



Cisco Service Control Management Suite Collection Manager ユーザ ガイド

Release 3.1
May 2007

このマニュアルに記載されている仕様および製品に関する情報は、予告なしに変更されることがあります。このマニュアルに記載されている表現、情報、および推奨事項は、すべて正確であると考えていますが、明示的であれ黙示的であれ、一切の保証の責任を負わないものとします。このマニュアルに記載されている製品の使用は、すべてユーザ側の責任になります。

対象製品のソフトウェア ライセンスおよび限定保証は、製品に添付された『Information Packet』に記載されています。添付されていない場合には、代理店にご連絡ください。

シスコシステムズが採用している TCP ヘッダー圧縮機能は、UNIX オペレーティングシステムの UCB (University of California, Berkeley) パブリックドメインバージョンの一部として、UCB が開発したプログラムを最適化したものです。All rights reserved. Copyright © 1981, Regents of the University of California.

ここに記載されている他のいかなる保証にもよらず、各社のすべてのマニュアルおよびソフトウェアは、障害も含めて「現状のまま」として提供されます。シスコシステムズおよびこれら各社は、商品性や特定の目的への準拠性、権利を侵害しないことに関する、または取り扱い、使用、または取引によって発生する、明示されたまたは黙示された一切の保証の責任を負わないものとします。

いかなる場合においても、シスコシステムズおよびその代理店は、このマニュアルの使用またはこのマニュアルを使用できないことによって起こる制約、利益の損失、データの損傷など間接的で偶発的に起こる特殊な損害のあらゆる可能性がシスコシステムズまたは代理店に知らされていても、それらに対する責任を一切負いかねます。

CCSP, the Cisco Square Bridge logo, Follow Me Browsing, and StackWise are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn, and iQuick Study are service marks of Cisco Systems, Inc.; and Access Registrar, Aironet, ASIST, BPX, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Empowering the Internet Generation, Enterprise/Solver, EtherChannel, EtherFast, EtherSwitch, Fast Step, FormShare, GigaDrive, GigaStack, HomeLink, Internet Quotient, IOS, IP/TV, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, LightStream, Linksys, MeetingPlace, MGX, the Networkers logo, Networking Academy, Network Registrar, Packet, PIX, Post-Routing, Pre-Routing, ProConnect, RateMUX, ScriptShare, SlideCast, SMARTnet, StrataView Plus, SwitchProbe, TeleRouter, The Fastest Way to Increase Your Internet Quotient, TransPath, and VCO are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0501R)

このマニュアルで使用している IP アドレスは、実際のアドレスを示すものではありません。マニュアル内の例、コマンド出力、および図は、説明のみを目的として使用されています。説明の中に実際のアドレスが使用されていたとしても、それは意図的なものではなく、偶然の一致によるものです。

Cisco Service Control Management Suite Collection Manager ユーザガイド

Copyright © 2007 Cisco Systems, Inc.

All rights reserved.



CONTENTS

はじめに	vii
マニュアルの変更履歴	viii
対象読者	x
マニュアルの構成	x
表記法	xi
関連資料	xii
マニュアルの入手方法、テクニカル サポート、およびセキュリティ ガイドライン	xii
Japan TAC Web サイト	xii

CHAPTER 1

概要	1-1
Cisco Service Control の概要	1-2
Cisco Service Control ソリューション	1-2
無線サービス プロバイダーでサービス コントロールを使用する場合	1-2
DSL プロバイダーおよび ISP でサービス コントロールを使用する場合	1-3
ケーブル MSO でサービス コントロールを使用する場合	1-3
Cisco Service Control の機能	1-4
SCE プラットフォーム	1-5
管理および収集	1-7
ネットワーク管理	1-7
加入者管理	1-8
サービス設定の管理	1-8
データ収集	1-8
Cisco Service Control Application for Broadband	1-9

CHAPTER 2

Collection Manager の機能	2-1
データ収集プロセス	2-2
RDR	2-3
CM ソフトウェア パッケージ	2-4
RDR サーバ	2-4
カテゴリライザ	2-4
プライオリティ キューおよび永続的バッファ	2-4

アダプタ	2-5
JDBC アダプタ	2-5
CSV アダプタ	2-5
TA アダプタ	2-5
TA アダプタ サイクル	2-6
RAG アダプタ	2-6
RAG アダプタ集計バケット	2-6
バケットのフラッシュ	2-7
HTTTPC アダプタ	2-7
データベース	2-8
バンドルされているデータベースの使用法	2-8
外部データベースの使用法	2-8

CHAPTER 3

Collection Manager のインストール	3-1
システム要件	3-2
システムの前提条件のチェック	3-2
Solaris の要件	3-3
ハードウェア	3-3
ソフトウェアおよび環境	3-3
ロケールおよび時間帯の設定	3-5
Red Hat Linux の要件	3-5
ハードウェア	3-5
ソフトウェアおよび環境	3-5
ロケールおよび時間帯の設定	3-6
ディストリビューションの内容	3-6
デフォルト設定	3-6
Collection Manager のインストール	3-7
Collection Manager ソフトウェアで使用されるポート	3-7
Sybase データベースのインストール	3-8
installsyb.sh によって実行される動作	3-8
前提条件	3-8
Collection Manager ソフトウェアのインストール	3-9
install-cm.sh スクリプト	3-9
Sybase データベースおよびサービス コントロール ソフトウェアのアンインストール	3-13
Sybase のアンインストール	3-13
サービス コントロール ソフトウェアのアンインストール	3-13
CM バージョン 3.1 へのアップグレード	3-14

CHAPTER 4

Collection Manager の管理	4-1
ユーティリティ スクリプトの使用法	4-2
CM の設定	4-3
サーバのアクティブ化	4-3
アダプタの制御	4-4
アダプタのイネーブル化	4-4
データベースの制御	4-4
SCE 接続の切断	4-5
カテゴライザの設定	4-6
システム状態のモニタ	4-7
定期チェッカのインストール	4-7
定期チェッカ スクリプト	4-7
定期チェッカ スクリプト	4-7
テスト	4-8
ユーザ管理	4-9
仮想リンクの管理	4-10
CM のモニタ	4-11
データベース容量の確認	4-11
RDR レートの確認	4-11
SCE 接続の確認	4-12
例：	4-12
サーバが動作可能であることの確認	4-12

CHAPTER 5

データベースおよび CSV リポジトリの管理	5-1
一般的なデータベース管理タスク	5-2
データベース テーブルのリスト化	5-2
古いレコードの定期削除の管理	5-3
定期削除の設定	5-4
定期削除コンフィギュレーション ファイルの適用	5-5
バンドルされたデータベースの管理	5-6
テーブルの削除	5-6
古いレコードの削除	5-6
データベースのバックアップ	5-7
データベースの復元	5-7
IP アドレスが変更された Sybase のアップデート	5-8
CSV リポジトリの管理	5-9
CSV リポジトリのファイル構造	5-9
CSV ファイル リポジトリの設定	5-9

カンマ エスケープの設定	5-11
プリント不可能文字のエスケープ設定	5-11

CHAPTER 6

データベースの設定	6-1
Oracle ユーザのクイック スタート ガイド	6-2
Velocity Template Language	6-3
データベース コンフィギュレーション ファイル	6-4
コンテキスト オブジェクト	6-5
tables オブジェクト	6-5
dbinfo オブジェクト	6-5
tools オブジェクト	6-5
アプリケーションの設定	6-6
作業例	6-7
マクロの定義	6-7
dbinfo の設定	6-8
SQL の定義	6-8
「drop table」のコード	6-8
「create table」のコード	6-8
「create indexes」のコード	6-9
「insert」のコード	6-9
メタデータ クエリーのコード	6-9
テストおよびデバッグ	6-10
文字列の解析	6-10
完全デバッグ情報の取得	6-10
スクリプトでの JDBC フレームワークの使用法	6-11
SCE 時間帯オフセットの表示および設定サンプル	6-11
Oracle のスケーラビリティに関するヒント	6-13
カスタム テーブルスペースの使用法	6-13
テーブル パーティショニングの使用法	6-14

APPENDIX A

コード サンプル	A-1
アプリケーションの設定	A-1
tables.xml ファイル	A-1
tables.dtd ファイル	A-3
アダプタの設定	A-5
ragadapter.conf ファイル	A-5
ragadapter.xml ファイル	A-5



はじめに

May 30, 2007, OL-7208-05-J

ここでは、『Cisco Service Control Management Suite Collection Manager ユーザガイド』の対象読者、本書の内容、表記法、資料および技術サポートの取得方法について説明します。このマニュアルでは、Cisco Service Control ソリューション、Service Control Engine (SCE) プラットフォーム、および関連コンポーネントの概要に関して基本的な知識があることが前提となります。

このイントロダクションでは、次の内容について説明します。

- [マニュアルの変更履歴 \(p.viii\)](#)
- [対象読者 \(p.x\)](#)
- [マニュアルの構成 \(p.x\)](#)
- [表記法 \(p.xi\)](#)
- [関連資料 \(p.xii\)](#)
- [マニュアルの入手方法、テクニカルサポート、およびセキュリティ ガイドライン \(p.xii\)](#)

マニュアルの変更履歴

Cisco Service Control のリリース	Part Number	発行日
リリース 3.1.0	OL-7208-05	2007 年 5 月

変更内容

- 仮想リンクのセマンティックを管理する新しいスクリプトが追加されました。「[仮想リンクの管理](#)」(p.4-10)を参照してください。
- Solaris 10 のサポートが追加され、Solaris 8 のサポートが削除されました。
- MySQL 5 のサポートが追加されました。
- Oracle 10 のサポートが追加されました。
- バンドルされている Solaris 用 Sybase データベースがバージョン 15 にアップグレードされました。
- Sybase インストール スクリプトから --expert モードが削除されました。ユーザは、通常のインストール モードから DB (データベース) 容量を確認できます。
- 定期削除メカニズムおよび dbtables.sh スクリプトが、外部データベースに対してサポートされるようになりました。「[一般的なデータベース管理タスク](#)」(p.5-2)を参照してください。
- リリース 3.1 へのアップグレード手順では、以前のバージョンの設定が維持されます。「[install-cm.sh オプション](#)」(p.3-9) および「[CM バージョン 3.1 へのアップグレード](#)」(p.3-14)を参照してください。

Cisco Service Control のリリース	Part Number	発行日
リリース 3.0.5	OL-7208-04	2007 年 11 月

変更内容

以下のセクションが追加されました。

- [ユーザ管理](#) (p.4-9)
- [アダプタのイネーブル化](#) (p.4-4)
- [カテゴリの設定](#) (p.4-6)

Cisco Service Control のリリース	Part Number	発行日
リリース 3.0.3	OL-7208-03	2006 年 5 月

変更内容

以下の新機能が追加されました。

- check_prerequisites.sh スクリプト「[システムの前提条件のチェック](#)」(p.3-2)を参照してください。
- monitor.sh スクリプト「[システム状態のモニタ](#)」(p.4-7)を参照してください。

以下の廃止予定機能を削除しました。

- installsyb.sh スクリプトの --legacy フラグ

以下のセクションが追加されました。

- [プリント不可能文字のエスケープ設定](#) (p.5-11)

Cisco Service Control のリリース	Part Number	発行日
リリース 3.0	OL-7208-02	2005 年 12 月

変更内容

以下の新機能が追加されました。

- [HTTPC アダプタ \(p.2-7\)](#)

以下の廃止予定機能を削除しました。

- DB アダプタ

以下のセクションが追加されました。

- [コード サンプル \(p.A-1\)](#)

Cisco Service Control のリリース	Part Number	発行日
リリース 2.5.5	OL-7208-01	2005 年 2 月

変更内容

- このマニュアルの初版

対象読者

このマニュアルは、Cisco Service Control Management Suite (SCMS) Collection Manager (CM) のインストールおよび設定を実際に行うネットワークまたはコンピュータ技術者を対象としています。また、加入者指向環境でサービス プロバイダー オペレータによる拡張機能を実行可能にする CM の日常業務を担当するオペレータも対象としています。

マニュアルの構成

このマニュアルで説明する内容は、次のとおりです。

章	タイトル	説明
第 1 章	概要	Cisco Service Control ソリューションの機能概要を説明します。
第 2 章	Collection Manager の機能	CM コンポーネントの機能について詳細に説明します。
第 3 章	Collection Manager のインストール	CM とそのデータベースのインストール手順、および CM の起動方法について説明します。
第 4 章	Collection Manager の管理	ユーティリティ スクリプトを使用して CM パラメータや情報を表示およびアップデートする方法について説明します。
第 5 章	データベースおよび 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 データベースおよび CSV リポジトリを管理する方法について説明します。
第 6 章	データベースの設定	ご利用のデータベースと連携するように CM を設定する方法について説明します。
付録 A	コード サンプル	コンフィギュレーション ファイルのコードのサンプル一覧を提供します。

表記法

このマニュアルでは、次の表記法を使用しています。

- 太字は、コマンド、キーワード、およびボタンを示しています。
- イタリック体は、ユーザが値を指定するコマンド入力に使用されます。
- screen フォントは、画面に表示される情報に使用されます。
- 太字の screen フォントは、ユーザが入力する情報に使用されます。
- 縦棒 (|) は、選択要素の区切りを表します。
- 角カッコ ([]) は、省略可能な要素を表します。
- 波カッコ ({ }) 必須の選択肢を表します。
- 角カッコ内の波カッコ ([{ }]) は、省略可能な要素内の必須選択肢を表します。



(注)

「注釈」です。役立つ情報や、このマニュアル以外の参照資料などを紹介しています。



ワンポイント・アドバイス

「時間の節約に役立つ操作」です。記述されている操作を実行すると時間を節約できます。



注意

「要注意」の意味です。機器の損傷またはデータ損失を予防するための注意事項が記述されています。



警告

「危険」の意味です。人身事故を予防するための注意事項が記述されています。機器の取り扱い作業を行うときは、電気回路の危険性に注意し、一般的な事故防止対策に留意してください。

関連資料

この『Cisco SCMS Collection Manager ユーザガイド』は、次のシスコ製品マニュアルと併せて使用してください。

- 『Cisco Service Control Application for Broadband User Guide』
- 『Cisco Service Control Application for Broadband Reference Guide』
- 『Cisco SCA BB Service Configuration API Programmer Guide』
- 『Cisco Service Control Application Reporter User Guide』

マニュアルの入手方法、テクニカル サポート、およびセキュリティ ガイドライン

マニュアルの入手方法、テクニカル サポート、マニュアルに関するフィードバックの提供、セキュリティ ガイドライン、および推奨されるエイリアスと一般のシスコ製品のマニュアルの詳細に関しては、次の URL で毎月更新される『What's New in Cisco Product Documentation』を参照してください。シスコの新規版および改訂版の技術マニュアルの一覧が示されています。

<http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html>

Japan TAC Web サイト

Japan TAC Web サイトでは、利用頻度の高い TAC Web サイト (<http://www.cisco.com/tac>) のドキュメントを日本語で提供しています。Japan TAC Web サイトには、次の URL からアクセスしてください。

<http://www.cisco.com/jp/go/tac>

サポート契約を結んでいない方は、「ゲスト」としてご登録いただくだけで、Japan TAC Web サイトのドキュメントにアクセスできます。

Japan TAC Web サイトにアクセスするには、Cisco.com のログイン ID とパスワードが必要です。ログイン ID とパスワードを取得していない場合は、次の URL にアクセスして登録手続きを行ってください。

<http://www.cisco.com/jp/register/>



概要

この章では、Cisco Service Control ソリューションの概要を示します。まず、Cisco Service Control の概要について、さらにサービス コントロール機能について説明します。

また、Service Control Engine (SCE) プラットフォームのハードウェア機能、および Cisco Service Control ソリューションを構成するシスコ固有のアプリケーションについても簡単に説明します。

- [Cisco Service Control の概要 \(p.1-2 \)](#)
- [Cisco Service Control の機能 \(p.1-4 \)](#)
- [SCE プラットフォーム \(p.1-5 \)](#)
- [管理および収集 \(p.1-7 \)](#)
- [Cisco Service Control Application for Broadband \(p.1-9 \)](#)

Cisco Service Control の概要

- [Cisco Service Control ソリューション \(p.1-2\)](#)

Cisco Service Control ソリューション

Cisco Service Control ソリューションは、サービス プロバイダーが直面するさまざまなサービス コントロールの課題を解決する専用ハードウェア、および特定のソフトウェア ソリューションが組み合わさって実現されます。SCE プラットフォームの設計目的は、インターネット /IP トラフィックの調査、分析、および制御をサポートすることです。

サービス プロバイダーはサービス コントロールを使用することにより、既存インフラストラクチャに投資しながら、利益を上げる仕組みを新規に作成することができます。また、マルチギガビットワイヤ回線速度で IP ネットワーク トラフィックを分析、課金、および制御することもできます。さらに、収益力の高いコンテンツベース サービスを探して実現し、配信できるようにするための必要なツールにもなり得ます。

電気通信業界の低迷が示すように、IP サービス プロバイダーのビジネス モデルには、利益を上げるために再編が必要です。プロバイダーは、より巨大なデータ リンクを構築するために莫大な資金を投下し、多額の負債を抱え、コストは上昇し続けました。その一方で、アクセスおよび帯域幅という商品の価格は継続的に下落し、利益は消滅しました。現在、サービス プロバイダーは、自身のネットワーク上を往来するトラフィックやサービスからより多くの収入を得るには、付加価値のあるサービスを提供する必要があることを認識しています。ただし、IP サービスから実際に利益を得るには、データ リンク上でこれらのサービスを単に提供するだけでなく、詳細なモニタリングと精度、リアルタイム制御、およびサービス提供時のサービス認識が必要となります。シスコが提供する Service Control ソリューションを使用すると、サービス プロバイダーはこのギャップを埋めることができます。

- [無線サービス プロバイダーでサービス コントロールを使用する場合 \(p.1-2\)](#)
- [DSL プロバイダーおよび ISP でサービス コントロールを使用する場合 \(p.1-3\)](#)
- [ケーブル MSO でサービス コントロールを使用する場合 \(p.1-3\)](#)

無線サービス プロバイダーでサービス コントロールを使用する場合

無線サービス プロバイダーは、加入者に 2.5G および 3G ベースのデータ サービスを提供することに成功しています。これらのサービスでは、ビジネス モデルを維持したり、新規サービスを迅速に提供したりするためにきわめて重要とされる、Average Revenue Per User (ARPU; ユーザあたりの平均売上高) を大幅に増加させることが期待されています。

これらのデータ サービスでは、サービスを提供したり、加入者にこれらのサービスを課金するための新しい方法が必要なのです。Cisco Service Control ソリューションを使用すると、以下のことが可能になります。

- 複数の課金モデルのサポート
- リアルタイム サービス コントロールによる課金漏れの回避
- 柔軟な支払プラン 後払い、前払い、MRC、利用回数による支払
- 各種アプリケーションのコンテンツベースの課金
- 登録制の段階的アプリケーション サービス

DSL プロバイダーおよび ISP でサービス コントロールを使用する場合

個人宅およびビジネス向けのブロードバンドユーザ層をターゲットとする DSL プロバイダーおよび ISP は、広範な IP サービスの提供によって差別化を図りながら、既存インフラストラクチャから最大限の収益を上げる新しい方法を見つける必要があります。

シスコ製品を使用すると、既存ネットワークに新たなサービス インテリジェンス レイヤおよび制御機能が追加され、以下のことが可能になります。

- ネットワーク使用状況の詳細表示
- アプリケーション SLA または Acceptable Use Policy (AUP; アクセプタブル ユース ポリシー) の自動適用
- 異なるタイプの顧客、コンテンツ、またはアプリケーション向けのさまざまなサービス レベルの実装
- エンドツーエンドのサービス コントロールのための、ネットワーク エッジからネットワーク コアまでの配置
- 既存のネットワーク要素および BSS/OSS システムと、シスコ製ソリューションとの統合の簡素化

ケーブル MSO でサービス コントロールを使用する場合

ケーブル MSO は、多数のユーザ宅に高速ケーブル モデム サービスを配置してきました。現在、ケーブル MSO は差別化されたサービスを導入し、ブロードバンド インフラストラクチャを介したサービス配信の完全管理に対応したサービス コントロールを実装することにより、ブロードバンド アクセスを提供するだけのビジネスから脱却する必要があります。Cisco Service Control ソリューションを使用すると、以下のことが可能になります。

- 容量計画のための、加入者レベルおよび集約レベルでのネットワーク トラフィックのレポートおよび分析
- AUP に違反しているネットワーク不正使用者の識別
- ピアツーピア トラフィック、NNTP (ニュース) トラフィック、およびスパム不正使用者の識別および管理。AUP の適用
- 加入者宅でのサービス使用の制限、および複数の (未払い) PC の使用制限
- 直感的な利用が可能な階層型アプリケーション サービスおよびアプリケーション SLA の保証
- 加入者の管理および課金のための標準 OSS または従来の OSS との完全な統合

Cisco Service Control の機能

Cisco Service Control ソリューションの中心には、専用ネットワーク ハードウェア デバイスである SCE があります。Service Control ソリューションを提供するためのアプリケーションを広範にサポートする、SCE プラットフォームのコア機能には、次のようなものがあります。

- 加入者およびアプリケーション認識機能 アプリケーションレベルで IP トラフィックを調査することにより、加入者単位での使用状況およびコンテンツを詳細かつリアルタイムに認識および制御することができます。
 - 加入者の認識 IP フローと特定の加入者を対応付けて、プラットフォーム経由でトラフィックを送信している各加入者の状態を維持したり、この加入者トラフィックに適切なポリシーを適用することができます。

加入者認識機能を実現するには、DHCP や RADIUS サーバなどの加入者管理リポジトリと統合するか、RADIUS または DHCP トラフィックのスニффングを通じて行います。

- アプリケーション認識 アプリケーション プロトコル レイヤ (レイヤ 7) までのトラフィックを認識および分析できます。

バンドルされたフローを使用して実装されたアプリケーション プロトコル (制御およびデータフローを使用して実装された FTP [ファイル転送プロトコル] など) の場合、SCE プラットフォームはフロー間のバンドリング接続を認識して、適切に処理します。

- アプリケーションレイヤでのステートフルなリアルタイム トラフィック制御 詳細な帯域幅の測定やシェーピング、クォータ管理とリダイレクション、アプリケーションレイヤでのステートフルなリアルタイム トラフィック トランザクション処理の使用など、高度な制御機能を実行できます。そのためには、適応性の高いプロトコルおよびアプリケーション レベル インテリジェンスが必要です。
- プログラム可能性 新規プロトコルを迅速に追加して、常に変化するサービス プロバイダー環境に新規サービスおよびアプリケーションを容易に適応させることができます。プログラム可能性を実現するには、Cisco Service Modeling Language (SML) を使用します。

プログラム可能であるということは、新規サービスを迅速に配置し、ネットワーク、アプリケーション、またはサービスの拡張に合わせて容易にアップグレードできるということです。
- 堅牢で柔軟なバックオフィス統合 サービス プロバイダーで使用されている、プロビジョニングシステム、加入者リポジトリ、課金システム、OSS システムなどのサードパーティ製システムと統合できます。SCE には、詳しいマニュアルが付属したさまざまな公開 API が用意されていて、迅速かつ強固な統合プロセスを実行できます。
- スケーラブルな高性能サービス エンジン これらすべての操作をワイヤ スピードで実行することのできる機能

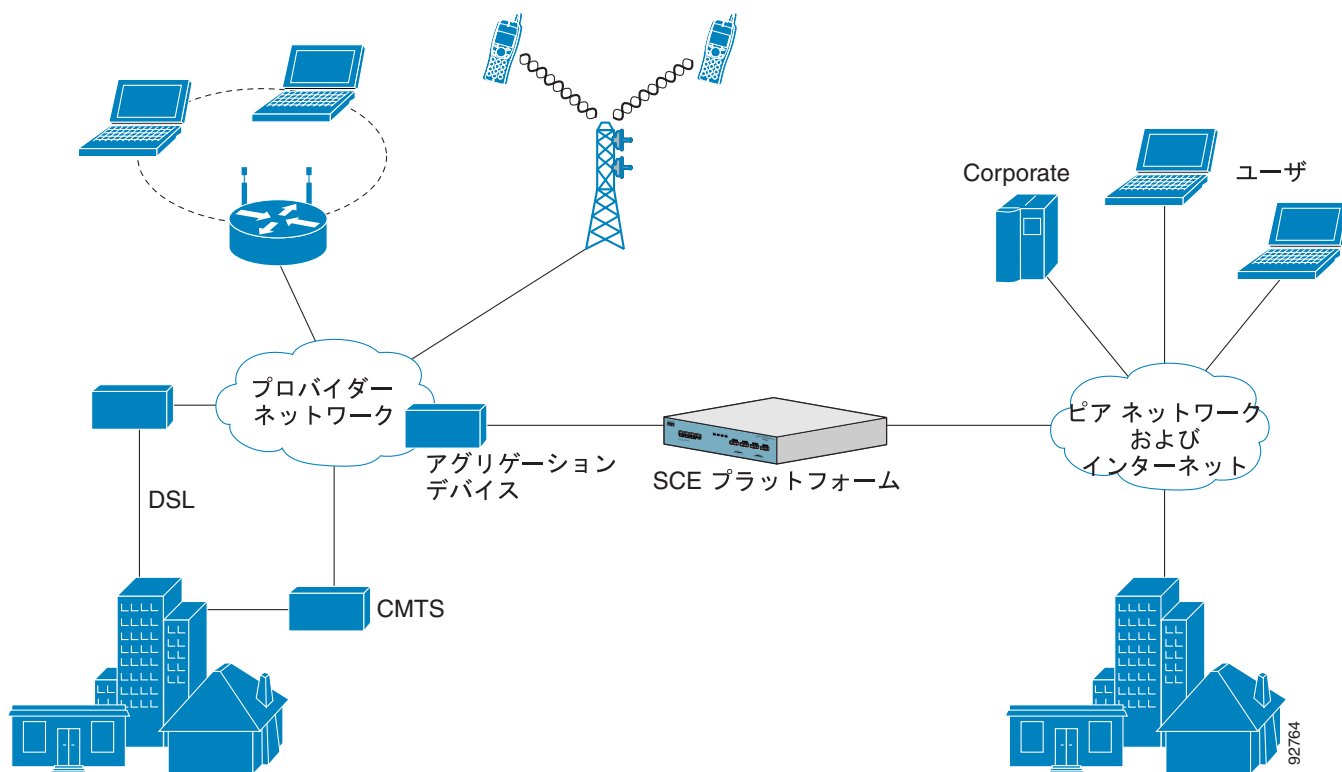
SCE プラットフォーム

プログラム可能なネットワーク デバイスである SCE ファミリーには、IP トラフィックのアプリケーションレイヤ ステートフル フロー インスペクションを実行したり、設定可能な規則に基づいてトラフィックを制御する機能があります。SCE は、ASIC (特定用途向け IC) コンポーネントおよび RISC (縮小命令セット コンピュータ) プロセッサを利用する専用ネットワーク デバイスです。これにより、パケットをカウントするだけでなく、ネットワーク トラフィックの内容を詳細に調べることができます。双方向トラフィック フローにプログラム可能なステートフル インスペクションを実行したり、これらのフローとユーザ所有権を対応付けることにより、SCE プラットフォームはネットワークの使用状況をリアルタイムで分類できます。この情報は SCE プラットフォームの高度なトラフィック制御および帯域幅シェーピング機能の基礎となります。多くの帯域幅シェイパ機能が適用されない条件下でも、SCE プラットフォームは次のような制御およびシェーピング オプションを提供します。

- レイヤ7のステートフルワイヤ速度パケット インスペクションおよび分類
- 次のような 600 を超えるプロトコルおよびアプリケーションの安定したサポート
 - 一般 HTTP、HTTPS、FTP、TELNET、NNTP、SMTP、POP3、IMAP、WAP など
 - P2P ファイル共有 FastTrack-KazaA、Gnutella、BitTorrent、Winny、Hotline、eDonkey、DirectConnect、Piolet など
 - P2P VoIP Skype、Skinny、DingoTel など
 - ストリーミングおよびマルチメディア RTSP、SIP、HTTP ストリーミング、RTP/RTCP、など
- プログラム可能なシステム コアによる、柔軟性のあるレポートおよび帯域幅の制御
- 透過的なネットワークおよび BSS/OSS と既存ネットワークの統合
- 加入者の認識による、トラフィックおよび使用状況と特定の顧客との関連付け

次の図に、ネットワーク内の SCE プラットフォーム配置例を示します。

図 1-1 ネットワーク内の SCE プラットフォーム



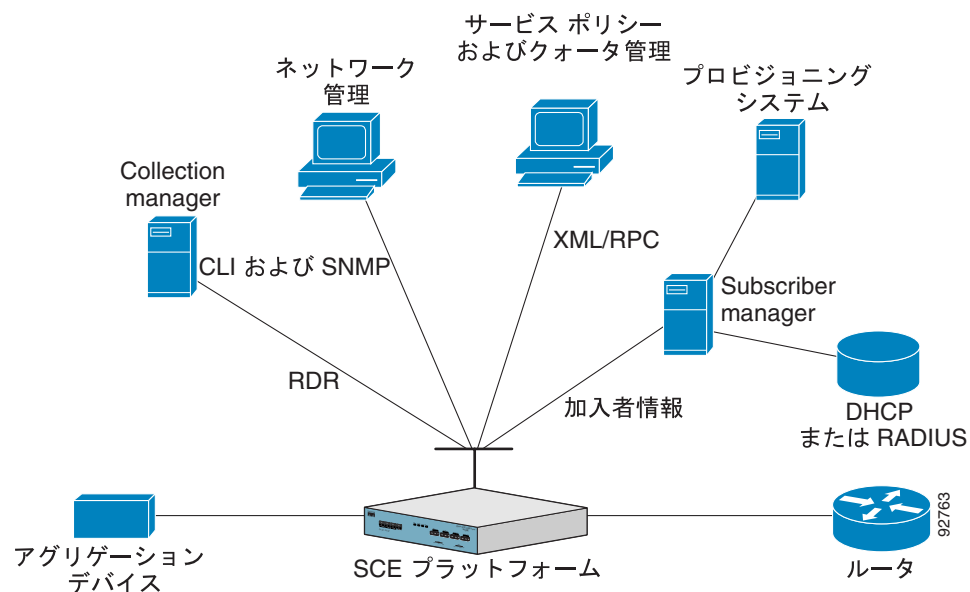
管理および収集

Cisco Service Control ソリューションには、サービス コントロール ソリューションのあらゆる面を管理する、次の管理コンポーネントを備えた完全な管理インフラストラクチャが含まれています。

- ネットワーク管理
- 加入者管理
- サービス コントロール管理

これらの管理インターフェイスの設計目的は、一般的な管理基準に準拠して、既存 OSS インフラストラクチャとの統合を容易にすることです。

図 1-2 サービス コントロールの管理インフラストラクチャ



ネットワーク管理

シスコは、完全なネットワーク FCAPS 管理（障害、設定、アカウントティング、パフォーマンス、セキュリティ）を提供します。

ネットワーク管理用のインターフェイスが2つ用意されています。

- CLI (コマンドライン インターフェイス) コンソール ポートまたは Telnet 接続を通じてアクセス可能な CLI は、設定およびセキュリティ機能に使用します。
- SNMP (簡易ネットワーク管理プロトコル) (SNMP トラップによる) 障害管理およびパフォーマンス モニタリング機能を提供します。

加入者管理

Cisco Service Control Application for Broadband (SCA BB) が加入者毎に個別のポリシーを実施して、個別の加入者ベースで使用状況を追跡している場合、OSS と SCE プラットフォーム間のブリッジングに Cisco Service Control Management Suite (SCMS) Subscriber Manager (SM) がミドルウェアソフトウェアとして使用されている場合があります。加入者情報は SM データベースに格納され、実際の加入者配置に従って、複数のプラットフォーム間で配信できます。

SM は、ネットワーク ID と加入者 ID のマッピングによって加入者認識機能を実現しています。SM は RADIUS や DHCP サーバなどの AAA (認証、認可、アカウントिंग) デバイスと統合された専用統合モジュールを使用して、加入者情報を取得します。

加入者情報は、次のいずれかの方法で取得することができます。

- **プッシュモード** 加入者がログオンすると、SM は加入者情報を SCE プラットフォームに自動的にプッシュします。
- **プルモード** SCE プラットフォームからのクエリーの応答で、SM は加入者情報を SCE プラットフォームに送信します。

サービス設定の管理

サービス設定管理は、サービス コントロール アプリケーションの一般的なサービス定義を設定する機能です。トラフィック分類、課金およびレポート、および制御の設定を含むサービス コンフィギュレーション ファイルが作成されて、SCE プラットフォームに適用されます。SCA BB アプリケーションは、これらのコンフィギュレーション ファイルを SCE プラットフォームに自動的に配信するツールです。このシンプルで標準ベースの手法により、大規模なネットワークにある複数のデバイスを簡単に管理することができます。

サービス コントロールは、これらのファイルの編集と作成を行うための簡単に使用できる GUI (グラフィカル ユーザー インターフェイス) と、その作成を自動化する API のセットを提供しています。

データ収集

SCE プラットフォームの分析およびデータ処理機能を実行すると、Raw Data Record (RDR) が生成されます。これらの RDR は Cisco Service Control Management Suite Collection Manager (CM) で処理されます。CM ソフトウェアは、1 つ以上の SCE プラットフォームから RDR を取得する収集システムが実装されたものです。このソフトウェアはこれらのレコードを収集し、いずれかのアダプタで処理します。各アダプタは、RDR に特定のアクションを実行します。

RDR には、システム設定に応じてさまざまな情報および統計情報が含まれます。RDR には主に 3 つのカテゴリがあります。

- **トランザクション RDR** トランザクションがネットワーク トラフィックで検出された単一イベントである場合に、トランザクションごとに生成されるレコード。トランザクション ID は、特定のアプリケーションまたはプロトコルに依存します。
- **加入者使用状況 RDR** 定義期間中に加入者によって生成されたトラフィックを記述する、加入者単位で生成されるレコード
- **リンク RDR** 定義期間中にリンク上で伝達されるトラフィックを記述する、リンク単位で生成されるレコード

Cisco Service Control Application for Broadband

シスコでは、SCE プラットフォームの最上部で動作し、サービス プロバイダーが直面する IP ネットワーク コントロールの問題に対処するソリューションを提供しています。そのソリューションが Cisco Service Control Application for Broadband (SCA BB) です。

SCA BB を使用すると、サービス プロバイダーは複雑かつ検知の難しいネットワーク アプリケーション プロトコル (P2P など) を検出し、ビジネスおよびサービスの配信要件に従って制御できます。また、サービス プロバイダーが収益を上げたり、エンド カスタマーに競争力のあるサービスを提供するために使用する、差別化された階層型サービスを作成できます。SCA BB のプログラム可能なアプリケーション検出機能および加入者認識機能により、ネットワーク内にある特定の地点から階層型サービスを使用できるようになります。SCA BB はネットワークの変更やアップグレードを必要としません。既存のすべての IP ネットワーク スイッチ、ルータ、およびインフラストラクチャと互換性があります。



Collection Manager の機能

この章では、Collection Manager (CM) コンポーネントの機能について詳細に説明します。

この章では、Cisco Service Control Management Suite (SCMS) CM の機能について説明します。ここでは、Service Control Engine (SCE) プラットフォームが生成し CM に送信する Raw Data Record (RDR)、および CM ソフトウェア パッケージのコンポーネントの概要について説明します。また、RDR を格納するために使用されるデータベースの概要も示します。

- [データ収集プロセス \(p.2-2\)](#)
- [RDR \(p.2-3\)](#)
- [CM ソフトウェア パッケージ \(p.2-4\)](#)
- [アダプタ \(p.2-5\)](#)
- [データベース \(p.2-8\)](#)

データ収集プロセス

Cisco SCE プラットフォームは、Cisco Service Control Application for Broadband (SCA BB) などの、SEC プラットフォームで実行中のアプリケーションが仕様を定義した RDR を生成します。

RDR はシンプルで信頼性の高い RDR-Protocol を使用して、SCE プラットフォームから送信されます。Service Control ソリューションへのデータ レコード収集の統合作業では、収集システムへの RDR-Protocol サポート (わかりやすい開発プロセス) の実装も行われます。

CM が SCE プラットフォームから RDR を受信したあと、CM ソフトウェア モジュールは、プリセット カテゴリおよびタイプとプライオリティに基づいて各種タイプの RDR を認識してソートし、永続バッファに RDR をキューします。

その後、1 つまたは複数の CM アダプタが各 RDR を処理します。各アダプタが、RDR に関する特定の機能を実行します (RDR をローカル マシンに CSV 形式ファイルで保存したり、RDBMS (リレーショナル データベース管理システム) アプリケーションに送信したり、カスタム操作を実行したりします)。

プリインストールされているユーティリティ スクリプトを使用すると、CM の動作に影響する多数のパラメータを決定できます。

RDR

RDR は、SCE プラットフォームで作成されるレポートです。RDR、そのフィールド、および意味のリストは、固有の Service Control Protocol (SCP) アプリケーションごとに異なります。各 RDR タイプには RDR タグという固有の ID があります。

次に、SCP アプリケーションで作成可能な RDR の例を示します。

- 定期加入者使用状況レポート SCE プラットフォームは加入者認識ネットワーク デバイスで、加入者毎の使用状況をレポートすることができます。

これらの RDR には、一般的に加入者 ID (OSS 加入者 ID 等)、トラフィック タイプ (HTTP、ストリーミング、ピアツーピアトラフィック) および使用状況カウンタ (合計アップストリームおよびダウンストリーム量など) が含まれています。これらのタイプの使用状況レポートは、使用状況ベースの課金方式を使用する場合や、ネットワーク分析や容量計画に必要です。

SCA BB アプリケーション加入者使用状況 RDR は次のようなカテゴリに含まれます。

- トランザクション レベル レポート SCE プラットフォームは、配置されたリンク上で実行される各ネットワーク トランザクションのステートフル追跡を実行します。このステートフル性を使用して、SCP は複数の OSI レイヤ 7 プロトコル (HTTP、RTSP、SIP、Gnutella など) を追跡して、各アプリケーション レベル属性についてレポートします。

通常、これらの RDR には、基本的なレイヤ 3 ~ 4 属性 (送信元 IP、宛先 IP、ポート番号など) からレイヤ 7 プロトコル依存属性 (ユーザエージェント、HTTP のホスト名、SMTP メール送信者の電子メール アドレスなど) のトランザクションレベル パラメータ、および一般的なパラメータ (時刻、トランザクション有効期限など) が含まれます。これらの RDR は、コンテンツベース課金方式や詳細な使用状況統計情報にとって重要です。

SCA BB アプリケーション トランザクション RDR は次のようなカテゴリに含まれます。

- SCP アプリケーション アクティビティ レポート SCP アプリケーションでは、ネットワークトラフィックにさまざまなアクションを実行するように SCE プラットフォームをプログラムできます。これらのアクションには、トランザクションのブロック、特定の速度と制限へのトラフィックのシェーピング、アプリケーションレベルのリダイレクションの実行などが含まれています。そのような操作が実行される際に、SCP アプリケーションが RDR を生成する可能性があります。

SCA BB アプリケーション Breaching RDR および Blocking RDR は次のようなカテゴリに含まれます。(使用状況が何らかのクォータを超えたために) システムによって加入者へのアクティブな適用が変更された場合に、Breaching RDR が生成されます。(現在のサービス コンフィギュレーションに含まれている規則に従って) SCE プラットフォームがネットワーク トランザクションをブロックした場合に、Blocking RDR が生成されます。

CM ソフトウェア パッケージ

CM ソフトウェア パッケージには、一連の処理モジュールおよびソート モジュールが含まれています。たとえば、次のコンポーネントなどです。

- [RDR サーバ \(p.2-4\)](#)
- [カテゴリザ \(p.2-4\)](#)
- [プライオリティ キューおよび永続的バッファ \(p.2-4\)](#)

RDR サーバ

各着信 RDR は SCE プラットフォームから着信するため、RDR サーバが着信タイムスタンプと送信元 SCE プラットフォームの ID をこれに付加して、RDR をカテゴリザに送信します。

カテゴリザ

カテゴリザでは、RDR タグに従って各 RDR を分類します。また、各 RDR の宛先アダプタ、および RDR を送信する場合に経由するプライオリティ キューを決定します。

RDR は複数のアダプタに対応付けることができます。ユーザの要件に基づいて、有資格技術者がフローをコンフィギュレーション ファイルに定義します。

プライオリティ キューおよび永続的バッファ

各アダプタには各プライオリティ キューが 1 つ以上あり、各プライオリティ キューに永続的バッファが 1 つ割り当てられています。

プライオリティ キューで、プライオリティ レベルに従って各 RDR がキューイングされ、アダプタで処理されるまで永続的バッファに格納されます。

永続的バッファは不揮発性ストレージ領域で、万が一ハードウェア、ソフトウェア、または電源障害が発生してもシステムでの RDR の処理が保証されます。

アダプタ

アダプタは、目的のシステム要件を満たすように RDR を変換して、要求時に RDR を配布するソフトウェア モジュールです。現在、システムには次のアダプタが同梱されています。

- JDBC アダプタ
- CSV アダプタ
- TA アダプタ
- Real-Time Aggregating (RAG) アダプタ
- HTTPC アダプタ

アダプタの中には、データベースヘータを送信したり、CSV ファイルに書き込むものもあります。データベース テーブルの構造、およびこれらの CSV ファイルの場所と構造については、『Cisco Service Control Application for Broadband Reference Guide』で説明しています。

各アダプタには独自のコンフィギュレーション ファイルがあり、すべてのコンフィギュレーション ファイルは構造内で類似しています。サンプルの RAG アダプタ コンフィギュレーション ファイルについては、「ragadapter.conf ファイル」(p.A-5) を参照してください。

- [JDBC アダプタ \(p.2-5\)](#)
- [CSV アダプタ \(p.2-5\)](#)
- [TA アダプタ \(p.2-5\)](#)
- [RAG アダプタ \(p.2-6\)](#)
- [HTTPC アダプタ \(p.2-7\)](#)

JDBC アダプタ

JDBC アダプタは RDR を受け入れて処理し、データベースに格納します。

このアダプタの設計目的は、JDBC 準拠の任意のデータベース サーバとの互換性を保つことであり、そのデータベース サーバに応じてレコードが変換されます。JDBC アダプタは、リモート マシンで稼働するデータベースを利用するように設定できます。

JDBC アダプタは、以下のデータベースをサポートするように事前設定されています。

- Sybase ASE 12.5 および 15.0
- Oracle 9.2 および 10
- MySQL 4.1 および 5

CSV アダプタ

CSV アダプタは、RDR を受信して処理し、ディスク上のファイルに CSV 形式で書き込みます。FTP (ファイル転送プロトコル) などの標準メカニズムを使用して、サービス プロバイダーの OSS またはサードパーティ製課金システムでこれらのレコードを取得し、拡張アカウントリングおよびネットワーク トラフィック分析レコードを生成することもできます。

TA アダプタ

TA アダプタは、加入者使用状況 RDR を受信し、含まれているデータを集計して、上位レポートをデータベースに出力し、また集計した (上位のコンシューマだけではなく) すべての加入者の日次統計を CSV ファイルに出力します。複数のメトリックの上位加入者リストを定期的に出力します (直前の 1 時間における上位 50 のトラフィック量またはセッション コンシューマなど)。

■ アダプタ

このアダプタは永続的な保存状態（ディスクに保存）を維持して、障害時のデータ損失を最小限にします。

TA アダプタは、JDBC アダプタのインフラストラクチャを使用し、JDBC 準拠データベースを使用してローカルまたはリモートで設定することができます。

TA アダプタ サイクル

TA アダプタは、短期サイクルと長期サイクルの2種類のサイクルで動作します。サイクルは、最後に TA アダプタが集計情報をデータベースおよび CSV ファイルに出力することのできる固定インターバルです。短期サイクルのインターバルは 1 時間、長期サイクルのインターバルは 24 時間（毎日深夜）です。（分単位で定義される）インターバルと開始および終了時間を設定可能です。



(注) 長期サイクル インターバルは、短期サイクル インターバルの乗数でなければなりません。

各サイクルのアクティビティは、次のようにわずかに異なります。

- 短期サイクル 短期サイクルの終わりに、アダプタは次の処理を実行します。
 - 直前のサイクルの集約上位レポートを、短期サイクル データベース テーブルに追加します。
 - 停電の発生に備えて、現在のステータス ファイルを保存します。
- 長期サイクル 長期サイクルの終わりに、アダプタは次の処理を実行します。
 - 直前のサイクルの集約上位レポートを、短期サイクル データベース テーブルに追加します。
 - 停電の発生に備えて、現在のステータス ファイルを保存します。
 - 長期サイクル期間の集計統計情報を含む CSV ファイルを作成します。

RAG アダプタ

RAG アダプタは1つ以上のタイプの RDR を処理し、あらかじめ指定されたフィールド位置のデータをバケットに累積します。バケットの内容は CSV ファイルに書き込まれます。

- [RAG アダプタ集計バケット \(p.2-6\)](#)
- [バケットのフラッシュ \(p.2-7\)](#)

RAG アダプタ集計バケット

RAG アダプタ集計バケットは、RDR 内のフィールドの値を組み合わせてインデックス化されます。インデックス関係は、1対1または他對1になります。

バケット識別フィールドの値は、RDR タイプごとに設定されたクロージャ（同等クラス）を使用して処理されます。

例：

```
Bucket-identifying field = field number 3
Closures: 4 = 4,5,6; 10 = 8,10,11
Value in field 3 = 4, 5, or 6; field reported as 4
Value in field 3 = 8, 10, or 11; field reported as 10
```

特定のフィールド値をモニタして、バケットに格納された最初の RDR の値に対する変化を調べるように、アダプタを設定できます。モニタするフィールドごとに、値の変化が検出時に動作が実行されます。サポートされている動作は次のとおりです。

- この RDR を集約しないでバケットを終了し（チェックポイント）、この RDR で新しいバケットを開始します。
- ユーザ ログに警告を発行します。

バケット、クロージャ、トリガ、およびトリガ動作が XML ファイルに定義されます。XML ファイル例については、「[ragadapter.xml ファイル](#)」(p.A-5) を参照してください。

バケットのフラッシュ

バケットがフラッシュされると、CSV ファイルに CSV 行として書き込まれます。

バケットをフラッシュするためのトリガー（チェックポイント）は、次のうち、発生時間が最も早いものになります。

- バケット作成後の経過時間が設定値に到達
- バケット内の累積フィールドの量が設定値を超過
- アダプタまたは CM 全体がダウン
- 一部のフィールドに（バケットの内容と比較して）新しい値を持つ RDR がバケットに到着

CSV ファイルを閉じるためのトリガーは、次のうち、発生時間が最も早いものになります。

- ファイル作成後の経過時間が設定値に到達
- ファイル内の行数が設定値に到達
- アダプタまたは CM 全体がダウン

HTTPC アダプタ

HTTPC アダプタは RDR を受け入れて処理し、それを HTTP でポリシー サーバへ送信します。

HTTPC アダプタは、さまざまなポリシー サーバ モードと特定フローで必要な動作に従って、いろいろな HTTP 要求を設定するのに使用することができます。

HTTPC アダプタは、フローが開始されるポリシー サーバへの信号と、フローが終了するポリシー サーバへの信号に対する 2 つのタイプの RDR のみを受信します。

データベース

CM は、バンドルされているデータベースまたは外部データベースを使用して、システムの SCE プラットフォームから供給される RDR を格納することができます。

- [バンドルされているデータベースの使用方法 \(p.2-8\)](#)
- [外部データベースの使用方法 \(p.2-8\)](#)

バンドルされているデータベースの使用方法

バンドルモードでは、CM は Sybase Adaptive Server Enterprise データベースを使用します。これは、トランザクション集約型のエンタープライズアプリケーションをサポートし、オンラインでの情報の格納および取得が可能で、必要に応じて情報を蓄積することができます。

Sybase データベースは、他の CM コンポーネントと同じサーバ内にあります。小さくてシンプルなテーブルのグループから構成されるシンプルなスキーマを使用します。JDBC アダプタは、変換済の RDR をデータベースに送信して、これらのテーブルに格納します。レコードには、標準データベース クエリーおよびレポート ツールを使用してアクセスできます (シスコでは、サービスコントロール レポート ツール用の情報として、加入者の使用状況、ネットワーク リソース分析、およびトラフィック分析についてのレポートを生成できるテンプレートベースのレポート ツールを提供しています。『Cisco Service Control Application Reporter User Guide』を参照してください)。

データベース メンテナンスは、オペレーティング システムのコマンドとスクリプトを使用して実行されます。CM は、古いレコードをバンドルされているデータベースから自動消去する機能をサポートしています。デフォルトでは、レポート テーブルに保存された期間が 2 週間を超えるすべてのレコードは、自動的に消去されます。レコードは 1 時間おきにポーリングされます。データベース メンテナンスを設定するには、`dbperiodic.py` ユーティリティ スクリプトを使用します。詳細については、「[古いレコードの定期削除の管理](#)」(p.5-3) を参照してください。

外部データベースの使用方法

JDBC 準拠データベース (Oracle や MySQL 等) を JDBC アダプタとともに CM で使用することができます。この場合、ローカルまたはリモートでデータベースを使用することができます。データベースを使用するように JDBC アダプタを設定し、データベース パックを設定して CM にデータベースのパラメータ (IP アドレスやポート等) を供給します。また、このデータベース用の JDBC ドライバを用意して、データベースに接続するときにアダプタで使用する必要があります。外部データベースと連動した CM の設定の詳細については、「[データベースおよび CSV リポジトリの管理](#)」(p.5-1) を参照してください。



Collection Manager のインストール

この章では、Collection Manager (CM) とそのデータベースのインストール手順、および CM の起動方法について説明します。

- [システム要件 \(p.3-2\)](#)
- [Collection Manager のインストール \(p.3-7\)](#)
- [Sybase データベースおよびサービス コントロール ソフトウェアのアンインストール \(p.3-13\)](#)
- [CM バージョン 3.1 へのアップグレード \(p.3-14\)](#)

システム要件

CM およびそのデータベースは、サーバ プラットフォームで稼働するソフトウェア コンポーネントです。以下の設定のいずれかにインストールすることができます。

- Solaris 9 または Solaris 10 が稼働する Sun SPARC マシン (「Solaris の要件」 [p.3-3] を参照)
- Red Hat Enterprise Linux 3.0 または Red Hat Enterprise Linux 4.0 が稼働する IA32 マシン (「Red Hat Linux の要件」 [p.3-5] を参照)
- システムの前提条件のチェック (p.3-2)
- Solaris の要件 (p.3-3)
- Red Hat Linux の要件 (p.3-5)
- ディストリビューションの内容 (p.3-6)
- デフォルト設定 (p.3-6)

システムの前提条件のチェック

CM ディストリビューションには、`install_scripts` ディレクトリにスクリプト `check_prerequisites.sh` が含まれていて、システムが CM またはバンドルされている Sybase データベースをインストールするための要件に合致しているかどうかを判別するのに役立ちます。

スクリプトが、CM または Sybase をインストールできるかどうかを全体的にチェックします。チェックされる主な前提条件は次のとおりです。

- CPU の速度
- RAM の容量
- オペレーティングシステムのバージョン (Solaris 9 または 10、Red Hat Enterprise Linux 3.0 または 4.0)
- 追加の必須またはオプション パッケージ
- パス内でインストールされ実行可能な Python
- CM および Sybase ホーム用の空き容量
- 全 NIC の名前
- Sybase カーネル パラメータ
- ロケールおよび時間帯フォーマット

```
check_prerequisites.sh [ --sybhome=SYBHOME ] [ --cmhome=CMHOME ] [ --datadir=DATADIR ]
```

表 3-1 check_prerequisites.sh スクリプト オプション

<code>--sybhome=SYBHOME</code>	Sybase インストールのホーム ディレクトリ
<code>--datadir=DATADIR</code>	Sybase データ ファイルのデータ ディレクトリ (Datadir インストールメソッド用)
<code>--cmhome=CMHOME</code>	CM インストールのホーム ディレクトリ

Solaris の要件

Collection Manager 3.1.0 以降を、以下のセクションで列挙されている要件に適合する Solaris が動作する Sun SPARC マシンにインストールすることができます。

- [ハードウェア \(p.3-3\)](#)
- [ソフトウェアおよび環境 \(p.3-3\)](#)
- [ロケールおよび時間帯の設定 \(p.3-5\)](#)

ハードウェア

- 500 MHz 以上の CPU
- CPU につき 1 GB 以上の RAM
- ハード ディスク :
 - 18 GB 以上のハード ディスク × 1
 - (バンドル インストール用の推奨) Sybase データを格納するための 2 台めのハードディスク (18 GB 以上)
- 100 BaseT ネットワーク インターフェイス

ソフトウェアおよび環境

- Solaris 5.9 64 ビット ビルド 04/01 以上 (現時点では Solaris 5.9 および 5.10 だけをサポート)
- Solaris Core Installation
- 次の追加パッケージをインストールする必要があります。

システム	SUNWbash	GNU Bourne-Again シェル (bash)
システム	SUNWgzip	GNU Zip (gzip) 圧縮ユーティリティ
システム	SUNWzip	Info-Zip (zip) 圧縮ユーティリティ
システム	SUNWlibC	Sun Workshop Compilers Bundled libC
システム	SUNWlibCx	Sun WorkShop Bundled 64-bit libC

- bundled モードで CM と Sybase を併用する場合は、次のパッケージもインストールする必要があります。

システム	SUNWipc	Interprocess Communication (IPC; プロセス間通信)
------	---------	--

- (任意) 次のパッケージをインストールできます (sys-unconfig などの sysadmin アプリケーション用)

システム	SUNWadmap	システム管理アプリケーション
システム	SUNWadmc	システム管理コア ライブラリ

- Python スクリプトを使用するには、Python インタープリタ バージョン 2.2.1 以上がシステムになければなりません。このインタープリタを入手するには、次のパッケージをインストールするという方法があります。

アプリケーション	SMCpython (Solaris 9)	Python
	SMCpython (Solaris 10)	

- Python パッケージは、次の2つの追加パッケージをインストールする必要があります。

アプリケーション	SMClibgcc	libgcc
アプリケーション	SMCncurs	ncurses

- これらのパッケージは、<http://sunfreeware.com/> からダウンロードできます。
ルート (/) パーティションには、これらのパッケージをインストールするために 104 MB の空き容量が必要です。
- Sun の最新の推奨パッチを適用する必要があります。
 - Solaris 9 については、<http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/xos-9&nav=pub-patches> を参照してください。
 - Solaris 10 については、<http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/xos-10&nav=pub-patches> を参照してください。
 - Java については、<http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/J2SE> を参照してください。

- Sybase を使用する場合は、Sybase が推奨する最新の Solaris パッチをインストールする必要があります。
 - <http://my.sybase.com/detail?id=1016173> を参照してください。

- CM をインストールするパーティションに、8 GB 以上の空き容量が必要です (CSV の保存および永続的バッファに使用します)。

- (バンドルされている Sybase をインストールする場合) Sybase ホーム ディレクトリ用に 1 パーティションに 3 GB 以上の空き容量が必要です。

- (バンドルされている Sybase をインストールする場合) 1 パーティションに、Sybase データおよびログを保持するための適切な空き容量が必要です (これらのサイズはインストール時に設定できます)。

- (任意、およびバンドルされている Sybase をインストールする場合のみ) sudo パッケージを (たとえば <http://sunfreeware.com> から) インストールして、sudoers ファイルに次の行を設定します。

```
scmscm ALL= NOPASSWD:XXX/scripts/dbconf.sh
```

XXX は scmscm のホーム ディレクトリです。

sudo をインストールしない場合: まれに Sybase がクラッシュした場合に、CM が自動回復できなくなります。

- (従来の (3.0 以前の) Service Control Application Suite (SCAS) Reporter を使用する予定で、バンドルされている Sybase をインストールする場合) SCA Reporter が認証できるように、いずれかの FTP サーバをポート 21 で待ち受ける必要があります。

- (バンドルされている Sybase をインストールする場合) /etc/hosts またはその他のアクティブなネーミング サービス内で、マシン NIC に設定されたすべての IP アドレスにホスト名が関連付けられていることを確認してから、インストールする必要があります (これは Sybase Adaptive Server Enterprise の制限です)。

- (Sybase をインストールする使用する場合) カーネルに少なくとも以下を設定する必要があります。

- shmmax に 512,000,000 バイト

- さらに、起動時に IPC モジュールをロードする必要があります。そのためには、ファイル /etc/system に次の行を挿入します。

```
forceload:sys/semsys
forceload:sys/shmsys
```

- データベースの定期削除を使用する場合、scmscm ユーザは cron ジョブをスケジューリングし、実行できる必要があります。

ロケールおよび時間帯の設定

- CM および Sybase を正しく動作させるには、US English ロケールを使用する必要があります。ロケールを設定する最も簡単な方法は、`/etc/TIMEZONE` コンフィギュレーション ファイルに次の行を挿入することです（このファイルの変更を有効にするには、再起動する必要があります）。

```
LANG=en_US
```

また、Solaris にもこのロケールをインストールする必要があります。ディレクトリ `/usr/lib/locale/en_US` が存在するかを調べることで、このロケールがインストールされていることを確認します。このディレクトリが存在しない場合は、Solaris CD からロケール ファイルをインストールします。

- OS 時間帯 を POSIX 形式で GMT とのオフセットとして設定しないでください。問題が生じることがあります。`/etc/TIMEZONE` コンフィギュレーション ファイル内で、次のように（サポートされている）国名で時間帯を設定する方法を推奨します。

```
TZ=Japan
```

ディレクトリ `/usr/share/lib/zoneinfo` 内にその国名が表示されているかを調べることで、国名が時間帯設定としてサポートされているか確認します。

GMT オフセットを使用しなければいけない場合、次のように `:Etc/` 接頭辞を付加することで `zoneinfo` フォーマットを使用します。

```
TZ=:Etc/GMT+5
```

Red Hat Linux の要件

Collection Manager 3.1.0 以降を、以下のセクションで列挙されている要件に適合する Red Hat Linux が動作する i386 にインストールすることができます。

- ハードウェア (p.3-5)
- ソフトウェアおよび環境 (p.3-5)
- ロケールおよび時間帯の設定 (p.3-6)

ハードウェア

- 800 MHz 以上の CPU
- CPU につき 1 GB 以上の RAM
- ハード ディスク：
 - 18 GB 以上のハード ディスク × 1
 - （バンドル インストール用の推奨）Sybase データを格納するための 2 台めのハード ディスク（18 GB 以上）
- 100 BaseT ネットワーク インターフェイス

ソフトウェアおよび環境

- Red Hat Linux 3.0 または 4.0
- Red Hat Enterprise 「Base」のインストール
- （バンドルされている Sybase を使用する場合）次の追加パッケージをインストールする必要があります。

```
compat-libstdc++
```

- このパッケージは、Red Hat インストール CD に収録されています。

- Red Hat の最新の推奨パッチを適用する必要があります。
- (バンドルされている Sybase を使用する場合) Sybase が推奨する最新の Solaris パッチをインストールする必要があります。
- CM をインストールするパーティションに、8 GB 以上の空き容量が必要です (CSV の保存および永続的バッファに使用します)。
- (バンドルされている Sybase をインストールする場合) Sybase ホーム ディレクトリ用にいずれかのパーティションに 1 GB 以上の空き容量が必要です。(任意、およびバンドルされている Sybase をインストールする場合のみ) sudo パッケージをインストールして、sudoers ファイルに次の行を設定します。

```
scmscm ALL= NOPASSWD:XXX/scripts/dbconf.sh
```

 XXX は scmscm のホーム ディレクトリです。
 sudo をインストールしなかった場合、まれに Sybase がクラッシュすると、CM は自動回復できなくなります。
- (従来の (3.0 以前の) SCAS Reporter を使用する予定で、バンドルされている Sybase をインストールする場合) SCA Reporter が認証できるように、いずれかの FTP サーバをポート 21 で待ち受ける必要があります。
- (バンドルされている Sybase をインストールする場合) /etc/hosts またはその他のアクティブなネーミング サービス内で、マシン NIC に設定されたすべての IP アドレスにホスト名が関連付けられていることを確認してから、インストールする必要があります (これは Sybase Adaptive Server Enterprise の制限です)。
- (Sybase をインストールする使用する場合) カーネルに少なくとも以下を設定する必要があります。
 - shmmax に 512,000,000 バイト
- データベースの定期削除を使用する場合、scmscm ユーザは cron ジョブをスケジューリングし、実行できる必要があります。

ロケールおよび時間帯の設定

- CM および Sybase を正しく動作させるには、US English ロケール (en_US) を使用する必要があります。

ディストリビューションの内容

CM インストール キットには、CM および Sybase データベースをインストールするためのインストール スクリプトが収録されています。

これには、次のものも含まれています。

- サポート ファイル収集用スクリプト
- 定期的な Sybase メンテナンス用スクリプト

デフォルト設定

CM の設定は、インストール時に行われます。これらの設定は、イネーブルにする必要があるアダプタ、アダプタの場所、プライオリティ キューのパラメータ、RDR タイプ (RDR タグ値別) ごとのターゲット アダプタ、およびさまざまなロギング ポリシーなどです。これらの設定は、資格のある担当者だけが行ってください。

Collection Manager のインストール

このセクションでは、CM バージョン 3.1.0 以降および Sybase データベースを Solaris または Red Hat Linux が稼働するコンピュータにインストールする方法について説明します。

- [Collection Manager ソフトウェアで使用されるポート \(p.3-7\)](#)
- [Sybase データベースのインストール \(p.3-8\)](#)
- [Collection Manager ソフトウェアのインストール \(p.3-9\)](#)




Collection Manager ソフトウェアで使用されるポート

次の表では、CM ソフトウェアおよび関連コンポーネント (Sybase データベースなど) が待ち受ける TCP/UDP ポートを説明します。このテーブルは、ネットワーク管理者がソフトウェアの動作、およびソフトウェアとセキュリティ ポリシーとの関係を理解する場合に役立つことがあります。

下記のポートは、デバイスが常に待ち受けているポートです。これらのポート番号に対してアクセスを許可しておきます。許可がない場合、処理内容においては失敗することがあります。

処理 (ファイル転送など) によっては、記載されたポート以外のポートをデバイスが一時的に開くことがあります。処理が終了すると自動的にポートが閉じます。

表 3-2 CM が継続的に待ち受けるポート

ポート番号	説明
33000	データ収集用 RDR を送信するために、SCE デバイスが使用します。
21	CM マシンの CM ユーザを認証するために、従来の (3.0 以前の) SCAS Reporter が使用します。
33001	内部 CM  (注) アクセスが必要なのは、ローカル マシンからのみです。外部アクセスの場合はブロックできます。
9092	HTTP 技術者用インターフェイス
4100	(バンドルされている Sybase インストール用) ODBC/JDBC による Sybase データベース接続。データベースへのアクセスが必要です。
1099 ~ 1120	RMI。データ コレクタとサービス コントロール管理サーバ間の管理インターフェイスとして使用します。
22000	CM の FTP サーバ  (注) データ転送の場合、FTP トランザクションはプロトコルによるネゴシエーションに従って、その他のポート (22001 ~ 22100) で待ち受けることがあります。
7787	管理ユーザ ログの内部ログ  (注) アクセスが必要なのは、ローカル マシンからのみです。外部アクセスの場合はブロックできます。
14375	シンボル定義 (values.ini) を CM に送信するために、Cisco Service Control Application Suite for Broadband (SCAS BB) Console で使用します。

Sybase データベースのインストール

Sybase をインストールしない場合 (unbundled モードで作業している場合) は、「[Collection Manager ソフトウェアのインストール](#)」(p.3-9) を参照してください。



(注)

インストールの任意の時点で、Sybase インストール作業を元に戻す場合は (まれに、停電によってインストールが中断された場合など)、次の手順を実行してください。

1. root ユーザとしてログオンします。
2. `kill -u sybase` を入力して、すべての Sybase プロセスを終了します。
3. `userdel -r sybase` を入力して、Sybase ユーザおよびホーム ディレクトリを削除します。
4. Sybase インストール プロセスを最初からやり直します。

installsyb.sh によって実行される動作

`installsyb.sh` スクリプトは、Sybase データベースをインストールします。このスクリプトは、次の動作を実行します。

- `/etc/system` で Sybase の `shmem` 設定を確認します。この設定が存在しない場合は、スクリプトで設定が挿入され (ユーザに確認プロンプトを表示したあと) 再起動されます。
- ユーザ `sybase` およびグループ `sybase` を追加します。
- 使用しているプラットフォームの Sybase インストーラを起動します。
- Sybase ユーザおよびパスワードを含む Sybase サーバを構築します。
- Sybase を開始します。
- Collection Manager データベース構造を作成する SQL スクリプトを実行します。この作業は Sybase を何度か再起動する必要があり、完了まで長時間かかります。

前提条件

root ユーザとしてログオンして、システムまたはローカル ネットワーク上で利用可能なディストリビューション キット コンテンツを作成します。

サマリー手順

1. ディレクトリをディストリビューション キット ルートの `sybase` に変更します。
2. スクリプト `installsyb.sh` を実行します。

詳細手順

ステップ 1 ディレクトリをディストリビューション キット ルートの `sybase` に変更します。

ステップ 2 スクリプト `installsyb.sh` を実行します。

スクリプトの使用法は、次のとおりです。

```
installsyb.sh --sybhome=SYBHOME{ --datadir=DATADIR}
```

- `SYBHOME` は Sybase ユーザのホーム ディレクトリです (1 GB の空き容量が必要)。

- 以下のデータ場所オプションを選択します。
 - `--datadir=DATADIR` を指定します。DATADIR は、すべての Sybase データが格納されるディレクトリです。
このディレクトリを含むパーティションには、15 GB 以上の空き容量が必要です。
- DATADIR を指定した場合、すべての Sybase データは通常のファイルとして該当ディレクトリに格納されます。デフォルト サイズは、データ用に 10 GB、ログ用に 3 GB、Sybase テンポラリストレージ用に 3 GB です。ディレクトリの所有権は、インストール中に Sybase ユーザに変更されます。

Collection Manager ソフトウェアのインストール



(注)

インストールの任意の時点で、Sybase インストール作業を元に戻す場合は(まれに、停電によってインストールが中断された場合など)、次の手順を実行してください。

1. root ユーザとしてログオンします。
2. `pkill -u sybase` を入力して、すべての Sybase プロセスを終了します。
3. `userdel -r sybase` を入力して、Sybase ユーザおよびホーム ディレクトリを削除します。
4. Sybase インストール プロセスを最初からやり直します。

install-cm.sh スクリプト

`install-cm.sh` スクリプトは、Collection Manager Server をインストールするのに使用されます。

install-cm.sh オプション

`install-cm.sh` スクリプトの使用法に関するメッセージは、次のとおりです。

```
Usage: install-cm.sh [-h] (-d CMDIR | -o)
Options: -d CMDIR   select directory for ~scmscm
              (must not exist and must be on 8 GB free partition)
         -o         upgrade the existing installation
              while preserving the current configuration
              (can't be used with -d)
         -h         print this help and exit
```

Description of the options:

```
-d CMDIR
Used to designate the directory of the newly created
scmscm user's home. Should be the name of a
non-existing directory, whose parent resides on a
partition where at least 8 GB is free.
As an alternate to this option, you can specify -o :
-o
Use this option when you wish to upgrade the existing
installation while preserving the current configuration.
(can't be used with -d)
```

install-cm.sh によって実行される動作

install-cm.sh スクリプトは、次の手順を実行します

- 必要に応じて、scmscm ユーザおよび scmscm グループを作成します。
- オプションとしてこのユーザのホームを作成します。
- scmscm のホームに CM ファイルおよびスクリプトを読み込みます。
- 次の追加コンポーネントをインストールします。
 - ~scmscm/cm/lib にプライベート JRE
- /etc/init.d および /etc/rcX.d 内に sybase および scmscm ユーザのブート スクリプト シンボリック リンクを作成します。

サマリー手順

1. ディストリビューション キットのルートにある install-scripts ディレクトリに移動します。
2. install-cm.sh スクリプトを実行します。
3. スクリプトの完了後、scmscm ユーザのためのパスワードを設定します。
4. Topper/Aggregating (TA) アダプタに割り当てられるメモリ サイズを拡張します。
5. ファイル ~scmscm/cm/config/cm.conf を開きます。
6. [adapter_mem] セクション内で、TAAdapter を含む設定を検索します。
7. デフォルト値 (512 MB) を、より大きな値に変更します。
8. ファイルを保存して、閉じます。
9. Real-Time Aggregating (RAG) アダプタに割り当てられるメモリ サイズを拡張します。
10. ファイル ~scmscm/cm/config/cm.conf を開きます。
11. [adapter_mem] セクション内で、RAGAdapter を含む設定を検索します。
12. デフォルト値 (512 MB) を、より大きな値に変更します。
13. ファイルを保存して、閉じます。
14. アプリケーションが使用する各アダプタに対して、アプリケーションを示すようにアダプタを設定します。
15. データベース テーブルの定期削除手順をインストールし、アクティブにします。
16. 定期削除手順をインストールします。
17. 定期削除手順の自動呼出しをアクティブにします。
18. Service Control Engine (SCE) デバイスの時間帯を設定します。
19. CM を起動します。

詳細手順

ステップ 1 ディストリビューション キットのルートにある install-scripts ディレクトリに移動します。

ステップ 2 install-cm.sh スクリプトを実行します。

install-cm.sh スクリプト オプションの詳細については、「install-cm.sh オプション」(p.3-9) を参照してください。

スクリプトに関する追加情報については、「install-cm.sh によって実行される動作」(p.3-10) を参照してください。

ステップ3 スクリプトの完了後、scmscm ユーザのためのパスワードを設定します。

次のコマンドを実行して、scmscm ユーザのパスワードを設定します。

```
passwd scmscm
```

選択したパスワードを記録します。

ステップ4 TA アダプタに割り当てられるメモリ サイズを拡張します。

TA アダプタを利用するアプリケーションを実行する場合は、このアダプタに割り当てられたメモリ サイズを拡張しなければならない場合があります。拡張が必要かどうかは、CM が処理する加入者数によって決まります。メモリ割り当てを拡張するには、次の手順を実行します。

- a. ファイル `~scmscm/cm/config/cm.conf` を開きます。
- b. `[adapter_mem]` セクション内で、`TAAdapter` を含む設定を検索します。
- c. デフォルト値 (512 MB) を、より大きな値に変更します。
たとえば、1,024 MB のメモリを割り当てる場合は、値 `-Xmx1024M` を使用します。
- d. ファイルを保存して、閉じます。

ステップ5 RAG アダプタに割り当てられるメモリ サイズを拡張します。

RAG アダプタを利用するアプリケーションを実行するときには、このアダプタに割り当てられたメモリ サイズを拡張しなければならない場合があります。拡張が必要かどうかは、CM が処理する加入者数、および RAG アダプタの設定によって決まります。設定を変更するには、次の手順を実行します。

- a. ファイル `~scmscm/cm/config/cm.conf` を開きます。
- b. `[adapter_mem]` セクション内で、`RAGAdapter` を含む設定を検索します。
- c. デフォルト値 (512 MB) を、より大きな値に変更します。
たとえば、1,024 MB のメモリを割り当てる場合は、値 `-Xmx1024M` を使用します。
- d. ファイルを保存して、閉じます。



(注)

外部データベースを使用するには、データベースの接続先 CM がイネーブルになるように、`dbpack` を設定する必要もあります。これを実行する方法の詳細については、「[データベースおよび CSV リポジトリの管理](#)」(p.5-1) を参照してください。

ステップ6 アプリケーションが使用する各アダプタに対して、アプリケーションを示すようにアダプタを設定します。

- JDBC アダプタ ファイル `~scmscm/cm/config/jdbcadapter.conf` を編集し、`[app]` セクションの値 `app_conf_dir` を、目的のアプリケーションを示すように変更します。
デフォルトでは、この値は `apps/scasbb/3.1.0` に設定されています。
TA アダプタ ファイル `~scmscm/cm/config/taadapter.conf` を編集し、`[app]` セクションの値 `app_conf_dir` を、目的のアプリケーションを示すように変更します。デフォルトでは、この値は `apps/scasbb/3.1.0` に設定されています。

ステップ7 データベース テーブルの定期削除手順をインストールし、アクティブにします。

(定期削除の動作の設定に関する詳細については、「古いレコードの定期削除の管理」[p.5-3] を参照してください。)



(注)

レポートがデータベースに送信されて定期削除手順をインストールし、アクティブにしていない場合、2 番目のディスクがオーバーフローする可能性があります。

a. 定期削除手順をインストールします。

scmscm ユーザとしてログオンし、CM を起動して、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.
```

b. 定期削除手順の自動呼出しをアクティブにします。

次のコマンドを実行します。

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

ステップ8 SCE デバイスの時間帯を設定します。

次のコマンドを使用して、時間帯を設定します。

```
~scmscm/cm/bin/jselect-sce-tz.sh --offset=offset-in-minutes from GMT
```

たとえば、SCE デバイスが GMT+2 の場所にある場合は、次のコマンドを使用します。

```
~scmscm/cm/bin/jselect-sce-tz.sh --offset=120
```

SCE が GMT-10 の場所にある場合は、次のコマンドを使用します。

```
~scmscm/cm/bin/jselect-sce-tz.sh --offset=-600
```



(注)

(サマータイムへの移行時に時間帯をアップデートする場合など) SCE の時間帯がアップデートするたびに、スクリプトを実行する必要があります。

ステップ9 CM を起動します。

次のコマンドを実行して、CM を起動します。

```
~scmscm/cm/bin/cm start
```

Sybase データベースおよびサービス コントロール ソフトウェアのアンインストール

- [Sybase のアンインストール \(p.3-13\)](#)
- [サービス コントロール ソフトウェアのアンインストール \(p.3-13\)](#)

Sybase のアンインストール

サマリー手順

1. root ユーザとしてログインします。
2. Sybase のアンインストール
3. `/etc/system` を編集して、Sybase の `shmem` 設定を削除します。

詳細手順

ステップ 1 root ユーザとしてログインします。

ステップ 2 Sybase のアンインストール

次のコマンドを実行して、Sybase をアンインストールします。

```
pkill -u sybase userdel -r sybase rm /etc/rc*.d/[SK]*sybase
```

ステップ 3 `/etc/system` を編集して、Sybase の `shmem` 設定を削除します。

サービス コントロール ソフトウェアのアンインストール

サマリー手順

1. root ユーザとしてログインします。
2. サービス コントロール ソフトウェアのアンインストール

詳細手順

ステップ 1 root ユーザとしてログインします。

ステップ 2 サービス コントロール ソフトウェアのアンインストール

次のコマンドを実行して、サービス コントロール ソフトウェアをアンインストールします。

```
pkill -u scmscm userdel -r scmscm rm /etc/rc*.d/[SK]*scmscm
```

CM バージョン 3.1 へのアップグレード

サマリー手順

1. CM を停止します。
2. `install-cm.sh` スクリプトを使用して新規 CM をインストールします。

詳細手順

ステップ 1 CM を停止します。

ステップ 2 `install-cm.sh` スクリプトを使用して新規 CM をインストールします。

アップグレードする場合は、`-o` オプションを使用して既存の設定を保存します。

現在の `scmscm` ユーザが使用されます。

アップグレード後最初に CM を起動する際に、3.1 の新規データベーステーブルが自動的に作成されます。



(注) バージョン 3.1 へのアップグレードは、バージョン 3.x からのみ実行できます。



Collection Manager 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) パラメータを表示およびアップデートする方法について説明します。

Telnet や SHH など CM を経由して接続するマシンは、CM をモニタし、管理するためのユーティリティ スクリプトを使用することができます。ユーティリティ スクリプトは CM のインストール ディレクトリにあります。

データベースおよび CSV リポジトリの管理については、「[データベースおよび CSV リポジトリの管理](#)」(p.5-1) を参照してください。

- [ユーティリティ スクリプトの使用法](#) (p.4-2)
- [CM の設定](#) (p.4-3)
- [カテゴライザの設定](#) (p.4-6)
- [システム状態のモニタ](#) (p.4-7)
- [ユーザ管理](#) (p.4-9)
- [仮想リンクの管理](#) (p.4-10)
- [CM のモニタ](#) (p.4-11)

ユーティリティ スクリプトの使用法

ここでは、ユーティリティ スクリプトの一般的な使用手順を示します。

- スクリプトを呼び出すには、特に明記している場合を除き、scmscm ユーザとしてログインします。これらのスクリプトを root ユーザとして実行しようとする、エラーが発生します。
- スクリプトの説明、およびすべてのフラグとパラメータの説明を表示するには、help フラグを指定してスクリプトを起動します。



(注)

help フラグはスクリプトごとに多少異なっています。CM の管理用スクリプトは `--help` を使用し、データベースの管理用スクリプトは `-h` を使用します。各スクリプトの定義を参照してください。

次に、`dbperiodic.py` スクリプトの説明を表示する例を示します。

```
>~scmscm/scripts/dbperiodic.py --helpUsage:
~scmscm/scripts/dbperiodic.py --load
load configuration from
/export/home/scmscm/db_maint/dbperiodic.conf
~scmscm/scripts/dbperiodic.py --loadfile=FILE
load configuration from FILE
~scmscm/scripts/dbperiodic.py --dump
print the current configuration in INI format to standard output
~scmscm/scripts/dbperiodic.py --help
print this help message
```



(注)

データ収集用ソフトウェアを制御およびモニタするためのスクリプトには、Python スクリプト言語を使用するものがあります。Python の詳細については、<http://www.python.org> を参照してください。

CM の設定

ユーティリティ スクリプトは以下の処理に使用します。

- 起動時にアクティブになるサーバの指定
- データベースの起動または終了
- アダプタの起動または終了
- Service Control Engine (SCE) 接続の切断

CM の設定に使用するスクリプトは、次のとおりです。

- `~scmscm/setup/on-boot.py`
- `~scmscm/scripts/adapterconf.py`
- `~scmscm/scripts/dbconf.sh`
- `~scmscm/scripts/sceconf.py`

データベースおよび CSV リポジトリの管理のスクリプトについては、「[データベースおよび CSV リポジトリの管理](#)」(p.5-1) を参照してください。

以下のファイルも、CM の設定に使用されます。

- **cm.conf** CM の一般設定、CM 起動時にオンになるアダプタが含まれています。「[アダプタのイネーブル化](#)」(p.4-4) を参照してください。
- **queue.conf** アダプタ キューの設定。特定のアダプタと関連づけられている RDR タグが含まれています。「[カテゴリ化の設定](#)」(p.4-6) を参照してください。

サーバのアクティブ化

起動時にサーバ (CM または Sybase) をアクティブにするように設定するには、`on-boot.py` スクリプトを使用します。

```
~scmscm/setup/on-boot.py --cm=flag --sybase=flag
```

次のシステム再起動時に変更が有効になります。



(注)

スクリプトをパラメータなしで実行して、各コンポーネントの現在の起動ステータスを確認します。

scmscm として次のスクリプトを実行し、CM を再起動します。

```
~scmscm/cm/bin/cm restart
```

表 4-1 on-boot.py オプション

<code>--cm={ on off }</code>	起動時に CM をアクティブまたは非アクティブにします。
<code>--sybase={ on off }</code>	起動時に Sybase サーバをアクティブまたは非アクティブにします。

次に、起動時に CM および Sybase がアクティブになるように設定する例を示します (これがスクリプトのデフォルト設定です)。

```
>~scmscm/setup/on-boot.py --cm=on --sybase=on
```

アダプタの制御

設定されたアダプタをシャットダウンまたは起動したり、現在稼働中の CM アダプタを表示するには、`adapterconf.py` スクリプトを使用します。

```
~scmscm/scripts/adapterconf.py--op=action[--adapter=adapter name]
```

表 4-2 adapterconf.py オプション

<code>--op=start</code>	adapter パラメータで指定されたアダプタを起動します。
<code>--op=stop</code>	adapter パラメータで指定されたアダプタをシャットダウンします。
<code>--op=list</code>	現在動作している CM アダプタを表示します。
<code>adapter=adapter name</code>	動作しているアダプタを識別します。start および stop アクションの場合だけ使用します。
<code>--help</code>	このスクリプトのオプションを表示します。

アダプタをシャットダウンするには、scmscm ユーザとして次のスクリプトを実行します。

```
~scmscm/scripts/adapterconf.py--op=stop--adapter=adapter name
```

アダプタを起動するには、scmscm ユーザとして次のスクリプトを実行します。

```
~scmscm/scripts/adapterconf.py--op=start--adapter=adapter name
```

アダプタのイネーブル化

`cm.conf` ファイルの該当行の先頭にあるコメント記号を削除することで、CM 起動時にアダプタをオンに定義することができます。

次に、CM 起動時に RAG アダプタをオンにするように定義する例を示します。

```
adapter.4 = com.cisco.scmscm.adapters.rag.RAGAdapter
```

次に、CM 起動時に CSV アダプタをオフにしたままにするように定義する例を示します。

```
#adapter.2 = com.cisco.scmscm.adapters.CSVAdapter
```



(注) `adapter.<number>` の値は、対応するアダプタの `queue.conf` ファイルに定義されている `adapter_id` パラメータ値と一致していなければいけません。

データベースの制御

CM データベースをシャットダウンまたは起動する場合、またはデータベースの動作ステータスを表示するには、`dbconf.sh` スクリプトを使用します。

```
~scmscm/scripts/dbconf.sh--op=action
```

このスクリプトは、バンドルされているデータベースでのみ使用することができます。



(注)

このスクリプトが動作するのは、sudo パッケージがインストールされている場合だけです。sudo をインストールしない場合に、Sybase を起動または終了するには、root ユーザとしてログインし、`/etc/init.d/sybase` スクリプトを実行する必要があります。

表 4-3 dbconf.sh オプション

<code>--op=start</code>	CM データベースを起動します。
<code>--op=stop</code>	CM データベースをシャットダウンします。
<code>--op=status</code>	データベースの現在の動作ステータスを表示します。

CM データベースをシャットダウンするには、scmscm ユーザとして次のコマンドを実行します。

```
~scmscm/scripts/dbconf.sh--op=stop
```

CM データベースを起動するには、scmscm ユーザとして次のコマンドを実行します。

```
~scmscm/scripts/dbconf.sh--op=start
```

SCE 接続の切断

特定の SCE で接続を切断するには、`sceconf.py` スクリプトを使用します。

```
~scmscm/scripts/sceconf.py--op=drop--ip=IP address
```

CM の HTTP アダプタが動作中の場合のみ、このスクリプトを使用することができます。

このスクリプトは、SCE 接続に関する情報を表示する場合にも使用されます ([SCE 接続の確認 \[p.4-12\]](#) を参照)。

表 4-4 sceconf.py オプション

<code>Adapter=IP address</code>	指定された IP アドレスの接続を切断します。
<code>--help</code>	このスクリプトのオプションを表示します。

SCE 接続を切断するには、scmscm ユーザとして次のコマンドを実行します。

```
~scmscm/scripts/sceconf.py--op=drop--ip=IP address
```

カテゴリザの設定

カテゴリザでは、RDR タグに従って各 RDR を分類します。アダプタの RDR タグを tags パラメータ（カンマで区切られた RDR タグの一覧）に追加することで、RDR が特定のアダプタにルーティングされます。この設定は、`queue.conf` ファイルに含まれています。

次に、RDR タグ 4042321920 および 4042321922 を Topper/Aggregator アダプタに送信するように設定する例を示します。

```
# Topper/Aggregator Adapter
[topper-hi]
adapter_id=3
priority=3
warning_size=40000
maximum_size=50000
tags=4042321920,4042321922
```



(注) `adapter_id` パラメータの値は、対応するアダプタの `cm.conf` ファイルに定義されている `adapter.<number>` と一致していなければいけません。

システム状態のモニタ

CM には、システムをモニタして、事前定義された、潜在的に問題を含む状態に対してアラートを発行する、小さくて拡張可能なフレームワークが含まれています。

CM のモニタに使用するスクリプトは、次のとおりです。

- `~scmscm/setup/monitor/setup-monitor.sh`
- `~scmscm/setup/monitor/monitor.sh`
- [定期チェッカのインストール \(p.4-7\)](#)
- [定期チェッカ スクリプト \(p.4-7\)](#)

定期チェッカのインストール

cron (定期スケジューラ) サブシステム内で、定期チェッカ スクリプト `monitor.sh` のエントリを作成 (または削除) するには、`setup-monitor.sh` スクリプトを使用します。

```
~scmscm/setup/monitor/setup-monitor.sh -a flag[-i flag]
```

表 4-5 setup-monitor.sh オプション

<code>-a { install uninstall }</code>	cron 内で <code>monitor.sh</code> のエントリを作成 / 削除します。
<code>-i { 30m 1h 12h 24h }</code>	<code>monitor.sh</code> を 30 分、1 時間、12 時間、または 24 時間毎に実行します。

次に、30 分ごとに実行するように `monitor.sh` をインストールする例を示します。

```
$ ./setup-monitor.sh -a install -i 30m
```

次に、`monitor.sh` をアンインストールする例を示します。

```
$ ./setup-monitor.sh -a uninstall
```

定期チェッカ スクリプト

- [定期チェッカ スクリプト \(p.4-7\)](#)
- [テスト \(p.4-8\)](#)

定期チェッカ スクリプト

定期チェッカ スクリプト `monitor.sh` は、実行中のシステムをさまざまな角度からモニタする一連のサブスクリプトを呼び出します。

```
~scmscm/setup/monitor/monitor.sh { -a | TEST_NAME } [ -v ] [ -d ]
```

このスクリプトは、可能であるもののコマンドラインで実行することを想定していません。テスト結果は、syslog サブシステムに送信され、`/var/log/messages` ファイルに記録されます。

表 4-6 monitor.sh オプション

<code>-a</code>	すべてのテストを実行します。
<code>TEST_NAME</code>	1 つ以上のテストの名前。テスト名はテスト ファイル名から先頭の桁と後ろの <code>.sh</code> を取り除いたものです。
<code>-v</code>	詳細モードの結果を出力しません (成功したテストを記録します)。
<code>-d</code>	結果を画面に出力しません (デフォルトで、結果は <code>syslog</code> に送信されます)。

実行されるテストは、以下のフォーマットで結果を返します。

```
STATUS: Message
```

- STATUS PASS または FAIL
- Message 短い通知ステータス メッセージ

たとえば、**FAIL:db "apricot" has only 1523 free blocks**

次に、使用可能なすべてのテストを実行して、システム出力を画面に出力する例を示します。

```
$ ./monitor.sh -d -aTest: 01free_db.sh. Status: PASS. Message: db apricot has 1532 free blocks
Test: 02cm_is_up.sh. Status: FAIL. Message: cm process is not running
```

次に、インストールされたデータベースに十分な空き容量があることをチェックするための単一テストを実行する例を示します。

```
$ ./monitor.sh -d free_dbTest: 01free_db.sh. Status: PASS. Message: db apricot has 1532 free blocks
```

テスト

以下のテストは、`monitor.sh` を使用して実行することができます。

- `db_up` バンドルされている Sybase データベースが実行中かどうかをチェックします。
- `cm_up` CM アプリケーションが実行中かどうかをチェックします。
- `free_db` データベースに 10 % 以上の空き容量があるかどうかをチェックします。
- `free_log` データベース トランザクション ログに 70 % 以上の空き容量があるかどうかをチェックします。
- `cm_persistent_buffers` 各 CM アダプタの永続的バッファに含まれているファイル数が 500 ファイル未満かどうかをチェックします。

これらの全テストのスクリプトは、`~/setup/monitor/tests` ディレクトリにあります。

`test_name` というテストを呼び出す際に、スクリプトは `NNtest_name.sh` というファイルが検出されることを予想します。NN はスクリプト全体のプライオリティを指定したものです。たとえば、テスト `free_db` は、ファイル `01free_db.sh` にマッピングされます。

ユーザ管理

CM は、p3rpc ユーティリティを使用して、認証済の RPC コールについてユーザを管理します。

コマンドのフォーマットは、p3rpc OPERATION [OPTIONS] です。

次の表は、p3rpc の動作とオプションについてまとめたものです。

表 4-7 p3rpc の動作

動作	説明
<code>--set-user--username=username--password=password</code>	ユーザ名とパスワードを追加しアップデートします。
<code>--validate-password--username=username--password=password</code>	ユーザ名とパスワードを確認します。
<code>--delete-user--username=username</code>	ユーザ設定を削除します。
<code>--show-users</code>	すべての設定済ユーザを表示します。

仮想リンクの管理

スクリプトが CM ディストリビューションに含まれるため、特定の SCE に設定される仮想リンクおよびインデックスを管理できます。

仮想リンクを表示または設定するには、`update_vlinks.sh` スクリプトを使用します。

```
~scmscm/cm/bin/update_vlinks.sh--sce=SCE IP address[ --file=file| --show]
```

表 4-8 update_vlinks.sh オプション

<code>--sce=SCE IP--file=file</code>	指定された SCE の供給された CSV 形式ファイルにデータを持つ VLINK_INI テーブルをアップデートします。
<code>--sce=SCE IP--show</code>	指定された SCE のエントリに関して、VLINK_INI テーブルにクエリーを実行します。
<code>--help</code>	このスクリプトのオプションを表示します。

仮想リンクの詳細を設定するには、scmscm ユーザとして次のコマンドを実行します。

```
~scmscm/cm/bin/update_vlinks.sh--sce=SCE IP address--file=file
```

CSV ファイル形式は、link id (正の整数)、link direction (0 = アップストリーム、1 = ダウンストリーム)、name (文字列) です。

次の確認手順がファイル上で実行されます。

- ファイルが存在する。
- 各方向で仮想リンク ID が重複していない。
- 仮想リンク ID が 0 ~ 1024 の正の整数である。
- 方向は、0 (アップストリーム) または 1 (ダウンストリーム) である。
- 各方向で仮想リンク名が重複していない、または名前が空白ではない。
- 仮想リンク名が 256 文字以下で構成されている。34、39 (') および 96 (`) を除く、32 ~ 126 (含める) 間の ASCII コードによるすべてのプリント可能文字が使用されている。

ファイルが正常に確認されたあと、スクリプトは次の動作を実行します。

1. SCE IP フィールド内の SCE IP アドレスを含むすべてのエントリが、VLINK_INI テーブルから削除されます。
2. 次のフォーマットの 2 つのエントリが VLINK_INI テーブルに追加されます。
 - Timestamp, sce ip, 0, 0, "Default Virtual Link Up"
 - Timestamp, sce ip, 0, 1, "Default Virtual Link Down"
3. CSV ファイルは解析され、CSV ファイル内の各行は VLINK_INI テーブルの行エントリとして入力されます。

仮想リンクの詳細を表示するには、scmscm ユーザとして次のコマンドを実行します。

```
~scmscm/cm/bin/update_vlinks.sh--sce=SCE IP address--show
```

CM のモニタ

スクリプトを使用すると、CM に関連する、次のようなシステム統計情報をモニタできます。

- データベースの空き容量の割合
- CM に入力される RDR レート
- SCE プラットフォームの接続データ

CM のモニタに使用するスクリプトは、次のとおりです。

- `~scmscm/scripts/dbfree.sh`
- `~scmscm/scripts/rdr-rate.py`
- `~scmscm/scripts/sceconf.py`
- `~scmscm/setup/alive.sh`

次のスクリプトは CM を設定する場合に使用しますが（「[CM の設定](#)」 [p.4-3] を参照）、関連設定を表示する場合にも使用できます。

- `~scmscm/setup/on-boot.py`
- `~scmscm/scripts/adapterconf.py`
- `~scmscm/scripts/dbconf.sh`

データベース容量の確認

データベース レポート テーブルおよび関連するトランザクション ログ内の空き容量の割合を表示するには、`dbfree.sh` スクリプトを使用します。

```
~scmscm/scripts/dbfree.sh
```

このスクリプトは、バンドルされているデータベースでのみ使用することができます。

詳細手順

ステップ 1 `scmscm` ユーザとして、`dbfree.sh` スクリプトを実行します。

RDR レートの確認

CM に入力されるレポートの瞬間的な合計レートを表示するには、`rdr-rate.py` スクリプトを使用します。

```
~scmscm/scripts/rdr-rate.py
```

出力は、直前の 5 秒間に CM に着信した（すべての送信元からの）RDR の 1 秒間の合計レートを、単精度浮動小数点で表したものです。

CM の HTTP アダプタが動作中の場合のみこのスクリプトを使用することができます。

詳細手順

ステップ 1 `scmscm` ユーザとして、`rdr-rate.py` スクリプトを実行します。

SCE 接続の確認

SCE 接続に関する情報を表示するには、`sceconf.py` スクリプトを使用します。

```
~scmscm/scripts/sceconf.py --op=list
```

CM の HTTP アダプタが動作中の場合のみ、このスクリプトを使用することができます。

このスクリプトは、特定の SCE との接続を切断する場合にも使用します「SCE 接続の切断」(p.4-5)を参照してください。

詳細手順

ステップ 1 scmscm ユーザとして、`sceconf.py` スクリプトを実行します。

```
~scmscm/scripts/sceconf.py --op=list
```

例：

次に、SCE 接続の出力例を示します。

```
>~scmscm/scripts/sceconf.py --op=listIP                               Rate                               Peak
-----
10.1.6.93                      0.71798986                        0.718
10.1.9.36                      0.14420895                        0.1442139
10.1.9.35                      0.0                                0.027929332
10.1.12.11                     0.0                                0.0
```

サーバが動作可能であることの確認

サーバが適切に機能していることを確認するには、`alive.sh` スクリプトを使用します。

```
~scmscm/setup/alive.sh
```

スクリプトで、次のコンポーネントの動作が確認されます。

- CM
- データベース (バンドルされているデータベースの場合)
- レポートテーブル (バンドルされているデータベースの場合)

停止しているコンポーネントがある場合、スクリプトはエラーメッセージを発行します。

詳細手順

ステップ 1 scmscm ユーザとして、`alive.sh` スクリプトを実行します。



(注) 起動後、コンポーネントの初期化には時間がかかります。再起動後、5分待機してこのスクリプトを実行してください。



データベースおよび CSV リポジットリ 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 CSV リポジッ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 データベースにのみ適用可能です。



(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 スクリプトの使用法](#)」(p.4-2) 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.5-2)
- [バンドルされたデータベースの管理](#) (p.5-6)
- [CSV リポジットリの管理](#) (p.5-9)

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

- データベース テーブル リストの生成
- 古いファイルの定期削除に対するスケジュールの定義と適用

データベースに格納されているすべてのレコードに、Cisco Server Control Management Suite (SCMS) 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`

データベース テーブルのリスト化

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


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

適用可能な場合、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。

次の表は、`dbtables.sh` のスクリプト オプションについてまとめたものです。

表 5-1 `dbtables.sh` オプション

オプション	説明
<code>-l</code>	既存のテーブル名のみを表示します (統計情報なし)。
<code>-a</code>	リスト内の非レポート テーブルを含みます。
<code>-f</code>	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 データベースにのみ適用可能です。
<code>-t { sec_num }</code>	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。
<code>-h</code>	このヘルプ メッセージを出力し、終了します。

次に、`dbtables.sh` の出力例を示します。

```
>~scmscm/scripts/dbtables.shExecuting 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、次の一般的な手順を実行する必要があります。

- 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 ユーザとしてログオンし、CM を起動して、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.
```

- 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。
- `dbperiodic.py` ユーティリティ スクリプトを使用して新規設定を適用します。

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。



(注) 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(`/var/adm/messages`)に記録されます。



(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 でのみサポートされます。

- [定期削除の設定 \(p.5-4\)](#)
- [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 `~scmsem/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して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にすべてのフィールドが必要なわけではありません)。パラメータおよびその値は等号 (=) で区切ります。以下のテーブルに、定期削除コンフィギュレーション ファイルの例を示します。

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 = *
```

次の例では、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 `dbperiodic.py` オプション

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

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設定を出力します。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、実際のコンフィギュレーションとこのファイルの内容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ます。

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

次に、指定場所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ます。

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

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。

- 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

データベースに格納されているすべてのレコードに、Cisco Server Control Management Suite (SCMS) CM が 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 データベースのみ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/droptable.sh`
- `~scmscm/scripts/prunetable.sh`
- `~scmscm/scripts/sybback.sh`
- `~scmscm/scripts/sybrestore.sh`

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 droptable.sh オプション

オプション	説明
<code>table_name</code>	データベースから <code>table_name</code> を削除します。
<code>ALLTABLES</code>	データベースからすべてのテーブルを削除します。
<code>-f</code>	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)。
<code>-h</code>	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 prunetable.sh オプション

オプション	説明
<i>num_days</i>	削除されない最長保存期間 (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。

次に、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例を示します。

-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
```

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
```

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 ディレクトリに格納されます。

表 5-6 sybback.sh オプション

オプション	説明
-d <i>path_to_backup_directory</i>	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。
-h	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 sybrestore.sh オプション

オプション	説明
-d <i>path_to_restore_directory</i>	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。
-h	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` 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、マシン間で小さな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 の *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 アドレスが変更された Sybase サーバは、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
```


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 が出力した CSV ファイルのリポジトリを管理できます。これらのファイルは CSV アダプタによってディスクに書き込まれ、サービス プロバイダーの Operational Support System (OSS) またはサードパーティ製課金システムで使用されます。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 ファイルが削除されなくなります)。



(注) サードパーティ製アプリケーションに 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 アダプタが動作していなければいけません。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 \(p.5-11 \)](#)
- [プリント不可能文字のエスケープ設定 \(p.5-11 \)](#)

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、RDR タグの番号が設定されます (RDR タグは RDR タイプを示します)。各 RDR は、名前が RDR タグ番号と一致する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、`~scmscm/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ているかどうかの管理 (デフォルトでこのオプションはオフ)



(注) このスクリプトを使用する代わりに、`~scmscm/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のみ影響します。

表 5-8 csvconf.sh オプション

オプション	説明
<code>--list</code>	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。
<code>--clear</code>	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)。
<code>--maxlines=N</code>	CSV ファイルごとの RDR の最大数を N (1 ~ 20,000 の整数) に設定します。
<code>--maxfiles=M</code>	各サブディレクトリの CSV ファイルの最大数を M (10 ~ 10,000 の整数) に設定します。
<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 1000 に設定する例を示します。

```
>~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ます。

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

次に、リポジトリ内の古い 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に、カンマがフィールドの末尾を示さないように指定するには、エスケープシーケンスを使用します。

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つまたは複数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をバックスラッシュ (\) で置き換えます。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。この値の有効値は `backslash`、`quote`、または `url` です。

プリント不可能文字のエスケープ設定

オプションで、CSV アダプタでプリント不可能文字をエスケープすることができます。このオプションを有効化すると、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、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をバックスラッシュ (\) で置き換えます。

このオプションは、CSVAdapter ディレクトリ内の `csvadapter.conf` ファイル内で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。



データベースの設定

この章では、データベースと連動させるための Cisco Service Control Management Suite (SCMS) Collection Manager (CM) の設定方法、CM の機能を拡張させるためのデータベース インフラストラクチャの使用方法について説明します。

- [Oracle ユーザのクイック スタート ガイド \(p.6-2\)](#)
- [Velocity Template Language \(p.6-3\)](#)
- [データベース コンフィギュレーション ファイル \(p.6-4\)](#)
- [作業例 \(p.6-7\)](#)
- [テストおよびデバッグ \(p.6-10\)](#)
- [スクリプトでの JDBC フレームワークの使用法 \(p.6-11\)](#)
- [Oracle のスケーラビリティに関するヒント \(p.6-13\)](#)

Oracle ユーザのクイック スタート ガイド

CM とともに Oracle データベースを使用するには、Oracle が展開される IP アドレスやポートなどの基本的な接続パラメータを変更する必要があります。他の設定変更は不要です。次の手順では、必要な変更について説明しています。

1. CM が実行中の場合は、CM を停止します。
2. Oracle を使用するように JDBC アダプタを設定します。
 - a. テキスト エディタでファイル `~scmscm/cm/config/jdbcadapter.conf` を開きます。
 - b. スtring `db_template_dir` を検索します。

この String には、Sybase と Oracle に 1 つずつ、合計 2 つの行が含まれています。デフォルトでは、*Oracle* 行がコメント化されています。

- c. Oracle 行をコメント解除します。
- d. Sybase 行をコメント化します。
- e. 変更を保存します。

これは、以下のコード フラグメントで表現されています (変更後)。

```
#db_template_dir = dbpacks/sybase
db_template_dir = dbpacks/oracle/9204e/
```

3. Oracle を使用するように Topper/Aggregator (TA) アダプタを設定します。
 - a. テキスト エディタでファイル `~scmscm/cm/config/taadapter.conf` を開きます。
 - b. String `db_template_dir` を検索します。

この String には、Sybase と Oracle に 1 つずつ、合計 2 つの行が含まれています。デフォルトでは、*Oracle* 行がコメント化されています。

- c. Oracle 行をコメント解除します。
- d. Sybase 行をコメント化します。
- e. 変更を保存します。

これは、以下のコード フラグメントで表現されています (変更後)。

```
#db_template_dir = dbpacks/sybase
db_template_dir = dbpacks/oracle/9204e/
```

4. データベース接続パラメータを設定します。
 - a. テキスト エディタでファイル `~scmscm/cm/config/dbpacks/oracle/9204e/dbinfo.vm` を開きます。
 - b. 以下の行を変更して設定を反映させます。

```
#set ($dbinfo.options.host = "localhost")
#set ($dbinfo.options.port = "1521")
#set ($dbinfo.options.user = "pqb_admin")
#set ($dbinfo.options.password = "pqb_admin")
#set ($dbinfo.options.sid = "apricot")
```



(注) `dbinfo.vm` ファイルはシェル スクリプトではありません。ポンド記号 (#) は宣言の一部であり、コメント記号ではありません。

関連するパラメータは次のとおりです。

- Oracle がインストールされたマシンのホスト名または IP アドレス
- Oracle サーバが待ち受けしているポートの番号
- Oracle に対して認証するためのユーザ名およびパスワード
- CM で使用する既存の Oracle SID (サービス ID)

c. 変更を保存します。

5. CM を起動します。

Velocity Template Language

JDBC アダプタ フレームワークは Velocity Template Language (VTL) で記述されたマクロを使用して、データベース サーバに渡されるすべての SQL コードを生成します。以下のセクションでは、生成プロセスを制御するために使用されるコンフィギュレーション ファイルについて説明します。

Apache Jakarta Project の一部である VTL の完全な参照先については、Web サイト <http://jakarta.apache.org/velocity/vtl-reference-guide.html> を参照してください。

以下の表は、VTL 構造を簡単に説明したものです。

表 6-1 VTL 構造の概要

ディレクティブ	構文例	目的
#foreach	#foreach (\$item in \$collection) item is \$item #end	コレクション、配列、またはマップ に対して処理を反復します。
#if ... #else ... #elseif	#if (\$order.total == 0) No charge #end	条件ステートメント
#parse	#parse("header.vm")	指定されたテンプレートをロードお よび解析して、生成された出力に取り 込みます。
#macro	#macro(currency \$amount) \${formatter.currency(\$amount)} #end	新しいディレクティブおよび必要な すべてのパラメータを定義します。 結果は、あとでこのテンプレートで 使用する場合に解釈されます。
#include	#include("disclaimer.txt")	指定されたファイルをそのまま、生 成された出力に含めます。
#set	#set (\$customer = \${order.customer})	コンテキスト オブジェクトに値を割 り当てます。コンテキスト オブジェ クトが存在しない場合は、値が追加 されます。コンテキスト オブジェ クトが存在する場合は、値が置き換え られます。
#stop	#if (\$debug) #stop #end	テンプレート処理を停止します。

データベース コンフィギュレーション ファイル

データベース アクセス フレームワークを初期化する場合、最初に検索されるファイルは `main.vm` です。このファイルには、必要なすべてのデータベース SQL 定義または定義に対するポインタが格納されています。このファイルの検索場所は、CM で使用される `dbpack` によって決まります。`dbpack` は、特定のデータベース インストールに関連する一連のコンフィギュレーション ファイルです。(このコンフィギュレーション ファイルに関連づけられた) アダプタが `dbpack` を選択します。たとえば、次に示す `jdbcadapter.conf` ファイルの一部は、Oracle `dbpack` と連携するようにアダプタを設定します。

```
db_template_dir = dbpacks/oracle/9204e/
db_template_file = main.vm
```



(注) ディレクトリの場所は、メインの CM 設定ディレクトリ (通常は `~scmscm/cm/config`) に対して相対的に解釈されます

設定をモジュール化するには、`main.vm` ファイルがその他のファイルをポイントします。ただし、モジュール化が厳密に必要なわけではありません。このファイルには任意の定義を格納し、スクリプト内などで、あとで使用することができます。一部の定義は JDBC アダプタの動作に使用されるため、必須です。次の表に、これらの定義を示します。

表 6-2 必須 VM 定義

オブジェクト名	必須の定義
<code>\$table.sql.dropTable</code>	これらの設定は、テーブルごとに、指定の処理に対する SQL の生成方法を制御します。
<code>\$table.sql.createTable</code>	
<code>\$table.sql.createIndexes</code>	
<code>\$table.sql.insert</code>	
<code>\$table.sql.metaDataQuery</code>	
<code>\$dbinfo.driverjarfile</code>	JDBC ドライバの場所およびクラス名
<code>\$dbinfo.driver</code>	
<code>\$dbinfo.cmdSeparator</code>	複数の SQL ステートメントを区切るために使用されるパターン
<code>\$dbinfo.url</code>	データベースに接続するための URL、および任意の接続プロパティ
<code>\$dbinfo.connOptions</code>	

VTL 解析コンテキストに含まれる CM 設定を表すオブジェクトのうちのいくつかは、テンプレートで使用できるようになっています。これらのオブジェクトについては、次のセクションで説明します。

- [コンテキスト オブジェクト \(p.6-5\)](#)
- [アプリケーションの設定 \(p.6-6\)](#)

コンテキスト オブジェクト

VM テンプレートをロードして、CM コンポーネント (TA または JDBC アダプタ、スクリプトなど) で解析する前に、次の Java オブジェクトを使用して解析コンテキストを初期化します。

- [tables オブジェクト \(p.6-5\)](#)
- [dbinfo オブジェクト \(p.6-5\)](#)
- [tools オブジェクト \(p.6-5\)](#)

tables オブジェクト

tables オブジェクトは、データベースに格納する必要がある RDR の構造、データベース テーブルの構造とその格納場所、および CM で使用できるその他のデータベース テーブルの構造など、アプリケーションに関連するデータベース設定について説明します。このオブジェクトは、CM で使用されるデータベース テーブルの 1 つを各行で表す配列です。各テーブルの行には、次の情報を格納できます (一部のテーブルに関連しない項目もあります)。

- 論理名
- 物理名
- このテーブルに関連付けられた RDR タグ
- それぞれ次の属性を持つ、このテーブル内のフィールドまたはカラムのリスト
 - フィールド ID
 - フィールド名
 - フィールド固有のタイプ
 - 自由形式のフィールド オプション
- それぞれ次の属性を持つ、このテーブルのインデックス リスト
 - インデックス名
 - インデックス付きカラム名
 - 自由形式のインデックス オプション

tables オブジェクトの内容は、テンプレートをロードするときに調べたり、操作することができます。tables オブジェクトは、アプリケーション固有の XML コンフィギュレーション ファイルを使用して初期化されます。「[アプリケーションの設定](#)」(p.6-6)を参照してください。

dbinfo オブジェクト

dbinfo オブジェクトは、データベース接続を開くときに使用されるパラメータ、使用される SID やスキーマなど、データベース固有の設定について説明します。このオブジェクトには、データベース固有の設定オプションが保持されます。保持される情報は、次のとおりです。

- このデータベースのドライバとして使用される JDBC クラス名
- ドライバが格納された JAR ファイルの名前
- JDBC URL として表されたデータベースの場所
- 認証データ (ユーザおよびパスワード) などの自由形式の JDBC 接続オプション

tools オブジェクト

tools オブジェクトは、テンプレートを開発する場合や、コンテキスト データ構造を操作する場合に役立つユーティリティ メソッドをいくつか含むコンテナです。

オブジェクトのメソッドを `$tools.method(arg1, ..., argN)` として呼び出すことができます。`method` はメソッド名です。

次の表に、含まれているメソッドを示します。

表 6-3 tools オブジェクトのメソッド概要

メソッド名と引数	機能
<code>getTableByName (allTables, name)</code>	論理名が <code>name</code> に対応するデータベース テーブル オブジェクトを検出します。
<code>getTableByDbTabName (allTables, name)</code>	物理名が <code>name</code> に対応するデータベース テーブル オブジェクトを検出します。
<code>assignParams (sql, list_of_args)</code>	<code>sql</code> 文字列内の疑問符文字を、文字列で表される <code>list_of_args</code> パラメータ内の連続要素で置き換えます。このメソッドは、ベースとして JDBC Prepared Statement 文字列を使用する SQL 挿入ステートメントを作成するテンプレートで作業している場合に役立ちます。
<code>collapseWhitespace()</code>	複数の連続スペース文字からなるすべてのインスタンスを 1 つのスペースに変換して、先頭および末尾のスペースを削除します。このメソッドは、少なくとも 1 つの新規行と他のスペース文字が含まれている SQL が必要なデータベースで便利です (Sybase と Oracle はこれが不要です)。

これらのツールの使用例については、「[スクリプトでの JDBC フレームワークの使用法](#)」(p.6-11) を参照してください。

アプリケーションの設定

アプリケーションに関連するすべての設定は、次の項目が格納された特定のファイル (`tables.xml`) 内で行います。

- アプリケーションの名前およびバージョン
- 各データベース テーブルの名前とプロパティ、特に、データベース テーブルに格納されるアプリケーション RDR の構造
- 各データベース テーブル関連：
 - テーブルおよび RDR フィールドの名前および固有のタイプ
 - テーブル インデックスの名前およびプロパティ

この情報は、主にコンテキストを解析するテンプレートに `tables` オブジェクトを読み込む場合に使用されます。「[tables オブジェクト](#)」(p.6-5) を参照してください。

作業例

main.vm ファイルには、モジュール化をサポートするために他の VM ファイルの参照が含まれています（「データベース コンフィギュレーション ファイル」 [p.6-4] を参照）。名前が事前に決められている VM_global_library.vm ファイルを除き、これらの他のファイル名は任意です。定義が必要なすべてのマクロをこのファイルに格納して、マクロが正しい時期にロードされるようにする必要があります。この特別なファイルの詳細については、『Velocity User Guide』を参照してください。

以下のサンプルは、Oracle セットアップの main.vm のコンテンツを示したものです。

```
#parse ('dbinfo.vm')
#foreach ($table in $tables)
#set ($table.sql.dropTable = "#parse ('drop_table.vm')")
#set ($table.sql.createTable = "#parse ('create_table.vm')")
#set ($table.sql.createIndexes = "#parse ('create_indexes.vm')")
#set ($table.sql.insert = "#parse ('insert.vm')")
#set ($table.sql.metaDataQuery = "#parse ('metadata.vm')")
#end
```

このサンプルでは、必須のデータベースおよび SQL 定義（表 6-2）が別々のファイルに移動されていて、#parse ディレクティブを使用してロードおよび解析されます。

次のセクションでは、Oracle dbpack 内の各ファイルの内容を示します。定義の一部では、VM_global_library.vm ファイルに定義されたマクロを使用します。このファイルには、すべてのテンプレートで使用されるすべてのマクロ定義を格納する必要があります。

- [マクロの定義 \(p.6-7\)](#)
- [dbinfo の設定 \(p.6-8\)](#)
- [SQL の定義 \(p.6-8\)](#)

マクロの定義

次に、固有のタイプと SQL タイプ間のマッピングの定義例、およびリスト内の連続する要素間にカンマを挿入する optcomma マクロなどのユーティリティ マクロの定義サンプルを示します。

```
#macro (optcomma)#if ($velocityCount >1),#end#end
#macro (sqltype $field)
#set ($maxStringLen = 2000)
#if ($field.type == "INT8") integer
#elseif ($field.type == "INT16") integer
#elseif ($field.type == "INT32") integer
#elseif ($field.type == "UINT8") integer
#elseif ($field.type == "UINT16") integer
#elseif ($field.type == "UINT32") integer
#elseif ($field.type == "REAL") real
#elseif ($field.type == "BOOLEAN") char(1)
#elseif ($field.type == "STRING") varchar2(#if($field.size <=
$maxStringLen)$field.size #else $maxStringLen #end)
#elseif ($field.type == "TEXT") long
#elseif ($field.type == "TIMESTAMP") date
#end
#end
```

dbinfo の設定

以下のコード サンプルでは、URL および接続オプション（認証用）が唯一の必須フィールドです。コードの空白行は、読みやすくし、またあとで設定変更をしやすいするために、コードを個別のフィールドに分離するものです。

```
#set ($dbinfo.driver = "oracle.jdbc.OracleDriver")
#set ($dbinfo.driverjarfile = "ojdbc14.jar")
#set ($dbinfo.options.host = "localhost")
#set ($dbinfo.options.port = "1521")
#set ($dbinfo.options.user = "pqb_admin")
#set ($dbinfo.options.password = "pqb_admin")
#set ($dbinfo.options.sid = "apricot")
#set ($dbinfo.url =
"jdbc:oracle:thin:@$dbinfo.options.host:$dbinfo.options.port:$dbinfo.options.sid")
#set ($dbinfo.connOptions.user = $dbinfo.options.user)
#set ($dbinfo.connOptions.password = $dbinfo.options.password)
## the vendor-specific piece of SQL that will return the current
## date and time:
#set ($dbinfo.options.getdate = "sysdate")
```

SQL の定義

- 「drop table」のコード (p.6-8)
- 「create table」のコード (p.6-8)
- 「create indexes」のコード (p.6-9)
- 「insert」のコード (p.6-9)
- メタデータクエリーのコード (p.6-9)

「drop table」のコード

次のサンプル コードは、通常の SQL 構文を使用してテーブルを削除します。

```
drop table $table.dbtablename
```

「create table」のコード

次のサンプル コードは、通常の SQL 構文を使用してテーブルを作成します。テーブル作成用の特殊なディレクティブが必要となる任意のカスタマイズ済みデータベース設定は、この定義を使用して実装できます。たとえば、何らかの一意な tablespace 内にテーブルを作成したり、テーブルをパーティション化するように、この定義を変更することができます。

```
create table $table.dbtablename (
#foreach ($field in $table.fields)
#optcomma()$field.name #sqltype($field)
#if ("${field.options.notnull}" == "true")
not null
#end
#end)
#end)
```

「create indexes」のコード

次のコードは、通常の SQL 構文を使用してインデックスを作成します。インデックス作成用の特殊なディレクティブが必要となる任意のカスタマイズ済みデータベース設定は、この定義を使用して実装できます。たとえば、何らかの一意的な tablespace 内にインデックスを作成するように、この定義を変更することができます。

```
#foreach ($index in $table.indexes)
create index $index.name on $table.dbtabname ($index.columns)
#end
```

「insert」のコード

次のコードは、テーブル構造に対応する JDBC Prepared Statement イディオムを作成します。

```
insert into ${table.dbtabname} (
#foreach ($field in $table.fields)
#optcomma() ${field.name}
#end)
values (
#foreach ($field in $table.fields)
#optcomma()?
#end)
```

メタデータ クエリーのコード

次のコードは、テーブルのメタデータ（カラム名およびタイプ）を取得するために使用されるシンプルなクエリーを定義します。空の結果セットを戻すクエリーを使用することができます。

```
select * from ${table.dbtabname} where 1=0
```

テストおよびデバッグ

データベースの一連のテンプレートを作成する場合に、解析結果を直接参照できると便利です。この機能を実現するために、JDBC アダプタは CM メイン スクリプト `~scmscm/cm/bin/cm` による直接呼び出しをサポートしています。

このような呼び出しの一般構文は、次のとおりです。

```
~/cm/bin/cm invoke com.cisco.scmscm.adapters.jdbc.JDBCAdapter argument
```

`argument` は、次のセクションで説明されるフラグの 1 つです。このメカニズムは、CM が動作しているかどうかに関係なく使用することができます。

また、次のセクションに記載されたクエリーおよび更新の実行メソッドを使用すると、動作中のデータベースに対するテンプレート結果をテストできます。

- [文字列の解析 \(p.6-10\)](#)
- [完全デバッグ情報の取得 \(p.6-10\)](#)

文字列の解析

すべての文字列は、コンテキスト全体が適切な場合、VTL テンプレートとして解析できます。解析結果は、標準出力に表示されます。文字列を解析するには、`-parse` フラグを指定して、アダプタを呼び出します。次に、例をいくつか示します (応答は太字で示されています)。

```
$ ~/cm/bin/cm invoke com.cisco.scmscm.adapters.jdbc.JDBCAdapter -parse 'xxx'

xxx

$ ~/cm/bin/cm invoke com.cisco.scmscm.adapters.jdbc.JDBCAdapter -parse '$dbinfo.url'

jdbc:oracle:thin:@localhost:1521:apricot

$ ~/cm/bin/cm invoke com.cisco.scmscm.adapters.jdbc.JDBCAdapter -parse '$tools.getTableByName($tables, "LUR").sql.createTable'

create table RPT_LUR (
  TIME_STAMP      date
  ,RECORD_SOURCE  integer
  ,LINK_ID        integer
  ,GENERATOR_ID   integer
  ,SERVICE_ID    integer
  ,CONFIGURED_DURATION  integer
  ,DURATION       integer
  ,END_TIME       integer
  ,UPSTREAM_VOLUME  integer
  ,DOWNSTREAM_VOLUME  integer
  ,SESSIONS       integer
)
```

完全デバッグ情報の取得

テンプレートによって作成された `tables` および `dbinfo` 構造のすべての内容のダンプを表示するには、`-debug` フラグを使用します。このフラグを使用すると、これらの構造のすべてのフィールド、プロパティ、およびオプションの詳細が標準出力に出力されます。

スクリプトでの JDBC フレームワークの使用法

任意の SQL コマンドをデータベースに送信して実行したり、作成されたデータを表示したりすることができます。この方法は、定期的にデータベースをメンテナンスしたり、データベース テーブルの内容をモニタしたり、追加のデータベース テーブルを管理する場合などに役立つことがあります。

`update` 処理を実行するには、`-executeUpdate` フラグを指定してアダプタを呼び出します。クエリーを実行して、結果を表示するには、`-executeQuery` フラグを指定してアダプタを呼び出します。

- [SCE 時間帯オフセットの表示および設定サンプル \(p.6-11\)](#)

SCE 時間帯オフセットの表示および設定サンプル

次の `update` 処理のサンプルは、Service Control Engine (SCE) 時間帯オフセット設定を保持するデータベース テーブル内の値をプログラムで変更する方法を示しています。このテーブルの名前は、通常 `JCONF_SE_TZ_OFFSET` です。テーブルに別の名前が割り当てられている可能性があるため、ここでは論理名 `TZ` で呼びます。「[tables.xml ファイル](#)」(p.A-1) のリストを参照してください。

テーブルの有無を最初に確認してから更新する必要がないように、テーブルを削除してから (テーブルが存在しない場合のエラー ステータスは無視します) テーブルを再作成し、適切な値を挿入します。テーブルにはタイムスタンプ カラムが含まれているため、データベース内の現在日付を取得する必要があります。この処理は各データベース ベンダーで固有なため、この例ではテンプレート内で定義された設定済みの `getdate` 処理を呼び出しています。

ツール `assignParams` および `getTableByName` の使用法に注意して、SQL を生成してください。

```
#!/bin/bash
this=$0
tableName=TZ
usage () {
cat <<EOF
Usage:
$this --status      - show currently configured TZ offset
$this --offset=N    - set the offset to N minutes (-1440 <= N <= 1440)
$this --help        - print this message
EOF
}
query () {
~/cm/bin/cm invoke com.cisco.scmscm.adapters.jdbc.JDBCAdapter -executeQuery "$*"
}
update () {
~/cm/bin/cm invoke com.cisco.scmscm.adapters.jdbc.JDBCAdapter -executeUpdate "$*"
}
get_tz () {
query 'select * from $tools.getTableByName($tables, "TZ").dtablename'
}
set_tz () {
update '$tools.getTableByName($tables, "TZ").sql.dropTable'
update '$tools.getTableByName($tables, "TZ").sql.createTable'
update '$tools.assignParams($tools.getTableByName($tables, "TZ").sql.insert,
[$dbinfo.options.getdate, '$1'])'
}
case $1 in
--status)
get_tz
;;
--help)
usage
exit 0
;;
--offset=*)
n=$(echo $1 | egrep 'offset=[-]?[0-9]+$' | sed 's/.*=//')

```

■ スクリプトでの JDBC フレームワークの使用法

```
if [ "$n" ]; then
if [ "$n" -ge -1440 -a "$n" -le 1440 ]; then
set_tz $n &>/dev/null
ok=1
fi
fi
if [ ! "$ok" ]; then
usage
exit 2
fi
get_tz
;;
*)
usage
exit 3
;;
esac
```

実行したクエリーによって返された一連の結果は、適切なカラム ヘッダーを使用して、表形式で標準出力に表示されます。

Oracle のスケーラビリティに関するヒント

次の2つのセクションでは、CM でのデータベース処理のスケーラビリティを高める方法を示します。これらの方法は Oracle 特有であり、この機能が実現可能であることを示す単なるヒントとして示されています。

- [カスタム テーブルスペースの使用法 \(p.6-13\)](#)
- [テーブルパーティショニングの使用法 \(p.6-14\)](#)

カスタム テーブルスペースの使用法

複数のテーブルスペースが作成されていて、それらの間で CM テーブルを配信するとします。これを簡単に行うには、`tables.xml` ファイル内の各テーブルに対して使用するテーブルスペースを指定します。1 つのテーブルに対して、定義は次のようになります (特に太字のコードに注目してください)。

```
<rdrr name="LUR" dbtabname="RPT_LUR" tag="4042321925" createtable="true">
<options >
<option property="tablespace" value="tspace1" />
</options >
<fields>
<field id="1" name="TIME_STAMP" type="TIMESTAMP">
<!-- (other field declarations) -->
<field id="10" name="DOWNSTREAM_VOLUME" type="UINT32"/>
<field id="11" name="SESSIONS" type="UINT32"/>
</fields>
<indexes>
<index name="RPT_LUR_I1" columns="END_TIME">
<options>
<option property="clustered" value="true"/>
<option property="allowduprow" value="true"/>
<option property="tablespace" value="tspace2" />
</options>
</index>
</indexes>
</rdrr>
```

このサンプルでは、インデックスおよびテーブル自体のために、必要なテーブルスペース (`tspace1` および `tspace2`) が追加されています。CM 内のオプションの `tablespace` に意味は設定されていません。任意の新しいオプション名を使用できます。意味は、テンプレートでの今後の使用法に基づいて決定します。

正しいテーブルスペース内にテーブルを作成するには、`create_table.vm` を次のように変更します。

```
create table $table.dbtabname (
#foreach ($field in $table.fields)
#optcomma()$field.name #sqltype($field)
#if ("${field.options.notNull}" == "true")
not null
#end
#end)#if ("${table.options.tablespace}" != "") TABLESPACE $table.options.tablespace
#end
```

独自のテーブルスペース内にインデックスを作成するには、`create_indexes.vm` を次のように変更します。

```
#foreach ($index in $table.indexes)
create index $index.name on $table.dbtabname ($index.columns)
#if ("${index.options.tablespace}" != "") TABLESPACE $index.options.tablespace #end
#end
```

テーブルパーティショニングの使用法

週単位で特定のテーブルにローリングパーティショニングを実装するために、前のセクションの例のような `tables.xml` ファイル内に、テーブル用の `partitioned` オプションを作成することができます (「[カスタムテーブルスペースの使用法](#)」[p.6-13] を参照)。その後、次のような `create_table.vm` コードを追加します (特に太字のコードに注目してください)。

```
create table $table.dbtabname (
#foreach ($field in $table.fields)
#optcomma()$field.name #sqltype($field)
#if ("${field.options.notNull}" == "true")
not null
#end
#end)#if ("${table.options.partitioned}" != "") partition by range (timestamp)
(partition week_1 values less than (to_date ('01-JAN-2005 00:00:00', 'DD-MON-YYYY
HH24:MI:SS')), partition week_2 values less than (to_date ('08-JAN-2005
00:00:00', 'DD-MON-YYYY HH24:MI:SS')) partition week_3 values less than (to_date
('15-JAN-2005 00:00:00', 'DD-MON-YYYY HH24:MI:SS')) partition week_4 values less than
(to_date ('22-JAN-2005 00:00:00', 'DD-MON-YYYY HH24:MI:SS')) ); #end
```

Oracle では時間境界に非定数表現を指定できないため、テーブルを作成するときに値を組み込む必要があります。

`cron` ジョブを作成して、毎週パーティションをローリングする (古いパーティションを削除して、新しいパーティションを作成する) ことができます。この `cron` ジョブは、JDBC アダプタの CLI (コマンドライン インターフェイス) を呼び出し (「[スクリプトでの JDBC フレームワークの使用法](#)」[p.6-11] を参照) 適切な `alter table drop partition` および `alter table add partition` SQL コマンドを実行するスクリプトを実行します。



コード サンプル

この付録は、Cisco Service Control Management Suite (SCMS) Collection Manager (CM) および CM が受信するデータを処理するアダプタの設定に使用するファイルのサンプルで構成されています。

- [アプリケーションの設定 \(p.A-1\)](#)
- [アダプタの設定 \(p.A-5\)](#)

アプリケーションの設定

以下のセクションでは、データベース テーブルを設定するのに使用する XML ファイル (`tables.xml`) の一部と、XML ファイルの構造を確認するのに使用される DTD ファイルを示しています。

- [tables.xml ファイル \(p.A-1\)](#)
- [tables.dtd ファイル \(p.A-3\)](#)

tables.xml ファイル

次に、Cisco Service Control Application for Broadband `tables.xml` ファイルの一部を示します。

```
<?xml version="1.0" encoding="ISO8859_1"?>
<!DOCTYPE dbtabconf PUBLIC "-//P-Cube//Engage DB RDR Configuration 2.1.0//EN"
"dbtables.dtd">
<dbtabconf>
<fileversion>
...
</fileversion>
<application name="Engage" version="2.1"/>
<dbtables>
<rdr name="SUR" dbtabname="RPT_SUR" tag="4042321922" createtable="true">
<fields>
<field id="1" name="TIME_STAMP" type="TIMESTAMP">
<options>
<option property="source" value="timestamp"/>
</options>
</field>
<field id="2" name="RECORD_SOURCE" type="INT32">
<options>
<option property="source" value="recordsource"/>
</options>
</field>
<field id="3" name="SUBSCRIBER_ID" type="STRING" size="64"/>
<field id="4" name="PACKAGE_ID" type="INT32"/>
<field id="5" name="SERVICE_ID" type="INT32">
<options>
<option property="notnull" value="true"/>
</options>
```

```

</field>
<field id="6" name="MONITORED_OBJECT_ID" type="INT32"/>
<field id="7" name="BREACH_STATE" type="INT32"/>
<field id="8" name="REASON" type="INT32"/>
<field id="9" name="CONFIGURED_DURATION" type="INT32"/>
<field id="10" name="DURATION" type="INT32"/>
<field id="11" name="END_TIME" type="INT32"/>
<field id="12" name="UPSTREAM_VOLUME" type="UINT32"/>
<field id="13" name="DOWNSTREAM_VOLUME" type="UINT32"/>
<field id="14" name="SESSIONS" type="UINT32"/>
</fields>
<indexes>
<index name="RPT_SUR_I1" columns="END_TIME">
<options>
<option property="clustered" value="true"/>
</options>
</index>
</indexes>
</rdr>
<rdr name="LUR" dbtabname="RPT_LUR" tag="4042321925" createtable="true">
<fields>
<field id="1" name="TIME_STAMP" type="TIMESTAMP">
<options>
<option property="source" value="timestamp"/>
</options>
</field>
<field id="2" name="RECORD_SOURCE" type="INT32">
<options>
<option property="source" value="recordsource"/>
</options>
</field>
<field id="3" name="LINK_ID" type="INT32"/>
<field id="4" name="GENERATOR_ID" type="INT32"/>
<field id="5" name="SERVICE_ID" type="INT32"/>
<field id="6" name="CONFIGURED_DURATION" type="INT32"/>
<field id="7" name="DURATION" type="INT32"/>
<field id="8" name="END_TIME" type="INT32"/>
<field id="9" name="UPSTREAM_VOLUME" type="UINT32"/>
<field id="10" name="DOWNSTREAM_VOLUME" type="UINT32"/>
<field id="11" name="SESSIONS" type="UINT32"/>
</fields>
<indexes>
<index name="RPT_LUR_I1" columns="END_TIME">
<options>
<option property="clustered" value="true"/>
<option property="allowduprow" value="true"/>
</options>
</index>
</indexes>
</rdr>
<aggtable name="TOP_HOURLY" dbtabname="RPT_TOPS_PERIOD0" aggperiod="0">
<fields>
<field id="1" name="RECORD_SOURCE" type="INT32"/>
<field id="2" name="METRIC_ID" type="INT8"/>
<field id="3" name="SERVICE_ID" type="INT8"/>
<field id="4" name="TIME_STAMP" type="TIMESTAMP"/>
<field id="5" name="AGG_PERIOD" type="INT8"/>
<field id="6" name="SUBSCRIBER_ID" type="STRING" size="64"/>
<field id="7" name="CONSUMPTION" type="UINT32"/>

</fields>
<indexes>
<index name="RPT_TOPS_PERIOD0_I1" columns="TIME_STAMP">
<options>
<option property="clustered" value="true"/>
<option property="allowduprow" value="true"/>
</options>
</index>
</indexes>

```

```

</aggtable>
<table name="TZ" dbtabname="JCONF_SE_TZ_OFFSET">
<fields>
<field id="1" name="TIME_STAMP" type="TIMESTAMP"/>
<field id="2" name="OFFSET_MIN" type="INT16"/>
</fields>
</table>
</dbtables>
</dbtabconf>

```

テーブル (RDR テーブル、集計テーブル、または追加テーブル) ごとに、フィールド、インデックスなどが示されます。



(注)

テーブル、インデックス、またはフィールドには、テンプレートからアクセスできる任意のフリーテキスト オプションを設定できます。

XML ファイルは実行時に、次のセクションで示す単純な DTD に対して検証されます。

tables.dtd ファイル

次に、`tables.xml` 定義ファイルを検証するために使用する DTD ファイルを示します。

```

<?xml version="1.0" encoding="ISO8859_1"?>
<!ELEMENT dbtabconf (fileversion, application, db?, dbtables)>
<!ELEMENT fileversion (#PCDATA)>
<!ELEMENT application EMPTY>
<!ATTLIST application
name CDATA #REQUIRED
version CDATA #REQUIRED
>
<!ELEMENT db (options)>
<!ELEMENT dbtables (rdr*, aggtable*, table*)>
<!ELEMENT table (options?, fields, indexes?)>
<!ATTLIST table
name CDATA #REQUIRED
dbtabname CDATA #REQUIRED
createtable (true | false) "true"
inserttodb (true | false) "false"
>
<!ELEMENT aggtable (options?, fields, indexes?)>
<!ATTLIST aggtable
name CDATA #REQUIRED
dbtabname CDATA #REQUIRED
aggperiod CDATA #REQUIRED
createtable (true | false) "true"
>
<!ELEMENT rdr (options?, fields, indexes?)>
<!ATTLIST rdr
name CDATA #REQUIRED
dbtabname CDATA #REQUIRED
tag CDATA #REQUIRED
createtable (true | false) "true"
inserttodb (true | false) "true"
>
<!ELEMENT fields (field+)>
<!ELEMENT field (options?)>
<!-- the id attribute below is presumably a numeric index, but it is for future
use, we currently don't look at it, as the order is imposed in the XML -->
<!ATTLIST field
id CDATA #REQUIRED
name CDATA #REQUIRED

```

```
type CDATA #REQUIRED
size CDATA #IMPLIED
>
<!ELEMENT indexes (index+)>
<!ELEMENT index (options?)>
<!ATTLIST index
name CDATA #REQUIRED
columns CDATA #REQUIRED
create (true | false) "true"
>
<!ELEMENT options (option+)>
<!ELEMENT option EMPTY>
<!ATTLIST option
property CDATA #REQUIRED
value CDATA #REQUIRED
>
```

DTD および XML ファイルの場所および名前は、アダプタのコンフィギュレーション ファイル内でアダプタごとに個別に設定できます。

アダプタの設定

次のセクションでは、Real-Time Aggregation (RAG) アダプタの設定に使用するコンフィギュレーション ファイル (**ragadapter.conf**) および関連 XML ファイル (**ragadapter.xml**) を示します。

他のアダプタのコンフィギュレーション ファイルは、RAG アダプタ コンフィギュレーション ファイルと類似しています。RAG アダプタにのみ関連 XML ファイルがあります。

- [ragadapter.conf ファイル \(p.A-5\)](#)
- [ragadapter.xml ファイル \(p.A-5\)](#)

ragadapter.conf ファイル

RAG アダプタの一般的なメンテナンスは、ファイル `~scmscm/cm/config/ragadapter.conf` を使用して実行されます。次に、RAG アダプタ コンフィギュレーション ファイルのサンプルを示します。

```
#
# RAGAdapter main configuration file
#
[config]
xml_filename = ~/cm/config/ragadapter.xml
[housekeeper]
interval_sec = 10
[db]
operations_timeout = 60
batch_size = 10
transaction_size = 15
commit_interval = 6
blocking_connects = true
db_template_file = main.vm
db_template_dir = dbpacks/sybase/ase12.5.1
[app]
app_conf_file = dbtables.xml
app_dtd_file = dbtables.dtd
app_conf_dir = apps/scasbb/3.1.0
```

ragadapter.xml ファイル

次のコードは、RAG アダプタの設定サンプルです。

```
<?xml version="1.0"?>
<!DOCTYPE ragadapterconf [
<!ELEMENT ragadapterconf (fileversion, config)>
<!ELEMENT fileversion (#PCDATA)>
<!ELEMENT config (aggregations, sinks)>
<!ELEMENT aggregations (aggregation+)>
<!ELEMENT aggregation (bucketident, closures, accumulators, monitors)>
<!-- ATTENTION: aggregation
id CDATA #REQUIRED
intag CDATA #REQUIRED
outtag CDATA #REQUIRED
sinkid CDATA #REQUIRED
-->
<!ELEMENT bucketident (field+)>
<!ELEMENT closures (closure*)>
<!ELEMENT closure (closurespec+)>
<!-- ATTENTION: closure
field CDATA #REQUIRED
-->
<!ELEMENT closurespec (equivvalue+)>
<!-- ATTENTION: closurespec
type (string | int | long | double) #REQUIRED
primaryvalue CDATA #REQUIRED
-->
```

```

>
<!ELEMENT equivvalue EMPTY>
<!ATTLIST equivvalue
val CDATA #REQUIRED
>
<!ELEMENT accumulators (field+)>
<!ELEMENT monitors (changemonitor | maxmonitor | timeoutmonitor)*>
<!ELEMENT changemonitor EMPTY>
<!ATTLIST changemonitor
action (warn | checkpoint) #REQUIRED
field CDATA #REQUIRED
active (true | false) #REQUIRED
>
<!ELEMENT maxmonitor EMPTY>
<!ATTLIST maxmonitor
action (warn | checkpoint) #REQUIRED
field CDATA #REQUIRED
maxvalue CDATA #REQUIRED
active (true | false) #REQUIRED
>
<!ELEMENT timeoutmonitor EMPTY>
<!ATTLIST timeoutmonitor
action (warn | checkpoint) #REQUIRED
maxsec CDATA #REQUIRED
active (true | false) #REQUIRED
>
<!ELEMENT field EMPTY>
<!ATTLIST field
index CDATA #REQUIRED
type (string | int | long | double) #REQUIRED
>
<!ELEMENT sinks (csvsink | dbsink | generalsink)+>
<!ELEMENT csvsink EMPTY>
<!ATTLIST csvsink
id CDATA #REQUIRED
classname CDATA #REQUIRED
filenameformat CDATA #REQUIRED
dirname CDATA #REQUIRED
maxagesec CDATA #REQUIRED
maxlines CDATA #REQUIRED
usequotes (true | false) #REQUIRED
active (true | false) #REQUIRED
>
<!ELEMENT dbsink EMPTY>
<!ATTLIST dbsink
id CDATA #REQUIRED
classname CDATA #REQUIRED
active (true | false) #REQUIRED
>
<!ELEMENT generalsink EMPTY>
<!ATTLIST generalsink
id CDATA #REQUIRED
classname CDATA #REQUIRED
active (true | false) #REQUIRED
>
]>
<ragadapterconf>
<fileversion>
$File: ragadapter.xml $ $Revision: #3 $
$Author: ronv $
$DateTime: 2005/08/15 15:48:23 $
</fileversion>
<config>
<aggregations>
<aggregation id="NUR's by subscriber and subs usage counter"
intag="4042321920" outtag="71070" sinkid="csv1">
<bucketident>
<!-- SUBSCRIBER_ID=0, SUBS_USG_CNT_ID=2 -->
<field index="0" type="string"/>

```



```

<field index="2" type="int" />
</bucketident>
< closures>
< closure field="0">
< closurespec type="string" primaryvalue="GuyM">
< equivvalue val="RonK" />
< equivvalue val="OmerT" />
< equivvalue val="GuyM" />
< /closurespec>
< closurespec type="string" primaryvalue="OdedE">
< equivvalue val="NimrodR" />
< equivvalue val="YossiO" />
< equivvalue val="LironL" />
< /closurespec>
< /closure>
< closure field="2">
< closurespec type="int" primaryvalue="15">
< equivvalue val="5" />
< equivvalue val="6" />
< equivvalue val="7" />
< /closurespec>
< /closure>
< / closures>
< accumulators>

<!-- up=8, down=9, sessions=10 -->
<field index="8" type="long" />
<field index="9" type="long" />
<field index="10" type="long" />
< /accumulators>
<!-- nothing to monitor for change in NUR really.
For sake of testing, let's warn if DURATION changes.-->
< monitors>
< maxmonitor action="checkpoint" field="8" maxvalue="10000" active="true" />
< maxmonitor action="checkpoint" field="9" maxvalue="10000" active="true" />
< changemonitor action="warn" field="6" active="true" />
< timeoutmonitor action="checkpoint" maxsec="60" active="true" />
< /monitors>
< /aggregation>
< aggregation id="NUR's by subscriber only"
intag="4042321920" outtag="71071" sinkid="dbsink1">
< bucketident>
< field index="0" type="string" />
< /bucketident>
< closures />
< accumulators>
< field index="8" type="long" />
< field index="9" type="long" />
< field index="10" type="long" />
< /accumulators>
< monitors>
< timeoutmonitor action="checkpoint" maxsec="60" active="true" />
< /monitors>
< /aggregation>
< /aggregations>
< sinks>
< csvsink id="csv1"
classname="com.cisco.scmscm.adapters.rag.sinks.CSVSink"
filenameformat="yyyy-MM-dd_HH-mm-ss-SSS'.csv'"
dirname="~/cm/adapters/RAGAdapter/csvfiles"
maxagesec="300" maxlines="1000" usequotes="true" active="true" />
< dbsink id="dbsink1"
classname="com.cisco.scmscm.adapters.rag.sinks.JDBCSink" active="false" />
< /sinks>
< /config>
< /ragadapterconf>

```

