



CHAPTER 4

インストールとアップグレード

概要

この章では、Cisco Service Control Management Suite Subscriber Manager (SCMS SM) のインストール方法について、また、アップグレードとアンインストールの方法について説明します。この章では、インストール、アップグレード、およびアンインストールに関連するトピックも扱います。

Subscriber Manager のインストール、アップグレード、アンインストールの方法

この章では、Subscriber Manager のインストール、アップグレード、またはアンインストールの手順を説明します。

- 「SM のインストールに関する情報」 (P.4-1)
- 「Subscriber Manager のインストール」 (P.4-11)
- 「インストールの確認」 (P.4-18)
- 「Subscriber Manager の設定」 (P.4-19)
- 「追加のインストール手順を実行する方法」 (P.4-19)
- 「Subscriber Manager のアップグレード方法」 (P.4-20)
- 「Subscriber Manager のダウングレード方法」 (P.4-37)
- 「Subscriber Manager のアンインストール方法」 (P.4-37)

SM のインストールに関する情報

Subscriber Manager (SM) のインストールは自動プロセスです。このプロセスでは、シスコが提供する SM 配布ファイルのルートに存在するインストール スクリプトを実行します。



(注)

Solaris の場合：手順には、`/etc/system` ファイルの変更も必要です。この変更は手動で行うか、他の自動ユーティリティを使用します。



(注) Linux の場合：手順には、`/etc/sysctl.conf` ファイルの変更も必要です。この変更は手動で行うか、他の自動ユーティリティを使用します。



(注) Linux の場合：インストール手順を実行する前に、SELinux がディセーブルであることを確認します。SELinux に関してアップデートされた情報については、RedHat の Web サイトを参照してください。

- 「インストールの概要」(P.4-2)
- 「配布ファイルの内容」(P.4-2)
- 「ドキュメンテーション」(P.4-6)
- 「システム要件」(P.4-6)
- 「インストール手順」(P.4-8)
- 「インストール スクリプトによるシステム変更に関する情報」(P.4-9)
- 「高度なシステム メモリ設定に関する情報」(P.4-10)

インストールの概要

インストール手順によって、次のコンポーネントがインストールされます。

- SM および Command-Line Utilities (CLU; コマンドライン ユーティリティ)
- TimesTen データベースおよび Data Source Name (DSN; データ ソース名)
- Java Runtime Environment (JRE)
- SM Veritas Cluster Agent

インストール手順には、次の操作も含まれます。

- pcube ユーザおよびグループの設定
- 起動およびシャットダウン スクリプトの追加
- TimesTen のシステム設定（手動で実行またはスクリプトを使用）
- レプリケーション方式の設定（CLU を使用して実行）（クラスタ セットアップだけに関連）

インストールと設定が完了したら、SM を使用して加入者をシステムに通知できます。



(注) Login Event Generator および Service Control Application for Broadband (SCA BB) アプリケーションは現在、SM にデフォルトでインストールされます。

配布ファイルの内容

SCMS SM コンポーネントは、次の 3 つの配布ファイルで提供されます。

- SM for Solaris
- SM for Linux
- Login Event Generator (LEG)

各配布ファイルは tar ファイルとして提供されます。この tar ファイルは gzip で圧縮され、拡張子 `.tar.gz` が付いています。

表 4-1 に Solaris および Linux 対応の SM インストール配布ファイルの内容を示します。

表 4-1 SM 配布ファイルの内容

パス	名前	説明
DIST_ROOT	—	クロスプラットフォーム ファイル
	dhcp_forwarder.tar.gz	DHCP Forwarder ディストリビューション
	hooks.sh	アップグレードに関するユーザ定義機能
	install	標準インストール手順の説明
	install-forwarder.sh	DHCP Lease Query LEG と一緒に使用する DHCP Forwarder のインストール スクリプト
	install-sm.sh	SM インストール スクリプト
	install-vcs-agents.sh	Veritas Cluster Server (VCS) エージェント インストール スクリプト
	linux-def.sh	Linux 固有の定義 (Linux 配布ファイル内に限る)
	solaris-def.sh	Solaris 固有の定義 (Solaris 配布ファイル内に限る)
	MANIFEST	CD 情報
	p3sm.sh	起動およびシャットダウン スクリプト
	Prerequisites	最低限のシステム要件リスト
	sm-common.sh	汎用のインストール スクリプト
	sm-dist.tar.gz	SM ディストリビューション
	tt-sysconf.sh	TimesTen システム設定スクリプト
	uninstall-sm.sh	SM アンインストール スクリプト
	upgrade-sm.sh	SM アップグレード スクリプト
	vcs-agents-dist.tar.gz	VCS エージェント ディストリビューション
	install-def.cfg	SM のインストールまたはアップグレードの前に設定できる、いくつかの SM パラメータの定義が含まれています。
	verify-subscriber.sh	SM を実行せずに加入者が存在するかどうかを確認するためのスクリプト。クラスタアップグレードの実行時に使用します。
	cluster-upgrade.sh	クラスタ インストール用のアップグレード スクリプト

表 4-1 SM 配布ファイルの内容 (続き)

パス	名前	説明
DIST_ROOT/Java/	—	Java Runtime Environment ファイル
	jre1.5.0_14-linux.tar.gz	32 ビット対応の JRE (Linux 配布ファイル内に限る)
	jre1.5.0_14-linux64.tar.gz	Linux 64 ビット対応の JRE (Linux 64 ビット配布ファイル内に限る)
	LICENSE	JRE ライセンス
DIST_ROOT/TimesTen/	—	TimesTen ファイル
	pqb_resp_uninst.txt	TimesTen アンインストール用の応答ファイル
	pqb-odbc-ini.txt	Open Database Connectivity (ODBC) の定義
	pqb-response70.txt	TimesTen インストール用の応答ファイル
	pqb-sys-odbc-ini.txt	Open Database Connectivity (ODBC) の定義
	timesten70520.linux86.tar.Z	Linux 対応の TimesTen (Linux 32 ビット配布ファイル内に限る)
	timesten70520.linux86-64.tar.gz	Linux 対応の TimesTen (Linux 64 ビット配布ファイル内に限る)
timesten70500.sparc64.tar.Z	Solaris 64 ビット対応の TimesTen (Solaris 配布ファイル内に限る)	

表 4-1 SM 配布ファイルの内容 (続き)

パス	名前	説明
DIST_ROOT/scripts	.bash_profile	bash 用のユーザ プロファイル
	.cshrc	C シェル用のユーザ プロファイル
	scmssm.sh	bash 用のユーザ プロファイル ユーティリティ ファイル
	scmssm.csh	C シェル用のユーザ プロファイル ユーティリティ ファイル
	dropRep.sql	アップグレード手順時に使用する SQL ファイル
	install-dsn.sh	TimesTen DSN 設定スクリプト
	installjava.sh	JRE インストール スクリプト
	install-tt.sh	TimesTen インストール スクリ プト
	get_name.sql	アップグレード手順時に使用する SQL ファイル
	p3db	インストールおよびアップグ レード手順で使用するユーティ リティ ファイル
	deleteLastPqiInfo.sql	アップグレード手順時に使用する SQL ファイル
clu-common	インストールおよびアップグ レード手順で使用するユーティ リティ ファイル	

表 4-2 に、LEG 配布ファイルの内容を示します。

表 4-2 LEG 配布ファイルの内容

パス	名前	説明
DIST_ROOT	—	クロスプラットフォーム ファイ ル
	MANIFEST	配布情報
DIST_ROOT/bgp_leg	—	Border Gateway Protocol (BGP) LEG ファイル
	bgp_leg.tar.gz	BGP LEG ディストリビュー ション
	Install	LEG インストール手順の説明
	install-bgp-leg.sh	BGP LEG インストール スクリ プト
	linux-def.sh	Linux 固有の定義
	sm-common.sh	汎用のインストール スクリプト
	solaris-def.sh	Solaris 固有の定義

表 4-2 LEG 配布ファイルの内容 (続き)

パス	名前	説明
DIST_ROOT/cnr_leg	—	Cisco Network Register (CNR) LEG ファイル
	cnr-leg-dist.tar.gz	CNR LEG ディストリビューション
	Install	LEG インストール手順の定義
DIST_ROOT/sce_api	—	SCE 加入者 Application Programming Interface (API; アプリケーションプログラミングインターフェイス) ファイル
	Readme	API セットアップ手順の説明
	sce-java-api-dist.tar.gz	API ディストリビューション
DIST_ROOT/sm_api	—	SM API ファイル
	Readme	API セットアップ手順の説明
	sm-c-api-dist.tar.gz	C API ディストリビューション
	sm-java-api-dist.tar.gz	Java API ディストリビューション

ドキュメンテーション

SM インストール配布ファイルには、次のドキュメントが含まれています。

- 『Manifest』: 配布ファイルを作成する元となったすべてのコンポーネントのバージョンおよびビルド番号
- 『Install』: SCMS SM 標準インストール手順
- 『Prerequisites』: SM インストールに関する最低限のシステム要件

システム要件

SM は次のプラットフォームにインストールできます。

- Solaris: 64 ビットバージョンの Solaris 9 または Solaris 10 と 64 ビットバージョンの Java Virtual Machine 搭載の Sun SPARC マシン (64 ビット)。表 4-3 および表 4-4 を参照してください。
- Linux : 32 ビットまたは 64 ビットバージョンの Linux と 32 ビットまたは 64 ビットバージョンの Java Virtual Machine 搭載の、32 ビットまたは 64 ビットの CPU を備えた Intel ベースのマシン。表 4-3 および表 4-5 を参照してください。



注意

SM は専用のマシンで実行する必要があります。SM は Collection Manager と同じマシンで実行できません。

マシンは、表 4-3、表 4-4、および表 4-5 に示されているシステム要件を満たしている必要があります。



(注)

表 4-3 には、最低限の仕様が示されています。特定のパフォーマンスとキャパシティ要件を保証するため、これらの仕様を確認してください。

表 4-3 最低限のシステム ハードウェア要件

項目	要件
CPU	<ul style="list-style-type: none"> SUN SPARC、64 ビット、500 MHz 以上 (Solaris の場合) Intel プロセッサ、32 ビットまたは 64 ビット、1 GHz 以上 (Linux Red Hat の場合)
RAM	1 GB 以上 (表 4-6 または表 4-8 を参照)
ディスクの空き領域	<p>合計 3 GB 以上。内訳は次のとおりです。</p> <ul style="list-style-type: none"> VARDIR (SM データベース リポジトリ) をインストールするパーティションに 1 GB 以上の空き領域 PCUBEDIR (SM ファイル) をインストールするパーティションに 0.5 GB 以上の空き領域 /tmp をマウントするパーティションに 200 MB 以上の空き領域
ネットワーク インターフェイス	<p>設定にクラスタが含まれるかどうかによって、次のように異なります。</p> <ul style="list-style-type: none"> クラスタなし：1 つの 100BASE-T イーサネット クラスタあり：6 つの 100BASE-T イーサネット
CD-ROM ドライブ	推奨



(注) Veritas Cluster Server に関するハードウェアおよびソフトウェアのシステム要件については、「[Veritas Cluster Server](#)」(P.E-1) を参照してください。

表 4-4 Solaris システムのソフトウェア要件

項目	要件
OS	<p>Solaris 5.9 64 ビット以降：現在、64 ビット バージョンの Solaris 5.9 および 5.10 だけがサポートされています。</p> <p>Solaris Core Installation</p>
システム パッケージ	<p>必須：</p> <ul style="list-style-type: none"> SUNWbash : GNU Bourne-Again シェル (bash) SUNWgzip : GNU Zip (gzip) 圧縮ユーティリティ SUNWzip : Info-Zip (zip) 圧縮ユーティリティ SUNWlibC : Sun Workshop Compilers Bundled libC SUNWlibCx : Sun WorkShop Bundled 64 ビット libC (Solaris 10 には不要) super user do (sudo) パッケージ <p>オプション：</p> <ul style="list-style-type: none"> SUNWadmap : システム管理アプリケーション SUNWadmc : システム管理コア ライブラリ



(注) SUN 提供の最新パッチを適用することを強く推奨します。最新パッチは SUN のパッチ関連 Web サイトからダウンロードできます。

表 4-5 Red Hat システム ソフトウェア要件

項目	要件
OS	Red Hat Enterprise Linux AS/ES 4.0/5.0、現在、32 ビットバージョンと 64 ビットバージョンの両方がサポートされています。 Red Hat Core Installation
システム パッケージ	必須 : <ul style="list-style-type: none"> • GNU Bourne-Again シェル (bash-2.05b-29.i386.rpm) • GNU データ圧縮プログラム (gzip-1.3.3-9.i386.rpm) • ファイルの圧縮およびパッケージング ユーティリティ (zip-2.3-16.i386.rpm) • Red Hat Linux 6.2 下位互換性に対応する標準 C++ ライブラリ (compat-gcc-7.3-2.96.122.i386.rpm) • super user do (sudo) パッケージ C API との統合用 : <ul style="list-style-type: none"> • GNU cc および gcc C コンパイラ (gcc-3.2.3-20.i386.rpm) • GNU gcc コンパイラ用の C++ サポート (gcc-3.2.3-20.i386.rpm)



(注) Red Hat 提供の最新パッチを適用することを強く推奨します。



(注) 32 ビットバージョンと 64 ビットバージョンの両方の Linux がサポートされています。

インストール手順

SM 配布ファイルのルートにあるインストール スクリプトを実行することにより、すべてのインストールを実行できます。

ほとんどの場合、インストールの実行に必要なスクリプトは SM インストール スクリプトだけです。

インストール スクリプトは、実行中の重要な手順を説明するメッセージを表示します。これらのメッセージは、今後の参照用にシステム ログにも送信されます。システム ログメッセージの詳細については、「[スクリプトメッセージのロギング](#)」(P.4-9) を参照してください。

SM が稼働中のマシンや、SM がすでにインストールされている（稼動していない場合も含め）ディレクトリに SM をインストールしようとする、インストールに失敗し、SM をアップグレードするように要求されます。「[Subscriber Manager のアップグレード方法](#)」(P.4-20) を参照してください。

適用される特定のインストール手順は、必要な SM トポロジによって異なります。

スタンドアロン トポロジに対するインストール手順については、「[Subscriber Manager のインストール](#)」(P.4-11) を参照してください。

クラスタ トポロジに対するインストール手順については、「SM クラスタのインストール」(P.4-19) を参照してください。

インストール スクリプトによるシステム変更に関する情報

このセクションでは、SM のインストールによって自動的に適用されるシステム変更について説明します。SM のインストールによって専用のユーザとグループ、および起動とシャットダウンのスクリプトが追加されます。

- 「スクリプト メッセージのロギング」(P.4-9)
- 「pcube ユーザおよびグループ」(P.4-9)
- 「起動およびシャットダウン スクリプト」(P.4-9)
- 「ユーザ pcube 用の Bash および C シェル」(P.4-10)

スクリプト メッセージのロギング

スクリプト メッセージは、次のようにシステム ログに記録されます。

- Solaris の場合：インストール スクリプトはすべてのメッセージをシステム ログに記録します。システム ログ ファイルは通常、`/var/adm/messages` に保存されます。メッセージは、**user.info syslog** カテゴリに記録されます。
- Linux の場合：インストール スクリプトはすべてのメッセージをシステム ログに記録します。システム ログ ファイルは通常、`/var/log/messages` に保存されます。メッセージは、**user.info syslog** カテゴリに記録されます。

pcube ユーザおよびグループ

インストール時に **pcube** という名前のユーザが専用のグループと一緒に作成されます（まだ存在していない場合）。このユーザは、インストールされるすべての SM および CLU ファイルを所有します。このユーザのホーム ディレクトリは、インストール時に選択したインストール ディレクトリになります。セキュリティ上、このユーザは最初にロックされたパスワードと一緒に作成されます。新しいパスワードを割り当てる必要があります。

起動およびシャットダウン スクリプト

SM はブート時に実行レベル 2 で起動し、この実行レベルを離れると停止します（たとえば、マシンがシャットダウンされた場合など）。

インストーラにより、起動およびシャットダウン用に次のファイルがインストールされます。

- Solaris の場合：

```
-rwxr--r-- 1 root  other  /etc/init.d/p3sm
lrwxrwxrwx 1 root  other  /etc/rc0.d/K44p3sm ->/etc/init.d/p3sm
lrwxrwxrwx 1 root  other  /etc/rc1.d/K44p3sm ->/etc/init.d/p3sm
lrwxrwxrwx 1 root  other  /etc/rc2.d/S92p3sm ->/etc/init.d/p3sm
lrwxrwxrwx 1 root  other  /etc/rcS.d/K44p3sm ->/etc/init.d/p3sm
```

- Linux の場合：

```
-rwxr--r-- 1 root  other  /etc/init.d/p3sm
lrwxrwxrwx 1 root  other  /etc/rc0.d/K44p3sm ->/etc/init.d/p3sm
lrwxrwxrwx 1 root  other  /etc/rc1.d/K44p3sm ->/etc/init.d/p3sm
lrwxrwxrwx 1 root  other  /etc/rc2.d/S92p3sm ->/etc/init.d/p3sm
lrwxrwxrwx 1 root  other  /etc/rc3.d/S92p3sm ->/etc/init.d/p3sm
lrwxrwxrwx 1 root  other  /etc/rc5.d/S92p3sm ->/etc/init.d/p3sm
lrwxrwxrwx 1 root  other  /etc/rc6.d/K44p3sm ->/etc/init.d/p3sm
```

TimesTen インストーラでは、同様の起動およびシャットダウン スクリプトが作成されます。

ユーザ **pcube** 用の Bash および C シェル

SM は、`~pcube/sm/server/bin` にある CLU を使用して制御されます。このようなシェルプロファイルが存在しない場合、インストールおよびアップグレード スクリプトによって、ユーザ **pcube** のパス環境変数に CLU ディレクトリを設定するプロファイルが作成されます。

この操作は、`DIST_ROOT/scripts/` の下にある SM ディストリビューションからこれらのプロファイル スクリプトのコンテンツ (`scmssm * ファイル`) をコピーすることにより、手動で実行することもできます。

高度なシステム メモリ設定に関する情報

- 「[/var/TimesTen/sys.odbc.ini の設定](#)」 (P.4-10)
- 「[SM プロセス メモリ の設定](#)」 (P.4-10)

`/var/TimesTen/sys.odbc.ini` の設定

一部のインストールでは、必要に応じてデータベースを実行できるように TimesTen パラメータを変更する必要があります。ただし、デフォルト値が要件に適している場合は変更しないでください。

データベース サイズの設定

システムで 100,000 を超える加入者をサポートする必要がある場合、`sys.odbc.ini` ファイルの `[Pcube_SM_Repository]` セクションにある `PermSize` パラメータと `TempSize` パラメータの値を設定します。

[ステップ 2](#) のシステム メモリ設定の決定を参照してください。

たとえば、次のとおりです。

```
PermSize=500
TempSize=150
```



(注)

データベース サイズを変更する場合は、次の変更も行う必要があります。

Solaris : `/etc/system` ファイル内のパラメータ `shmsys:shminfo_shmmax` の値が、`PermSize` と `TempSize` の合計値よりも大きくなるように設定します。

Red Hat : `/etc/sysctl.conf` ファイル内のパラメータ `kernel.shmmax` の値が、`PermSize` と `TempSize` の合計値よりも大きくなるように設定します。

SM プロセス メモリ の設定

デフォルトでは、SM プロセスに 256 MB の RAM メモリが使用されます。ただし、一部のアプリケーション コンポーネントの設定では、SM プロセスが正しく機能するように、追加のメモリを割り当てる必要があります。必要なメモリ サイズ (メガバイト単位) を指定する環境変数

`PCUBE_SM_MEM_SIZE` の設定を利用して、SM プロセスに定義されたメモリ サイズを割り当てるよう、SM 起動スクリプトに指示します。

ユーザ **pcube** に対してこの環境変数のメモリ サイズ値を設定することも、ユーザ **pcube** のルート ディレクトリ (`~pcube/sm.sh`) にある `sm.sh` ファイルに必要なプロセス メモリ サイズを設定することもできます。

次の例は、SM プロセスに 512 MB のメモリ サイズを定義する `sm.sh` ファイルの行を示しています。

```
PCUBE_SM_MEM_SIZE=512
```



(注) **PCUBE_SM_MEM_SIZE** 環境変数が存在しない場合は、この行を **sm.sh** ファイルに追加する必要があります。

インストール スクリプトを実行する前に、**install-def.sh** スクリプト ファイル内に **PCUBE_SM_MEM_SIZE** を設定できます。これにより、正しい値が設定された状態で SM をインストールできます。

メモリ スワッピングによるパフォーマンス低下を防ぐため、マシンに SM プロセス、SM データベース、およびこのマシンで実行するその他すべてのアプリケーション用の十分な RAM があることを確認してください。

インストールに適したメモリ値を判断するには、[ステップ 2](#) のシステム メモリ設定の決定を参照してください。

Subscriber Manager のインストール

このセクションでは、Subscriber Manager のインストール方法について説明します。



(注) ハイ アベイラビリティ セットアップ ([「SM クラスタ」 \(P.2-13\)](#) を参照) では、SM Cluster VCS エージェントをインストールする必要があります。[「SM クラスタ エージェントのインストール」 \(P.4-20\)](#) を参照してください。

前提条件

[「システム要件」](#) のディスク スペース要件を満たしていることを確認します。

インストール手順

ステップ 1 配布ファイルを抽出します。

SM をインストールするには、インストールするマシン、またはそのマシンにマウントされているディレクトリに、配布ファイルをロードして抽出します。

- a. シスコの Web サイトから配布ファイルをダウンロードします。
- b. File Transfer Protocol (FTP; ファイル転送プロトコル) を使用して、配布ファイルを SM にロードします。
- c. **gunzip** コマンドを使用してファイルを解凍します。

```
gunzip SM_dist_<version>_B<build number>.tar.gz
```
- d. **tar** コマンドを使用して tar ファイルを抽出します。

```
tar -xvf SM_dist_<version>_B<build number>.tar
```

ステップ 2 システム メモリを設定を決定します。

最大加入者数に従って、システム メモリの設定要件を設定します。システム メモリを設定するには、Quota Manager なしと Quota Manager ありの 2 つの方法があります。

表 4-6 と表 4-7 に、サポートされる加入者の数に基づいたメモリ設定の推奨値を示します。これらの設定は、Quota Manager がディセーブルである場合に適用されます。

表 4-6 メモリ コンフィギュレーション パラメータと加入者数 : Quota Manager なし

最大加入者数	(TimesTen メモリ設定) 共有メモリ	(TimesTen メモリ設定) PermSize	(TimesTen メモリ設定) TempSize
100,000	512 MB	200 MB	100 MB
500,000	1024 MB	512 MB	256 MB
1,000,000	1280 MB	768 MB	256 MB
2,000,000	2048 MB	1536 MB	256 MB
3,000,000	2560 MB	2048 MB	256 MB
4,000,000	3328 MB	2816 MB	256 MB
5,000,000	3840 MB	3328 MB	256 MB
10,000,000	6912 MB	6400 MB	256 MB
15,000,000	9984 MB	9472 MB	256 MB
20,000,000	13312 MB	12800 MB	256 MB

表 4-7 SM プロセス メモリの設定とキャッシュ サイズ : Quota Manager なし

キャッシュ サイズ	SM プロセス メモリの設定
100,000	256 MB
500,000	512 MB
1,000,000	768 MB
2,000,000	1280 MB
3,000,000	1792 MB
4,000,000	2048 MB
5,000,000	2560 MB
10,000,000	5120 MB
15,000,000	7680 MB
20,000,000	10240 MB



(注) SCE の最大数は 200 です。



注意

表内の SM プロセス RAM は、次の条件で計算されています。

- SM ごとに 40 の SCE 接続 : SCE を追加するたびに、SM プロセス メモリの設定に 25 MB を追加します。
- SM に対する 20 の Proprietary Remote Procedure Call (PRPC; 専用のリモート プロシージャ コール) (SM API/CNR LEG) 接続 : 接続を追加するたびに SM プロセス メモリの設定に 25 MB を追加します。

- サービス コントロール ソリューションの仮想リンク機能を使用する場合は、100,000 の加入者を追加するたびに Perm Size 設定に 60 MB を追加します。



(注)

32 ビット Linux インストールの場合、追加の制限があります。

- 最大加入者数は 200 万です。
- SM プロセス メモリの設定の最大値は 1.8 GB です。
- TimesTen 共有メモリ設定と SM プロセス メモリ設定の合計値が 3 GB を超えないようにします。

表内の列の説明：

- 最大加入者数：SM がサポートする必要がある最大加入者数。
- キャッシュ サイズ：SM プロセスが維持する加入者レコード参照の数。デフォルト値は 100,000 レコードです。



(注)

デフォルトのキャッシュ サイズを拡大してパフォーマンスを向上することができますが、これには追加のメモリが必要です。デフォルト設定は、ほとんどのインストールに適しています。

- SM プロセス メモリの設定：SM プロセス自体に必要なメモリ設定。SM プロセス メモリの設定の追加情報については、表 4-7 を参照してください。
- TimesTen を正しく実行するために必要な設定。追加情報については、ステップ 3 の共有メモリの設定を参照してください。

上記の表に該当する最大加入者数がない場合は、「最大加入者数」の高いほうの近接値に指定された設定を使用してください。たとえば、加入者が 1,200,000 の場合は、2,000,000 加入者に指定された値を使用します。

表 4-8 と表 4-9 に、サポートされる加入者の数に基づいたメモリ設定の推奨値を示します。これらの設定は、Quota Manager がイネーブルである場合に適用されます。

表 4-8 メモリ コンフィギュレーション パラメータと加入者数：Quota Manager あり

最大加入者数	(TimesTen メモリ設定) 共有メモリ	(TimesTen メモリ設定) PermSize	(TimesTen メモリ設定) TempSize
500,000	1280 MB	768 MB	256 MB
1,000,000	1792 MB	1280 MB	256 MB
2,000,000	3072 MB	2560 MB	256 MB
3,000,000	4096 MB	3584 MB	256 MB
4,000,000	5376 MB	4864 MB	256 MB
5,000,000	6400 MB	5888 MB	256 MB
10,000,000	12032 MB	11520 MB	256 MB
15,000,000	17664 MB	17152 MB	256 MB
20,000,000	23552 MB	23040 MB	256 MB

表 4-9 SM プロセス メモリ の設定とキャッシュ サイズ : Quota Manager あり

キャッシュ サイズ	SM プロセス メモリ の設定
100,000	512 MB
500,000	512 MB
1,000,000	768 MB
2,000,000	1280 MB
3,000,000	1792 MB
4,000,000	2048 MB
5,000,000	2560 MB
10,000,000	5120 MB
15,000,000	7680 MB
20,000,000	10240 MB



(注)

SCE の最大数は 200 です。



注意

表内の SM プロセス RAM は、次の条件で計算されています。

- SM ごとに 20 の SCE 接続 : SCE を追加するたびに、SM プロセス メモリ の設定を 50 MB 追加します。
- SM に対する 20 の PRPC (SM API/CNR LEG) 接続 : 接続を追加するたびに SM プロセス メモリ の設定に 25 MB を追加します。
- サービス コントロール ソリューションの仮想リンク機能を使用する場合は、100,000 の加入者を追加するたびに Perm Size 設定に 60 MB を追加します。



(注)

32 ビット Linux インストールの場合、追加の制限があります。

- 最大加入者数は 2 百万です。
- SM プロセス メモリ の設定の最大値は 1.8 GB です。
- TimesTen 共有メモリ設定と SM プロセス メモリ設定の合計値が 3 GB を超えないようにします。

ステップ 3

共有メモリを設定します。

TimesTen では、オペレーティング システムのカーネル コンフィギュレーション ファイルを一部変更する必要があります。

- Solaris の場合、ファイル `/etc/system` を変更します。
- Linux の場合、ファイル `/etc/sysctl.conf` を変更します。

この変更で、SM マシンの共有メモリとセマフォ リソースをデフォルトよりも増やします。これらの変更に関する追加情報については、TimesTen の資料を参照してください。



(注)

スクリプトによって現在のファイル設定が「手動による変更」手順に記載された値に上書きされるため、**tt-sysconf.sh** スクリプトを実行する前に **/etc/system** または **/etc/sysctl.conf** ファイルを確認することを推奨します。現在のファイル設定の一部または全部を保持する場合は、手動による変更でコンフィギュレーション ファイルを編集してください。

自動または手動で変更します。

- 必要な変更を自動的に行うには、**tt-sysconf.sh** スクリプトを実行します。

ルート ユーザは、次に示すように、このスクリプト ファイルを引数なしで呼び出す必要があります。

```
# tt-sysconf.sh
```

- 必要な変更を手動で行うには、次の操作を行います。



(注)

SM で 100,000 を超える加入者をサポートする必要がある場合は、コンフィギュレーション ファイルを手動で編集する必要があります。共有メモリ サイズに影響するのは、システム サイズ要件だけです。システムに適した設定値を判断するには、[ステップ 2](#) のシステム メモリ設定の決定の表を参照してください。

- Solaris の場合、**/etc/system** ファイルに次の行を追加し、共有メモリ サイズを設定することで、必要な変更を手動で行います。

```
*---- Begin settings for TimesTen
set semsys:seminfo_semmni = 20
set semsys:seminfo_semmnl = 100
set semsys:seminfo_semmns = 2000
set semsys:seminfo_semmnu = 2000
set shmsys:shminfo_shmmax = 0x20000000
*---- End of settings for TimesTen
```

- Linux の場合、**/etc/sysctl.conf** ファイルに次の行を追加し、共有メモリ サイズを設定することで、必要な変更を手動で行います。

```
*---- Begin settings for TimesTen
kernel.shmmax = 536870912
kernel.sem = 250 32000 100 100
*---- End of settings for TimesTen
```

ステップ 4 **install-def.cfg** ファイルを編集します。



(注)

この手順は、SM インストールの実行時に任意で行います。ただし、パラメータ値のいずれかをデフォルト値に設定しない場合は、ファイルを編集することを推奨します。

install-def.cfg ファイルには、SM のインストールまたはアップグレードの前に事前に設定できるいくつかのパラメータが含まれています。これらのパラメータは、インストールまたはアップグレードルーチンによって関連する SM コンフィギュレーション ファイルにコピーされます。デフォルトでは、すべてのパラメータがコメントアウトされ、デフォルト値が使用されます。

ファイルには、次のパラメータが含まれます。

- max_subscribers_num

[SM Definitions] セクションにあります。SM がサポートする最大加入者数を定義します。最大加入者数を設定するには、このパラメータを使用するか、**p3sm.cfg** コンフィギュレーションファイルの **max_number_of_subscribers** パラメータを設定します。[Data Repository] セクション (P.A-17) を参照してください。

SM データベースに保存できる最大加入者数には制限があります。Solaris は 2000 万、Linux は 200 万の加入者がそれぞれ上限となります。

SM のデフォルト設定では、最大 200,000 の加入者をサポートします。

- **sm_memory_size**

[SM Definitions] セクションにあります。SM プロセスに割り当てるメモリの量を MB 単位で定義します。ここでパラメータを設定することも、**~pcube** フォルダの下にある **sm.sh** ファイル内の **PCUBE_SM_MEM_SIZE** を編集することもできます。

- **database_perm_size**

[Database Definitions] セクションにあります。データベースに割り当てる **PermSize** を MB 単位で定義します。ここでパラメータを設定することも、**/var/TimesTen/sys.odb.ini** ファイル内の **PermSize** を編集することもできます。

- **database_temp_size**

[Database Definitions] セクションにあります。データベースに割り当てる **TempSize** を MB 単位で定義します。ここでパラメータを設定することも、**/var/TimesTen/sys.odb.ini** ファイル内の **TempSize** パラメータを編集することもできます。

ステップ 5 **install-sm.sh** スクリプトを実行します。



(注) **install-sm.sh** スクリプトはカスタマイズ可能です。



(注) **/etc/motd** ファイルが存在する場合は、このスクリプトを実行できません。**install-sm.sh** スクリプトを実行する前に、このファイルを移動または削除してください。

マシンのシェル プロンプトから、配布ファイルを抽出するディレクトリに移動して、**install-sm.sh** スクリプトを実行します。

```
# install-sm.sh [command options]
```

表 4-10 に、コマンド オプションを示します。

表 4-10 **install-sm.sh** のオプション

オプション	説明
-d	<p>~pcube のインストール ディレクトリを指定します。</p> <p>このディレクトリは、既存のディレクトリにしないでください。</p> <p>このディレクトリは、「/」で始まるフルパス名として指定する必要があります。</p> <p>デフォルトは /opt/pcube です。</p>
-o	<p>インストール ディレクトリとして、ユーザ pcube の既存のホーム ディレクトリを指定します。</p> <p>(注) オプション -d および -o は相互に排他的です。</p>

表 4-10 install-sm.sh のオプション (続き)

オプション	説明
-v	<p>データ ストレージのディレクトリを指定します。</p> <p>このディレクトリは、既存のディレクトリにしないでください。</p> <p>このディレクトリは、1 GB 以上の空き容量を持つパーティションに置く必要があります。</p> <p>このディレクトリは、「/」で始まるフルパス名として指定する必要があります。</p> <p>デフォルトは InstallDirectory/var です。</p>
-h	このメッセージを表示します。

スクリプトは、次の手順を実行します。

- 引数の有効性と、ディスク容量が十分であることを確認します。
- ユーザ **pcube** およびグループ **pcube** を追加 (またはその存在を確認) します。
- SM および CLU ディレクトリ構造に **pcube** ホーム ディレクトリを移植します。
- **pcube** ホームをターゲット ディレクトリとして、JRE インストール スクリプトを起動します。JRE インストールは、既存の Java インストールに影響しません。
- **pcube** ホームをターゲット ディレクトリとして、TimesTen インストール スクリプトを起動します。
- **pcube** ホームをターゲット ディレクトリとして、TimesTen の SM DSN を作成します。-v オプションを使用して、指定のディレクトリに TimesTen の SM DSN をインストールすることができます。
- /etc に起動スクリプトおよびシャットダウン スクリプトを作成します。
- SM がインストールされている実際のフォルダに依存する環境変数が含まれた、シェルのプリアンブル **~pcube/sm.sh** を作成します。

install-sm.sh スクリプトの例

これらの例は、**install-sm.sh** スクリプトを使用して SM をインストールする方法を示します。

- 「SM および CLU のインストール : 例」 (P.4-17)
- 「デフォルト ディレクトリへの SM および CLU のインストール : 例」 (P.4-17)

SM および CLU のインストール : 例

この例では、デフォルトのデータ ストレージ ディレクトリを使用して、**/usr/local/pcube** というディレクトリに SM および CLU をインストールします。

```
# install-sm.sh -d /usr/local/pcube
```

デフォルト ディレクトリへの SM および CLU のインストール : 例

この例では、ユーザ **pcube** のデフォルト ディレクトリに SM および CLU をインストールします。

```
# install-sm.sh -o
```

ステップ 6 pcube ユーザのパスワードを設定します。

インストール スクリプトを正常に実行した後で、**# passwd pcube** コマンドを実行して **pcube** ユーザのパスワードを設定します。



(注) 選択したパスワードを覚えておくことが重要です。

ステップ 7 コンピュータをリブートします。

インストールを完了するには、コンピュータをリブートする必要があります。

ステップ 8 (任意) Multiprotocol Label Switching (MPLS; マルチプロトコル ラベル スイッチング) /Virtual Private Network (VPN; バーチャルプライベート ネットワーク) BGP LEG をインストールします。

詳細については、『[Cisco SCMS SM LEGs User Guide](#)』の「[Installing the MPLS/VPN BGP LEG](#)」の章を参照してください。

ステップ 9 PRPC 認証用のユーザを追加します。

SM に接続するときに SCA BB でユーザ名とパスワードを要求されるので、PRPC 認証用にユーザを追加する必要があります。

PRPC 認証用のユーザを追加するには、**p3rpc** コマンドライン ユーティリティを使用します。たとえば、次のとおりです。

```
-bash-3.1$ p3rpc --set-user --username=username --password=password
```

クラスタ インストールの場合、両方のデバイスがインストールされた後で、次の例のとおり **--remote** オプションを使用します。

```
-bash-3.1$ p3rpc --set-user --username=username --password=password
--remote=OTHER_SM_IP[:port]
```

インストール中のトラブルシューティングについては、「[トラブルシューティング](#)」(P.D-1) を参照してください。

インストールの確認

インストールが成功したことを確認するには、**p3sm** コマンドなどの CLU ユーティリティを実行して、SM の一般情報を表示します。

ステップ 1 マシンのシェル プロンプトから、**~pcube/sm/server/bin** ディレクトリに変更します。

ステップ 2 **p3sm** コマンドを実行します。

次の **p3sm** コマンドは、次のような SM の現在のステータスを表示します。

```
-bash-3.1$ p3sm --sm-status
```



(注) インストール後、数分待機してからこのコマンドを実行して、SM を動作可能にします。

このコマンドの出力に、SM が実行中であることが示されます。

インストール中にエラーがあった場合、コマンドの出力にそれらのエラーの説明が示されます。

Subscriber Manager の設定

SM をインストールした後は、特定のニーズに合わせて SM を設定できます。この時点では特に、次のパラメータを扱います。

- `topology` : クラスタまたはスタンドアロン
- `introduction_mode` : プルまたはプッシュ
- `support_ip_ranges` : インストールしたセットアップに IP 範囲を使用するかどうか

SM を設定するには、標準的なテキスト エディタを使用して `p3sm.cfg` コンフィギュレーション ファイルを編集します。コンフィギュレーション ファイルについては、「設定と管理」モジュールと「コンフィギュレーション ファイル オプション」モジュールに詳しい説明があります。`p3sm.cfg` コンフィギュレーション ファイルの編集を終えたら、`p3sm` ユーティリティを使用して新しい設定で SM をアップデートします。

ステップ 1 マシンのシェル プロンプトから、`p3sm` コマンドを実行します。

次の `p3sm` コマンドでは、コンフィギュレーション ファイルをロードし、それに従って、次のように SM の設定をアップデートします。

```
-bash-3.1$ p3sm --load-config
```

追加のインストール手順を実行する方法

次の手順は、「Subscriber Manager のインストール」(P.4-11) で説明した手順の補足です。

- 「SM クラスタのインストール」(P.4-19) : 初めて 2 つの SM ノードをインストールする場合に使用します。
- 「SM クラスタ エージェントのインストール」(P.4-20) : Veritas Cluster Server (VCS) を使用するハイ アベイラビリティ セットアップを使用する場合にインストールする必要があります。

SM クラスタのインストール

SM クラスタのインストールは、2 台のマシンへの SM のインストールとよく似ています。

ステップ 1 両方のマシンに Veritas Cluster Server ソフトウェアをインストールします。

ステップ 2 両方のマシンに SM をインストールします。

詳細については、「Subscriber Manager のインストール」(P.4-11) を参照してください。

ステップ 3 SM トポロジ パラメータを `cluster` に設定します。

詳細については、「Subscriber Manager の設定」(P.4-19) を参照してください。

ステップ 4 レプリケーション方式を設定します。

次の CLU を実行することによって、冗長マシンにデータストアを複製するためのレプリケーション方式を設定する必要があります。

```
p3cdb --set-rep-scheme
```

ステップ 5 SM VCS エージェントをインストールします。

詳細については、「Subscriber Manager のインストール」(P.4-11) を参照してください。

ステップ 6 VCS を設定します。

詳細については、「[Veritas Cluster Server](#)」の章を参照してください。

SM クラスタ エージェントのインストール

インストール配布ファイルには、クラスタ トポロジ内の SM 関連リソースのモニタと制御をサポートするためにカスタマイズされた、一組の Veritas Cluster Agent が含まれています。クラスタ エージェントは VCS bin ディレクトリの下にインストールする必要があります。



(注) /etc/motd ファイルが存在する場合は、このスクリプトを実行できません。install-vcs-agents.sh スクリプトを実行する前に、このファイルを移動または削除してください。

ステップ 1 マシンのシェル プロンプトから、install-vcs-agents.sh スクリプトを実行します。

```
# install-vcs-agents.sh [command-options]
```

表 4-11 に、コマンド オプションを示します。

表 4-11 install-vcs-agents.sh のオプション

オプション	説明
-d	エージェントのインストール ディレクトリを指定します。これは、VCS の bin ディレクトリである必要があります。 このディレクトリは、既存のディレクトリにしてください。 このディレクトリは、「/」で始まるフル パス名として指定する必要があります。 デフォルト : /opt/VRTSvcs/bin
-h	ヘルプ メッセージを出力し、終了します。

スクリプトは、次の手順を実行します。

- インストール ディレクトリが存在することを確認します。
- エージェント配布ファイルを指定のディレクトリに抽出します。
- インストール ディレクトリからすべてのエージェント ディレクトリに VCS default-script-agent-executable をコピーします。

Subscriber Manager のアップグレード方法

Subscriber Manager では、以前にインストールした SM バージョンと、新規インストールでのフェールオーバーの要件（または要件なし）に応じた、複数のタイプのアップグレード手順をサポートしています。

アップグレード手順には、次の 3 つのタイプがあります。

- 「[スタンドアロンセットアップからのアップグレード](#)」 (P.4-22)

- 「スタンドアロンセットアップからクラスタセットアップへのアップグレード」(P.4-25)
- 「クラスタセットアップのアップグレード」(P.4-27)

データ複製手順

データ複製手順を使用すると、ユーザはマシンから別のマシンへデータベース全体を複製またはコピーして、最後にレプリケーション エージェントを実行することでデータベースの同期を維持することができます。前のセクションで説明したアップグレード手順のいくつかは、この手順を使用します。

この手順の詳細については、「データベース複製の回復」(P.3-6) を参照してください。

VLAN マッピングを使用した加入者の自動アップグレード



(注) 次に、リリース 3.1.5 よりも前のリリースからのアップグレードに関する情報を示します。

SM の 3.1.5 リリースには、新規の管理対象エンティティとして VPN が導入されました。この新しいエンティティは、VLAN マッピングが加入者ではなく VPN に関係するようになったことを意味します。アップグレード手順中、SM によって加入者の VLAN-ID を持つ VPN が自動的に作成され、加入者は新しい VPN 内の全範囲の IP マッピングと関連付けられます。

たとえば、VLAN-ID = 15 の加入者「sub1」については、VLAN-ID=15 を持つ VPN 15 と、マッピング 0.0.0.0/0@VLAN-ID を持つ加入者が作成されます。

RADIUS Listener の自動アップグレード



(注) 次に、リリース 3.1.5 よりも前のリリースからのアップグレードに関する情報を示します。

SM の 3.1.5 リリースには、RADIUS Listener に対する操作機能が新たに導入されました。

アップグレード手順中に、SM により、次のルールに従ってコンフィギュレーション ファイル内の RADIUS セクションが変更されます。

- **radius_attribute** プロパティと **radius_attribute_type** プロパティが新しいセクションに移動される。
- **radius_attribute** プロパティと **radius_attribute_type** プロパティを置き換えるため、新しいフィールド プロパティが追加される。
- **strip_type=remove_suffix** プロパティが **field_manipulation.<field name>=(*)<strip_character>.*** に置き換えられる。
- **strip_type=remove_prefix** プロパティが **field_manipulation.<field name>=.*<strip_character>(.*)** に置き換えられる。
- **use_default** プロパティおよびデフォルト値が **mapping_table.^\$=<default>** に置き換えられる。
- **radius_attribute_vendor_id** プロパティおよび **radius_sub_attribute** プロパティが **radius_attribute(radius_attribute_vendor_id; radius_sub_attribute)** 形式に置き換えられる。

スタンドアロン セットアップからのアップグレード



(注) バージョン 2.x からバージョン 3.6 にアップグレードするには、まず、バージョン 3.0 または 3.1 にアップグレードする必要があります。

次のアップグレード手順は、リリース 3.x ソフトウェア バージョンからのサポートになります。リリース 2.x ソフトウェア バージョンから SM をアップグレードするには、ユーザはまず、SM 環境をリリース 3.1.x ソフトウェア バージョンにアップグレードしてから、リリース 3.6.0 への追加アップグレードを実行する必要があります。

リリース 2.x ソフトウェア バージョンからリリース 3.1.x ソフトウェア バージョンへのアップグレード方法については、リリース 3.1.x のマニュアルを参照してください。

この手順は、SM 3.0.x 以降に適用されます。このアップグレード手順には、サービスのダウンタイムが必要です。



(注) スタンドアロン セットアップからクラスタ セットアップへのアップグレード手順については、「[スタンドアロン セットアップからクラスタ セットアップへのアップグレード](#)」(P.4-25) を参照してください。

必要なメモリの設定

SM のアップグレードを準備するため、[ステップ 3](#) の共有メモリの設定で説明した手順に従って、SM でシステムのカーネル コンフィギュレーション ファイルを設定します。

ステップ 1 配布ファイルを抽出します。

SM をアップグレードするには、インストールされているマシン、またはそのマシンにマウントされているディレクトリに、配布ファイルをロードして抽出します。

- a. シスコの Web サイトから配布ファイルをダウンロードします。
- b. FTP を使用して、配布ファイルを SM にロードします。
- c. **gunzip** コマンドを使用してファイルを解凍します。

```
gunzip SM_dist_<version>_B<build number>.tar.gz
```

- d. **tar** コマンドを使用して tar ファイルを抽出します。

```
tar -xvf SM_dist_<version>_B<build number>.tar
```

ステップ 2 ステート交換をディセーブルにします。

バージョン 2.x からアップグレードする場合は、SM と SCE プラットフォーム間のステート交換をディセーブルにします。これには、SM コンフィギュレーション ファイル (**p3sm.cfg**) を編集し、**save_subscriber_state=false** を設定してから、次のコマンドを使用してコンフィギュレーション ファイルをロードします。

```
-bash-3.1$ p3sm --load-config
```



(注) この CLU は、ユーザ **pcube** として使用する必要があります。

ステップ 3 **install-def.cfg** ファイルを編集します。

install-def.cfg コンフィギュレーション ファイルを編集し、**ステップ 3** の共有メモリの設定に記載されている推奨事項に従って **PermSize** パラメータと **TempSize** パラメータを設定します。詳細については、**ステップ 4** の **install-def.cfg** ファイルの編集を参照してください。

ステップ 4 **upgrade-sm.sh** スクリプトを実行します。

非クラスタ セットアップからアップグレードするため、**SM** ディストリビューションには以前のバージョンからのアップグレードを実行するアップグレード スクリプトが用意されています。このアップグレード手順スクリプトでは、加入者データベースと、ネットワーク要素、ドメイン、およびアプリケーション固有のコンポーネントを含む **SM** 設定全体が保存されます。



(注) Solaris の場合 : Solaris 上の **SM** の以前のバージョンでは、32 ビットまたは 64 ビットの Java Virtual Machine (JVM) とデータベースが使用されていました。**SM** は現在、64 ビットの JVM およびデータベースと一緒にインストールされます。64 ビットにアップグレードするかどうかを選択することはできません。



(注) Linux の場合 : Linux システムでは、**SM 2.5.x** および **3.x** リリースからのアップグレードに限られます。Linux プラットフォームには、32 ビットまたは 64 ビットの JVM およびデータベースを使用できます。



(注) **/etc/motd** ファイルが存在する場合は、このスクリプトを実行できません。**upgrade-sm.sh** スクリプトを実行する前に、このファイルを移動または削除してください。

a. マシンのシェル プロンプトから **upgrade-sm.sh** スクリプトを実行します。

```
# upgrade-sm.sh [command-options]
```

表 4-12 **upgrade-sm.sh** のオプション

オプション	説明
-h	アップグレード メッセージを表示します。

スクリプトは、次の手順を実行します。

- 既存の **SM** バージョンを検出します。
- 新規の **SM** バージョンを検出します。
- Java がマシンにインストールされていることを確認します。
- ユーザ **pcube** が存在することを確認します。
- バージョン 3.0.x 以降の **SM** がシステムに存在することを確認します。
- 現在の **SM** を停止します (実行している場合)。
- 加入者データベースの既存のコンテンツを外部ファイルにバックアップします。
- TimesTen データベースを削除します。
- **SM** コンフィギュレーション ファイルをバックアップします。
- **SM** および TimesTen のアップデート バージョンをインストールします。
- **SM** およびデータベースのコンフィギュレーション ファイルをアップグレードするための別個のプログラムを起動します。

- 加入者データベースのバックアップ コンテンツを復元します。
- アップグレードした SM を起動します。

SM のアップグレード：例

この例では、次のように SM をアップグレードし、現在のデータベースを維持します。

```
# upgrade-sm.sh
```



(注) アップグレード手順後の SM のレポートは不要です。

ステップ 5 PRPC 認証用のユーザを追加します。

3.0.5 よりも前の SM バージョンからアップグレードする場合は、PRPC 認証用のユーザを追加する必要があります。これは、SM に接続する際に SCA BB でユーザ名とパスワードを要求されるためです。

PRPC 認証用のユーザを追加するには、**p3rpc CLU** を使用します。たとえば、次のとおりです。

```
-bash-3.1$ p3rpc --set-user --username=username --password=password
```

ステップ 6 古いステート情報を削除します。

バージョン 2.x からアップグレードする場合は、次に示すように **pcube** ユーザとして SM CLU を実行することにより、古い加入者ステート情報をすべて削除します。

```
-bash-3.1$ p3subsd --clear-all-states
```

ステップ 7 古い加入者プロパティを削除します（方法 A）。

バージョン 2.x からアップグレードする場合は、古い加入者プロパティをすべて削除します。



(注) いずれの CLU コマンドもユーザ **pcube** として実行する必要があります。

- 既存の加入者を csv ファイルにエクスポートします。

```
-bash-3.1$ p3subsd --export -o csv-file
```

- 加入者データベースをクリアします。

```
-bash-3.1$ p3subsd --clear-all
```

- csv ファイルから古いプロパティを削除します。

削除されるプロパティのリストについては、表 4-13 を参照してください。

- 改訂されたファイルから加入者をインポートします。

```
-bash-3.1$ p3subsd --import -f csv-file
```

ステップ 8 古い加入者プロパティを削除します（方法 B）。

バージョン 2.x からアップグレードする場合は、古い加入者プロパティをすべて削除します。



(注) いずれの CLU コマンドもユーザ **pcube** として実行する必要があります。

- p3subsd** コマンドを実行することによって、SM データベースから古いプロパティを削除します。

```
-bash-3.1$ p3subsd --remove-property --property=prop
```

削除される古いプロパティは、表 4-13 にリスト表示されています。

- b. すべての SCE を再同期化します。

```
-bash-3.1$ p3sm --resync-all
```

表 4-13 削除される古いプロパティ

QP 調整パラメータ	適用可能なバージョン
QpLimit[1]–QpLimit[16]	バージョン 2.5.5 以降
QpSet[1]–QpSet[16]	バージョン 2.5.5 以降
QpAdd	バージョン 2.5.5 よりも前のバージョン 2.5 バージョン 2.1.7 よりも前のバージョン 2.1
QpDelta[1]–QpDelta[16]	バージョン 2.5.5 よりも前のバージョン 2.5 バージョン 2.1.7 よりも前のバージョン 2.1

ステップ 9 SCE プラットフォームを設定します。

カスケード型 SCE セットアップを使用する場合は、「[SCE.XXX] セクション」(P.A-15) にある説明に従って **p3sm.cfg** ファイルでカスケード型 SCE のペアを設定します。

スタンドアロン セットアップからクラスタ セットアップへのアップグレード



(注) バージョン 2.x からバージョン 3.6 にアップグレードするには、まず、バージョン 3.0 または 3.1 にアップグレードする必要があります。

次のアップグレード手順は、リリース 3.x ソフトウェア バージョンからのサポートになります。リリース 2.x ソフトウェア バージョンから SM をアップグレードするには、まず、その SM 環境をリリース 3.1.x ソフトウェア バージョンにアップグレードしてから、リリース 3.6.0 への追加アップグレードを行う必要があります。

リリース 2.x ソフトウェア バージョンからリリース 3.1.x ソフトウェア バージョンへのアップグレード方法については、リリース 3.1.x のマニュアルを参照してください。

このセクションでは、スタンドアロン セットアップをクラスタ セットアップにアップグレードする基本的な手順について説明します。この手順は、バージョン 3.0.x 以降の SM に適用されます。このアップグレード手順には、サービスのダウンタイムが必要です。



(注) この手順では、できるだけ SM のダウンタイムを短くすることを試みます。したがって、加入者サービスが問題でない場合は、新しいマシンのインストールおよび新しいマシンのアップグレードの手順を代わりに使用してください。

次の手順では、SM-A は SM バージョン 3.0.x 以降を実行する元の SM マシンであり、SM-B は冗長性を追加する新規の SM マシンです。

ステップ 1 両方のマシンに VCS をインストールします。

ステップ 2 SM-B をインストールします。

SM-B をインストールするには、「Subscriber Manager のインストール」(P.4-11) に記載されている手順を実行します。

ステップ 3 SM-A をアップグレードします。

SM-A をアップグレードするには、「スタンドアロン セットアップからのアップグレード」(P.4-22) に記載されている手順を実行します。



(注) この手順から、アップグレード手順が完了するまで、SM では加入者は処理されません。

ステップ 4 SM-A の SM 設定を SM-B に複製します (~pcube/sm/server/root/config フォルダからすべてのコンフィギュレーション ファイルをコピーします)。

SM-A から SM-B に **p3sm.cfg** コンフィギュレーション ファイルを手動でコピーします。コンフィギュレーション ファイルをロードするには、「SM 設定のリロード (p3sm)」(P.5-16) を参照してください。

ステップ 5 加入者データベースを複製します。

データ複製手順については、「データ複製手順」(P.4-21) に説明があります。

冗長マシンにデータ ストアを複製するためのレプリケーション方式を設定します。



(注) この CLU は、ユーザ **pcube** として両方のマシンで実行する必要があります。

```
-bash-3.1$ p3db --set-rep-scheme
```

ステップ 6 クラスタを作成します。

a. クラスタをサポートするように、SM-A と SM-B を設定します。

b. SM-B をスタンバイにします。

CLU コマンド **p3cluster --standby** を使用します。

c. SM-A が確実にアクティブになるようにします。

CLU コマンド **p3cluster --active** を使用します。

d. VCS を設定します。

e. セットアップで VCS を実行します。

ステップ 7 クラスタの仮想 IP にログインを送信するように、LEG アプリケーションを設定します。

ステップ 8 環境をテストするには、次の手順を実行します。

a. アクティブ マシンに加入者を作成します。

```
p3subs --add -s testCluster1
```

b. Veritas Cluster Manager を使用してフェールオーバーを手動で起動し、スタンバイ SM がアクティブになり、アクティブ SM がスタンバイになるまで待機します。

c. /opt/VRTSvcs/bin から次の VCS CLU コマンドを実行します。

```
# hagrps -switch service group name to System
```

d. testCluster1 が新しいアクティブ マシンに存在することを確認します。

```
--p3subsdb --show-all
```

e. 新しいアクティブ マシンに加入者を作成します。

```
p3subs --add -s testCluster2
```

- f. testCluster1 が新しいアクティブ マシンに存在することを確認します。

```
--p3subsdb --show=all
```

クラスタ セットアップのアップグレード

- 「クラスタ セットアップ バージョン 3.x からのアップグレード」 (P.4-27)
- 「クラスタ セットアップ バージョン 2.x からバージョン 3.0 または 3.1 へのアップグレード」 (P.4-33)



(注)

バージョン 2.x からバージョン 3.6 にアップグレードするには、まず、バージョン 3.0 または 3.1 にアップグレードする必要があります。

クラスタ セットアップ バージョン 3.x からのアップグレード

このセクションでは、SM バージョン 3.0 以降のクラスタ セットアップからクラスタ セットアップにアップグレードするための基本的な手順について説明します。



(注)

この手順には、サービスのダウンタイムはありません。

クラスタ セットアップからアップグレードする際のアップグレード手順には、次の 3 つのハイ レベルな手順が含まれます。

1. スタンバイ マシンでアップグレード手順を実行します。
2. アップグレードした SM で手動によるフェールオーバーを実行します。
3. フェールオーバーの実行後にスタンバイになる SM で、アップグレード手順を実行します。

ステップ 1

両方のマシンで、システムのカーネル コンフィギュレーション ファイルを設定します。

アップグレード手順を開始する前に、両方のマシンでシステムのカーネル コンフィギュレーション ファイルを設定する必要があります。

- a. スタンバイ SM で、システムのカーネル コンフィギュレーション ファイルを設定します。

設定手順については、[ステップ 3](#) の共有メモリの設定に説明があります。

- b. スタンバイ SM をリブートします。

- c. Veritas Cluster Manager を使用してフェールオーバーを手動で起動し、スタンバイ SM がアクティブになり、アクティブ SM がスタンバイになるまで待機します。

/opt/VRTSvcs/bin から次の VCS CLU コマンドを実行します。

```
# hagrps -switch service group name to System
```

- d. 新しいスタンバイ SM で、手順 a と b を繰り返します。

ステップ 2

配布ファイルを抽出します。

SM をアップグレードするには、インストールされているマシン、またはそのマシンにマウントされているディレクトリに、配布ファイルをロードして抽出します。

- a. シスコの Web サイトから配布ファイルをダウンロードします。

- b. FTP を使用して、配布ファイルを SM にロードします。

c. **gunzip** コマンドを使用してファイルを解凍します。
`gunzip SM_dist_<version>_B<build number>.tar.gz`

d. **tar** コマンドを使用して tar ファイルを抽出します。
`tar -xvf SM_dist_<version>_B<build number>.tar`

ステップ 3 VCS モニタリングを停止します。

a. *root* ユーザとしてログインします。

b. SM の VCS モニタリングを停止します。

`/opt/VRTSvcs/bin` から次の VCS CLU コマンドを使用して、VCS モニタリングを停止します。

```
#!/hastop -local
```

ステップ 4 `install-def.cfg` ファイルを編集します。

`install-def.cfg` コンフィギュレーション ファイルを編集し、**ステップ 3** の共有メモリの設定に記載されている推奨事項に従って `PermSize` パラメータと `TempSize` パラメータを設定します。詳細については、**ステップ 4** の `install-def.cfg` ファイルの編集を参照してください。

ステップ 5 データベース レプリケーションを一時停止します。



(注) 第 1 の SM マシンをアップグレードするときだけ、次のコマンドを実行します。

a. アクティブマシンで、配布ファイルを抽出した場所にディレクトリを変更します。

b. `scripts/p3db --rep-pause` CLU を実行します。

c. `~pcube/sm/server/bin/p3db --rep-status` CLU を実行して、レプリケーションが「pause」状態であることを確認します。

d. スタンバイ ワークステーションに戻ります。

ステップ 6 `cluster-upgrade.sh` スクリプトを実行します。

クラスタ セットアップからクラスタ セットアップにアップグレードするため、SM バージョン 3.6.0 には以前のバージョンからのアップグレードを実行するためのアップグレード スクリプトが用意されています。このアップグレード手順スクリプトでは、加入者データベースと、ネットワーク要素、ドメイン、およびアプリケーション固有のコンポーネントを含む SM 設定全体が保存されます。



(注) Solaris の場合 : Solaris 上の SM の以前のバージョンでは、32 ビットまたは 64 ビットの Java Virtual Machine (JVM) とデータベースが使用されていました。SM バージョン 3.0.3 から、SM は 64 ビットの JVM およびデータベースと一緒にインストールされます。64 ビットにアップグレードするかどうかを選択することはできません。



(注) Linux の場合 : Linux システムでは、SM 3.x リリースからのアップグレードだけがサポートされています。Linux プラットフォームには、32 ビットまたは 64 ビットの JVM およびデータベースを使用できます。

a. スタンバイ マシンのシェル プロンプトから、**cluster-upgrade.sh** スクリプトを実行します。

```
# cluster-upgrade.sh [command-options]
```

スクリプトは、次の手順を実行します。

- 既存の SM バージョンを検出します。
- 新規の SM バージョンを検出します。
- Java がマシンにインストールされていることを確認します。
- ユーザ **pcube** が存在することを確認します。
- バージョン 3.x 以降の SM がシステムに存在することを確認します。
- **install-def.cfg** (存在する場合) で設定した値を確認します。
- 現在の SM を停止します (実行している場合)。
- 加入者データベースの既存のコンテンツを外部ファイルにバックアップします。
- TimesTen データベースを削除します。
- SM コンフィギュレーション ファイルをバックアップします。
- SM および TimesTen のアップデート バージョンをインストールします。
- SM およびデータベースのコンフィギュレーション ファイルをアップグレードするための別個のプログラムを起動します。
- 加入者データベースのバックアップ コンテンツを復元します。スクリプトを第 2 のマシンでアクティブ化すると、現在アクティブな SM には最新のデータが含まれているため、現在アクティブな SM からデータベースのコンテンツがコピーされます。

表 4-14 に、cluster-upgrade スクリプトのコマンド オプションを示します。

表 4-14 cluster-upgrade.sh のオプション

オプション	説明
-h	アップグレード メッセージを表示します。
-1	第 1 のマシンでスクリプトをアクティブ化する際は、このオプションを使用します。
-2	第 2 のマシンでスクリプトをアクティブ化する際は、このオプションを使用します。

cluster-upgrade.sh スクリプトの実行例

- 「第 1 の SM のアップグレード : 例」 (P.4-29)
- 「第 2 の SM のアップグレード : 例」 (P.4-29)

第 1 の SM のアップグレード : 例

この例では、第 1 の SM をアップグレードし、現在のデータベースを維持します。

```
# cluster-upgrade.sh -1
```



(注)

アップグレード手順後の SM のレポートは不要です。

第 2 の SM のアップグレード : 例

この例では、第 2 の SM をアップグレードし、現在のデータベースを維持します。

```
# cluster-upgrade.sh -2
```

cluster-upgrade.sh の実行後に、SM を起動しないでください。

ステップ 7 cluster-upgrade.sh スクリプトが完了するまで待機します。

ステップ 8 レプリケーションを停止し、SM を起動します。



(注) 次の手順は、第 1 のマシンでアップグレードするときだけに実行します。

Database (DB; データベース) スキーマが変更されているので、最初に SM をロードするときにスタンバイ マシンへの変更を複製する必要はありません。

a. SM のレプリケーションを停止します。

```
# ./p3db --rep-stop
```

b. SM を起動します。

```
# ./p3sm --start --wait
```

c. p3sm CLU コマンドを使用して、SM が起動して状態を確認するときを確認します。

```
-bash-3.1$ p3sm --sm-status
```



(注) SMS-STATUS が障害状態を示す場合、アップグレードを停止する必要があります。

ステップ 9 マシンのシェル プロンプトから、**install-vcs-agents.sh** スクリプトを実行します。**install-vcs-agents.sh** スクリプトの実行の追加情報については、「[SM クラスタ エージェントのインストール](#)」(P.4-20) を参照してください。

```
# install-vcs-agents.sh [command-options]
```

ステップ 10 VCS モニタリングを再起動します。

a. /opt/VRTSvcs/bin から次の VCS CLU コマンドを実行します。

```
# ./hastart
```



(注) この操作によって、アクティブ マシン上のデータベース方式を更新するレプリケーション エージェントが起動するため、この操作をいったん実行したら、ダウングレード操作を行うことはできなくなります。

VCS モニタリングは、SM プロセスを初期状態で自動的に起動します。

b. p3sm CLU コマンドを使用して、SM が起動するときを確認します。

```
-bash-3.1$ p3sm --sm-status
```

c. p3cluster CLU コマンドを使用して、SM をスタンバイ状態に設定します。

```
-bash-3.1$ p3cluster --standby
```



(注) アップグレード後の SM のブート時間は、データベース インデックスを初期化するための時間がかかるため、通常よりも長くなります。

ステップ 11 データベース レプリケーションを続行します。



(注) 第1のマシンをアップグレードするときに、[ステップ 5](#) を実行した場合に限り、このコマンドを実行します。

- a. アクティブマシンで、配布ファイルを抽出した場所にディレクトリを変更します。
- b. `scripts/p3db --rep-continue CLU` を実行します。
- c. `~pcube/sm/server/bin/p3db --rep-status CLU` を実行して、レプリケーションが「start」状態であることを確認します。
- d. スタンバイ ワークステーションに戻ります。

ステップ 12 変更したデータが複製されていることを確認します。

アップグレード スクリプトの実行中に変更したすべてのデータが複製されるまで待機します。

- アクティブ SM で、`p3subs CLU` を使用してダミー加入者を追加します。

```
-bash-3.1$ p3subs --add -s dummySub
```



(注) 第2の SM をアップグレードするときは、第1の SM のアップグレードでレプリケーションによって `dummySub` という名前が追加されているので、これ以外の名前を付けた加入者を追加します。

- スタンバイ SM で `verify-subscriber.sh` スクリプトを実行し、加入者が複製されていることを確認します。

```
#!/verify-subscriber.sh dummySub
```



(注) `verify-subscriber.sh` スクリプトは、`root` ユーザとして実行します。

ステップ 13 (任意) MPLS/VPN BGP LEG をインストールします。

詳細については、『*Cisco SCMS SM LEGs User Guide*』の「[Installing the MPLS/VPN BGP LEG](#)」の章を参照してください。

ステップ 14 Veritas Cluster Manager を使用してフェールオーバーを手動で起動し、スタンバイ SM がアクティブになり、アクティブ SM がスタンバイになるまで待機します。

`/opt/VRTSvcs/bin` から次の VCS CLU コマンドを実行します。

```
# hagrps -switch service group name -to System
```

`hagrps CLU` の詳細については、Veritas Cluster Server のドキュメンテーションを参照してください。

手動でフェールオーバーを実行した後は、アップグレード手順を実行したスタンバイ SM がアクティブ SM になります。以前のアクティブ SM は新しいスタンバイ SM になります。

ステップ 15 スタンバイ SM でアップグレード手順を繰り返します。

第2の SM をアップグレードするには、[ステップ 2](#) の「配布ファイルを抽出します」から手順を繰り返します。



(注) 第2の SM をアップグレードするときは、[ステップ 5](#) の「データベース レプリケーションを一時停止します」、[ステップ 8](#) の「レプリケーションを停止し、SM を起動します」、および [ステップ 11](#) の「データベース レプリケーションを続行します」は実行しないでください。

ステップ 16 データベース レプリケーション プロトコルのバージョンをアップグレードします。



(注) データベース レプリケーション プロトコルのバージョンをアップグレードするため、admin ユーザとして両方のマシンで次のコマンドを実行する必要があります。この操作は、両方の SM をアップグレードした後で実行してください。

- a. スタンバイ SM の VCS モニタリングを停止します。

/opt/VRTSvcs/bin から次の VCS CLU コマンドを使用します。

```
# ./hastop -local
```

- b. レプリケーション プロトコルを変更します。

スタンバイ SM で、次の CLU を実行します。

```
# p3db --upgrade-rep-protocol
```

このコマンドによって、次のアクションが実行されます。

- DB セキュリティ フラグの削除
- SM の停止
- DB デーモンの再起動
- SM の起動
- SM レプリケーションの起動

- c. VCS モニタリングを再起動します。

/opt/VRTSvcs/bin から次の VCS CLU コマンドを実行します。

```
# ./hastart
```

VCS モニタリングは、SM プロセスを初期状態で自動的に起動します。

- d. p3cluster CLU を使用して、SM をスタンバイ状態に設定します。

```
-bash-3.1$ p3cluster --standby
```

- e. Veritas Cluster Manager を使用してフェールオーバーを手動で起動し、スタンバイ SM がアクティブになり、アクティブ SM がスタンバイになるまで待機します。

/opt/VRTSvcs/bin から次の VCS CLU コマンドを実行します。

```
# hagrps -switch service group name -to System
```

hagrps CLU の詳細については、Veritas Cluster Server のドキュメンテーションを参照してください。

- f. 新しいスタンバイ SM で、ステップ 16 の「データベース レプリケーション プロトコル バージョンをアップグレードします」を繰り返します。

ステップ 17 PRPC 認証用のユーザを追加します。

3.0.5 よりも前の SM バージョンからアップグレードする場合は、PRPC 認証用のユーザを追加する必要があります。これは、SM に接続する際に SCA BB でユーザ名とパスワードを要求されるためです。

PRPC 認証用のユーザを追加するには、p3rpc CLU を使用します。たとえば、次のとおりです。

```
-bash-3.1$ p3rpc --set-user --username=username --password=password
--remote=OTHER_SM_IP[:port]
```

ステップ 18 SCE プラットフォームを設定します。

カスケード型 SCE セットアップを使用する場合は、「[SCE.XXX] セクション」(P.A-15) に説明されているとおりに p3sm.cfg ファイルでカスケード型 SCE のペアを設定します。

ステップ 19 ダミー加入者を削除します。

両方の SM を正常にアップグレードした後は、アップグレード中にレプリケーションを確認するために追加されたダミー加入者を削除することを推奨します。

新しいアクティブ SM で、次の CLU を実行します。

```
-bash-3.1$ p3subs --remove --subscriber=first dummy subscriber name
```

```
-bash-3.1$ p3subs --remove --subscriber=second dummy subscriber name
```

クラスタ セットアップ バージョン 2.x からバージョン 3.0 または 3.1 へのアップグレード

このセクションでは、SM バージョン 2.x 以降のクラスタ セットアップからクラスタ セットアップにアップグレードするための基本的な手順について説明します。



(注) バージョン 3.6 にアップグレードするには、まず、バージョン 3.0 または 3.1 にアップグレードする必要があります。



(注) この手順では、サービスのダウンタイムが発生します。

クラスタ セットアップからアップグレードする際のアップグレード手順には、次の 3 つのハイ レベルな手順が含まれます。

1. スタンバイ マシンでアップグレード手順を実行します。
2. アップグレードした SM で手動によるフェールオーバーを実行します。
3. フェールオーバーの実行後にスタンバイになる SM で、アップグレード手順を実行します。

ステップ 1 両方のマシンで、システムのカーネル コンフィギュレーション ファイルを設定します。

アップグレード手順を開始する前に、両方のマシンでシステムのカーネル コンフィギュレーション ファイルを設定する必要があります。

- a. スタンバイ SM で、システムのカーネル コンフィギュレーション ファイルを設定します。
設定手順については、[ステップ 3](#) の共有メモリの設定に説明があります。
- b. スタンバイ SM をリブートします。
- c. Veritas Cluster Manager を使用してフェールオーバーを手動で起動し、スタンバイ SM がアクティブになり、アクティブ SM がスタンバイになるまで待機します。

`/opt/VRTSvcs/bin` から次の VCS CLU コマンドを実行します。

```
# hagrps -switch service group name to System
```

- d. 新しいスタンバイ SM で、手順 a と b を繰り返します。

ステップ 2 配布ファイルを抽出します。

SM をアップグレードするには、インストールされているマシン、またはそのマシンにマウントされているディレクトリに、配布ファイルをロードして抽出します。

- a. シスコの Web サイトから配布ファイルをダウンロードします。
- b. FTP を使用して、配布ファイルを SM にロードします。

c. **gunzip** コマンドを使用してファイルを解凍します。
`gunzip SM_dist_<version>_B<build number>.tar.gz`

d. **tar** コマンドを使用して tar ファイルを抽出します。
`tar -xvf SM_dist_<version>_B<build number>.tar`

ステップ 3 VCS エージェントをアンインストールして、VCS モニタリングを停止します。

a. *root* ユーザとしてログインします。

b. VCS エージェントをアンインストールします。

VCS エージェントのアンインストールについては、「[VCS エージェントのアンインストール](#)」(P.4-38) に説明があります。使用するリソース名は、PcubeSm、OnOnlyProcess、および TimesTenRep です。

c. SM の VCS モニタリングを停止します。

`/opt/VRTSvcs/bin` から次の VCS CLU コマンドを使用して、VCS モニタリングを停止します。

```
#!/hastop -local
```

ステップ 4 ステート交換をディセーブルにします。

SM と SCE プラットフォーム間のステート交換をディセーブルにします。これには、SM コンフィギュレーションファイル (`p3sm.cfg`) を編集し、`save_subscriber_state=false` を設定してから、次のコマンドを使用してコンフィギュレーション ファイルをロードします。



(注) この CLU は、ユーザ **pcube** として使用する必要があります。

```
-bash-3.1$ p3sm --load-config
```

ステップ 5 古いレプリケーション方式をドロップします。

次の CLU を使用します。



(注) この CLU は、ユーザ **pcube** として使用する必要があります。

```
-bash-3.1$ p3sm --drop-rep-scheme
```

ステップ 6 `install-def.cfg` ファイルを編集します。

`install-def.cfg` コンフィギュレーション ファイルを編集し、[ステップ 3](#) の共有メモリの設定に記載されている推奨事項に従って `PermSize` パラメータと `TempSize` パラメータを設定します。詳細については、[ステップ 4](#) の `install-def.cfg` ファイルの編集を参照してください。

ステップ 7 `upgrade-sm.sh` スクリプトを実行します。

詳細については、[ステップ 4](#) の `upgrade-sm.sh` スクリプトの実行を参照してください。

ステップ 8 アプリケーションおよび LEG をアップグレードします。

アプリケーションまたは LEG 固有のアップグレード手順を実行します。

ステップ 9 レプリケーション方式を設定します。

次の CLU を使用して、冗長マシンにデータ ストアを複製するためのレプリケーション方式を設定します。



(注) この CLU は、ユーザ **pcube** として使用する必要があります。

```
-bash-3.1$ p3sm --set-rep-scheme
```

ステップ 10 VCS エージェントをインストールし、VCS モニタリングを設定して再起動します。

- a. SM VCS エージェントをインストールします。

SM VCS エージェントのインストールについては、「[SM クラスタ エージェントのインストール](#)」(P.4-20) に説明があります。

- b. VCS を設定します。

VCS の設定については、「[Veritas Cluster Server](#)」(P.E-1) に説明があります。

- c. VCS モニタリングを再起動します。

`/opt/VRTSvcs/bin` から次の VCS CLU コマンドを実行します。

```
#!/hastart
```

ステップ 11 古いステータス情報を削除します。

pcube ユーザとして次の SM CLU を実行することにより、古い加入者ステータス情報を削除します。

```
-bash-3.1$ p3subsd --clear-all-states
```

ステップ 12 古い加入者プロパティを削除します (方法 A)。

古い加入者プロパティをすべて削除します。



(注) いずれの CLU コマンドもユーザ **pcube** として実行する必要があります。

- a. 既存の加入者を csv ファイルにエクスポートします。

```
-bash-3.1$ p3subsd --export -o csv-file
```

- b. 加入者データベースをクリアします。

```
-bash-3.1$ p3subsd --clear-all
```

- c. csv ファイルから古いプロパティを削除します。

削除されるプロパティのリストについては、[表 4-13](#) を参照してください。

- d. 改訂されたファイルから加入者をインポートします。

```
-bash-3.1$ p3subsd --import -f csv-file
```

ステップ 13 古い加入者プロパティを削除します (方法 B)。

バージョン 2.x からアップグレードする場合は、古い加入者プロパティをすべて削除します。



(注) いずれの CLU コマンドもユーザ **pcube** として実行する必要があります。

- a. **p3subsd** コマンドを実行することによって、SM データベースから古いプロパティを削除します。

```
-bash-3.1$ p3subsd --remove-property --property=prop
```

削除される古いプロパティは、[表 4-13](#) にリスト表示されています。

- b. すべての SCE を再同期化します。

```
-bash-3.1$ p3sm --resync-all
```

- ステップ 14** フェールオーバーを手動で起動します。

Veritas Cluster Manager を使用してフェールオーバーを手動で起動し、スタンバイ SM がアクティブになり、アクティブ SM がスタンバイになるまで待機します。

/opt/VRTSvcs/bin から次の VCS CLU コマンドを実行します。

```
# hagrps -switch service group name to System
```

- ステップ 15** スタンバイ SM でアップグレード手順を繰り返します。

手動でフェールオーバーを実行した後は（[ステップ 14](#) の手動でのフェールオーバーの起動を参照）、アップグレード手順を実行したスタンバイ SM がアクティブ SM になります。以前のアクティブ SM は新しいスタンバイ SM になります。

第2の SM をアップグレードするには、[ステップ 2](#) の配布ファイルの抽出から [ステップ 12](#) の古い加入者プロパティの削除までの手順を繰り返します。

- ステップ 16** PRPC 認証用のユーザを追加します。

SM に接続するときに SCA BB でユーザ名とパスワードを要求されるので、PRPC 認証用にユーザを追加する必要があります。

PRPC 認証用のユーザを追加するには、**p3rpc** CLU を使用します。たとえば、次のとおりです。

```
-bash-3.1$ p3rpc --set-user --username=username --password=password
--remote=OTHER_SM_IP[:port]
```

- ステップ 17** SCE プラットフォームを設定します。

カスケード型 SCE セットアップを使用する場合は、「[\[SCE.XXX\] セクション](#)」 ([P.A-15](#)) に説明されているとおりに **p3sm.cfg** ファイルでカスケード型 SCE のペアを設定します。

追加のアップグレード手順

最大長 64 文字への加入者 ID のアップグレード

バージョン 3.0.5 では、加入者 ID の長さが 64 文字に増加されました。新規のインストールについては、加入者 ID の最大長は 64 文字になります。ただし、以前のバージョンからアップグレードすると、この長さは自動的に増加されません。

- ステップ 1** **p3subsd** CLU を使用して加入者データベースをエクスポートします。

```
-bash-3.1$ p3subsd --export --output=output file name
```

- ステップ 2** **p3db** CLU を使用してデータベースを破棄します。

```
-bash-3.1$ p3db --destroy-rep-db
```

- ステップ 3** SM を再起動します。

```
-bash-3.1$ p3sm --restart
```

- ステップ 4** **p3subsd** CLU を使用して、加入者をデータベースにインポートし直します。

```
-bash-3.1$ p3subsd --import --file=file name from Step 1
```



(注) SM データベースを破棄するため、この手順にはシステムのダウンタイムが必要です。さらに、再起動後は、すべての SCE からすべての加入者情報が自動的に失われます。これらの情報を復元できるのは、加入者を SM データベースにインポートし直した後に限られます。

Subscriber Manager のダウングレード方法

このセクションでは、SM を以前のバージョンにダウングレードする手順について説明します。

ステップ 1 「Subscriber Manager のアンインストール方法」(P.4-37) に記載されているアンインストール手順を実行します。

ステップ 2 「Subscriber Manager のインストール」(P.4-11) に記載されているインストール手順を実行します。



(注) `upgrade-sm.sh` および `cluster-upgrade.sh` アップグレードスクリプトでは、SM のダウングレードをサポートしていません。

Subscriber Manager のアンインストール方法

- 「Subscriber Manager のアンインストール」(P.4-37)
- 「VCS エージェントのアンインストール」(P.4-38)

Subscriber Manager のアンインストール

`uninstall-sm.sh` スクリプト

`uninstall-sm.sh` スクリプトを実行するには、マシンのシェル プロンプトから次のコマンドを入力します。

```
# uninstall-sm.sh [command-options]
```

表 4-15 に、コマンド オプションを示します。

表 4-15 `uninstall-sm.sh` スクリプトのオプション

オプション	説明
-n	SM データベースを削除しません。
-h	ヘルプ メッセージを表示します。

スクリプトは、次の手順を実行します。

- SM を停止します。
- `-n` フラグが使用されていない場合に、レプリケーション エージェント (クラスタ セットアップ内) を停止します。

- `-n` フラグが使用されていない場合に、データストアを破棄します。
- TimesTen データベースをアンインストールします。
- SM ディレクトリおよびブート ファイルを削除します。
- SM インストールの一部としてインストールされた Java を削除します。

ステップ 1 クラスタ セットアップを使用する場合は、SM の VCS モニタリングを停止します。
`/opt/VRTSvcs/bin` から次の VCS CLU コマンドを実行して、VCS モニタリングを停止します。
`# ./hastop -local`

ステップ 2 配布ルート ディレクトリから `uninstall-sm.sh` スクリプトを実行します。
`# ./uninstall-sm.sh`

詳細については、「[uninstall-sm.sh スクリプト](#)」(P.4-37) を参照してください。

ステップ 3 クラスタ セットアップを使用する場合は、VCS モニタリングを起動します。
`/opt/VRTSvcs/bin` から次の VCS CLU コマンドを実行して、VCS モニタリングを起動します。
`# ./hastart`

ステップ 4 クラスタ セットアップを使用する場合は、Veritas Cluster エージェントを削除します。
Veritas Cluster エージェントの削除については、「[VCS エージェントのアンインストール](#)」(P.4-38) に説明があります。

リソース名 `OnOnlyProcess`、`SubscriberManager`、および `TimesTenRep` を削除します。

ステップ 5 `userdel` コマンドを実行して、`pcube` ユーザを削除します。
`# userdel -r pcube`



(注) TimesTen をインストールしたままにする場合は、`pcube` ユーザを削除しないでください。

VCS エージェントのアンインストール

削除する各 Veritas Cluster エージェントに対して、次の手順を繰り返します。

ステップ 1 Veritas Cluster Manager を使用するか、`hares` CLU を使用して、VCS エージェントを削除します。
VCS エージェントは、Veritas Cluster Manager か次の CLU を使用して削除できます。

```
# hares -delete TimesTenDaemon
# hares -delete SM
# hares -delete ReplicationAgent
# hares -delete Network-NICs
# hares -delete Network-VIP
```

システム内のリソース名は異なる場合があります。`hares -list` を使用して、既存のリソース名を確認してください。

ステップ 2 `hatype` CLU を使用して、VCS リソース タイプを削除します。

システム内のタイプ名は異なる場合があります。`hatype -list` を使用して、既存のタイプ名を確認してください。

```
# hatype -delete OnOnlyProcess
# hatype -delete SubscriberManager
# hatype -delete TimesTenRep
```

ステップ 3 ディスクから VCS エージェントを削除します。

次のコマンドを使用して、VCS エージェントを削除します。

```
# rm -rf /opt/VRTSvcs/bin/OnOnlyProcess
# rm -rf /opt/VRTSvcs/bin/SubscriberManager
# rm -rf /opt/VRTSvcs/bin/TimesTenRep
```
