



CHAPTER 5

データベースとカンマ区切り形式リポジトリの管理

この章では、ユーティリティ スクリプトを使用して Collection Manager (CM) データベースおよび Comma Separated Value (CSV; カンマ区切り形式) リポジトリを管理する方法について説明します。ほとんどのデータベース管理タスクは、バンドルされた Sybase データベースにだけ適用可能です。



(注) ユーティリティ スクリプトの一般的な使用手順については、「[ユーティリティ スクリプトの使用方法](#)」(P.4-1) を参照してください。

- 「一般的なデータベース管理タスク」(P.5-1)
- 「バンドルされたデータベースの管理」(P.5-7)
- 「CSV リポジトリの管理」(P.5-8)

一般的なデータベース管理タスク

サポート対象のすべてのデータベースに適用可能なデータベース管理タスクは、次のとおりです。

- データベース テーブル リストの生成
- 古いレコードの定期削除スケジュールの定義と適用
- テーブルの削除
- テーブルからの古いレコードの手動削除

データベースに格納されているすべてのレコードに、Cisco Service Control Management Suite (SCMS) Collection Manager (CM) が Raw Data Record (RDR) を受信した時間を示すタイムスタンプが与えられます。このタイムスタンプは、データベース テーブルにさまざまなメンテナンス処理を実行する場合に使用します。

データベースの設定および維持に使用するスクリプトは、次のとおりです。

- ~scmscm/scripts/dbtables.sh
- ~scmscm/scripts/dbperiodic.py
- ~scmscm/db_maint/create_periodic_del_procs.sh
- ~scmscm/scripts/droptable.sh
- ~scmscm/scripts/prunetable.sh

データベース テーブルのリスト表示

データベース内のすべてのテーブルをリスト表示するには、**dbtables.sh** スクリプトを使用します。

```
~scmscm/scripts/dbtables.sh
```

該当する場合、テーブル内の行数と最古および最新のタイムスタンプが表示されます。

テーブルの実際の内容は Cisco Service Control Application (SCA) Reporter を使用して表示することができます。詳細については、『*Cisco Service Control Application Reporter User Guide*』を参照してください。

表 5-1 には、**dbtables.sh** スクリプト オプションがリスト表示されています。

表 5-1 dbtables.sh オプション

オプション	説明
-l	既存のテーブル名だけをリスト表示します (統計情報なし)。
-a	非レポート テーブルもリストに含まれます。
-f	JDBC ではなくクライアントを使用して、高速回線カウントをイネーブルにします。 (注) バンドルされた Sybase データベースにだけ適用可能です。
-t { sec_num }	応答に対する最大待機時間 (秒) です。デフォルトは、タイムアウトなしです。
-h	このヘルプ メッセージを出力し、終了します。

次に、**dbtables.sh** スクリプトの出力例を示します。

```
>~scmscm/scripts/dbtables.sh
Executing query ...
      name| num_lines|                min_time|                max_time|
-----+-----+-----+-----+
      RPT_SUR| 131000| 2006-10-30 16:46:42.24| 2007-02-15 12:00:32.216|
      RPT_LUR| 170000| 2007-04-10 15:25:45.31| 2007-04-11 07:06:05.45|
      RPT_VLUR| 4694| 2007-04-11 13:12:39.683| 2007-04-11 13:18:07.396|
      RPT_PUR| 116000| 2007-04-09 04:45:55.566| 2007-04-11 07:44:09.186|
      RPT_TR| 57766| 2007-04-11 13:12:39.683| 2007-04-11 13:18:07.396|
      RPT_MALUR| 109000| 2007-04-09 04:46:35.903| 2007-04-09 13:32:18.42|
      RPT_MEDIA| 120000| 2007-04-05 17:14:24.443| 2007-04-11 13:16:29.436|
      RPT_TOPS_PERIOD0| 194250| 2007-03-18 20:00:00.01| 2007-04-23 06:00:00.16|
      RPT_TOPS_PERIOD1| 46940| 2007-03-19 00:00:00.05| 2007-04-23 00:00:00.1|
```

古いレコードの定期削除の管理

古いレコードの定期削除を管理するには、次の手順を実行します。

1. CM のインストール時に定期削除手順がインストールされていない場合は、インストールします。
scmscm ユーザとしてログインし、CM を起動します。データベース テーブルが作成されるまで 1 ~ 2 分待機して、次のスクリプトを実行します。
~scmscm/db_maint/create_periodic_del_procs.sh.
2. 定期削除コンフィギュレーション ファイルを編集します。「定期削除の設定」(P.5-3) を参照してください。

3. 新しい設定を適用するには、**dbperiodic.py** ユーティリティ スクリプトを使用します。「[定期削除コンフィギュレーションファイルの適用 \(P.5-5\)](#)」を参照してください。

直前の削除期間がまだ終了していない場合は、テーブルの定期削除は実行されません。これにより、データベースに過度の負荷がかかって、アダプタの挿入パフォーマンスが低下することがなくなります。

同時に複数のテーブルが削減されるようにスケジューリングされている場合は、定期削除コンフィギュレーション ファイルに記述された順番にテーブルが処理されます。

設定を簡素化するために、1 つのスケジュールですべてのテーブルを連続的に定期削除するようスケジューリングすることもできます。

すべての定期削除アクティビティは、システム ログ ファイルに記録されます。Solaris の場合、システム ログ ファイルは **/var/adm/messages** ディレクトリにあります。Linux の場合、システム ログ ファイルは **/var/log/messages** ディレクトリにあります。次の例は、定期削除動作が実行されるときシステムのログ ファイルを示しています。

```
Feb 17 05:00:07 app-redhat79 logger: /opt/scmscm/db_maint/periodic_del.sh for hourly all -
running single del for RPT_LUR to 14 days
Feb 17 05:00:09 app-redhat79 logger: Starting periodic delete for RPT_LUR keeping 14 days
Feb 17 05:00:09 app-redhat79 logger: Deleted rows: 0 from the table RPT_LUR
```



(注) MySQL データベースを使用している場合、定期削除はバージョン 5 以降に限りサポートされます。

- 「[定期削除の設定 \(P.5-3\)](#)」
- 「[定期削除コンフィギュレーションファイルの適用 \(P.5-5\)](#)」

定期削除の設定

定期削除コンフィギュレーションファイル (**dbperiodic.conf**) は、デフォルトで **~scmscm/db_maint/** にあります。このファイルの構造は INI ファイルと似ています。セクションごとに、特定のテーブルセットに対する特定のデータ削減処理が記述されていて、指定されたスケジュールに従って削除処理が実行されます。



(注) ファイルを解析する場合、このファイルの各セクションの名前は使用されません。任意の名前を使用できます。

各セクションは、角カッコで囲まれたセクション名で開始し、次の表に示されたパラメータが続きます (コンフィギュレーションファイルの各セクションで、すべてのパラメータが必要であるわけではありません)。パラメータとパラメータ値は等号 (=) で区切ります。定期削除コンフィギュレーションファイルの例については、[表 5-2](#)を参照してください。

表 5-2 定期削除コンフィギュレーション ファイルのパラメータ

パラメータ名	説明	値	デフォルト	例
active	コンフィギュレーション ファイルのこのセクションを使用するかどうか	true/false	true	false
tablenames	このセクションが適用されるテーブルの名前	カンマで区切られたテーブルの名前、または * (すべてのテーブルを指定する場合)	* (すべて)	RPT_SUR,RPT_LUR
daystokeep	レコードを保持する日数	正の整数	14	30
minute	コンフィギュレーション ファイルのこのセクションで削除を実行する時期	0 ~ 59、*	0 *	0
hour		0 ~ 23、*	(すべて) *	0,4,8,12,16,20
day		1 ~ 31、*	(すべて) *	1
month		1 ~ 12、*	(すべて)	1,3,5,7,9,11



(注)

active および **daystokeep** を除くすべてのパラメータの値には、単一の値、カンマ区切り値のリスト、値の範囲 (ダッシュで区切った 2 つの値)、またはすべての有効値を表すアスタリスク (*) を指定できます。**tablenames** には範囲を指定できません。

次の例では、すべてのフィールドがデフォルト値に設定されます。

```
# This dbperiodic.conf file emulates the legacy style for periodic
# deletion. All tables are processed every hour on the hour, and
# records are kept for 14 days.
[hourly all]
active = true
tablenames = *
daystokeep = 14
minute = 0
hour = *
```

この例では、各テーブルにそれぞれ 10 日分のデータを残して、すべてのテーブルが午前 4:30 に削減されます。さらに、それぞれ 3 日分のデータを残して、リアルタイム テーブルが 1 時間おきに削減されます。

```
# This dbperiodic.conf file reduces all tables once a day and
# real-time tables once an hour.
[daily all]
active = true
tablenames = *
daystokeep = 10
minute = 30
hour = 4
[hourly real-time]
active = true
tablenames = RPT_SUR,RPT_LUR,RPT_PUR
daystokeep = 3
minute = 0
hour = *
```

定期削除コンフィギュレーション ファイルの適用

新規定期削除コンフィギュレーション ファイルをロードして適用したり、現在のファイルを表示したりするには、**dbperiodic.py** スクリプトを使用します。

```
~scmscm/scripts/dbperiodic.py [ --dump ] [ --load | --loadfile=path_to_dbperiodic.conf ]
```

スクリプトを使用して新規コンフィギュレーション ファイルをロードするには、ファイルを解析し、妥当性を確認し、scmscm ユーザの crontab を更新して変更を反映させます。表 5-3 では、dbperiodic.py オプションをリスト表示しています。

表 5-3 dbperiodic.py オプション

オプション	説明
--load	/export/home/scmscm/db_maint/dbperiodic.conf から定期削除コンフィギュレーションをロードします。
--loadfile=path to periodic delete configuration file	指定されたディレクトリから定期削除コンフィギュレーション ファイルをロードします。
--dump	定期削除コンフィギュレーションを出力します。
--h	このスクリプトのオプションを表示します。

次に、現在の定期削除コンフィギュレーションを出力する例を示しています。

```
~scmscm/scripts/dbperiodic.py --dump
```



(注)

このスクリプトは、ロード済みの定期削除コンフィギュレーションを出力します。現在の定期削除コンフィギュレーション ファイルがまだロードされていない場合は、コンフィギュレーションとスクリプト出力が異なることがあります。

次に、~scmscm/db_maint/dbperiodic.conf から定期削除コンフィギュレーション ファイルをロードする例を示しています。

```
~scmscm/scripts/dbperiodic.py --load
```

次に、指定された場所から定義削除コンフィギュレーション ファイルをロードする例を示しています。

```
~scmscm/scripts/dbperiodic.py --loadfile=path_to_periodic_delete_configuration_file
```

テーブルの削除

データベースからテーブルを 1 つ、または現在のテーブルをすべて削除するには、**droptable.sh** スクリプトを使用します。

```
~scmscm/scripts/droptable.sh [ -f ] tableParameter
```

表 5-4 では droptable.sh オプションをリスト表示しています。

表 5-4 droptable.sh オプション

オプション	説明
<i>table_name</i>	データベースから <i>table_name</i> を削除します。
ALLTABLES	データベースからすべてのテーブルを削除します。
-f	強制的に削除します（応答プロンプトやエラー報告は表示されません）。
-h	このスクリプトのオプションを表示します。

次に、確認プロンプトを表示しないで、データベースからテーブル RPT_SUR を削除する例を示しています。

```
~scmscm/scripts/droptable.sh -f RPT_SUR
```

次に、データベースからすべてのテーブルを削除する例を示しています。

```
~scmscm/scripts/droptable.sh ALLTABLES
```

古いレコードの削除

レコードのタイムスタンプに基づいてデータベース テーブルからレコードを削除するには、**prunetable.sh** スクリプトを使用します。

```
~scmscm/scripts/prunetable.sh [ -f ] num_days table_name
```

表 5-5 では prunetable.sh オプションをリスト表示しています。

表 5-5 prunetable.sh オプション

オプション	説明
<i>num_days</i>	削除されないレコードの最長保存期間（日数）。
<i>table_name</i>	レコードが削除されるテーブル。
-f	強制的に削除します（応答プロンプトやエラー報告は表示されません）。
-h	このスクリプトのオプションを表示します。

次の例では、テーブル RPT_SUR から、保存期間が 7 日を超えたすべてのレコードを削除する方法を示しています。

-f フラグが指定されていないため、確認プロンプトまたはエラー報告（またはその両方）が表示されることがあります。

```
>~scmscm/scripts/prunetable.sh 7 RPT_SUR
```

バンドルされたデータベースの管理

バンドルされたデータベースの管理には、次のものが含まれています。

- データベースのバックアップおよび復元
- IP アドレスが変更された Sybase のアップデート
- Sybase データベース統計情報のアップデート

データベースに格納されている各レコードに、Cisco Service Control Management Suite (SCMS) Collection Manager (CM) が Raw Data Record (RDR) を受信した時間を示すタイムスタンプが与えられます。このタイムスタンプは、データベース テーブルにさまざまなメンテナンス処理を実行する場合に使用します。

バンドルされた Sybase データベースだけを維持するのに使用するスクリプトは、次のとおりです。

- `~scmscm/scripts/sybback.sh`
- `~scmscm/scripts/sybrestore.sh`
- `~scmscm/db_maint/update_statistics.sh`

データベースのバックアップ

データベース内にある全テーブルのテキスト ファイル バックアップを作成するには、`sybback.sh` スクリプトを使用します。

```
~scmscm/scripts/sybback.sh -d path_to_backup_directory
```

スクリプトによりすべてのテーブルが ASCII ファイルに変換され、バックアップ ディレクトリにコピーされます。表 5-6 では `sybback.sh` オプションをリスト表示しています。

表 5-6 sybback.sh オプション

オプション	説明
<code>-d path_to_backup_directory</code>	バックアップ テキスト ファイルを指定したディレクトリに書き込みます。
<code>-h</code>	このスクリプトのオプションを表示します。

データベースの復元

`sybback.sh` スクリプトから作成されたバックアップ ファイルからデータベースを復元するには、`sybrestore.sh` スクリプトを使用します。

```
~scmscm/scripts/sybrestore.sh -d path_to_restore_directory
```

表 5-7 では `sybrestore.sh` オプションをリスト表示しています。

表 5-7 sybrestore.sh オプション

オプション	説明
<code>-d path_to_restore_directory</code>	指定されたディレクトリ内のテキスト ファイルを使用してデータベースを復元します。
<code>-h</code>	このスクリプトのオプションを表示します。



(注) **sybback.sh** スクリプトおよび **sybrestore.sh** スクリプトは、Sybase の実行可能なバックアップ メカニズムではありません。これらは、マシン間で小さいテーブルを転送するなど、少量のデータのバックアップと復元を行うために設計されたものです。



(注) 実行可能なバックアップ メカニズムが必要な場合は、*Sybase Backup Server* 製品のマニュアルを参照してください。

IP アドレスが変更された Sybase のアップデート

IP アドレスを変更する場合は、Sybase サーバをアップデートする必要があります。root ユーザとして、次のコマンドを実行します。

```
~scmscm/setup/syb_interfaces.sh
```

Sybase データベース統計のアップデート

毎週、夜間などのデータベース負荷が低いときに、`update_statistics.sh` スクリプトを実行することを推奨します。

```
~scmscm/db_maint/update_statistics.sh
```

CSV リポジトリの管理

ユーティリティ スクリプトを使用して、CM による CSV ファイル出力のリポジトリを管理できます。サービス プロバイダーの **Operational Support System (OSS)** あるいはサードパーティ製の課金システムがこれらのファイルを利用できるように、ファイルはカンマ区切り形式 (CSV) アダプタによってディスクに書き込まれます。ディスク オーバーフローを防止するために、CSV リポジトリのサイズをモニタする必要があります。



(注) バックアップ パラメータが **true** に設定されている場合、CSV ファイルの削除に失敗するとディスクのオーバーフローとなる可能性があります (CSV ファイルは削除されません)。



(注) サードパーティ製アプリケーションは、CSV ファイルを管理し、必要に応じてこれを削除する責任があります。

このスクリプトを正常に呼び出すためには、CM の HTTP アダプタが稼動している必要があります。アダプタがダウンしている場合、エラー メッセージが表示されます。

- 「CSV リポジトリのファイル構造」 (P.5-9)
- 「CSV ファイル リポジトリの設定」 (P.5-9)
- 「カンマ エスケープの設定」 (P.5-10)
- 「プリント不可能文字のエスケープ設定」 (P.5-10)

CSV リポジトリのファイル構造

CSV ファイルは複数のサブディレクトリに格納されます。各サブディレクトリには、Raw Data Record (RDR) タグの番号が設定されます (RDR タグは RDR のタイプを定義します)。各 RDR は RDR タグ番号と同じ名前のサブディレクトリに格納されます。RDR タグの詳細については、『[Cisco Service Control Application for Broadband Reference Guide](#)』を参照してください。

CSV ファイルには、ディレクトリごとに異なる番号が連番で (自動的に) 付加されます。親ディレクトリの場所を変更するには、**cm/config** ディレクトリ内の **cm.conf** ファイルを編集します。

CSV ファイル リポジトリの設定

csvconf.sh スクリプト、**~scm scm/scripts/csvconf.sh** を使用して次のことを実行します。

- リポジトリに現在格納されている RDR 数のリスト表示。
- CSV ファイルの最大数、および各ファイル内の最大レポート (行) 数の設定。
- 古い CSV ファイルが上書きされる時点でバックアップを作成するかどうかの制御。
- CSV ファイルの各行にこの RDR に送信された Service Control Engine (SCE) の IP を示すものが含まれるかどうかの制御 (デフォルトでこのオプションはオフ)。

表 5-8 では **csvconf.sh** オプションをリスト表示しています。



(注)

このスクリプトを使用する代わりに、**~scm scm/cm/config/csvadapter.conf** ファイルを編集できます。このファイルの変更を有効にするには、CM を再起動する必要があります。



(注)

CSV リポジトリ内のすべてのサブディレクトリに、同じ設定が適用されます。



(注)

これらのパラメータを設定しても、既存の CSV ファイルは変更されず、この後作成されるファイルにだけ影響します。

表 5-8 **csvconf.sh** オプション

オプション	説明
--list	CSV リポジトリの内容 (リポジトリに現在格納されている RDR 数) を表示します。
--clear	CSV リポジトリからすべてのファイルを削除します (このオプションでは、すべての CSV ファイルが削除されますが、ファイルが格納されているディレクトリは削除されません)。
--maxlines=N	CSV ファイルごとの RDR の最大数を N (1 ~ 20,000 の範囲の整数) に設定します。
--maxfiles=M	各サブディレクトリの CSV ファイルの最大数を M (10 ~ 10,000 の範囲の整数) に設定します。

表 5-8 csvconf.sh オプション (続き)

オプション	説明
<code>--backups={true false}</code>	古い CSV ファイルのバックアップをイネーブルまたはディセーブルにします。
<code>--recordsource={true false}</code>	CSV ファイルへのレコード送信元の追加をイネーブルまたはディセーブルにします。

次に、サブディレクトリごとの CSV ファイルの最大数を 1,000 に設定する例を示しています。

```
>~scmscm/scripts/csvconf.sh --maxfiles=1000
```

次に、CSV ファイルごとのレコードの最大数を 10,000 に設定する例を示しています。

```
>~scmscm/scripts/csvconf.sh --maxlines=10000
```

次に、CSV リポジトリからすべてのファイルを削除する例を示しています。

```
~scmscm/scripts/csvconf.sh --clear
```

次に、リポジトリの古い CSV ファイルのバックアップをディセーブルにする例を示しています。

```
~scmscm/scripts/csvconf.sh --backups=false
```

カンマ エスケープの設定

CSV ファイルのフィールド内にカンマを含める場合、エスケープ シーケンスを使用して、カンマの位置がフィールドの末尾ではないことを示します。

3 つのエスケープ方式がサポートされています。

- 単一引用符：単一引用符は、1 つ以上のカンマが含まれているフィールドを囲みます。既存 RDR 内にすでに存在する単一引用符は特に処理されません。
- URL：フィールドに含まれている各カンマを %2C で置き換えます。既存 RDR 内にすでに存在するこのシーケンスは特に処理されません。
- バックスラッシュ：フィールド内の各カンマの前にバックスラッシュ (\) が追加されます。既存 RDR 内にすでに存在するバックスラッシュは特に処理されません。

最初の 2 つのエスケープ方式は、Microsoft Excel と互換性があります。バックスラッシュ方式は Excel と互換性はありませんが、下位互換性のために残されています。

デフォルトでは、単一引用符が使用されます。エスケープ方式を変更するには、`escapeMethod` 属性値を変更します。この属性は、**CSVAdapter** ディレクトリ内の `csvadapter.conf` ファイル内にあります。この値は、`backslash`、`quote`、`url` のいずれかでなければなりません。

プリント不可能文字のエスケープ設定

CSV アダプタでプリント不可能文字をエスケープできます。このオプションをイネーブルにすると、アダプタのパフォーマンスに悪影響を与えます。デフォルトでは、このオプションはディセーブルです。

このオプションがイネーブルの場合、フィールド内に含まれている CR や LF などの各プリント不可能文字の前にバックスラッシュ (\) が追加されます。

CSVAdapter ディレクトリ内の `csvadapter.conf` ファイルで、このオプションをイネーブルにできます。このファイルの変更を有効にするには、CM を再起動する必要があります。