書き込みます。 これらの属性のタイプは同じである必要があります。1 つの Transfer エントリまたは 1 つの Transpose エントリのいずれかが必要です。

  例：

  次の例は、RS-8000 の VLAN 名をコピーし、それを Notes フィールドに書き込みます。

  Transpose 0x11564 = 0x2c604dd

  次のエントリは、RS-8000 のすべての SS-PortIf モデルから Notes と PollingStatus をコピーします。

```
Advanced_Transfer_Attributes {
Path = HASPART
Identifier = 0x1006a
If ModelType = 0x2c60006
Transfer = 0x11564
Transfer = 0x1154f
}
```

- Transfer_Notes

  指定された領域からのノート フィールドのコピーを有効にします。各行は、**Advanced_Transfer_Attributes** （このセクションで前述）として変換されます。

  例：

  ```
  Transfer_Notes {
  Device
  Interfaces
  Applications
  Modules
  Ports
  }
  ```

  指定したクラス（デバイス、インターフェースなど）のすべてのモデルのノート フィールドをコピーするように、1 つ以上のフィールドを指定できます。

第 3 章: DDM データベースの保守

このセクションには、以下のトピックが含まれています。

データベース セキュリティ (P. 71)
データベース サイズの管理 (P. 72)
データベース バックアップ (P. 73)
データベースのリストア (P. 77)
データベース保守と最適化 (P. 79)
DDM データベース クエリ (P. 81)
その他のデータベース ユーティリティ (P. 83)
アーカイブ マネージャ リソース (P. 83)
DDM およびアーカイブ マネージャのトラブルシューティング (P. 85)

データベース セキュリティ

分散データ マネージャ データベースのセキュリティを強化するには、CA Spectrum のインストール後に、以下のいずれかのプラットフォーム固有手順を実行します。

Windows の場合:

1. 管理者としてログインします。
2. Windows エクスプローラを使用して、*$SPECROOT*/SS/DDM ディレクトリに移動します。
3. .configrc プログラムを右クリックし、［プロパティ］を選択します。
4. ［セキュリティ］タブをクリックし、［アクセス許可］パネルを確認して、［詳細設定］ボタンをクリックします。
5. ［Administrator］または［CA Spectrum Users］グループだけがリストに表示されることを確認します。

   ［Everyone］または他のユーザやグループのエントリが表示される場合は削除します。親からの子オブジェクトへのアクセス許可の継承を可能にする設定をクリアすることが必要な場合があります。この設定はアクセス許可の［詳細設定］オプションにあります。

Solaris の場合:

1. root としてログインします。
2. *$SPECROOT*/SS/DDM ディレクトリに移動します。
3. 以下のコマンドを入力します。

```
chmod 600 .configrc
```

データベース サイズの管理

DDM データベースのサイズの管理は、データベースの効率的な実行に役立つ可能性があります。 DDM データベースのサイズを制御するには、イベントおよび統計ログ記録をネットワークの重要なコンポーネントに制限できます。 データベース サイズを管理するには、以下の手順に従います。

- 統計のログ記録を無効にすることにより、ログに記録されるデータを制限します。 .vnmrc ファイルで stats_logging_enabled パラメータを FALSE に設定します。

  注:

  - stats_logging_enabled のデフォルト値は FALSE です。 ただし、前のリリースからアップグレードした場合は、この設定を確認してください。
  - .vnmrc ファイルの詳細については、「分散 SpectroSERVER 管理者ガイド」を参照してください。

- 保存されるイベント データの量を制限します。履歴が重大ではないすべてのイベントについて、EventDisp ファイルで「E」ログ記録フラグを無効にします。

  注: 詳細については、「Event Configuration User Guide」を参照してください。

- 積極的に監視していないモデルのログ記録を無効にするか減らします。モデル情報ビューで［ログ率］（1 つのログ当たりのポーリングの数）をゼロに設定することにより、個別のモデルでのログを無効にできます。 または、属性エディタを使用して、特定のタイプのすべてのモデルのログ記録を無効にします。たとえば、ユーザ ポートのトラフィック統計をレポートすることに興味がない場合は、すべてのユーザ ポートで［ログごとのポーリング回数］をゼロに設定します。

  ログ記録頻度を減らすには、これらの方法のどちらかを使用して、ログ率を増やします。

  注: 詳細については、「IT インフラストラクチャのモデリング/管理 - 管理者ガイド」を参照してください。

データベース バックアップ

SpectroSERVER データベースと同様に、DDM データベースでも、ハードウェアまたはオペレーティング システムの障害の場合に完全なリカバリを可能にするために定期的なバックアップが必要です。 理想的には、どちらのデータベースも毎日バックアップし、最新のデータの可用性を保証し、かつデータベースの同期を維持します。

ただし、通常は DDM データベースの方が SpectroSERVER データベースよりはるかに大きく、バックアップにより多くの時間を要します。 また、バックアップ中はアーカイブ マネージャをシャットダウンする必要があります。 その結果、再起動時には、普通、履歴データの大量のバックログを処理する必要があります。バックログ処理は SpectroSERVER のパフォーマンスに悪影響を及ぼすことがあります。 これらの理由で、通常は、DDM データベースは週単位でバックアップし、SpectroSERVER データベースはより頻繁に（できれば毎日）バックアップすることを推奨します。

注: DDM データベースと SpectroSERVER データベースは、一方をリストアして他方をリストアしないと、同期しなくなる場合があります。 たとえば、日曜日に両方のデータベースをバックアップします。 月曜日にモデルを作成した後、システム クラッシュが発生し、日曜日の保存ファイルで SS データベースをリストアします。リストアされた SS データベースには作成したモデルが含まれなくなりますが、そのモデルの履歴レコードは DDM データベースに存在します。 さらに、モデルを再作成する場合、再作成されたモデルに対して生成された新しいレコードで元のモデルからの DDM レコードを調整できません。SpectroSERVER データベースをリストアする場合は常に、DDM データベースを常にリストアすることにより、この種の競合を回避します。

DDM データベースのバックアップは、ddm_save (P. 75) または ddm_backup (P. 76) ツールを使用していつでも実行できます。ただし、オンライン バックアップ機能を使用して、自動的に DDM データベースをバックアップすることもできます。オンライン バックアップは post_olb_script を使用します。SpectroSERVER データベースと DDM データベースの同期を維持することが重要なので、DDM のバックアップにはこのスクリプトが推奨される方法です。

重要: DDM データベースをバックアップする方法としてサポートされるのは、このセクションで説明するツール（post_olb_script、ddm_backup、ddm_save）だけです。 サードパーティのバックアップ ソフトウェアを使用すると、データベースが破損する可能性があります。

post_olb_script

このスクリプト ファイルを使用すると、オンライン バックアップを使用して実行される SpectroSERVER データベース バックアップと、カスタム スクリプトの実行を調整できます。 このスクリプトは <*$SPECROOT*>/SG-Support/CsScript ディレクトリにあります。デフォルト設定で実行した場合、post_olb_script は、オンライン バックアップが指定された曜日（デフォルトは日曜日）に完了した直後に DDM データベースをバックアップします。

post_olb_script ファイルをデフォルト設定で実行するには、ファイルを *$SPECROOT*/SG-Support/CsScript から *$SPECROOT*/custom/CsScript にコピーします。 テキスト エディタを使用し、ファイルの最後の 11 行の先頭からシャープ記号 (#) を削除することによってアクティブにします。DDM データベースのバックアップは、日曜日（target day=0）に実行されるオンライン バックアップの後にスケジュールされます。 したがって、オンライン バックアップを毎日実行するようにスケジュールした場合、DDM データベースは自動的に週単位でバックアップされます。

注: CA Spectrum アップグレード インストール中に、コピー操作が自動的に実行されます。 デフォルト設定に戻らないように、コピーの前にファイルがすでに存在しないことを確認してください。

より頻繁な DDM データベース バックアップを必要とする場合は、post_olb_script ファイルの最後の 5 行のみのコメントを外します。「target day」は指定されません。 したがって、DDM データベースは、オンライン バックアップを使用する SpectroSERVER データベースのバックアップのたびにバックアップされます。

詳細情報:

オンライン バックアップ (P. 12)

ddm_save

$SPECROOT/SS/DDM ディレクトリにある ddm_save ユーティリティは、指定するファイルへの履歴データベースの完全な保存を実行します。 コマンドを実行する前にコントロール パネルを使用してアーカイブ マネージャをシャットダウンし、保存が完了したら再起動します。

注: ddm_save ユーティリティは DDM データベースからデータを削除しません。指定されたファイルにコピーするだけです。

このユーティリティの形式は以下のとおりです。

```
ddm_save [-extension] [-quiet] <SAVE_FILE>
```

引数名の最初の文字が太字で示されている場合は、名前全体を入力するのではなく、その文字だけを使用してもかまいません。

-extension

デフォルトでは、ddm_save はファイルに .tgz ファイル拡張子を割り当てます。 この引数を指定すると、ファイルはファイル拡張子なしで保存されます。

-quiet

対話型/冗長モードを無効にします。

<SAVE_FILE>

保存されるデータベースの保存先ファイルの名前を指定します。

注: ddm_save コマンドは、.tgz のデフォルト ファイル拡張子で gzip された tar ファイル形式に必要なデータベース ファイルを保存します。

ddm_backup

$SPECROOT/SS/DDM ディレクトリにある ddm_backup ユーティリティは、アーカイブ マネージャをシャットダウンし、ddm_save コマンドを実行し、保存が完了するとアーカイブ マネージャを再起動します。

このユーティリティの形式は以下のとおりです。

```
ddm_backup <SAVE_FILE>
```

<SAVE_FILE>

保存されるデータベースの保存先ファイルの名前を指定します。 ファイルを保存する完全修飾ファイル パスと保存先ファイル名を指定します。 完全修飾ファイル パスを指定しないと、ファイルは DDM ディレクトリに保存されます。

詳細情報:

ddm_save (P. 75)

バックアップに関する注意事項

- より頻繁にバックアップを実行すると、データが失われるリスクが小さくなります。
- 履歴データをバックアップしても、DDM データベースからそのデータは削除されません。
- 履歴レコードは、.configrc ファイルの MAX_STAT_DAYS および MAX_EVENT_DAYS の設定に従って、DDM データベースからパージ（完全に削除）されます。 詳細については、「アーカイブ マネージャのリソース（P. 83）」を参照してください。

  デフォルト設定は 45 日です。 どんなレコードも指定された経過期間に到達すると、パージが毎日その後発生します。 その結果、最も古い日のデータは日単位でパージされます。
- ddm_save メソッドを使用する場合、バックアップが完了したらコントロール パネルを使用してアーカイブ マネージャを再起動する必要があります。

データベースのリストア

ハードウェアまたはオペレーティング システムで障害が発生した場合は、サポートされるバックアップ方法（post_olb_script、ddm_backup、ddm_save）のいずれかを使用して作成された保存ファイルのロードにより、DDM データベースをリストアします。

保存ファイルは、ここで説明する ddm_load ツールを使用してロードします。

ddm_load

$SPECROOT/SS/DDM ディレクトリにあるこのユーティリティを使用すると、DDM データベースをリストアできます。 このツールは、最初に DDM ディレクトリを初期化し（データをすべて削除します）、次に指定された保存ファイルをロードします。 保存ファイルまたはディレクトリをロードする場合は、初期化が自動的に行われます。 -initialize オプションは、DDM ディレクトリをクリアし、何もない状態から開始する場合にのみ必要です。 このオプションでは、関連する SpectroSERVER のランドスケープ ハンドルも必要です。

コマンドを実行する前に、コントロール パネルからアーカイブ マネージャをシャットダウンします。 リストアが完了したら再起動します。

コマンドの形式は以下のとおりです。

```
ddm_load [-quiet] [-initialize <LANDSCAPE_HANDLE>] [-events_init]
[-stats_init][<SAVE_FILE>]
```

引数名の最初の文字が太字で示されている場合は、文字列全体を入力するのではなく、その文字だけを使用してもかまいません。

-quiet

プロンプト（対話型モード）を無効にします。 スクリプトの内部からロード コマンドを実行する場合に役立ちます。

-initialize

DDM データベースからデータをすべて削除します。LANDSCAPE_HANDLE 引数と一緒にのみ使用します。

<LANDSCAPE_HANDLE>

データベースを初期化している SpectroSERVER のランドスケープ ハンドルを指定します。

-events_init

DDM データベース内のイベント レコードのみを初期化（削除）します。

-stats_init

DDM データベース内の統計レコードのみを初期化（削除）します。

<SAVE_FILE>

ロードするバックアップ ファイルの名前を指定します。 ddm_load ユーティリティは、新しい gzip された tar ファイル形式と、従来の固有ファイル形式の両方をサポートします。

リストアに関する注意事項

- データベースの同期を保つため、SpectroSERVER データベースをリストアするたびに DDM データベースもリストアする必要があります。
- 特定の保存ファイルのレコードの一部は、再ロードするまでにパージ日付に到達している可能性があります。その結果、アーカイブ マネージャが再起動されるとすぐに、それらはパージされます。
- リストアが完了したら、アーカイブ マネージャを再起動します。

データベース保守と最適化

以下では、DDM データベースの効率的な実行を維持するために実行できるスクリプトについて説明します。

db_maintenance.pl

$SPECROOT/SS/DDM/scripts ディレクトリにあるこのスクリプトは、参照されていないすべての統計レコードを DDM データベースから削除します。参照されていないレコードは、通常、モデルを削除すると作成されます。時間と共に、関連するレコードがパージされて、不要なものがデータベースに残ります。

重要: db_maintenance.pl スクリプトは、アーカイブ マネージャのダウン時にのみ実行できます。 このスクリプトを実行するメリットと、アーカイブ マネージャを停止するデメリットを、慎重に比較検討してください。

このスクリプトの定期的な実行をスケジュールしないでください。 非常に多くのモデルをランドスケープから削除した場合にのみ、実行します。このスクリプトの実行により実現されるパフォーマンスの向上は顕著ではありません。このスクリプトは、多くのモデルが破棄された場合に（全ランドスケープなど）、一般的な保守作業でのみ実行します。

このスクリプトを実行する前に db_optimize.pl スクリプトを実行することをお勧めします。 さらに、実稼働環境への影響が最も小さいときにこのスクリプトを実行することを推奨します。 このスクリプトは、大きくて最適化されていないデータベースまたは多くのイベント レコードがあるデータベースでは、非常に時間がかかることがあります。

このスクリプトの形式は以下のとおりです。

```
./db_maintenance.pl[-q]
```

-q

（クワイエット）プロンプト（対話型モード）を無効にします。

詳細情報:

db_optimize.pl (P. 80)

db_optimize.pl

このスクリプトは、*$SPECROOT*/SS/DDM/scripts ディレクトリにあります。DDM データベース内のすべてのテーブルを最適化する簡単な方法を提供します。

テーブルの最適化には 2 つの大きな利点があります。

- DDM データベースに送信されるクエリの速度が向上します。
- パージされたレコードから使用可能なディスク容量が回復されます。

パフォーマンスの要件に応じて、週単位または月単位でこのスクリプトを実行するようにスケジュールできます。 DDM データベースのサイズおよびフラグメンテーションのレベルによっては、このスクリプトの実行に相当な時間がかかる可能性があることに留意してください。

注: db_optimize.pl スクリプトでは、*$SPECROOT*/mysql/data/ddmdb ディレクトリ内の最大のファイルのサイズより大きい空きディスク領域が必要です。 このスクリプトを実行する前に、データベース ファイルをバックアップしておくことをお勧めします。

このスクリプトは、アーカイブ マネージャを停止しないで実行できます。ただし、テーブルが最適化されてから、アーカイブ マネージャがロックされたデータベースからリカバリできるまでに、10 分間の遅延があります。

このスクリプトの形式は以下のとおりです。

```
./db_optimize.pl[-q]
```

-q

（クワイエット）プロンプト（対話型モード）を無効にします。

DDM データベース クエリ

ここで説明するクエリは、どんなイベントが生成されているか、また、どのモデルがこれらのイベントを生成しているかを特定するのに役立ちます。特定のデバイスが過度のイベントを生成しているか、または統計データを保存する日数を変更する必要があるかどうかを判断するのに役立つクエリが用意されています。 これらのクエリは、DDM データベース イベント テーブルのサイズによっては、実行に時間がかかることがあります。

注: データが保存される日数を制御する MAX_STAT_DAYS パラメータ値の設定については、「アーカイブ マネージャ リソース (P. 83)」を参照してください。

MySQL にログインする方法

$SPECROOT/mysql/bin でコマンド プロンプトから次のコマンドを入力します。

```
mysql -uroot -proot ddmdb
```

UNIX では、以下の引数も渡します。

```
-S $SPECROOT/mysql/tmp/mysql.sock
```

mysql> プロンプトが表示されます。

DDM (ddmdb) データベースをクエリする方法

mysql> プロンプトで、次のクエリを入力します。

- テーブルの統計情報を表示する方法

  ```
  SHOW TABLE STATUS LIKE "event";
  ```

- イベントの総数を取得する方法

  ```
  SELECT COUNT(*) FROM event;
  ```

- 特定の日付より後に発生したイベントの数を取得する方法

  ```
  SELECT COUNT(*)
    FROM event
    WHERE utime >= UNIX_TIMESTAMP("yyyy-mm-dd");
  ```

- 最もよく生成される上位 10 イベントを取得する方法

```
SELECT HEX(type), COUNT(*) AS cnt
  FROM event
  GROUP BY type
  ORDER BY cnt DESC
  LIMIT 10;
```

- 最もイベントの多い上位 10 モデルを取得する方法

```
SELECT HEX(e.model_h), m.model_name, COUNT(*) AS cnt
  FROM event e, model m
  WHERE e.model_h=m.model_h
  GROUP BY e.model_h
  ORDER BY cnt DESC
  LIMIT 10;
```

- イベントが最も多い上位 10 日を取得する方法

```
SELECT DATE(FROM_UNIXTIME(UTIME)) AS x, COUNT(*) AS cnt
  FROM event
  GROUP BY x
  ORDER BY cnt DESC
  LIMIT 10;
```

- 過去 10 のイベント数を取得する方法

```
SELECT date(from_unixtime(utime)) AS x, COUNT(*) AS cnt
  FROM event
  GROUP BY x
  ORDER BY x DESC
  LIMIT 10;
```

その他のデータベース ユーティリティ

CA Spectrum には 3 つの MySQL データベース ユーティリティがあり、すべて *$SPECROOT*/mysql/bin ディレクトリにインストールされます。

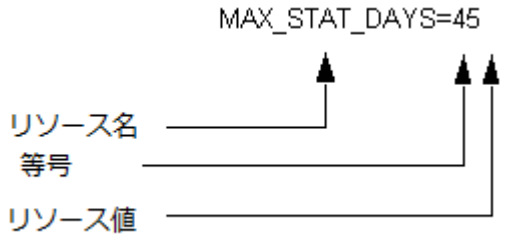
- myisamchk を使用すると、MySQL サーバが停止している間に、MyISAM テーブルを確認して修復できます。
- mysqlcheck は myisamchk に似ています。 ただし、テーブルを確認して修復するためにサーバを停止する必要がありません。 このユーティリティは、テーブルの最適化と分析も行います。
- mysqladmin を使用すると、サーバ設定の確認やデータベースの削除などの管理操作を実行できます。

注: これらの MySQL ユーティリティの使用法の詳細については、http://www.mysql.com のドキュメントを参照してください。

アーカイブ マネージャ リソース

アーカイブ マネージャ リソースは、履歴レコードの処理方法を制御する設定です。 また、これらのリソースにより、アーカイブ マネージャは SpectroSERVER および DDM データベースと通信できるようになります。

アーカイブ マネージャ リソースは .configrc ファイルで定義されており、このファイルは *$SPECROOT*/SS/DDM ディレクトリにあります。 次に示すように、リソースは「リソース = リソース値」の形式でリストされています。



.configrc ファイルでは、以下のアーカイブ マネージャ リソースが定義されています。

ARCH_MGR_SOCKET_NUMBER

アーカイブ マネージャが VNM および SSAPI クライアントからの要求をリスンするポートを示します。

デフォルト： 0xbafe

AUTO_REPAIR_DB

アーカイブ マネージャが破損している DDM データベースの修復を自動的に試行するかどうかを制御します。 デフォルトでは、このパラメータは .configrc ファイルにリストされていず、したがって有効ではありません。 自動修復機能を有効にするには、AUTO_REPAIR_DB=TRUE を .configrc ファイルに追加します。 自動修復機能を無効にするには、値を FALSE に設定するか、または .configrc ファイルからエントリを削除します。

デフォルト： 無効

DDM_DATABASE_NAME

アーカイブ マネージャと DDM データベースの間の通信に必要です。 デフォルト値は、CA Spectrum がインストールされると自動的に読み取られます。 この値の変更は推奨されません。

デフォルト： ddmdb

DDM_DATABASE_PORT

MySQL データベース サーバへの接続に使用されるポートを指定します。

DDM_DATABASE_HOSTNAME

アーカイブ マネージャと DDM データベースの間の通信に必要です。 デフォルト値は、CA Spectrum がインストールされると自動的に設定されます。 この値の変更は推奨されません。

デフォルト： local host

DDM_DATABASE_PASSWORD

アーカイブ マネージャと DDM データベースの間の通信に必要です。 デフォルト値は、インストールの間に自動的に読み取られます。 この値の変更は推奨されません。

デフォルト： CA Spectrum インストール所有者ユーザ名パスワード

DDM_DATABASE_USERNAME

アーカイブ マネージャと DDM データベースの間の通信に必要です。デフォルト値は、インストールの間に自動的に読み取られます。 この値の変更は推奨されません。

デフォルト： CA Spectrum インストール所有者ユーザ名

DDM_SOCKET_NUMBER

廃止されました。 このリソースは将来のリリースに表示されません。

LANDSCAPE_PRECEDENCE

現在サポートされていないリソース。

MAX_DB_CONNECTIONS

アーカイブ マネージャが要求の処理に使用できる同時 MySQL 接続の最大数を指定します。

デフォルト: 25

MAX_EVENT_DAYS

イベント データが保存される最大日数を指定します。MAX_EVENT_DAYS を超えた古いデータはパージされます。

デフォルト： 45

MAX_STAT_DAYS

統計データが保存される最大日数を指定します。MAX_STAT_DAYS を超えた古いデータはパージされます。

デフォルト： 45

TIME_TOLERANCE_IN_SECONDS

ログ データに対してレポートされるタイムスタンプと実際のログ記録時間の間の許容される最大誤差を指定します。 時間許容値が大きいほど、データを効率よく圧縮でき、CPU とディスクのリソースを節約できます。ただし、この値を大きくすると、正確性が犠牲になります。

デフォルト： 300

DDM およびアーカイブ マネージャのトラブルシューティング

このセクションでは、アーカイブ マネージャおよび DDM データベースに関連する問題の想定される原因および解決策を示します。

ARCHMGR.OUT のエラー メッセージ

症状

次のようなエラー メッセージが表示されます。

`<date/time> ERROR at ArchMgr.cc(591): Failed to open connection to SpectroSERVER at <hostname>, 0xbeef.` このエラーは、SpectroSERVER がダウンしていることを示します。 60 秒後に再度試行します。

解決方法

SpectroSERVER は準備ができていません。 アーカイブ マネージャは、SpectroSERVER が開始されると自動的に開始しますが、SpectroSERVER が準備ができるまで接続できません。 このメッセージは、初期起動時および SpectroSERVER のダウン時に常に表示されます。

症状

アーカイブ マネージャが接続または再接続すると、次のメッセージが表示されます。

<*date/time*> : ArchMgr has successfully registered with DDM name service - <*ip_address*> (<machine_name>).

症状

アーカイブ マネージャを開始しようとすると、次のメッセージが表示されます。

<*date/time*> : ArchMgr is shutting down...

解決方法

アーカイブ マネージャを開始しようとしたユーザの CA Spectrum ユーザ モデルが SpectroSERVER に含まれていないため、アーカイブ マネージャはシャットダウンされました。

症状

次のようなエラー メッセージが表示されます。

<*date/time*> : ArchMgr has successfully shut down.

このメッセージが消えない場合は、OneClick コンソールの［ユーザ］タブで、ユーザが ArchMgr プロセスの所有者として存在することを確認します。

症状

次のようなエラー メッセージが表示されます。

```
<date/time> : ArchMgr started as user '<user_name>'
<date/time> : ArchMgr validating database
Database corruption detected:
    ddmdb.statistic_ul64 - record delete-link-chain corrupted
    ddmdb.statistic_ul64 - Corrupt
Error opening the DDM database. One or more tables are corrupt.
Recovery options:
1. Run `ArchMgr -repair` to attempt recovery
2. Load a valid DDM savefile using `ddm_load <savefile>`
3. Initialize the DDM database using `ddm_load -i <LANDSCAPE_HANDLE>`
<date/time> : ArchMgr invalid database error.
```

解決方法

DDM データベースは破損しておりロードできません。 ddm_load コマンドを使用して、保存されている DDM データベースをロードするか、または DDM データベースを初期化できます。 また、ArchMgr -repair コマンドを使用して、既存のデータベース ファイルの修復を試みることができます。

ArchMgr -repair コマンドを実行するには、*$SPECROOT*/SS/DDM ディレクトリに移動して、次のように入力します。

./ArchMgr -repair

修復が正常に完了すると、以下のメッセージが表示されます。

“ArchMgr successfully repaired database.”

アーカイブ マネージャが破損した DDM データベースの自動修復を試行できるようにするには、.configrc ファイルで auto_repair_db オプションを使用します。 詳細については、「アーカイブ マネージャのリソース (P. 83)」を参照してください。

詳細情報:

ddm_load (P. 77)
アーカイブ マネージャ リソース (P. 83)

イベントがリストされない

症状

イベントが［イベント］タブにリストされず、［サーバ］（［接続ステータス］）ボタンが黄色で点滅します。 ボタンをダブルクリックすると、［接続ステータス］ダイアログ ボックスでは、イベント サービスと表示サービスについてはステータスが「ダウン」と表示され、ランドスケープ サービスについては「アップ」と表示されます。

解決方法

アーカイブ マネージャは、実行していないか、SpectroSERVER に接続されていません。 コントロール パネルの［制御］メニューで、アーカイブ マネージャ サービスが開始されていることを確認します。 開始されている場合は、*$SPECROOT*/SS/DDM/ARCHMGR.OUT ファイルでアーカイブ マネージャのエラーを確認します。