



## **Cisco BTS 10200 Softswitch CORBA アダプタ インターフェイス仕様 プログラマ ガイド**

Release 5  
February 16, 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。見当たらない場合には、代理店にご連絡ください。

シスコが採用している 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 パーミッションとして、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、すべてのマニュアルおよび上記各社のソフトウェアは、障害も含めて「現状のまま」として提供されます。シスコおよび上記各社は、商品性や特定の目的への適合性、権利を侵害しないことに関する、または取り扱い、使用、または取り引きによって発生する、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ません。

CCVP, the Cisco Logo, and the Cisco Square Bridge logo are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn is a service mark of Cisco Systems, Inc.; and Access Registrar, Aironet, BPX, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Enterprise/Solver, EtherChannel, EtherFast, EtherSwitch, Fast Step, Follow Me Browsing, FormShare, GigaDrive, GigaStack, HomeLink, Internet Quotient, IOS, iPhone, IP/TV, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, iQuick Study, LightStream, Linksys, MeetingPlace, MGX, Networking Academy, Network Registrar, Packet, PIX, ProConnect, RateMUX, ScriptShare, SlideCast, SMARTnet, StackWise, The Fastest Way to Increase Your Internet Quotient, and TransPath 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. (0612R)

*Cisco BTS 10200 Softswitch CORBA アダプタ インターフェイス仕様 プログラム ガイド*

Copyright © 2007 Cisco Systems, Inc.

All rights reserved.



## CONTENTS

<b>このマニュアルについて</b>	vii
対象読者	viii
インターフェイス	viii
XML インターフェイス	viii
データの一括取得 (ページング)	viii
接続とトランザクション	ix
技術情報の入手方法、サポートの利用方法、およびセキュリティ ガイドライン	ix

### CHAPTER 1

## **CORBA のアーキテクチャとアプリケーション プログラミング インターフェイス** 1-1

CORBA アダプタのアーキテクチャ	1-1
ORB の仕様	1-2
コンパイラ ツール	1-3
ORB の展開	1-3
NameService	1-9
BTS 10200 ソフトスイッチの IDL	1-10
Bts10200 の API	1-11
Bts10200 のセキュリティ API	1-11
Login	1-11
Logout	1-11
Bts10200 のプロビジョニング API	1-12
getCommandDoc	1-12
request	1-12
マクロ コマンド	1-13
動作とアトリビュート	1-13
マクロ コマンドの管理	1-13
CORBA/XML インターフェイス	1-15
外部インターフェイス	1-16
CAD インターフェイス	1-16
操作	1-17
cadexception	1-18

CHAPTER 2

**Extensible Markup Language の処理 2-1**

- XML およびコンポーネント 2-1
  - CORBA インターフェイス サーバントにおける XML 2-1
    - CIS の機能 2-1
    - ManagedObject 2-5
    - Request 2-6
    - Reply 2-6
  - CORBA インターフェイス サーバント アダプタ実装 2-7
    - Cisco BTS 10200 ソフトスイッチの IDL コード 2-7

CHAPTER 3

**CORBA の Secure Sockets Layer サポート 3-1**

- システム セキュリティ拡張のシステム コンテキスト 3-3
  - 依存関係 3-3
    - 縮小 Solaris イメージ 3-3
    - Java の SSL 実装 3-3
  - 証明書およびキー パスワード 3-4

CHAPTER 4

**プロキシ 4-1**

CHAPTER 5

**トラブルシューティング 5-1**

- cadexception 5-1
- CORBA ネットワーク設定の変更 5-5
- CORBA が Cisco BTS 10200 ソフトスイッチに接続できない 5-6
- CORBA と EPOM のトラブルシューティング手順 5-7
- CORBA と EPOM の特殊文字のトラブルシューティング : subscriber コマンド 5-9
  - subscriber noun と add verb A-1
  - 外部キー リレーション A-8
  - XML 要求バッチ ファイル B-1
  - CLI から CORBA XML へのトランザクション B-5
  - BTSxsd Java の機能 C-1
    - SDK C-1
    - Java C-2
    - CORBA インターフェイス サーバント C-2
      - 拡張機能 C-2
      - デュアル モード動作 C-2
  - 前提条件 C-2
  - OpenORB の設定 C-3
  - BTSxsd のビルド C-3

BTSxsdk の実行 C-4

INDEX

索引





## このマニュアルについて

このマニュアルでは、CORBA アダプタ (CAD) について説明します。CAD は、CORBA (Common Object Request Broker Architecture) を利用して Machine-to-Machine Interface (MMI; マシン間インターフェイス) を提供します。このアーキテクチャは、Object Management Group (OMG; オブジェクト管理グループ) という組織によって定義されています。

CAD インターフェイスは、Cisco BTS 10200 ソフトスイッチ製品をプロビジョニングする手段を提供するもので、機能は Command Line Interface (CLI; コマンドライン インターフェイス) アダプタに類似しています。CAD を通じて、Cisco BTS 10200 ソフトスイッチを一貫性のあるオブジェクト指向モデルに抽象化することができます。このインターフェイスの実際のオブジェクトモデルについては、このマニュアルでは説明しません。

CORBA インターフェイスで使用されるトークン (パラメータ) とその値の詳細については、『Cisco BTS 10200 Softswitch Command Line Interface Reference Guide』を参照してください。



(注)

このマニュアルでは、OpenORB を使用した CORBA について説明します。Visibroker を使用する場合は、『Cisco BTS 10200 Softswitch Release 3.2/3.3/3.5 Programmers Specification』を利用してください。

### 改版履歴

リリース番号	変更点
5.0	変更点はありません。
4.5.1	CORBA 仕様 Release 2.4 に準拠しました。「接続とトランザクション」項に情報を追加しました。「データの一括取得 (ページング)」項に情報を追加しました。アーキテクチャと設計に関する情報を新しく追加しました。「プロキシ」章およびトラブルシューティング情報を追加しました。Software Development Kit (SDK; ソフトウェア開発キット) による拡張機能の詳細については、付録 C「CORBA クライアント パッケージ (BTSxsdk) のサンプル実装」を参照してください。
4.5	「付録 C」を新しく追加しました。OpenORB を 1.3.0 から 1.3.1 に変更しました (jar ファイル名で使用されているものも含む)。
4.4.0/1	「CORBA の Secure Sockets Layer サポート」、「データの一括取得 (ページング)」、および「XML インターフェイス」の情報を追加しました。BTS 10200 IDL を更新しました。セキュリティとトラブルシューティングに関する第 4 章と第 5 章を新しく追加しました。
4.1	変更点はありません。
3.5.2	OpenORB の変更点が組み込まれました。

3.5	マクロ コマンドが導入されました。
3.3	CORBA の拡張機能が導入されました。
1.0	CORBA が導入されました。
すべて	無効な文字に関する表を追加しました。

## 対象読者

このマニュアルは、中級から上級の CORBA プログラマを対象としています。CORBA プログラミングについて、中程度以上の知識があることが前提になります。

## インターフェイス

この CORBA アダプタ (CAD) では、CORBA 仕様 2.4 で定義されている分散オブジェクトベースのアプリケーションの開発および展開に、OpenORB 1.3.1 インターフェイスを使用します。

## XML インターフェイス

この XML インターフェイスは、CORBA 自体から派生したものです。OpenORB パッケージは、Transmission Control Protocol (TCP) または User Datagram Protocol (UDP) を使用する Internet Inter-ORB Protocol (IIOP) を接続に使用します。NameService のナローイングによって、BTS10200 のオブジェクトも生成されます。ナローイングの詳細については、Cisco BTS 10200 ソフトスイッチ SDK パッケージのコード例で説明しています。このパッケージは、Cisco BTS 10200 ソフトスイッチ アプリケーションに付属しています。

アプリケーションレベルの個々の接続またはオブジェクトを作成するには、大量のオブジェクトプール化と多数のログインを通じて、有効なログイン キーを BTS10200 オブジェクトのインスタンスごとに取得する必要があります。この例については、付属の SDK パッケージで参照できます。

## データの一括取得 (ページング)

CLI の show コマンドを使用すると、データを一括で取得すること (ページング) ができます。大きなデータセットを表示する場合は、ページングの使用が必須です。ただし、データを初めて要求してページングがセットアップされるときは、パフォーマンスが低くなります。いったんページングされたデータを以降に要求した場合は、データがキャッシュされているため、高速で取得することができます。この処理は、セッション (キー) ごとに適用されます。該当のセッションに対して他の何らかのコマンドを実行すると、キャッシュされているデータはすべてフラッシュされます。最高のパフォーマンスを得るには、要求するページング済みのデータが、すべて連続している必要があります。詳細については、『Cisco BTS 10200 Softswitch Operations, Maintenance and Troubleshooting Manual』および『Cisco BTS 10200 Command Line Interface Reference Guide』を参照してください。



(注)

limit トークンについては、単一の show コマンドによるページングは、自動的に最大 500 行に制限されます。



## 接続とトランザクション

CORBA は、最大で 50 の同時セッションをサポートします。セッションとは、CORBA インターフェイスに対する有効なログインです。また、CORBA は 20 の同時トランザクションをサポートします。トランザクションとは、個々の要求です ( show や change など )。

各ログイン セッションには、個々の CLI と同様に、それぞれ固有のリソース セットが割り当てられます。これらのセッションは、アイドル状態になっていないかどうかを監査対象になります。セッションが 10 分以上アクティブになっていない ( コマンドを実行していない ) 場合、セッションはアイドル状態と見なされ、インターフェイスから削除されて、すべてのリソースが解放されます。



(注)

このアイドル期間は、以前のリリースとは別の値に変更されました。以前のデフォルトは 30 分で、現在は 10 分です。アイドル期間は、CIS アプリケーションの `bts.properties` ファイルで設定できます。このプロパティ ファイルは、Cisco BTS 10200 ソフトスイッチの `/opt/BTScis/etc` ディレクトリにあります。

## 技術情報の入手方法、サポートの利用方法、およびセキュリティ ガイドライン

技術情報の入手、サポートの利用、技術情報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、推奨するエイリアスおよび一般的なシスコのマニュアルに関する情報は、月刊の『*What's New in Cisco Product Documentation*』を参照してください。ここには、新規および改訂版のシスコの技術マニュアルもすべて記載されています。次の URL からアクセスできます。

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





# CORBA のアーキテクチャと アプリケーション プログラミング インターフェイス

この章では、Cisco BTS 10200 ソフトスイッチの CORBA (Common Object Request Broker Architecture) アダプタのアーキテクチャと 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) について説明します。

## CORBA アダプタのアーキテクチャ

この CORBA アダプタ (CAD) インターフェイスは、Cisco BTS 10200 ソフトスイッチの Element Management System (EMS; 要素管理システム) コンポーネントのアダプタ アーキテクチャを利用します。このアーキテクチャは、外部インターフェイスを EMS の共通インフラストラクチャに適応させることにより、各種のアダプタで Operations Administration Maintenance and Provisioning (OAM&P; 保守運用管理とプロビジョニング) の提供を可能にします。図 1-1 に CAD のアーキテクチャ全体を図示し、CORBA のアーキテクチャを示します。

このアーキテクチャは、セキュア CORBA と非セキュア CORBA が同時にアクティブになるデュアル モードのサポートを提供します。これは、以前のリリースではインストール オプションでした。非セキュア モードとセキュア モードが完全にサポートされます。このデュアル モード サポートの一部として、CIS アプリケーションを管理する Cisco BTS 10200 ソフトスイッチの EMS 上に 2 つの Java プロセスが存在します。各プロセスには、それぞれの固有機能を示す一意の名前が付けられています。

- 非セキュア : `-D_CIS_HIOP`
- セキュア : `-D_CIS_SSLIOP`

このデュアル モード サポートには副作用があります。両方のモードがアクティブになっている場合、新しいセキュア CORBA モードは以前のリリースと同様には動作しません。この新しい動作にはネーム スペースの衝突が含まれるため、問題を避けるために新しいセキュア POA コンテキストおよびネーム スペースが提供されています。

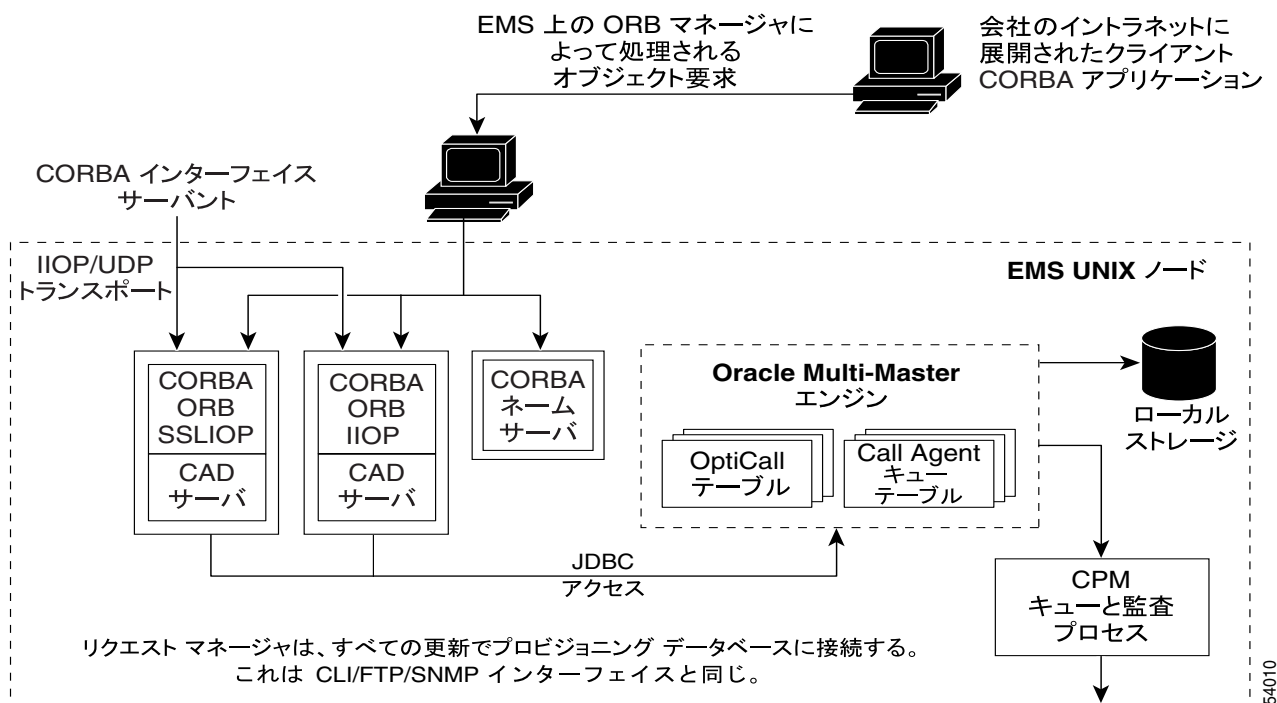
非セキュア モードとセキュア モードのどちらでも、アクティブな EM01 EMS アプリケーション上でのみ接続が許可されます。アクティブな EM01 アプリケーションが障害状態になるか、冗長メイトに手動でスイッチオーバーすると、自動的に接続がドロップされます。さらに、該当する EMS へ新しいクエリーが解決されないように、オブジェクトがローカル INS または NameService から削除されます。この動作によって、スタンバイ EMS からのプロビジョニングを防止し、従来の CORBA の問題を解決しています。

Virtual IP address (VIP; パーチャル IP アドレス) が設定されている場合、CORBA インストールは自動的に VIP を iiop.hostname として使用します。VIP が設定されていない場合は、最初の EMS 管理 IP アドレスが iiop.hostname として使用されます。これにより、NameService がアクティブな EMS 上のすべての IP アドレスをリスンできます。VIP の使用については、『Cisco BTS 10200 Softswitch CLI Reference Guide, Operations, Administration and Maintenance』を参照してください。

CORBA をインストールすると、CORBA SDK も EMS 上にインストールされます。インストーラでサンプルのテスト プログラムを実行すると、CORBA がインストールされたことを確認できます。詳細については、付録 C 「CORBA クライアント パッケージ (BTSxsdk) のサンプル実装」を参照してください。

クライアント アプリケーションの開発は、CIS アプリケーションの bts.properties ファイルの影響を受けます。ログイン セッションは、10 分間放置すると期限切れになります。つまり、セッションをアクティブな状態に保つには、10 分に 1 回、コマンドが各セッションに発行される必要があります。これは、接続プールの使用を伴うすべてのクライアント アプリケーションで重要です。

図 1-1 CORBA のアーキテクチャ



## ORB の仕様

CAD インターフェイスで使用される Object Request Broker (ORB; オブジェクト リクエスト ブローカ) は、OpenORB 1.3.1 準拠のパッケージです。この ORB は、Portable Object Adapter (POA; ポータブル オブジェクト アダプタ) など、この他の高度な機能もサポートしています。POA は CAD で使用される実装モデルです。

## コンパイラ ツール

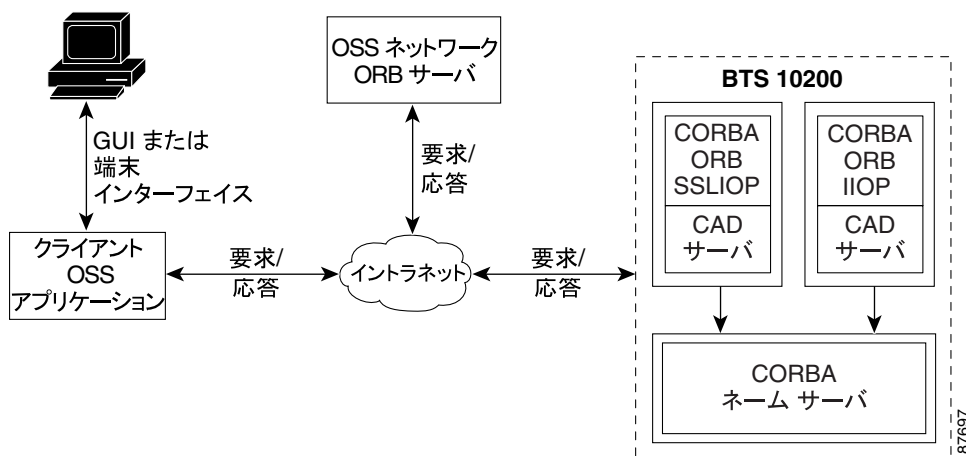
必要なコンパイラは、J2SE Development Kit (JDK) 1.4.1 以上です。Cisco BTS 10200 ソフトスイッチでは、コンパイルおよび Java Runtime Environment (JRE; Java ランタイム環境) に JDK1.5.0 (JDK5) を使用しています。クライアント側アプリケーションには、次のツールも必要になる場合があります。

- Xerces パーサー
- ECS レポート ビルダ

## ORB の展開

図 1-2 に、ORB の展開プロセスを示します。

図 1-2 ORB の展開



クライアント側アプリケーションに OpenORB を使用する場合、クライアント環境を正しく設定してアプリケーションを使用できる状態にするには、いくつかの基本的な手順を実行する必要があります。まず、OpenORB を Java のプライマリ ORB として使用するようクライアントマシン上の JVM を設定します。この設定は、実行時に実施することも、JVM の orb.properties で永続的に設定することもできます。後者の場合は、JVM インストールの所有者特権が必要です。通常、これはルートです。ルートユーザが次のコマンドを実行する必要があります。

```
java -jar openorb_1.3.1.jar
```

このコマンドによって、次の値を使用した orb.properties 設定が適切な場所に配置されます。

```
org.omg.CORBA.ORBClass=org.openorb.CORBA.ORB
org.omg.CORBA.ORBSingletonClass=org.openorb.CORBA.ORBSingleton
```

または、クライアントアプリケーションの各呼び出しに対する環境オーバーライドとして、この設定を提供する必要があります。

クライアント プログラマにとって有益となる値は、この他に OpenORB の DEBUG オプションがあります。これらのオプションは、クライアント アプリケーションによって生成されるデバッグ情報の量を制御するものです。デバッグ情報は、アプリケーションの標準出力に送信されます。このデータをプログラマがログ ファイルにキャプチャするには、基本的なシェル リダイレクト コマンドを使用してデータをリダイレクトします。Web サービスなど、この他の環境では操作が異なる場合があります。最も多くのデバッグ出力を取得できるのは、下に示したオプションです。これらのオプションを使用すると、IIOP/SSLIOIP メッセージのダンプが生成されます。このダンプは、Cisco BTS 10200 ソフトスイッチとの通信に関する問題のデバッグでは最も有益です。

```
-Dopenorb.debug.trace=DEBUG -Dopenorb.debug.level=HIGH
```

このオプションは、Java 呼び出しの一環で、クライアント アプリケーションに対する環境設定として提供されます。以降の例では、Interface Definition Language (IDL; インターフェイス定義言語) のコンパイルについて示すとともに、POA を検索してコンパイル済み IDL オブジェクトにバインドするために必要な Java コードを示します。

この一般的な OpenORB プロセスは、Cisco BTS 10200 ソフトスイッチと対話するなどのクライアント アプリケーションの開発でも重要な部分です。Cisco BTS 10200 ソフトスイッチで提供される IDL は、インターフェイス クラスにコンパイルしてからクライアント アプリケーションで使用する必要があります。



(注)

IDL とは、ある言語で記述されたプログラムまたはオブジェクトと、未知の言語で記述された別のプログラムとの通信を可能にする言語の一般的な呼称です。分散オブジェクト技術では、新しいオブジェクトが、送信先となる任意のプラットフォーム環境での実行方法を発見できる必要があります。ORB は、オブジェクト間のクライアント / サーバ関係において仲介者として機能するミドルウェア プログラムです。

このコード例では、Cisco BTS 10200 ソフトスイッチ製品で開発された Java パッケージ ツリーを使用しています。このコードは変更できます。他のクライアントでは、IDL インターフェイス オブジェクトを包含する別のパッケージ ツリーを指定できます。このスクリプトの詳細については、SDK を参照してください。

```
#!/bin/sh
#####
# Copyright (c) 2002, 2006 by Cisco Systems, Inc.
#
# AUTHOR: A. J. Blanchard
#
# DESC: Invoke the IDL compiler for the OpenORB package.
#
#####
set -e
set -a
#set -x

#
# List required jar files
#
CLASSPATH=./opt/BTSorb/lib/logkit.jar:/opt/BTSorb/lib/openorb-1.0.1.jar:/opt/BTSorb/lib/openorb_tools-1.3.1.jar:/opt/BTSorb/lib/xerces.jar:/opt/BTSorb/lib/avalon-framework.jar:/opt/BTSorb/lib/openorb_ots-1.3.1.jar:/opt/BTSorb/lib/openorb_pss-1.3.1.jar:/opt/BTSorb/lib/openorb_ins-1.3.1.jar:/opt/BTSorb/lib/openorb_tns-1.3.1.jar

export CLASSPATH

java -classpath $CLASSPATH org.openorb.compiler.IdlCompiler $1 -jdk1.4 -all -verbose -d ./
```

Java ファイルは、パッケージ ディレクトリ内で指定されたローカル ディレクトリ ツリーに生成されます。このパッケージ パスは、オブジェクト インターフェイス実装を発見するためのバインド ロジックで必要になります。

```
#!/bin/sh
#####
# Copyright (c) 2002, 2006 by Cisco Systems, Inc.
#
# AUTHOR: A. J. Blanchard
#
# DESC: Compile Java ORB programs with the required components from OpenORB.
#
#####
set -e
set -a
#set -x

CLASSPATH=./opt/BTSoorb/lib/logkit.jar:/opt/BTSoorb/lib/openorb-1.3.1.jar:/opt/BTSoorb/
lib/openorb_tools-1.3.1.jar:/opt/BTSoorb/lib/xerces.jar:$HOME/mb/devel/em/lib/cad.jar
:$HOME/mb/devel/em/lib/ecs-1.4.1.jar:/opt/BTSoorb/lib/openorb_ots-1.3.1.jar:/opt/BTSoo
rb/lib/openorb_pss-1.3.1.jar:/opt/BTSoorb/lib/openorb_ins-1.3.1.jar:/opt/BTSoorb/lib/o
penorb_tns-1.3.1.jar

export CLASSPATH

javac -classpath $CLASSPATH -d ./ $*
```

Java ファイルのパッケージをコンパイルして、必要なクラス ファイルを生成します。これらのクラ スファイルが、クライアントのクラスパスに存在している必要があります。



(注)

この例は、単一のディレクトリ内にあるすべての Java ファイルを 1 つのコマンドだけでビルドす る場合の一般的な方法を示しています。これは、Java コードを最も早く一括でコンパイルする方 法の 1 つです。

Cisco BTS 10200 ソフトスイッチは、CORBA インターフェイスの利用例を網羅した Software Development Kit (SDK; ソフトウェア開発キット) を提供しています。これらの例では、次のよう な多くのトピックを取り上げています。

- CLI
- バッチ ファイル処理
- マルチスレッドの同時使用
- SSL



(注)

下の例は、BTS 10200 オブジェクトの基本的な抽象化を示しています。この他のツールを使用する場合は、IDL オブジェクトを OpenORB ファイルとともに修正する必要があります。

```

package com.sswitch.oam.ccc;

import java.lang.*;
import java.io.*;
import java.util.*;
import java.text.*;
// CORBA stuff
import org.omg.CORBA.*;
import org.omg.PortableServer.*;
import org.omg.Messaging.*;
// XML Stuff
import org.apache.ecs.xml.*;
import org.apache.ecs.*;
import org.w3c.dom.*;
import org.xml.sax.*;
import org.apache.xml.serialize.*;
// BTS Code jar files...
import com.sswitch.oam.cad.*;
import com.sswitch.oam.xml.*;
import com.sswitch.oam.util.*;

/**
 * CorbaXmlIntf.java
 * Copyright (c) 2002, 2006 by Cisco Systems, Inc.
 * -- This is the client side driver stub. This allows the client application
 * to generate the Request object which is then digested in this class as a
 * XML document and sent as a request to the CORBA server. The results are
 * then returned to the user or the CORBA exception is thrown.
 *
 * @author A. J. Blanchard
 * @version 4.0
 * @since BTS 10200 4.0
 */

public class CorbaXmlIntf {

    /**
     * Class private data
     */
    private String []                objArgs;
    private org.omg.CORBA.ORB        objOrb;
    private org.omg.CosNaming.NamingContextExt objContext;

    private com.sswitch.oam.cad.Bts10200        objBts;
    private com.sswitch.oam.cad.Bts10200_Security objBtsSec;
    private org.omg.CORBA.StringHolder        objKey;

    /**
     * Generic Constructor for the test driver.
     */
    public CorbaXmlIntf(String[] args)
    {
        // Initialize the ORB.
        objOrb = org.omg.CORBA.ORB.init(args, null);
        objArgs = args;
        return;
    }

    /**
     * This is the primary execution method for the object. It performs the
     * actual request and calls for the print of the reply.
     */
    public void connect() throws CadExceptions
    {
        //
        // Log into the target machine with generic optiuser
        //
        try {

```



```

        bind();
        objKey = new org.omg.CORBA.stringholder();
        objBtsSec.login("btsadmin", "btsadmin", objKey);
        Log.info("BTS10200 Login successful: "+objKey.value);
    }
    catch (Exception e) {
        Log.error("Exception in CORBA Bind/Login = "+
            Util.stackTraceToString(e));
        throw new CadExceptions(1, e.toString());
    }
}

/**
 * This method generate the request to the CORBA server and returns
 * the reply or an exception if the interface throws an exception.
 * The argument "request" must be an XML formatted document.
 *
 * @param request    This XML request document.
 * @returns String   This is the XML formatted answer.
 */
public String    request(String request)
    throws CadExceptions
{
    String    answer=null;
    try {
        org.omg.CORBA.StringHolder reply = new org.omg.CORBA.StringHolder();

        // Issue request to BTS 10200
        objBts.request(request, objKey.value, reply);

        // Build an XMLReply from the document

        answer= reply.value;
    }
    catch (Exception e) {
        Log.warning("Request Command Exception:\n " +
            Util.stackTraceToString(e));
        throw new CadExceptions(1, e.toString());
    }
    return answer;
} // end request()

/**
 * This method generate the request for a command document to the
 * CORBA server and returns the reply or an exception if the interface
 * throws an exception.
 *
 * @param noun        This noun for the request.
 * @param verb        This verb for the request.
 * @returns String    This is the XML formatted answer.
 */
public String    getCommandDoc(String verb, String noun)
    throws CadExceptions
{
    String    answer=null;
    try {
        org.omg.CORBA.StringHolder reply = new org.omg.CORBA.StringHolder();

        // Issue request to BTS 10200
        objBts.getCommandDoc(noun, verb, objKey.value, reply);

        // Build an XMLReply from the document

        answer= reply.value;
    }
    catch (Exception e) {
        Log.warning("Request Command Exception:\n " +
            Util.stackTraceToString(e));
        throw new CadExceptions(1, e.toString());
    }
    return answer;
} // end getCommandDoc()

/**
 * This method generate the request for a command document to the
 * CORBA server and returns the reply or an exception if the interface
 * throws an exception.

```

```

*
* @param noun      This noun for the request.
* @param verb     This verb for the request.
* @returns String  This is the XML formatted answer.
*/
public String  getExtCommandDoc(String verb, String noun)
    throws CadExceptions
{
    String  answer=null;
    try {
        org.omg.CORBA.StringHolder reply = new org.omg.CORBA.StringHolder();

        // Issue request to BTS 10200
        objBts.getExtCommandDoc(noun, verb, objKey.value, reply);

        // Build an XMLReply from the document
        answer= reply.value;
    }
    catch (Exception e) {
        Log.warning("Request Command Exception:\n " +
                    Util.stackTraceToString(e));
        throw new CadExceptions(1, e.toString());
    }
    return answer;
} // end getExtCommandDoc()

/**
 * This module disconnects the user from the BTS 10200 CORBA interface.
 */
public void  disconnect()  throws CadExceptions
{
    objBtsSec.logout(objKey.value);
    return;
}

/*=====
 * Internal processing methods...
 *=====*/

/**
 * This method binds to the target CORBA objects for us to operate
 */
protected void  bind()
    throws org.omg.CORBA.ORBPackage.InvalidName,
           org.omg.CosNaming.NamingContextPackage.InvalidName,
           org.omg.CosNaming.NamingContextPackage.NotFound,
           org.omg.CosNaming.NamingContextPackage.CannotProceed
{
    org.omg.CosNaming.NameComponent[] nameComponent = null;
    org.omg.CORBA.Object result = null;

    insLocate();

    result = objContext.resolve(objContext.to_name("Bts10200_Security_poa"));
    objBtsSec = Bts10200_SecurityHelper.narrow(result);

    result = objContext.resolve(objContext.to_name("Bts10200_poa"));
    objBts = Bts10200Helper.narrow(result);
    Log.info("Basic POA(s) have been located and bound.");
    return;
}

/**
 * Load the name service and find the context for the CORBA objects.
 * Remember, the INS must be the one located on the BTS. This has the
 * object references. Use a 'corbaloc:' for now but later a migration
 * to URL for name service location would be good.
 */
protected void  insLocate()
    throws org.omg.CORBA.ORBPackage.InvalidName
{
    //System.out.println("Locate NameService in system.");
    org.omg.CORBA.Object initial_context_obj =
        objOrb.resolve_initial_references("NameService");
    objContext =
        org.omg.CosNaming.NamingContextExtHelper.narrow(initial_context_obj);
}

```

```
        Log.info("NameService found in initial context.");
        return;
    }
} // end CorbaXmlIntf
```

- 実際の実装では、POA 選択を動的に実行すること、および何らかの形でサイトへの誘導（たとえば、ソフトスイッチのホーム ロケーションやソフトスイッチ ID の一部への誘導）に基づいて実行することが可能です。いったん POA が選択されると、すべてのオブジェクト実装は同一になります。どのオブジェクトでもサイト固有の動作はありません。ただし、サイト固有アトリビュートは存在し、ローカル データベースの内容に基づいて導出されます。

## NameService

OpenORB の NameService モジュールは、NameService Specification Version 1.2 (2002 年 9 月) の Object Management Group (OMG; オブジェクト管理グループ) 準拠の実装を提供します。このモジュールは、Cisco BTS 10200 ソフトスイッチ上の CORBA の動作には必須です。クライアントは、NameService に接続し、corbaloc プロセスを通じて Cisco BTS 10200 ソフトスイッチ CORBA オブジェクトへの参照を取得します。アプリケーションのクライアント側で OpenORB を使用する場合は、次の構文を適用して NameService に接続します。

```
"corbaloc::1.2@<Host Name>:14001/NameService"
```

*corbaloc* 文字列は、/opt/BTSorb/config ディレクトリにある OpenORB.xml コンフィギュレーション ファイルで初期参照として指定できます。または、クライアント アプリケーションで必要に応じて動的に生成することができます。ホスト名は、IP アドレスにすることをお勧めします。

各 Cisco BTS 10200 ソフトスイッチ EMS には管理インターフェイスがあり、INS または NameService では、これらの両方の管理インターフェイスにわたって解決されるホスト名を使用する必要があります。これは必須条件です。ただし、クライアント側アクセスは、どのインターフェイスが使用されているかを意識しません。クライアントは、所定の管理インターフェイスがさまざまな理由でダウンする可能性があること、およびそれがネーム サービスへのアクセスに影響することを認識している必要があります。どのサブネットでも、スイッチやルータの障害は時間の経過に伴って自然に発生する可能性があり、これに対処するためには、両方の IP アドレスを使用し、ホスト名でリトライすることを推奨する必要があります。

各 Cisco BTS 10200 ソフトスイッチには、それぞれ独自のデュプレックス INS ペアがあります。各 INS は、Cisco BTS 10200 ソフトスイッチの片側からのオブジェクトを表しています。このリソース ロケーション文字列を使用して、NameService への参照を生成します。各 Cisco BTS 10200 ソフトスイッチにはそれぞれ独自のネーム サービス インスタンスがあり、ネーム サービスは EMS ごとに個別に利用できます。このモジュールの Cisco BTS 10200 ソフトスイッチのデフォルト UDP ポートは、14001 です。OpenORB モデルでは、この値はコンフィギュレーション ファイル OpenORB.xml で渡されます。ソフトウェア開発キット (SDK) には、OpenORB クライアント実装を使用した動作可能サンプルをビルドするためのコード例とともに、この設定の例が含まれています。

要求インターセプタを適用すると、複数の NameService モジュールを使用できます。プロキシ オブジェクトを使用することで、ForwardRequest (CORBA 3.0 spec. 1.3.1) プロトコルを使用して要求を転送できます。詳細については、第 4 章「プロキシ」を参照してください。

NameService モジュールを使用したオブジェクト解決の例を次に示します。ここでは、基本的な POA をローカル Cisco BTS 10200 ソフトスイッチへのルートレベル参照として使用していることに注意してください。

```
org.omg.CORBA.Object initial_context_obj =
    objOrb.resolve_initial_references("NameService");
objContext =
    org.omg.CosNaming.NamingContextExtHelper.narrow(initial_context_obj);
result = objContext.resolve(objContext.to_name("Bts10200_poa"));
objBts = Bts10200Helper.narrow(result);
```

## BTS 10200 ソフトスイッチの IDL

この IDL は、CAD インターフェイスのオブジェクトレベル インターフェイスを表現するために使用されます。このオブジェクト インターフェイスには、オブジェクトの属性と動作が含まれています。この項では、Cisco BTS 10200 ソフトスイッチの IDL の概要を示します。これらの IDL オブジェクトは、Cisco BTS 10200 ソフトスイッチのプロビジョニングに使用される XML 記述およびドキュメントへのアクセスを定義するものです。この XML ドキュメントの詳細については、後の章で扱います。CORBA は、ほとんどの部分でこれらの XML ドキュメントのトランスポートとして機能します。

bts10200.idl ファイルは、システム全体の全般的なデータ構造と型定義を含んでいます。また、エラー インターフェイス(例外)も保持しています。bts10200.idl ファイルの全文については、[P.2-7 の「Cisco BTS 10200 ソフトスイッチの IDL コード」](#)を参照してください。このファイルは、Cisco BTS 10200 ソフトスイッチで使用するために定義されたすべてのオブジェクトを含んでいます。各オブジェクトの内訳を次に示します。

- **Bts10200\_Security** : プライマリ セキュリティ オブジェクト。別のオブジェクトで 사용되는ログイン キーの作成に使用されます。Cisco BTS 10200 ソフトスイッチにアクセスするには、このオブジェクトが必要です。
- **Bts10200** : XML 記述ドキュメントのほか、プロビジョニング ドキュメントと制御ドキュメントの取得に使用される基本オブジェクト。
- **CadException** : Cisco BTS 10200 ソフトスイッチ CAD インターフェイスのすべてのエラー報告に使用されるオブジェクト。
- **Macro** : このオブジェクトは、独自の *show* コマンドや *display* コマンドを定義し、Cisco BTS 10200 ソフトスイッチ上で実行します。このオブジェクトによって、ユーザは複雑な関係を基にしてシンプルな表示コマンドを作成できるようになり、これらのコマンドを永続的に保存して、以後に「マクロ」コマンドとして再び呼び出すことができます。

## Bts10200 の API

この項では、CAD インターフェイスへの実際の API コールについて説明します。前提条件は、クライアント アプリケーションが Java 言語で開発されていることです。C++ を使用できないわけではありませんが、このマニュアルでは取り扱いません。

関連オブジェクトのメソッドを呼び出すごとに、挙げられているすべてのパラメータが必要です。

## Bts10200 のセキュリティ API

Cisco BTS 10200 ソフトスイッチのセキュリティ オブジェクト (Bts10200\_Security) は、Cisco BTS 10200 ソフトスイッチの CAD インターフェイスに関するさまざまなレベルのセキュリティをユーザに提供します。認可されたユーザは、セキュリティ キーを取得して、このキーを以後のすべてのトランザクションで使用できます。このオブジェクトは、インターフェイスの他のどの CORBA メソッド呼び出しよりも先に使用する必要があります。このキーが CAD インターフェイスで有効となるのは、ユーザのセッションが継続している間です。logout メソッドが一度でも呼び出されると、キーは無効になります。同様に、システムに対するアクセスが 10 分以上なかった場合はセキュリティ キーが期限切れになり、ユーザは自動的に CAD インターフェイスからログアウトされます。ユーザ名とパスワードは、CLI/MAC アダプタ インターフェイスで許容されるものと同値であり、同じ認可権限が適用されます。

この項の各メソッドは、Bts10200\_Security インターフェイスの一部です。挙げられているパラメータは、メソッドごとに必要であり、データを含んでいる必要があります。

### Login

login メソッドは、CORBA インターフェイス ユーザの認証を提供します。FTP アダプタや CLI アダプタと同一のユーザセキュリティを利用しています。このメソッドは、キーとして定義された文字列値を返します。このキーは、CAD インターフェイスに対するすべてのトランザクションで必要になります。これは、特定ユーザ固有の認可を示す認証キーです。このメソッドのシグニチャは、次のコードで定義されます。

```
int login (java.lang.String user, java.lang.String passwd, java.lang.StringHolder key)
throws CadExceptions
```

- **戻り値**：処理の成功または失敗を示すステータス。失敗は、ファシリティが使用不能であることを意味します。成功は、処理が完了したことを意味します。
- **例外**：CadException は、要求の処理で動作エラーが発生したことを意味します。これには、パラメータの型、範囲、およびデータベース アクセスに関する問題が含まれます。

### Logout

logout メソッドは、ログイン セッションを終了します。認証キーの有効性を破棄します。このメソッドが完了すると、そのキーはこの他のメソッド呼び出しに使用できなくなります。このメソッドのシグニチャは、次のコードで定義されます。

```
int logout (java.lang.String) throws CadExceptions
```

- **戻り値**：処理の成功または失敗を示すステータス。失敗が示された場合は、ファシリティが使用不能であることを意味します。成功の戻り値は、処理が完了したことを示します。
- **例外**：CadException は、要求の処理で動作エラーが発生したことを意味します。これには、パラメータの型、範囲、およびデータベース アクセスに関する問題が含まれます。

## Bts10200 のプロビジョニング API

Cisco BTS 10200 ソフトスイッチ オブジェクト (Bts10200) は、認可されたユーザに対して Cisco BTS 10200 ソフトスイッチ CLI エンジンへのプロビジョニング インターフェイス機能を提供します。第 2 章「Extensible Markup Language の処理」で説明するように、入力と出力は両方とも XML になります。CLI コマンドは、CORBA インターフェイスを通じて送信される前に解析され、XML ドキュメントに変換されます。次に、CORBA CIS サーバが CLI プロビジョニング コマンドを実行して、応答を XML ドキュメントで返送します。この項の各メソッドは、Bts10200 インターフェイスの一部です。挙げられているパラメータは、メソッドごとに必要であり、データを含んでいる必要があります。

### getCommandDoc

getCommandDoc メソッドは、コマンド記述の取得を提供します。このメソッドは、特定の名詞 (noun) / 動詞 (verb) の組み合わせについて、コマンドの構文とオプションを記述した XML ドキュメントを取得します。このメソッドのシグニチャは、次のコードで定義されます。

```
void getCommandDoc (java.lang.String noun, java.lang.String verb, java.lang.String key, org.omg.CORBA.StringHolder reply) throws CadExceptions
```

ここで、

- パラメータ *noun* : コマンドの noun です。
- パラメータ *verb* : コマンドの verb です。
- パラメータ *key* : ログインで取得された認可キーです。
- パラメータ *reply* : コマンドの構文とオプションが含まれた XML 応答です。
- 例外 (cadexception) : 要求の処理で動作エラーが発生したことを意味します。これには、パラメータの型、範囲、およびデータベース アクセスに関する問題が含まれます。

### request

request メソッドは、CORBA インターフェイスを通じて XML ドキュメント ベースのプロビジョニング要求を処理します。CORBA CIS サーバがプロビジョニング コマンドを実行して、応答を XML ドキュメントで返送します。このメソッドのシグニチャは、次のコードで定義されます。

```
void request (java.lang.String command, java.lang.String key, org.omg.CORBA.StringHolder reply) throws CadExceptions
```

ここで、

- パラメータ *command* : XML ドキュメントによるプロビジョニング コマンド要求です。
- パラメータ *key* : ログインで取得された認可キーです。
- パラメータ *reply* : CLI エンジンでのコマンド実行の XML 応答です。
- 例外 (cadexception) : 要求の処理で動作エラーが発生したことを意味します。これには、パラメータの型、範囲、およびデータベース アクセスに関する問題が含まれます。

## マクロ コマンド

Cisco BTS 10200 ソフトスイッチは、自身のデータベース内にある複雑なデータの高レベル（概要）ビューを提供する機能を必要とします。通常、データベース項目間の関係を特定するには、データベースに何回かコマンドを発行して、複数の要求を行う必要があります。このプロセスは低速でコストがかかり、Operations Support System（OSS; オペレーション サポート システム）管理システムの処理の進行を阻害します。このため、複数の操作を、データベースに対する単一の要求に縮小できる最適化されたアプローチが必要です。この要求では、一連の定義済みルールに基づいて正しい関連データが返される必要があります。

Cisco BTS 10200 ソフトスイッチの複雑なデータ関係を表示できる機能のことを、マクロと呼びます。マクロは、ユーザが定義した関係ルールを使用して、複数のコマンドによる複合クエリーを構築する単一のコマンドです。

マクロ コマンドは Cisco BTS 10200 ソフトスイッチ固有のものであり、シンプレックス構成とデュプレックス構成の両方で機能します。このドキュメントにおけるこのコマンドの主な目的は、CORBA アダプタ（CAD）機能の CORBA インターフェイス定義言語（IDL）インターフェイスでこのコマンドを利用することです。



(注)

マクロ コマンドは Release 3.5 で利用可能になりました。Release 3.2 および Release 3.3 では利用できません。

## 動作とアトリビュート

マクロ コマンド インターフェイスは、ユーザが Cisco BTS 10200 ソフトスイッチのデータベースで複数のテーブルを選択および定義することを可能にします。通常のコマンドは、単一のデバイスまたはテーブルの両方、あるいはその一方を操作するだけです。

マクロ コマンド インターフェイスを使用して、ユーザがデータベース内のテーブルに書き込むことはできません。この操作については、データベース テーブルに対する実行を禁止するルールおよび制約事項が多数あります。

## マクロ コマンドの管理

マクロ コマンドの管理は、マクロ コマンド定義を作成、変更、および削除するユーザ コマンドで構成されます。これらのユーザ コマンドによって、実際のマクロ コマンドを定義および操作し、標準のプロビジョニング コマンドとして実行します。この他の主な構成要素は、マクロの実行です。これは CORBA インターフェイスを通じて提供されます。マクロ コマンドの管理ルールを追加のユーザ指定データとともに使用すると、マクロ パラメータのインスタンス値が返されます。

## マクロの定義

マクロ コマンドのビルドに使用される値は、内部的に検証されます。厳密には、各マクロは複数の noun とその関連パラメータを持ったスーパーセット show コマンドです。マクロ コマンドの作成に使用される値は、実際のテーブル名やカラム名からではなく、すべて noun のパラメータから導出されます。これにより、Cisco BTS 10200 ソフトスイッチ スキーマの抽象化を維持できます。以降のマクロ コマンド管理の例では、マクロ コマンドの一般的な作成、変更、および削除方法を示します。例に続く各項では、マクロ定義で使用される実際の値、およびこれらの値に関する制約事項について詳しく説明します。

```
add macro id=CTXG_NUMBERS; \
  parameters=office_code.NDC, \
             office_code.EC, \
             dn2subscriber.DN, \
             subscriber.CTXG_ID; \
  rules="office_code.OFFICE_CODE_INDEX=\
dn2subscriber.OFFICE_CODE_INDEX, \
dn2subscriber.SUB_ID=subscriber.ID";
```

既存のマクロを編集するときは、変更しようとするパラメータを入力する必要があります。この場合、パラメータとルールは（両方または一方が）リスト形式になっています。リスト全体を再入力する必要があります。マクロ データベース エントリに格納されているリストが置換されます。このことを次の例で示します。

```
change macro id=CTXG_NUMBERS; \
  parameters= office_code.NDC, \
             office_code.EC, \
             dn2subscriber.DN, \
             subscriber.ID;
```

マクロが不要になった場合は、削除できます。削除は delete macro コマンドで実行します。必要なパラメータは、マクロ ID だけです。該当するマクロに関連するすべての定義が、データベースから削除されます。delete macro コマンドの書式は次のとおりです。

```
delete macro id=CTXG_NUMBERS;
```

マクロの表示に必要なのは、マクロ ID だけです。コマンドの静的コンポーネントがすべて返されます。このコマンドの書式を次の例に示します。

```
show macro id=CTXG_NUMBERS;
```

## マクロ ID

マクロ ID は、マクロ定義へのすべての参照で、ハンドルとして使用されます。マクロを呼び出す場合は、ID がマクロを呼び出すための noun になります。また、change macro、delete macro、show macro の各コマンドでは、ID は特定の定義を編集するために使用されるプライマリ キーまたはパラメータです。ID は文字型のフィールドであり、他のどのマクロとも重複しない必要があります。ID は、最長で 79 文字です。マクロ ID によって、必要な操作を記述するための多様性を十分に備えた一意のマクロ定義が提供されます。



## パラメータ リスト

パラメータ リストは、マクロ コマンドに表示される noun とパラメータを列挙するために使用されます。これらの noun とパラメータは、応答に配置される選択項目を構成します。パラメータ リスト フィールドの値は、関連する Cisco BTS 10200 コマンドの noun とパラメータのカンマ区切りリストとなるように設計されています。このリストは、入力時には次の形式になります。

```
parameters=<noun.parameter>,<noun.parameter>,...,<noun.parameter>;
```

リストの各項目は、noun、ピリオド、その noun のパラメータという形式になっています。これらの noun とパラメータは、要素管理システム (EMS) を通じて検証されます。リストの各項目は、常に noun とパラメータという完全な形式で指定する必要があります。これは、「ID」などの利用頻度の高いパラメータ名を再使用できるようにするためです。

パラメータには、暗黙的な序列はありません。マクロで定義された順序に従って、コマンド処理に渡されます。所定の noun が持つ他のパラメータからの追加データ項目が含まれることはありません。

## ルール

マクロ コマンドの最も重要な部分は、ルールです。これらのルールによって、Cisco BTS 10200 ソフトスイッチ データベースから選択されるデータの関係を規定します。これらのルールは、データベース選択文の「where」節に相当します。ルールは、操作に必要なデータ サブセットに絞り込むために役立ちます。マクロに適用できる基本的なルール セットは、2 つあります。and ルールと or ルール、および equivalence と not です。一般的なパラメータ ルールは、and ルールとして適用されます。これらのルールの定義は、次のとおりです。

- **and ルール**：マクロでデータを選択するための必要条件を表す、1 つまたは複数のデータ修飾子。単一の and ルールによって、「特定の noun およびパラメータが他の特定の noun およびパラメータと等しくなる」ことを必要条件として指定できます。
- **or ルール**：マクロでデータを選択するための十分条件を表す、1 つまたは複数のデータ修飾子。単一の or ルールによって、「特定の noun およびパラメータが他の特定の noun およびパラメータと等しくなる」ことを十分条件として指定できます。
- **equivalence および not**：マクロ コマンドのルール セクションに、2 つのバリエーションを導入できます。これらを使用すると、noun/パラメータ ペアを等価または不等として記述できます。この表現の一般的な構文は、等価の場合は =、不等の場合は != です。

## ユーザ入力

ユーザ入力コンポーネントは、ルール リストでは必須ではありません。これは、マクロ呼び出し時に提供されます。この追加入力は and ルールの入力として扱われます。このデータは、選択されるデータ サブセットを定義するデータを絞り込むためのものです。このルール データでは、等号構文を使用してデータ選択のバリエーションを表現できます。たとえば、ユーザは、特定の機能を持たないすべてのサブスクリイバを検索することができます。

## CORBA/XML インターフェイス

CORBA Interface Servant ( CIS; CORBA インターフェイス サーバント ) は、マクロ コマンド インターフェイスのための新しい IDL インターフェイスをサポートしている必要があります。これには、次の新しいコンポーネントを使用できることを含みます。CIS サブシステムは、マクロ コマンドの定義と実行をサポートしています。

- **マクロ コマンド定義**：「Bts10200.request(...);」インターフェイスを通じて通常のプロビジョニング要求として定義できるマクロが規定されます。

## ■ マクロ コマンド

- **IDL インターフェイス定義**: 標準の機能プロビジョニング インターフェイスから独立したマクロ コマンドにアクセスするための、新しい IDL メソッドが追加されています。このメソッドはマクロの実行専用です。
- **IDL インターフェイスの CIS 実装**: 認可キーと入力の引数の文字列を受け入れることで、他の CIS インターフェイス オブジェクトの標準的な動作に準拠する実装。マクロ実行への応答を示す文字列も返します。引数および書式は、次のとおりです。
  - **Request**: 要求に対する追加のユーザ指定ルールをページング ファシリティ パラメータとともに含んだ、XML Request ドキュメント。XML Request ドキュメントでは、noun キーが、要求を実行するための目的のマクロ名に設定されていることも必要です。このとき、verb キーは使用されません。デフォルトとしては、*show* が最適のオプションです。これにより、書き込みが許可されている場合に機能の競合が回避されます。
  - **Key**: Bts10200\_Security インターフェイスを通じて提供される実際の認証キーを含んだ、単純な文字列オブジェクト。
  - **Reply**: マクロの実行によって返されたデータを含んだ XML Reply ドキュメント。このドキュメントは、Bts10200 インターフェイスからの Reply XML ドキュメントと同じ形式に従っています。

## 外部インターフェイス

この項では、Cisco BTS 10200 ソフトウェアで提供され、Cisco BTS 10200 ソフトスイッチの CORBA アダプタ (CAD) の外部インターフェイスに反映される拡張機能について詳しく説明します。マクロ コマンドの実行に対しては、ページング パラメータを使用できます。ページング パラメータには、limit パラメータと start-row パラメータがあります。

## CAD インターフェイス

この項では、CAD インターフェイスの新しい IDL メソッドのシグニチャについて説明します。

```
interface Macro {
    //-----
    // Issue a Macro Command XML document
    //-----
    void          execute(in string          request,
                        in string          key,
                        out string         reply)
                    raises(CadExceptions);

}; // end Macro
```

### Request の形式

次の例は、P.1-14 の「マクロの定義」で例として定義した CTXG\_NUMBERS の実行を要求するマクロ XML Request ドキュメントを示しています。追加のページング パラメータが付加されています。このパラメータは、単一の応答で返されるデータの量を制限します。大きな XML ドキュメントは、インターフェイスでエラーになります。現在のデフォルト制限値は、1 要求あたり 500 レコードと定義されています。

```
<Request Noun="CTXG_NUMBERS" Verb="show">
  <Entry Key=" subscriber.ctxg_id" Value="rcdn_grp"/>
  <Entry Key="limit" Value="1"/>
</Request>
```

## Reply の形式

XML Reply ドキュメントの例を次に示します。この応答は、「Request の形式」の項にある例に基づいています。

```
<Reply id="Reply">
  <Status>true</Status>
  <Reason>Success: Entry 1 of 5002 returned.</Reason>
  <Size>1</Size>
  <AbsoluteSize>5002</AbsoluteSize>
  <StartRow>1</StartRow>
  <DataTable>
    <Row id="0">
      <Column id="NDC">601</Column>
      <Column id="EC">227</Column>
      <Column id="DN">1013</Column>
      <Column id="CTXG_ID">rcdn_grp</Column>
    </Row>
  </DataTable>
</Reply>
```

## 操作

この項では、Cisco BTS 10200 マクロ コマンドによって加えられる操作ユーザ インターフェイスへの変更について説明します。マクロ コマンドの実行に対しては、標準的なページング パラメータを使用できません。これらは大きなデータ セットに適用されます。

追加コマンドのためのオペレータ インターフェイスを使用して、Cisco BTS 10200 ソフトスイッチのマクロ コマンドを管理できます。これらのコマンドは CLI インターフェイスから使用できます。また、これらの同じコマンドを CORBA および一括プロビジョニング インターフェイスから使用することもできます。

生成可能なユーザ コマンドを表 1-1 に示します。この表の [R] は、必須であることを表します。これらは、noun (id) とパラメータのさまざまな組み合わせによるユーザ定義コマンドです。

表 1-1 ユーザ定義のマクロ コマンド

noun	verb	オプション	説明
macro	add	id [R]	マクロの識別子。この識別子は 1 ~ 79 文字で指定できます。他のどの ID とも重複しないようにする必要があります。
macro	add	parameters [R]	マクロ実行の結果として返される実データのカンマ区切りリスト。
macro	add	rules [R]	マクロでのデータの表示基準。
macro	change	id [R]	マクロの識別子。
macro	change	parameters	マクロ実行の結果として返される実データのカンマ区切りリスト。
macro	change	rules	マクロでのデータの表示基準。
macro	show	id [R]	表示されるマクロを定義します。
macro	delete	id [R]	マクロの識別子。

## cadexception

返される可能性のある基本的な cadexception は、次のとおりです。サンプルコードの中で与えられている番号は、説明に含まれているテキストを指しています。サンプルコードを使用すると、このテキストが返されます。推奨される対処については、必要に応じて第5章「トラブルシューティング」の「cadexception」の項を参照してください。

### エラー メッセージ No Error

**説明** 0 はエラーではないため、これはプレースホルダです。

**サンプルコード** `public static final int EM_NONE=0;`

### エラー メッセージ CIS Error

**説明** このエラーは、ORB の相互動作や、その他の実行時例外に関係している可能性のある内部処理エラーに対して使用されます。

**サンプルコード** `public static final int EM_ERROR=1;`

### エラー メッセージ CIS No Data

**説明** このエラーは、`show` コマンドからデータが何も返されなかったことを示します。これは、実際にはエラーではない可能性があります。例外がスローされるのは、NULL オブジェクトを返すよりも明確なためです。

**サンプルコード** `public static final int EM_NODATA=2;`

### エラー メッセージ User Security Error

**説明** ユーザセキュリティに障害が発見されました。これは、無効なユーザ名またはパスワードによるログインが原因になった可能性があります。また、ユーザ ID の検証に失敗したセキュリティ システムの内部エラーが原因で発生した可能性もあります。

**サンプルコード** `public static final int EM_USERSEC=5;`

### エラー メッセージ Permission Error

**説明** コマンドの実行が試みられましたが、そのコマンドの認可テストに合格しませんでした。ユーザは、このコマンドの実行権限を持っていません。

**サンプルコード** `public static final int EM_PERMISSION=6;`

### エラー メッセージ Error Message: Block Error

**説明** スイッチに対するすべてのプロビジョニングがブロックされました。コマンドは完全にウェルフォームドであり、接続はまだ有効です。これは、単に BTS 10200 がメンテナンスモードになっていることを示します。

**サンプルコード** `public static final int EM_BLOCK=7;`

**エラー メッセージ** Linkage Error

**説明** リンケージに失敗しました。

**サンプル コード** `public static final int EM_LINKAGE=10;`

**エラー メッセージ** Exception In Initializer Error

**説明** このメソッドによって実行された初期化が失敗しました。

**サンプル コード** `public static final int EM_INIT=11;`

**エラー メッセージ** Class Not Found Exception

**説明** クラスが見つかりません。

**サンプル コード** `public static final int EM_NOTFOUND=12;`

**エラー メッセージ** Illegal Access Exception

**説明** クラスまたはイニシャライザがアクセス不能です。

**サンプル コード** `public static final int EM_ACCESS=13;`

**エラー メッセージ** Instantiation Exception

**説明** クラスが、抽象クラス、インターフェイス、配列クラス、基本タイプ、または void を表しています。または、インスタンス化が他の何らかの理由で失敗しています。

**サンプル コード** `public static final int EM_INSTANCE=14;`

**エラー メッセージ** Security Exception

**説明** 新しいインスタンスを作成する権限がありません。

**サンプル コード** `public static final int EM_SECURITY=15;`

**エラー メッセージ** Invalid Request Exception

**説明** 要求が有効でないか、初期化できません。

**サンプル コード** `public static final int EM_REQUEST=16;`

**エラー メッセージ** Invalid Noun Exception

**説明** コマンドの noun が見つからないか無効です。

**サンプル コード** `public static final int EM_NOUN=17;`

**エラー メッセージ** Invalid Verb Exception

**説明** コマンドの verb が見つからないか無効です。

**サンプル コード** `public static final int EM_VERB=18;`

**エラー メッセージ** SQL Exception

**説明** 制約に違反している場合、この他にテーブル競合がある場合、リソースに問題がある場合、またはこの他の何らかの Oracle 関連の原因がある場合は、データベースにアクセスできません。

**サンプル コード** `public static final int EM_DATABASE=19;`

**エラー メッセージ** Invalid Value Exception

**説明** パラメータ値が、テキスト長、ピックリストなどの有効な範囲を超えたか、他の何らかの制約事項に違反しました。

**サンプル コード** `public static final int EM_VALUE=20;`

**エラー メッセージ** Invalid Key Exception

**説明** 一部のデータ値の記述に、無効なキーまたはトークンが使用されました。

**サンプル コード** `public static final int EM_KEY=21;`

**エラー メッセージ** Missing Parameter Exception

**説明** 1 つまたはそれ以上の必須パラメータが、コマンド パラメータ データに含まれていませんでした。

**サンプル コード** `public static final int EM_PARAM=22;`



## Extensible Markup Language の処理

この章では、CORBA ( Common Object Request Broker Architecture ) アダプタでの XML ( Extensible Markup Language ) の処理について説明します。

### XML およびコンポーネント

CORBA アダプタの主要コンポーネントには、XML の他に CORBA Interface Servant ( CIS; CORBA インターフェイス サーバント ) Java プログラムがあります。また、Managed Object ( MO; 管理対象オブジェクト ) Java パッケージの関連コンポーネントにも依存しています。必須のパッケージは次のとおりです。

- Apache XML
- Xerces ( パーサー )
- ECS ( XML ドキュメント ビルディング ツールキット )

XML および CIS Java パッケージは、Cisco BTS 10200 ソフトスイッチの中心となる機能コンポーネントです。このソフトスイッチに適用される、さらに規模の大きいオブジェクト モデルはありません。これよりも大きいオブジェクト モデルは、シスコシステムズ標準モデルが作成されるまでは存在しません。このモデルは、さまざまなアプリケーションのポケット テレフォニーに应用することができます。

### CORBA インターフェイス サーバントにおける XML

この項では、CIS での XML の使用方法について説明します。この項で使用する用語は、XML 仕様で使用されている用語に準拠しています。これは、要素、サブ要素、アトリビュートなどの用語の使用で混乱が生じないようにするためです。

### CIS の機能

XML ドキュメントの構造をクライアント側で検証するために、スキーマが提供されています。これらのスキーマは、次の項目をカバーしています。

- *ManagedObject*
- *Request*
- *Reply*

*ManagedObject* のスキーマは、次の形式に従っています。

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

  <xs:element name="ManagedObject">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="MOAttribute" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="Verb" type="xs:string" use="required"/>
      <xs:attribute name="id" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>

  <xs:element name="MOAttribute">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Required"/>
        <xs:element ref="Type"/>
        <xs:element ref="Default"/>
        <xs:element ref="Width"/>
        <xs:element ref="HelpText"/>
        <xs:element ref="Label"/>
        <xs:element ref="Parser" minOccurs="0"/>
        <xs:element ref="Permitted" minOccurs="0"/>
        <xs:element ref="Fk" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="id" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>

  <xs:element name="Required" type="xs:boolean"/>

  <xs:element name="Type">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="single"/>
        <xs:enumeration value="text"/>
        <xs:enumeration value="multi"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>

  <xs:element name="Default" type="xs:string"/>

  <xs:element name="HelpText" type="xs:string"/>

  <xs:element name="Label" type="xs:string"/>

  <xs:element name="Noun" type="xs:string"/>

  <xs:element name="Param" type="xs:string"/>

  <xs:element name="Parser">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="JavaScript"/>
        <xs:element ref="RegExp"/>
      </xs:sequence>
      <xs:attribute name="id" use="required" type="xs:string"/>
    </xs:complexType>
  </xs:element>

  <xs:element name="JavaScript" type="xs:string"/>
  <xs:element name="RegExp" type="xs:string"/>

  <xs:element name="Permitted" type="xs:string"/>

  <xs:element name="Width" type="xs:int" />

```



```
<xs:element name="Fk">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Noun"/>
      <xs:element ref="Param"/>
      <xs:element ref="Fk" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>

</xs:schema>
```

*Request* のスキーマは、次の形式に従っています。

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <xs:element name="Request">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Entry" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="Verb" type="xs:string" use="required"/>
      <xs:attribute name="Noun" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="Entry">
    <xs:complexType>
      <xs:attribute name="Key" type="xs:string" use="required"/>
      <xs:attribute name="Value" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

*Reply* のスキーマは、次の形式に従っています。

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

  <xs:element name="Reply">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Status"/>
        <xs:element ref="Reason"/>
        <xs:element ref="Size"/>
        <xs:element ref="AbsoluteSize"/>
        <xs:element ref="StartRow"/>
        <xs:element ref="DataTable"/>
      </xs:sequence>
      <xs:attribute name="id" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>

  <xs:element name="Status" type="xs:boolean"/>

  <xs:element name="Reason" type="xs:string"/>

  <xs:element name="Size" type="xs:integer"/>

  <xs:element name="AbsoluteSize" type="xs:integer"/>

  <xs:element name="StartRow" type="xs:integer"/>

  <xs:element name="DataTable">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Row" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="Row">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Column" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="id" use="required" type="xs:integer"/>
    </xs:complexType>
  </xs:element>

  <xs:element name="Column" >
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="id" use="required"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

Interface Definition Language (IDL; インターフェイス定義言語) を使用すると、各 noun/verb の組み合わせの XML 記述ドキュメントにアクセスできます。たとえば、**add subscriber** コマンドは、このコマンドの要素とアトリビュートを定義するための対応 XML ドキュメントを生成します。IDL は、ウェルフォームドであるが未検証の XML ドキュメントに基づいたコマンド処理を可能にします。

IDL を使用すると、サポートされている Media Gateway (MGW; メディア ゲートウェイ) デバイスにコマンド アクセスできます。このコマンド文字列は、スキーマ内に定義された XML アクセス形式には準拠していません。CIS は、MGW の内部コマンド構造に対してネイティブである MGW コマンド文字列をサポートしています。

Cisco BTS 10200 ソフトスイッチ内で発生する XML ドキュメントは、すべて動的に生成されます。これは、すべてのコマンド記述ドキュメントを含みます。

## ManagedObject

*ManagedObject* は、*MOAttribute* という 1 つの要素を持っています。また、*ManagedObject* は、*ManagedObject* の *id* と *verb* という 2 つのアトリビュートを持っています。*id* は、何らかの操作の対象となるオブジェクトを表します。*verb* は実行される操作を示します。たとえば、*subscriber* や *termination* は有効な *id* です。これは必須のアトリビュートです。

このスキーマのさまざまな部分およびその値について、次のリストで説明します。

- **id** : このアトリビュートは、何らかの操作の対象となるオブジェクトを表します。
- **Verb** : このアトリビュートは、所定の *ManagedObject* に対して実行する操作を定義します。これは必須のアトリビュートであり、文字型データで構成されます。
- **MOAttribute** : *ManagedObject* は、この要素を 0 個、1 個、または複数保持することができます。この要素は *id* という 1 つのアトリビュートを持っています。この文字型データは、要素のラベルとして機能します。これらの要素の順序が、特定の動作を意味することはありません。任意の順序で記述できます。
- **Required** : このサブ要素には、*true* と *false* の 2 つの値のいずれかが定義されます。
- **Type** : このサブ要素は、*MOAttribute* が保持するのが単一値であるか、複数値であるか、またはテキストであるかを定義します。*multiple* オプションまたは *single* オプションは、選択肢のリストが *Permitted* 要素で提供されることを意味します。
- **Default** : このサブ要素は情報提供用です。*MOAttribute* のデフォルト値を示します。
- **Width** : このサブ要素は、データの合計フィールド幅を示します。たとえば、*MOAttribute* が説明である場合、これは説明文の長さを示します。
- **HelpText** : このサブ要素は、*MOAttribute* の性質を示す簡単なテキストを提供します。
- **Permitted** : このサブ要素は、*MOAttribute* の許容値または許容範囲を指定します。
- **Parser** : このサブ要素は、どの種類の検証が必要であるかを指定します。このサブ要素にはアトリビュートが 1 つだけあります。このアトリビュートは、文字型データで構成される *id* フィールドです。サブ要素を次に示します。
  - **JavaScript** : このサブ要素は、検証または正規表現一致検索を実行できる JavaScript を示します。
  - **RegExp** : このサブ要素は、正規表現を文字型データ形式で定義します。

## Request

*Request* スキーマは、1 つの要素、0 個 ~ 複数個の *Entry* 要素、および 2 つの属性とその値で構成されます。

このスキーマのさまざまな部分およびその値について、次のリストで説明します。

- **Noun** : この属性は、何らかの操作の対象となる項目を定義します。これは文字型データとして表現されます。
- **Verb** : この属性は、「Noun」属性に対して実行する操作を定義します。これは文字型データとして表現されます。

*Entry* 要素は、空値になることができます。また、2 つの属性を含むことができます。これらの属性は、次のように定義されます。

- **Key** : これは、*ManagedObject* の *MOAttribute* から導出された *id* の値です。これは文字型データとして表現されます。
- **Value** : これは、上で指定されたキーに割り当てるためのクライアント導出値です。*Value* は文字型データとして表現されます。また、このキー / 値ペアの導出元となった *MOAttribute* のサブ要素に一致している必要があります。

## Reply

*Reply* スキーマは、*Request* への応答として生成される戻りデータの構造を定義します。*Reply* は 3 つの要素を保持し、属性はありません。これらの要素は、次のように定義されます。

- **Status** : *Reply* は 1 つの *Status* 要素を保持します。この要素の許容値は 2 つあります。この要素に適用されるのは、true または false のいずれかです。
- **Reason** : *Reason* 要素は文字型データを保持します。この要素は、コマンド処理でのエラーの原因を説明するものです。または、成功通知を返します。
- **DataTable** : この要素は、下に示す 1 つの属性と 1 つのサブ要素を持っています。この要素は、要求の実行結果データのコンテナとして使用されます。各 *Reply* が *DataTable* 要素を保持できます。
  - **Row** : このサブ要素は、1 つのデータ項目全体を定義します。*DataTable* は、1 つまたはそれ以上の *Row* サブ要素を保持できます。*Row* は 1 つの属性を持っています。この文字型データは行 ID を定義します。ID は、返される行の数に基づいた連続値です。*id* 属性は必須です。
  - **Col** : 各 *Row* は、*Col* と呼ばれるサブ要素を保持します。このサブ要素は 1 つの属性を持っています。この要素の値は、文字型データとして表現されます。*Col* の属性は *id* です。この属性は文字型値として表現されます。これは、*MOAttribute* で使用されるものと同じ *id* 値です。これは必須の属性です。

## CORBA インターフェイス サーバント アダプタ実装

CIS は、外部インターフェイスを指定するアダプタ実装です。この項では、CIS プログラムとクライアント側プログラム間でのドキュメント交換の構造について、詳細を示します。外部インターフェイスで必ず問題になる点の1つは、ここで説明したすべてのドキュメントがウェルフォームドであるものの、検証されていないことです。つまり、スキーマが XML ドキュメントの一部として埋め込まれていません。スキーマを埋め込むことで、パーサー パッケージを使用してドキュメントの構造を検証できます。しかし、他のお客様がスキーマを必要とする場合、XML スキーマに対する変更を加えにくくなります。クライアント側が、このドキュメントに含まれているスキーマを使用して検証を実行することは可能です。

### Cisco BTS 10200 ソフトスイッチの IDL コード

この項では、Cisco BTS 10200 ソフトスイッチの CORBA アダプタ (CAD) インターフェイスのシステム IDL ファイルについて説明します。この IDL は、Cisco BTS 10200 ソフトスイッチ Release 4.x に適用されます。

```
// Copyright (c) 2002, 2006 by Cisco Systems, Inc.
//=====
//
// Name:          bts10200.idl
// Author:       A. J. Blanchard
// Description:
// This is the IDL for the entire provisioning infrastructure of the
// BTS 10200. The text strings are all XML well-formed documents. The
// current procedure is to maintain separate schema(s). This allows later
// migration to schemas and away from schema for document validation.
//
// All commands are expressed as XML documents. The template document for
// each NOUN/VERB pair is accessible from the a separate method. This XML
// interface is table oriented and follows the same nomenclature and syntax
// as the other BTS 10200 adapter interfaces.
//
//
//=====

#ifndef bts10200_idl
#define bts10200_idl

//-----
// Set up modules to match java package tree for the OAM&P
//-----

module cad {

    typePrefix cad "oam.sswitch.com";

    //-----
    // Exceptions
    //-----
    exception CadExceptions {
        long    error_code;
        string  error_string;
    };

    interface Bts10200_Security {

        //-----
        // Create a session key.
        //-----
        void    login(in string          name,
                     in string          password,
                     out string         key)
                raises(CadExceptions);
    };
};
#endif
```

```

//-----
// Destroy a session key.
//-----
void      logout(in string          key)
            raises(CadExceptions);

}; // end Bts10200_Security

interface Bts10200 {

//-----
// Fetch a command (noun/verb) XML document
//-----
void      getCommandDoc(in string          noun,
                        in string          verb,
                        in string          key,
                        out string         xml_doc)
            raises(CadExceptions);

//-----
// Fetch an Extended command (noun/verb) XML document - with foreign keys
//-----
void      getExtCommandDoc(in string       noun,
                            in string       verb,
                            in string       key,
                            out string      xml_doc)
            raises(CadExceptions);

//-----
// Issue a command XML document (add, change, delete, show)
//-----
void      request(in string          xml_request,
                  in string          key,
                  out string         xml_reply)
            raises(CadExceptions);

}; // end Bts10200

interface Macro {

//-----
// Process a Macro Command XML document (add, change, delete, show)
//-----
void      execute(in string          request,
                  in string          key,
                  out string         reply)
            raises(CadExceptions);

}; // end Macro

}; // end cad
#endif // end bts10200_idl

```



## CORBA の Secure Sockets Layer サポート

この章では、CORBA( Common Object Request Broker Architecture )における Secure Sockets Layer( SSL ) サポートについて説明します。

このシステムは、CORBA アダプタの CORBA Interface Servant( CIS; CORBA インターフェイス サーバント)で SSL モジュールを使用することにより、セキュアな CORBA トランスポートを提供します。Object Management Group (OMG; オブジェクト管理グループ)機構は、Security Attribute Service ( SAS ) プロトコルが定義された Common Secure Interoperability Specification Version 2 ( CSIv2 ) を定義しています。SAS によって、相互運用可能な認証、委任、および特権を導入できます。

SAS プロトコルは、自身のプロトコル要素を、接続ベースのトランスポート上でやり取りされる General Inter-ORB Protocol ( GIOP ) 要求メッセージおよび応答メッセージのサービス コンテキストで交換します。このプロトコルの使用対象となるのは、SSL/Transport Layer Security( TLS )や Internet Inter-ORB Protocol ( IIOOP ) over SSL ( SSLIOP ) を使用して得られるトランスポート層セキュリティによって、メッセージ保護 (つまり、整合性や機密性)、およびサーバからクライアントに対する認証を実施している環境です。このプロトコルは、クライアント認証、委任、および特権に関する機能を提供します。これらの機能を適用すると、基礎となっているトランスポートの対応する不十分な箇所を補完できます。SAS プロトコルは、セキュア トランスポートが統合される部分の上位で高レベル プロトコルとして機能することにより、相互運用を支援します。CIS の SAS 実装は、次の機能を提供します。

- 一般的なトランスポート層セキュリティ メカニズムの使用に基づいた (たとえば、SSL/TLS によって提供される) セキュアな相互運用性。
- GIOP の入力要求と出力要求の引数を保護するために、トランスポート層で必要に応じて提供されるメッセージ保護。
- トランスポート層で必要に応じて提供される、クライアントによるターゲットの認証。ターゲットが意図したターゲットであることを確認するために、ターゲットの身元を確認します。
- 予備的な要求を発行しなくても、意図したターゲットとの機密アソシエーションをクライアントが確立できる、トランスポート層セキュリティ。
- トランスポート層セキュリティ メカニズムを使用して認証を受けることができないクライアントのサポート。SAS プロトコルは、トランスポート層よりも上位でのクライアント認証を提供します。
- GIOP サービス コンテキストを使用したセキュリティ コンテキストの形成をサポートするために、SAS プロトコルでセキュリティ コンテキストの確立に必要なメッセージは、方向ごとに多くとも1つです。
- 単一の要求 / 応答ペアが持続している間だけ存在するセキュリティ コンテキストのサポート。
- 複数の要求 / 応答ペアで再利用できるセキュリティ コンテキストのサポート。

この実装は、OpenORB CORBA ディストリビューションで提供されるモジュール拡張を通じて提供されます。

**注意**

---

セキュリティに関する予期しない問題を回避するため、CORBA アダプタのデフォルト転送プロトコルには SSL を使用してください。

---

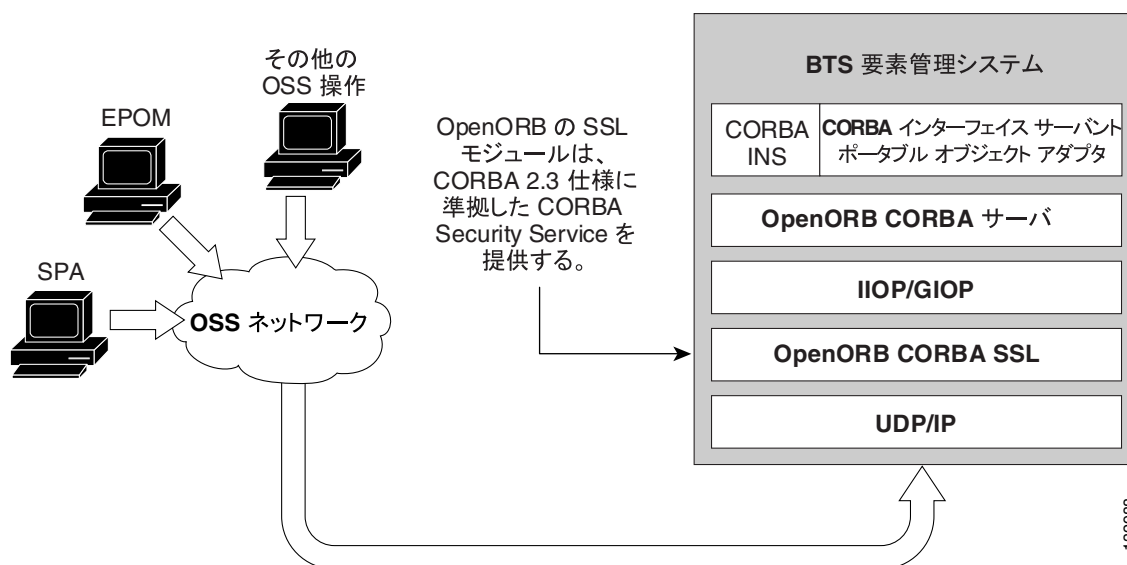


## システム セキュリティ拡張のシステム コンテキスト

この項では、システム セキュリティ拡張のシステム コンテキストについて説明します。システム セキュリティ拡張は、CORBA の SSL モジュール、およびセキュリティ証明書の交換の組み合わせで構成されます。

図 3-1 に、システム セキュリティ拡張の影響を受ける関連システム コンポーネントの概要を示します。この概要では、Cisco BTS 10200 ソフトスイッチで影響を受けるシステム コンポーネントを示しています。例外は、ベース Solaris OS プラットフォームです。

図 3-1 CORBA の Secure Sockets Layer 実装



### 依存関係

この項では、セキュリティ拡張機能のアプリケーション コンポーネントにおける依存関係を示します。依存関係の大部分は、外部コンポーネントに基づいています。

### 縮小 Solaris イメージ

このアプリケーション コンポーネントには、特定の依存関係は存在しません。ただし、依存関係は Cisco BTS 10200 ソフトスイッチのセキュリティ ニーズに基づいています。

### Java の SSL 実装

CIS アプリケーション プログラムは、CORBA のセキュリティ サポートを OpenORB の SSL コンポーネント モジュール実装に依存しています。この OpenORB モジュールは、Sun Microsystems が提供している Java の JSSE (Java Secure Sockets Extension) 実装を必要とします。

## 証明書およびキー パスワード

この項では、証明書、および SSL CORBA のキー パスワードについて説明します。SSL CORBA の主な機能は、暗号化されたトランスポートです。この実装では、公開鍵を使用した自己署名証明書だけを使用し、すべてのユーザが同じキー パスワードを使用します。

この項では、SSL CORBA のキー ストア、トラストストア、および証明書が配置されるディレクトリ構造、およびこれらのファイルが従う必要のある命名規則についても説明します。Cisco BTS 10200 ソフトスイッチでは、証明書とキー ストアは Cisco BTS 10200 CIS パッケージ内に構築されます。これらは次の場所にあります。

```
/opt/BTScis/cert
```

証明書とキー ストアは、次の場所にある CORBA SDK パッケージ BTSxsdk でも使用できます。

```
/opt/BTSxsdk/cert
```

トラストストアとキー ストアは次のように命名する必要があります。

```
bts10200_ks bts10200_ts
```

使用する必要のある必須キー パスワードは、Chillan です（大文字と小文字が区別されます）。

cis-install.sh を使用した CORBA のインストールでは、*SSLIOP enabled* がデフォルトです。クライアントシステムは、前述のキー パスワード、キー ストア、および証明書を使用しない限りアクセスできません。自動的な冗長化は行われないため、キー ストアとトラストストアは、両方の Element Management System (EMS; 要素管理システム) 上にある必要があります。



(注)

以降で示すコマンドは、キー ストアとトラストストアを Cisco BTS 10200 ソフトスイッチ上とクライアント側に構築します。キー ストアとトラストストアは、デフォルトで Cisco BTS 10200 ソフトスイッチと BTSxsdk に構築され展開されますが、手動で再構築または置換することができます。ここでは、手動による操作はお勧めしません。

パスワード : Chillan

**ステップ 1** キーを生成します（有効期間は 8 年）。

```
keytool -genkey -alias bts10200 -keyalg RSA -validity 2840 -keystore bts10200_ks
```

**ステップ 2** 証明書をエクスポートします。

```
keytool -export -alias bts10200 -keystore bts10200_ks -rfc -file bts10200.cer
```

**ステップ 3** 証明書をトラストストアにインポートします。

```
keytool -import -alias bts10200 -file bts10200.cer -keystore bts10200_ts
```



## プロキシ

プロキシ ソフトウェアは、CORBA Software Development Kit ( SDK; ソフトウェア開発キット ) パッケージの拡張機能です。この拡張機能によって、いくつかの新機能をクライアントアプリケーションに簡単に統合できます。また、プロキシ ソフトウェアは、改訂された新しい SSL、または Cisco BTS 10200 ソフトスイッチへのセキュア CORBA インターフェイスでのみ動作するように設計されています。

使用可能になる機能を次に示します。

- 冗長な管理インターフェイスを、単一の Cisco BTS 10200 ソフトスイッチの Element Management System ( EMS; 要素管理システム ) ノード上に抽象化する。各 EMS は、プロビジョニング用の 2 つの物理ネットワーク インターフェイス カード ( NIC ) を備えています。CORBA/SSL パッケージは、オブジェクトインスタンスを個別に各インターフェイスにバインドすることによって、これらの両方のインターフェイスを利用します。プロキシは、これらのインターフェイスをそれぞれプローブして、どちらのインターフェイスが機能しているか、あるいは両方とも機能しているかどうかを調べます。要求された Cisco BTS 10200 ソフトスイッチ オブジェクトを返した最初のインターフェイスがプライマリ インターフェイスとして宣言され、トランザクションにはこのインターフェイスだけが使用されます。CORBA プロトコル IIOP/SSLIIOP は、単一インターフェイス上での一貫した対話を必要とするため、これらのインターフェイス上の所定の接続で「ロードシェアリング」を行うことはできません。
- クライアント アプリケーションから見た EMS の冗長性を抽象化する。クライアント アプリケーションの多くは、Cisco BTS 10200 ソフトスイッチのデュプレックス性を完全には考慮していません。結果として、EMS で手動のスイッチオーバーやフェールオーバーが必要になった場合、これらのアプリケーションは何らかの複雑な再初期化を実行する必要があります。これは時間のかかる処理であり、エラーも発生しやすくなります。プロキシを使用すると、インターフェイスから返されるオブジェクトに問題がある場合、プロキシへの新しいアクセスがあったときに適切な ( 新しいアクティブな ) EMS オブジェクトが返されるような抽象化が可能になります。Cisco BTS 10200 ソフトスイッチ Release 4.5.1 の CORBA は、フェールオーバーが発生した場合、自身のオブジェクトを NameService からアンバインドし、現在のログイン セッションを終了します。結果として、プロキシは NameService を継続的にプローブして、アクティブな EMS ノードとスタンバイ EMS ノードに対する変更がないかどうかを確認し、フェールオーバーを感知します。アプリケーション クライアントは、これを CORBA からの PERMISSION\_DENIED エラーまたは関連する CORBA COMM エラーを使用して検出します。BTS 10200 のこのリリースは Notification Service を含んでいないため、非同期通知は実行されません。

- 複数の Cisco BTS 10200 ソフトスイッチ インスタンス、つまり「複合体」を抽象化する機能。それぞれの複合体は、どの Cisco BTS 10200 ソフトスイッチが参照されているかを特定するための、CLLI コード、sensor-id、元の CORBA site-id などの一意の識別子を持っている必要があります。この ID は、プロキシ ロジックの内部で、特定の Cisco BTS 10200 ソフトスイッチのオブジェクトと接続情報を発見するための一意のキーまたはロケータとして使用されます。これにより、複数の Cisco BTS 10200 ソフトスイッチ（現時点では数百まで）へのアクセスを単一のプロキシ インスタンスで管理できます。この抽象化によって、同じネットワークで他のプロキシ インスタンスを使用できなくなることはありません。他のプロキシ インスタンスが、同じ一連の Cisco BTS 10200 ソフトスイッチ複合体と対話することもできます。相互排他性に関する要件は、プロキシにはありません。
- EMS 上の両方の管理インターフェイスを完全に利用して、冗長性を実現します。バーチャル IP が導入されていない場合、NameService および CORBA アプリケーションは各 EMS 上の両方の管理インターフェイスを利用します。



# トラブルシューティング

この章では、基本的な cadexception、デバッグ、およびネットワーク設定の手順について説明します。

## cadexception

返される可能性のある基本的な cadexception は、次のとおりです。サンプルコードの中で与えられている番号は、説明に含まれているテキストを指しています。サンプルコードを使用すると、このテキストが返されます。

### エラー メッセージ No Error

**説明** 0 はエラーではないため、これはプレースホルダです。

**サンプルコード** `public static final int EM_NONE=0;`

**推奨処置** 必要な作業はありません。エラーではありません。

### エラー メッセージ CIS Error

**説明** このエラーは、ORB の相互動作や、その他の実行時例外に関係している可能性のある内部処理エラーに対して使用されます。

**サンプルコード** `public static final int EM_ERROR=1;`

**推奨処置** コマンドを再試行します。問題が引き続き発生する場合は、「[CORBA ネットワーク設定の変更](#)」の項の説明に従って、NameService と CORBA アダプタを再起動します。

### エラー メッセージ CIS No Data

**説明** show コマンドからデータが何も返されませんでした。これは、実際にはエラーではない可能性があります。例外がスローされるのは、NULL オブジェクトを返すよりも明確なためです。

**サンプルコード** `public static final int EM_NODATA=2;`

**推奨処置** 同じコマンドを CLI を使用して実行し、該当する noun (id) のデータが存在することを確認します。適切な構文については、『Cisco BTS 10200 Softswitch Command Line Interface Reference Guide』を参照してください。

**エラー メッセージ** User Security Error

**説明** ユーザ セキュリティに障害が発見されました。これは、無効なユーザ名またはパスワードによるログインが原因になった可能性があります。また、ユーザ ID の検証に失敗したセキュリティ システムの内部エラーが原因で発生した可能性もあります。

**サンプル コード** `public static final int EM_USERSEC=5;`

**推奨処置** ユーザ名とパスワードが正しいこと、および Cisco BTS 10200 ソフトスイッチへのアクセスに必要な権限を持っていることを確認します。同じユーザ名とパスワードで CLI にログインして、二重に確認します。

**エラー メッセージ** Permission Error

**説明** コマンドの実行が試みられましたが、そのコマンドの認可テストに合格しませんでした。ユーザは、このコマンドの実行権限を持っていません。

**サンプル コード** `public static final int EM_PERMISSION=6;`

**推奨処置** 要求の実行に必要なコマンド特権をユーザが持っていることを確認します。同じユーザとして、同じコマンドを CLI を使用して実行し、二重に確認します。コマンド特権については、『Cisco BTS 10200 Softswitch Command Line Interface Reference Guide』を参照してください。

**エラー メッセージ** Error Message: Block Error

**説明** Cisco BTS 10200 ソフトスイッチがメンテナンス モードになっているため、スイッチに関するすべてのプロビジョニングがブロックされました。コマンドは完全にウェルフォームドであり、接続はまだ有効です。

**サンプル コード** `public static final int EM_BLOCK=7;`

**推奨処置** 後でもう一度実行します。

**エラー メッセージ** Linkage Error

**説明** リンケージに失敗しました。

**サンプル コード** `public static final int EM_LINKAGE=10;`

**推奨処置** これはインストール エラーです。Cisco TAC にお問い合わせください。

**エラー メッセージ** Exception In Initializer Error

**説明** 初期化メソッドが失敗しました。

**サンプル コード** `public static final int EM_INIT=11;`

**推奨処置** コマンドをもう一度実行します。問題が引き続き発生する場合は、「CORBA ネットワーク設定の変更」の項の説明に従って、NameService と CORBA アダプタを再起動します。

**エラー メッセージ** Class Not Found Exception

**説明** クラスが見つかりません。

**サンプルコード** `public static final int EM_NOTFOUND=12;`

**推奨処置** CLASSPATH が設定されていること、およびすべての jar ファイルが CLASSPATH で指定された場所に存在することを確認します。CLASSPATH は、EMS 上の /opt/BTScis/bin/cis3 および /opt/ems/bin/Runtime.sh に設定されています。

**エラー メッセージ** Illegal Access Exception

**説明** クラスまたはイニシャライザがアクセス不能です。

**サンプルコード** `public static final int EM_ACCESS=13;`

**エラー メッセージ** Instantiation Exception

**説明** このクラスが、抽象クラス、インターフェイス、配列クラス、基本タイプ、または void を表す場合、あるいはインスタンス化が他の何らかの理由で失敗した場合にスローされます。

**サンプルコード** `public static final int EM_INSTANCE=14;`

**推奨処置** CLASSPATH が設定されていること、およびすべての jar ファイルが CLASSPATH で指定された場所に存在することを確認します。CLASSPATH は、EMS 上の /opt/BTScis/bin/cis3 および /opt/ems/bin/Runtime.sh に設定されています。

**エラー メッセージ** Security Exception

**説明** 新しいインスタンスを作成する権限がありません。

**サンプルコード** `public static final int EM_SECURITY=15;`

**推奨処置** CORBA アダプタがルートとして動作していることを確認します。EMS 上で `ps -ef | grep cis3 and not the user` コマンドを実行して確認します。

**エラー メッセージ** Invalid Request Exception

**説明** 要求が有効でないか、初期化できません。

**サンプルコード** `public static final int EM_REQUEST=16;`

**推奨処置** XML 要求の構文が、Cisco BTS 10200 ソフトスイッチの標準に準拠した適切なものであることを確認します。

**エラー メッセージ** Invalid Noun Exception

**説明** コマンドの noun が見つからないか無効です。

**サンプルコード** `public static final int EM_NOUN=17;`

**推奨処置** 要求の noun (id) が有効なものであること、および Cisco BTS 10200 ソフトスイッチの現行ソフトウェアリリースに対して適切であることを確認します。EMS 上で同じコマンドを CLI を使用して実行し、二重に確認します。構文については、『Cisco BTS 10200 Softswitch Command Line Interface Reference Guide』を参照してください。

**エラー メッセージ** Invalid Verb Exception

**説明** コマンドの verb が見つからないか無効です。

**サンプルコード** `public static final int EM_VERB=18;`

**推奨処置** 要求の verb が有効なものであること、および Cisco BTS 10200 ソフトスイッチの現行ソフトウェア リリースに対して適切であることを確認します。EMS 上で同じコマンドを CLI を使用して実行し、二重に確認します。構文については、『Cisco BTS 10200 Softswitch Command Line Interface Reference Guide』を参照してください。

**エラー メッセージ** SQL Exception

**説明** データベースにアクセスできない場合、制約に違反している場合、この他にテーブル競合がある場合、リソースに問題がある場合、またはこの他の何らかの Oracle 関連の原因がある場合にスローされます。

**サンプルコード** `public static final int EM_DATABASE=19;`

**推奨処置** EMS 上でリレーショナルデータベースが動作していることを確認します。次のコマンドを使用して確認します。

1) `ps -ef | grep mysqld` を実行します。動作中のプロセスが存在することを確認します。  
 2) ルートとして `nodestat` を実行し、Oracle データベースが動作していることを確認します。  
 どちらも動作していない場合は、『Cisco BTS 10200 Operations and Maintenance Manual』を参照して、これらのプロセスを再起動します。

**エラー メッセージ** Invalid Value Exception

**説明** パラメータ値が、テキスト長、ピックリストなどの範囲を超えたか、他の何らかの制約事項に違反しました。

**サンプルコード** `public static final int EM_VALUE=20;`

**推奨処置** XML 要求に含まれているすべての値が、これらの値に関する制約事項に従っていることを確認します。同じコマンドを同じ値で CLI を使用して実行し、二重に確認します。

**エラー メッセージ** Invalid Key Exception

**説明** 一部のデータ値の記述に、無効なキーまたはトークンが使用されました。

**サンプルコード** `public static final int EM_KEY=21;`

**推奨処置** XML 要求に含まれているすべてのキーおよび値が、有効なものであることを確認します。同じコマンドを同じ値で CLI を使用して実行し、二重に確認します。

**エラー メッセージ** Missing Parameter Exception

**説明** 1 つまたはそれ以上の必須パラメータが、コマンド パラメータ データに含まれていませんでした。

**サンプルコード** `public static final int EM_PARAM=22;`

**推奨処置** 必須のキーと値が、すべて有効なものであることを確認します。同じコマンドを同じ値で CLI を使用して実行し、二重に確認します。



## CORBA ネットワーク設定の変更

以前のリリースでは、オペレータが IP アドレスまたはホスト名のいずれかを入力するように求められました。この Release 4.5.1 では、Virtual IP( VIP; バーチャル IP)が設定されている場合、CORBA インストールは自動的に VIP を選択します。VIP が設定されていない場合は、最初の管理 IP アドレスが選択されます。このように運用するには、クライアントマシンにホスト名を追加する必要があります。Cisco BTS 10200 ソフトスイッチでの名前使用は、次の手順で変更することもできます。通常は、IP アドレスに変更して、クライアント側ネットワークで名前解決の問題が発生しないようにすることをお勧めします。



(注)

ホスト名または IP アドレスは、クライアント側に送信される Interoperable Object Reference ( IOR )の一部として使用されます。名前または IP アドレスを解決してクライアントにルーティングする必要があるのは、このためです。

クライアント側ネットワークでの使用に最適なホスト名または IP アドレスを特定し、次の手順を実行して、CORBA ネットワーク設定を変更します。

**ステップ 1** /etc/inittab ファイルを編集し、最後の 2 エントリの名前を変更して、目的の設定に合せます。次の例を参考にしてください。

```
ns:3:respawn:/sbin/ins3 <VIP or EMS Management IP or hostname>
cs:3:respawn:/sbin/cis3 <VIP or EMS Management IP or hostname> <EMS Management IP #1>
< EMS Management IP #2>
```

**ステップ 2** ディレクトリ /opt/BTSoorb/config/.(ピリオドはドメインディレクトリを完全修飾)で、setConfig.sh スクリプトを実行します。XML コンフィギュレーションファイルは、OpenORB アーキテクチャにおけるバンドルされたリソースファイルです。このコマンドは、変更された OpenORB.xml ファイルを使用して JAR ファイルを更新します。

**ステップ 3** NameService ( ins3 ) と CORBA サーバントプログラムを、挙げられている順序で強制終了して再起動します。

```
pkill ins3
pkill cis3
```

NameService ( ins3 ) と CORBA サーバント ( cis3 ) プログラムは、自動的に再起動します。

**ステップ 4** ptree コマンドと kill コマンドを使用して、Java の子プロセスを強制終了します。孤立した Java プログラムは、予期しない動作の原因になる場合があります。

## CORBA が Cisco BTS 10200 ソフトスイッチに接続できない

CORBA が Cisco BTS 10200 ソフトスイッチに接続できない場合は、Element Management System (EMS; 要素管理システム) 内に設定されている内容と照合するための、クライアント側のホスト名解決が必要です。

次の手順を実行して、問題を解決します。

---

**ステップ1** エントリを呼び出したときに /etc/inittab ファイルで使用されている名前を特定します。

```
ns:3:respawn:/sbin/ins3 <VIP or EMS Management IP or hostname>
cs:3:respawn:/sbin/cis3 <VIP or EMS Management IP or hostname> <EMS Management IP #1>
< EMS Management IP #2>
```

**ステップ2** 次の引数を追加して debug-trace コマンドを起動し、クライアント側のデバッグ情報を取得します。Java 引数は次のとおりです。

```
-Dopenorb.debug.trace=DEBUG -Dopenorb.debug.level=HIGH
```

**ステップ3** EMS サーバのホスト名が解決されない場合、例外がスローされます。「priems01」や「priems\_nms1」などのデフォルト名が使用されていた場合、これらの名前はクライアント ネットワークでは解決されません。CORBA パッケージをアンインストールし、再インストールして、IP アドレスを受け入れるように EMS を修正します。

---

## CORBA と EPOM のトラブルシューティング手順

この項では、CORBA とともに Extensible Provisioning and Operations Manager (EPOM) をトラブルシューティングするためのいくつかの手順について説明します。UNIX と Java コマンドの知識が必要です。

**注意**

EPOM は EMS と共存できません。

問題が発生した場合は、次の手順を実行します。

**ステップ 1** CORBA アプリケーションが動作していることを確認します。

```
ps -ef | grep cis
```

**ステップ 2** cis.log の中にエラーまたは手掛かりがないかどうかを確認します。

```
more /opt/ems/log/CIS.log
```

**ステップ 3** EPOM のログにエラーまたは手掛かりがないかどうかを確認します。

```
more var/opt/CSCOepom/logs
```

Cisco BTS 10200 ソフトスイッチの CORBA サービスの設定方法に問題がある場合は、次の手順を実行します。

**ステップ 1** Cisco BTS 10200 ソフトスイッチのホスト名が CORBA 設定で提供されていることを確認します。

- ホスト名が提供されている場合は、Cisco BTS 10200 ソフトスイッチの IP アドレスに変更し、Cisco BTS 10200 ソフトスイッチ上の CORBA サービスを再起動します。
- ホスト名が提供されていない場合は、他の作業を行う前に CORBA アプリケーションを終了し、再インストールします。

**ステップ 2** 次のコマンドを使用して、ポート 683 および 14001 が Cisco BTS 10200 ソフトスイッチ サーバ上で待ち受け状態かどうかを確認します。

```
netstat -a |grep 683  
netstat -a |grep 14001.
```

**ステップ 3** Cisco BTS 10200 ソフトスイッチ サーバ上の CIS.log に対して tail -f を実行します。

```
tail -f /opt/ems/log/CIS.log
```



**(注)** EPOM のログ ファイルは /var/opt/CSCOepom/logs にあります。EPOM サーバ上の trace.log ファイルと localhost ファイルを確認します。

**ステップ 4** EPOM GUI の Config ボタンをクリックし（何回かクリックが必要な場合があります）、次のコマンドを実行します。

```
netstat -a |grep 14001 and netstat -a |grep 683
```

**ステップ 5** ユーザ名とパスワードが、Cisco BTS 10200 ソフトスイッチ インベントリを EPOM に追加したときと同じものであることを確認します。ユーザ名とパスワードは、デフォルトではそれぞれ optiuser です。CLI を使用してシステムにログインし、このことを確認します。

---

## CORBA と EPOM の特殊文字のトラブルシューティング : subscriber コマンド

表 5-1 は、subscriber コマンドに含まれている特殊文字に対する CORBA と EPOM の応答を示しています。

表 5-1 特殊文字に対する CORBA と EPOM の応答

文字	使用例	CORBA と EPOM の応答
' (一重引用符)	<code>btsadmin&gt;change subscriber id=x1-6-00-00-ca-ac-ef-98_02;name=Joe'</code>	<p><b>エラー メッセージ</b> BtsException: IDL:oam.sswitch.com/cad/CadExceptions:1.0 , Invalid parameter value. name=Joe'; contains one of the following invalid characters: ('')</p> <p><b>説明</b> 一重引用符は、文字列を区切るための予約文字です。MTA 宛ての caller-ID メッセージでは、この目的で使用されます。caller-ID メッセージを解析する MTA でエラーが発生するため、名前に一重引用符を使用することはできません。</p>
" (二重引用符)	<code>btsadmin&gt;change subscriber id=x1-6-00-00-ca-ac-ef-98_02;name=Joe"</code>	<p><b>エラー メッセージ</b> org.omg.CORBA.UNKNOWN: Server Exception: Unregistered vendor exception #0 vmcid: 0x0 minor code: 0 completed</p> <p><b>説明</b> 二重引用符は、文字列を区切るための予約文字です。MTA 宛ての caller-ID メッセージでは、この目的で使用されます。caller-ID メッセージを解析する MTA でエラーが発生するため、名前に二重引用符を使用することはできません。</p>

表 5-1 特殊文字に対する CORBA と EPOM の応答 (続き)

文字	使用例	CORBA と EPOM の応答
; (セミコロン)	<p>例 1 :</p> <pre>btsadmin&gt;change subscriber id=x1-6-00-00-ca-ac-ef-98_02;name=Joe;;</pre> <p>Reply : Success: Transaction 914661109242987301 was processed.</p> <pre>btsadmin&gt;show subscriber id=x1-6-00-00-ca-ac-ef-98_02 ID=x1-6-00-00-ca-ac-ef-98_02 CATEGORY=INDIVIDUAL NAME=Joe STATUS=ACTIVE COUNTRY=USA PRIVACY=NONE RING_TYPE_DN1=1 TERM_ID=aaln/S1/1 MGW_ID=iad-11 PIC1=NONE PIC2=NONE PIC3=NONE GRP=N USAGE_SENS=Y SUB_PROFILE_ID=subpf1 TERM_TYPE=TERM IMMEDIATE_RELEASE=N TERMINATING_IMMEDIATE_REL=N SEND_BILLING_DN=N SEND_BDN_AS_CPN=N SEND_BDN_FOR_EMG=N</pre>	<p><b>エラー メッセージ</b> ID=x1-6-00-00-ca-ac-ef-98_02  CATEGORY=INDIVIDUAL  NAME=Joe;  STATUS=ACTIVE  COUNTRY=USA  PRIVACY=NONE  RING_TYPE_DN1=1  TERM_ID=aaln/S1/1  MGW_ID=iad-11  PIC1=NONE  PIC2=NONE  PIC3=NONE  GRP=N  USAGE_SENS=Y  SUB_PROFILE_ID=subpf1  TERM_TYPE=TERM  IMMEDIATE_RELEASE=N  TERMINATING_IMMEDIATE_REL=N  SEND_BILLING_DN=N  SEND_BDN_AS_CPN=N  SEND_BDN_FOR_EMG=N</p> <p>Reply : Success: Entry 1 of 1 returned.</p> <p><b>説明</b> 最初の例では、2つのセミコロン (;) は単に2つのヌルパラメータを区切っているだけです。2番目の例では、最後のパラメータはEnterキーで終端されているヌルパラメータです。CLIでは、常にEnterキーによって最後のパラメータが終端されます。</p> <p>セミコロンの間にパラメータを記述しない状態で、複数のセミコロンでコマンドを終端することは可能です。複数のセミコロンを他の場所に配置すると、キューが異常停止します。</p>

表 5-1 特殊文字に対する CORBA と EPOM の応答 (続き)

文字	使用例	CORBA と EPOM の応答
; (セミコロン)	<p>例 2 :</p> <pre>btsadmin&gt;change subscriber id=x1-6-00-00-ca-ac-ef-98_02;name=Joe;</pre> <p>Reply : Success: Transaction 914661178134430501 was processed.</p> <pre>btsadmin&gt;show subscriber id=x1-6-00-00-ca-ac-ef-98_02</pre> <p>ID=x1-6-00-00-ca-ac-ef-98_02</p> <p>CATEGORY=INDIVIDUAL</p> <p>NAME=Joe</p> <p>STATUS=ACTIVE</p> <p>COUNTRY=USA</p> <p>PRIVACY=NONE</p> <p>RING_TYPE_DN1=1</p> <p>TERM_ID=aaln/S1/1</p> <p>MGW_ID=iad-11</p> <p>PIC1=NONE</p> <p>PIC2=NONE</p> <p>PIC3=NONE</p> <p>GRP=N</p> <p>USAGE_SENS=Y</p> <p>SUB_PROFILE_ID=subpf1</p> <p>TERM_TYPE=TERM</p> <p>IMMEDIATE_RELEASE=N</p> <p>TERMINATING_IMMEDIATE_REL=N</p> <p>SEND_BILLING_DN=N</p> <p>SEND_BDN_AS_CPN=N</p> <p>SEND_BDN_FOR_EMG=N</p> <p>Reply : Success: Entry 1 of 1 returned.</p>	<p><b>エラー メッセージ</b> ID=x1-6-00-00-ca-ac-ef-98_02</p> <p>CATEGORY=INDIVIDUAL</p> <p>NAME=Joe;</p> <p>STATUS=ACTIVE</p> <p>COUNTRY=USA</p> <p>PRIVACY=NONE</p> <p>RING_TYPE_DN1=1</p> <p>TERM_ID=aaln/S1/1</p> <p>MGW_ID=iad-11</p> <p>PIC1=NONE</p> <p>PIC2=NONE</p> <p>PIC3=NONE</p> <p>GRP=N</p> <p>USAGE_SENS=Y</p> <p>SUB_PROFILE_ID=subpf1</p> <p>TERM_TYPE=TERM</p> <p>IMMEDIATE_RELEASE=N</p> <p>TERMINATING_IMMEDIATE_REL=N</p> <p>SEND_BILLING_DN=N</p> <p>SEND_BDN_AS_CPN=N</p> <p>SEND_BDN_FOR_EMG=N</p> <p>Reply : Success: Entry 1 of 1 returned.</p> <p><b>説明</b> 最初の例では、2つのセミコロン (;) は単に2つのヌルパラメータを区切っているだけです。2番目の例では、最後のパラメータはEnterキーで終端されているヌルパラメータです。CLIでは、常にEnterキーによって最後のパラメータが終端されます。</p> <p>セミコロンの中にパラメータを記述しない状態で、複数のセミコロンでコマンドを終端することは可能です。複数のセミコロンを他の場所に配置すると、キューが異常停止します。</p>
% (パーセント記号)	<pre>btsadmin&gt;change subscriber id=x1-6-00-00-ca-ac-ef-98_02;name=Joe%</pre>	<p><b>エラー メッセージ</b> BtsException:</p> <p>IDL:oam.sswitch.com/cad/CadExceptions:1.0, Invalid parameter value. name=Joe%; Enter at least 1 character, but not more than 32 characters.</p> <p><b>説明</b> パーセント記号はワイルドカードです。たとえば、show subscriber id=x1-6-00% は id が x1-6-00 で始まるすべてのサブスクリイバを表示します。したがって、パーセント記号は subscriber という noun では有効な文字ですが、verb である add または change で使用した場合は無効です。</p>

表 5-1 特殊文字に対する CORBA と EPOM の応答 (続き)

文字	使用例	CORBA と EPOM の応答
- (ハイフン)	<pre>btsadmin&gt;change subscriber id=x1-6-00-00-ca-ac-ef-98_02;name=Joe-</pre>	<p><b>エラー メッセージ</b> Success: Transaction 914661231396356901 was processed.</p> <p><b>説明</b> これは有効な文字ですが、発信者識別情報の表示に影響しません。この文字を使用することはお勧めしません。</p>
_ (アンダースコア)	<pre>btsadmin&gt;change subscriber id=x1-6-00-00-ca-ac-ef-98_02;name=Joe_</pre>	<p><b>エラー メッセージ</b> Failure: NAME cannot contain an _ character</p> <p><b>説明</b> これは有効な文字ですが、発信者識別情報の表示に影響しません。この文字を使用することはお勧めしません。</p>
&	<pre>btsadmin&gt;change subscriber id=x1-6-00-00-ca-ac-ef-98_02;name=Joe&amp;</pre> <p>Reply : Success: Transaction 914661254499435301 was processed.</p>	<p><b>エラー メッセージ</b> org.omg.CORBA.UNKNOWN: Server Exception: Unregistered vendor exception #0 vmcid: 0x0 minor code: 0 completed</p> <p><b>説明</b> アンパサンド (&amp;) は XML の予約文字です。このトランザクションは、XML を使用して CORBA インターフェイスを通じて送信された場合、失敗します。</p>
@#\$^&*(){} \/>.,:[]~! (アンパサンドが使用されている)	<pre>btsadmin&gt;change subscriber id=x1-6-00-00-ca-ac-ef-98_02;name=Joe!@#\$^ &amp;*(){} \/&gt;.,:[]~</pre> <p>Reply : Success: Transaction 914661290390225701 was processed.</p>	<p><b>エラー メッセージ</b> org.omg.CORBA.UNKNOWN: Server Exception: Unregistered vendor exception #0 vmcid: 0x0 minor code: 0 completed.</p> <p><b>説明</b> アンパサンド (&amp;) は XML の予約文字です。このトランザクションは、XML を使用して CORBA インターフェイスを通じて送信された場合、失敗します。</p>
@#\$^&*(){} \/>.,:[]~! (アンパサンドが使用されていない)	<pre>btsadmin&gt;change subscriber id=x1-6-00-00-ca-ac-ef-98_02;name=Joe!@#\$^ *(){} \/&gt;.,:[]~</pre>	<p><b>エラー メッセージ</b> Successful</p> <p><b>説明</b> これらの文字は有効です。</p>





# XML 記述ドキュメント

---

この付録では、XML 記述ドキュメントの例を示します。

## subscriber noun と add verb

次のサンプル XML 記述ドキュメントは、noun が subscriber、verb が add である例を示しています。

```
=====
<ManagedObject Verb="add" id="subscriber"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="ManagedObject.xsd">
  <MOAttribute id="dn1">
    <Required>>false</Required>
    <Type>text</Type>
    <Default>[null]</Default>
    <Width>14</Width>
    <HelpText>Enter at least 1, but not more than 14 characters from the following
set: {0123456789-}.</HelpText>
    <Label>Dn1</Label>
    <Parser id="GenericDNParser">
      <JavaScript>TBD</JavaScript>
      <RegExp>TBD</RegExp>
    </Parser>
  </MOAttribute>
  <MOAttribute id="tg">
    <Required>>false</Required>
    <Type>text</Type>
    <Default>[null]</Default>
    <Width>32</Width>
    <HelpText>Enter at least 1 character, but not more than 32 characters.</HelpText>
    <Label>Tg</Label>
    <Parser id="TextParser">
      <JavaScript>TBD</JavaScript>
      <RegExp>TBD</RegExp>
    </Parser>
  </MOAttribute>
  <MOAttribute id="policy_id">
    <Required>>false</Required>
    <Type>text</Type>
    <Default>[null]</Default>
    <Width>16</Width>
    <HelpText>Enter at least 1 character, but not more than 16 characters.</HelpText>
    <Label>POLICY_ID</Label>
    <Parser id="TextParser">
      <JavaScript>TBD</JavaScript>
      <RegExp>TBD</RegExp>
    </Parser>
  </MOAttribute>
  <MOAttribute id="category">
    <Required>>false</Required>

```

```

    <Type>single</Type>
    <Default>[INDIVIDUAL]</Default>
    <Width>15</Width>
    <HelpText>Enter one of the following values: [INDIVIDUAL, MLHG, MLHG_INDIVIDUAL,
MLHG_PREF_INDIV, CTXG, CTXG_INDIVIDUAL, PBX, CTXG_TG, CTXG_MLHG, RACF, IVR]</HelpText>
    <Label>Category</Label>
    <Permitted>[INDIVIDUAL, MLHG, MLHG_INDIVIDUAL, MLHG_PREF_INDIV, CTXG,
CTXG_INDIVIDUAL, PBX, CTXG_TG, CTXG_MLHG, RACF, IVR]</Permitted>
  </MOAttribute>
  <MOAttribute id="ss_number">
    <Required>>false</Required>
    <Type>text</Type>
    <Default>[null]</Default>
    <Width>11</Width>
    <HelpText>Enter a Social Security Number in the form ###-##-#### where # is digit
from 0-9.</HelpText>
    <Label>Ss Number</Label>
    <Parser id="SocSecParser">
      <JavaScript>TBD</JavaScript>
      <RegExp>TBD</RegExp>
    </Parser>
  </MOAttribute>
  <MOAttribute id="ctxg_id">
    <Required>>false</Required>
    <Type>text</Type>
    <Default>[null]</Default>
    <Width>16</Width>
    <HelpText>Enter at least 1 character, but not more than 16 characters.</HelpText>
    <Label>Ctxg Id</Label>
    <Parser id="TextParser">
      <JavaScript>TBD</JavaScript>
      <RegExp>TBD</RegExp>
    </Parser>
  </MOAttribute>
  <MOAttribute id="name">
    <Required>>false</Required>
    <Type>text</Type>
    <Default>[null]</Default>
    <Width>32</Width>
    <HelpText>Enter at least 1 character, but not more than 32 characters.</HelpText>
    <Label>Name</Label>
    <Parser id="TextParser">
      <JavaScript>TBD</JavaScript>
      <RegExp>TBD</RegExp>
    </Parser>
  </MOAttribute>
  <MOAttribute id="mlhg_pref_list_id">
    <Required>>false</Required>
    <Type>text</Type>
    <Default>[null]</Default>
    <Width>16</Width>
    <HelpText>Enter at least 1 character, but not more than 16 characters.</HelpText>
    <Label>Mlhg Pref List Id</Label>
    <Parser id="TextParser">
      <JavaScript>TBD</JavaScript>
      <RegExp>TBD</RegExp>
    </Parser>
  </MOAttribute>
  <MOAttribute id="address2">
    <Required>>false</Required>
    <Type>text</Type>
    <Default>[null]</Default>
    <Width>32</Width>
    <HelpText>Enter at least 1 character, but not more than 32 characters.</HelpText>
    <Label>Address2</Label>
    <Parser id="TextParser">
      <JavaScript>TBD</JavaScript>
      <RegExp>TBD</RegExp>
    </Parser>
  </MOAttribute>

```

```

</MOAttribute>
<MOAttribute id="address1">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[null]</Default>
  <Width>32</Width>
  <HelpText>Enter at least 1 character, but not more than 32 characters.</HelpText>
  <Label>Address1</Label>
  <Parser id="TextParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
<MOAttribute id="city">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[null]</Default>
  <Width>16</Width>
  <HelpText>Enter at least 1 character, but not more than 16 characters.</HelpText>
  <Label>City</Label>
  <Parser id="TextParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
<MOAttribute id="terminating_immediate_rel">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[N]</Default>
  <Width>3</Width>
  <HelpText>Enter a boolean value of Y for yes or N for no.</HelpText>
  <Label>Terminating Immediate Release</Label>
  <Parser id="BooleanParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
<MOAttribute id="billing_dn">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[null]</Default>
  <Width>32</Width>
  <HelpText>Enter at least 1, but not more than 32 characters from the following
set: {0123456789-}.</HelpText>
  <Label>Billing Dn</Label>
  <Parser id="GenericDNParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
<MOAttribute id="language">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[null]</Default>
  <Width>16</Width>
  <HelpText>Enter at least 1 character, but not more than 16 characters.</HelpText>
  <Label>Language</Label>
  <Parser id="TextParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
<MOAttribute id="email">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[null]</Default>
  <Width>64</Width>
  <HelpText>Enter an email address in the form text@text where text is a set of
characters with no spaces.</HelpText>

```

```

    <Label>Email</Label>
    <Parser id="EmailParser">
      <JavaScript>TBD</JavaScript>
      <RegExp>TBD</RegExp>
    </Parser>
  </MOAttribute>
  <MOAttribute id="mlhg_id">
    <Required>>false</Required>
    <Type>text</Type>
    <Default>[null]</Default>
    <Width>16</Width>
    <HelpText>Enter at least 1 character, but not more than 16 characters.</HelpText>
    <Label>MLHG ID</Label>
    <Parser id="TextParser">
      <JavaScript>TBD</JavaScript>
      <RegExp>TBD</RegExp>
    </Parser>
  </MOAttribute>
  <MOAttribute id="tgn_id">
    <Required>>false</Required>
    <Type>text</Type>
    <Default>[null]</Default>
    <Width>8</Width>
    <HelpText>Enter a number from 0 to 99999999.</HelpText>
    <Label>Trunk Group Number ID</Label>
    <Parser id="DecimalParser">
      <JavaScript>TBD</JavaScript>
      <RegExp>TBD</RegExp>
    </Parser>
  </MOAttribute>
  <MOAttribute id="mgw_id">
    <Required>>false</Required>
    <Type>text</Type>
    <Default>[null]</Default>
    <Width>32</Width>
    <HelpText>Enter at least 0 characters, but not more than 32 characters.</HelpText>
    <Label>Media Gateway ID</Label>
    <Parser id="TextParser">
      <JavaScript>TBD</JavaScript>
      <RegExp>TBD</RegExp>
    </Parser>
  </MOAttribute>
  <MOAttribute id="status">
    <Required>>false</Required>
    <Type>single</Type>
    <Default>[ACTIVE]</Default>
    <Width>17</Width>
    <HelpText>Enter one of the following values: [ACTIVE, TEMP_OOS, TEMP_DISCONNECTED,
TEMP_UNAVAILABLE]</HelpText>
    <Label>Status</Label>
    <Permitted>[ACTIVE, TEMP_OOS, TEMP_DISCONNECTED, TEMP_UNAVAILABLE]</Permitted>
  </MOAttribute>
  <MOAttribute id="term_id">
    <Required>>false</Required>
    <Type>text</Type>
    <Default>[null]</Default>
    <Width>32</Width>
    <HelpText>Enter at least 1 character, but not more than 32 characters.</HelpText>
    <Label>Termination ID</Label>
    <Parser id="TextParser">
      <JavaScript>TBD</JavaScript>
      <RegExp>TBD</RegExp>
    </Parser>
  </MOAttribute>
  <MOAttribute id="usage_sens">
    <Required>>false</Required>
    <Type>text</Type>
    <Default>[Y]</Default>
    <Width>3</Width>

```

```
<HelpText>Enter a boolean value of Y for yes or N for no.</HelpText>
<Label>Usage Sens</Label>
<Parser id="BooleanParser">
  <JavaScript>TBD</JavaScript>
  <RegExp>TBD</RegExp>
</Parser>
</MOAttribute>
<MOAttribute id="id">
  <Required>>true</Required>
  <Type>text</Type>
  <Default>[null]</Default>
  <Width>30</Width>
  <HelpText>Enter at least 1 character, but not more than 30 characters.</HelpText>
  <Label>ID</Label>
  <Parser id="TextParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
<MOAttribute id="grp">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[N]</Default>
  <Width>3</Width>
  <HelpText>Enter a boolean value of Y for yes or N for no.</HelpText>
  <Label>Grp</Label>
  <Parser id="BooleanParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
<MOAttribute id="sub_profile_id">
  <Required>>true</Required>
  <Type>text</Type>
  <Default>[null]</Default>
  <Width>16</Width>
  <HelpText>Enter at least 1 character, but not more than 16 characters.</HelpText>
  <Label>Sub Profile Id</Label>
  <Parser id="TextParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
<MOAttribute id="country">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[USA]</Default>
  <Width>16</Width>
  <HelpText>Enter at least 1 character, but not more than 16 characters.</HelpText>
  <Label>Country</Label>
  <Parser id="TextParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
<MOAttribute id="cos_restrict_id">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[null]</Default>
  <Width>16</Width>
  <HelpText>Enter at least 1 character, but not more than 16 characters.</HelpText>
  <Label>COS Restrict ID</Label>
  <Parser id="TextParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
<MOAttribute id="qos_id">
  <Required>>false</Required>
```

```

<Type>text</Type>
<Default>[null]</Default>
<Width>16</Width>
<HelpText>Enter at least 1 character, but not more than 16 characters.</HelpText>
<Label>QOS ID</Label>
<Parser id="TextParser">
  <JavaScript>TBD</JavaScript>
  <RegExp>TBD</RegExp>
</Parser>
</MOAttribute>
<MOAttribute id="term_type">
  <Required>>false</Required>
  <Type>single</Type>
  <Default>[TERM]</Default>
  <Width>5</Width>
  <HelpText>Enter one of the following values: [TERM, TG, ROUTE, RG]</HelpText>
  <Label>TERM TYPE</Label>
  <Permitted>[TERM, TG, ROUTE, RG]</Permitted>
</MOAttribute>
<MOAttribute id="ring_type_dn1">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[1]</Default>
  <Width>1</Width>
  <HelpText>Enter a number from 1 to 3.</HelpText>
  <Label>Ring Type Dn1</Label>
  <Parser id="DecimalParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
<MOAttribute id="immediate_release">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[N]</Default>
  <Width>3</Width>
  <HelpText>Enter a boolean value of Y for yes or N for no.</HelpText>
  <Label>Immediate Release</Label>
  <Parser id="BooleanParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
<MOAttribute id="sip_url">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[null]</Default>
  <Width>32</Width>
  <HelpText>Enter at least 1 character, but not more than 32 characters.</HelpText>
  <Label>Sip Url</Label>
  <Parser id="TextParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
<MOAttribute id="zipcode">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[null]</Default>
  <Width>10</Width>
  <HelpText>Enter at least 1 character, but not more than 10 characters.</HelpText>
  <Label>Zipcode</Label>
  <Parser id="TextParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
<MOAttribute id="pic3">
  <Required>>false</Required>

```

```
<Type>text</Type>
<Default>[null]</Default>
<Width>4</Width>
<HelpText>Enter a PIC value as four numeric characters, NPIC, or NONE.</HelpText>
<Label>Pic3</Label>
<Parser id="PicParser">
  <JavaScript>TBD</JavaScript>
  <RegExp>TBD</RegExp>
</Parser>
</MOAttribute>
<MOAttribute id="pic2">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[null]</Default>
  <Width>4</Width>
  <HelpText>Enter a PIC value as four numeric characters, NPIC, or NONE.</HelpText>
  <Label>Pic2</Label>
  <Parser id="PicParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
<MOAttribute id="privacy">
  <Required>>false</Required>
  <Type>single</Type>
  <Default>[NONE]</Default>
  <Width>4</Width>
  <HelpText>Enter one of the following values: [FULL, NAME, NONE]</HelpText>
  <Label>Privacy</Label>
  <Permitted>[FULL, NAME, NONE]</Permitted>
</MOAttribute>
<MOAttribute id="pic1">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[null]</Default>
  <Width>4</Width>
  <HelpText>Enter a PIC value as four numeric characters, NPIC, or NONE.</HelpText>
  <Label>Pic1</Label>
  <Parser id="PicParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
<MOAttribute id="state">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[null]</Default>
  <Width>16</Width>
  <HelpText>Enter at least 1 character, but not more than 16 characters.</HelpText>
  <Label>State</Label>
  <Parser id="TextParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
</ManagedObject>
```

## 外部キー リレーション

次のサンプル XML 拡張形式記述ドキュメントは、所定のコマンドの外部キー リレーションを保持しています。

```

=====
<ManagedObject Verb="add" id="subscriber"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="ManagedObject.xsd">
  <MOAttribute id="dn1">
    <Required>false</Required>
    <Type>text</Type>
    <Default>[null]</Default>
    <Width>14</Width>
    <HelpText>Enter at least 1, but not more than 14 characters from the following
set: {0123456789-}.</HelpText>
    <Label>Dn1</Label>
    <Parser id="GenericDNParser">
      <JavaScript>TBD</JavaScript>
      <RegExp>TBD</RegExp>
    </Parser>
  </MOAttribute>
  <MOAttribute id="tg">
    <Required>false</Required>
    <Type>text</Type>
    <Default>[null]</Default>
    <Width>32</Width>
    <HelpText>Enter at least 1 character, but not more than 32 characters.</HelpText>
    <Label>Tg</Label>
    <Parser id="TextParser">
      <JavaScript>TBD</JavaScript>
      <RegExp>TBD</RegExp>
    </Parser>
  </MOAttribute>
  <MOAttribute id="policy_id">
    <Required>false</Required>
    <Type>text</Type>
    <Default>[null]</Default>
    <Width>16</Width>
    <HelpText>Enter at least 1 character, but not more than 16 characters.</HelpText>
    <Label>POLICY_ID</Label>
    <Parser id="TextParser">
      <JavaScript>TBD</JavaScript>
      <RegExp>TBD</RegExp>
    </Parser>
  </MOAttribute>
  <MOAttribute id="category">
    <Required>false</Required>
    <Type>single</Type>
    <Default>[INDIVIDUAL]</Default>
    <Width>15</Width>
    <HelpText>Enter one of the following values: [INDIVIDUAL, MLHG, MLHG_INDIVIDUAL,
MLHG_PREF_INDIV, CTXG, CTXG_INDIVIDUAL, PBX, CTXG_TG, CTXG_MLHG, RACF, IVR]</HelpText>
    <Label>Category</Label>
    <Permitted>[INDIVIDUAL, MLHG, MLHG_INDIVIDUAL, MLHG_PREF_INDIV, CTXG,
CTXG_INDIVIDUAL, PBX, CTXG_TG, CTXG_MLHG, RACF, IVR]</Permitted>
  </MOAttribute>
  <MOAttribute id="ss_number">
    <Required>false</Required>
    <Type>text</Type>
    <Default>[null]</Default>
    <Width>11</Width>
    <HelpText>Enter a Social Security Number in the form ###-##-#### where # is digit
from 0-9.</HelpText>
    <Label>Ss Number</Label>
    <Parser id="SocSecParser">
      <JavaScript>TBD</JavaScript>
      <RegExp>TBD</RegExp>

```



```

    </Parser>
  </MOAttribute>
  <MOAttribute id="ctxg_id">
    <Required>>false</Required>
    <Type>text</Type>
    <Default>[null]</Default>
    <Width>16</Width>
    <HelpText>Enter at least 1 character, but not more than 16 characters.</HelpText>
    <Label>Ctxg Id</Label>
    <Parser id="TextParser">
      <JavaScript>TBD</JavaScript>
      <RegExp>TBD</RegExp>
    </Parser>
    <Fk id="CENTREX_GRP_PK">
      <Noun>centrex_grp</Noun>
      <Param>id</Param>
    </Fk>
  </MOAttribute>
  <MOAttribute id="name">
    <Required>>false</Required>
    <Type>text</Type>
    <Default>[null]</Default>
    <Width>32</Width>
    <HelpText>Enter at least 1 character, but not more than 32 characters.</HelpText>
    <Label>Name</Label>
    <Parser id="TextParser">
      <JavaScript>TBD</JavaScript>
      <RegExp>TBD</RegExp>
    </Parser>
  </MOAttribute>
  <MOAttribute id="mlhg_pref_list_id">
    <Required>>false</Required>
    <Type>text</Type>
    <Default>[null]</Default>
    <Width>16</Width>
    <HelpText>Enter at least 1 character, but not more than 16 characters.</HelpText>
    <Label>Mlhg Pref List Id</Label>
    <Parser id="TextParser">
      <JavaScript>TBD</JavaScript>
      <RegExp>TBD</RegExp>
    </Parser>
    <Fk id="MLH_PREF_LIST_PK">
      <Noun>mlhg_pref_list</Noun>
      <Param>id</Param>
    </Fk>
  </MOAttribute>
  <MOAttribute id="address2">
    <Required>>false</Required>
    <Type>text</Type>
    <Default>[null]</Default>
    <Width>32</Width>
    <HelpText>Enter at least 1 character, but not more than 32 characters.</HelpText>
    <Label>Address2</Label>
    <Parser id="TextParser">
      <JavaScript>TBD</JavaScript>
      <RegExp>TBD</RegExp>
    </Parser>
  </MOAttribute>
  <MOAttribute id="address1">
    <Required>>false</Required>
    <Type>text</Type>
    <Default>[null]</Default>
    <Width>32</Width>
    <HelpText>Enter at least 1 character, but not more than 32 characters.</HelpText>
    <Label>Address1</Label>
    <Parser id="TextParser">
      <JavaScript>TBD</JavaScript>
      <RegExp>TBD</RegExp>
    </Parser>

```

```

</MOAttribute>
<MOAttribute id="city">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[null]</Default>
  <Width>16</Width>
  <HelpText>Enter at least 1 character, but not more than 16 characters.</HelpText>
  <Label>City</Label>
  <Parser id="TextParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
<MOAttribute id="terminating_immediate_rel">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[N]</Default>
  <Width>3</Width>
  <HelpText>Enter a boolean value of Y for yes or N for no.</HelpText>
  <Label>Terminating Immediate Release</Label>
  <Parser id="BooleanParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
<MOAttribute id="billing_dn">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[null]</Default>
  <Width>32</Width>
  <HelpText>Enter at least 1, but not more than 32 characters from the following
set: {0123456789-}.</HelpText>
  <Label>Billing Dn</Label>
  <Parser id="GenericDNParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
<MOAttribute id="language">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[null]</Default>
  <Width>16</Width>
  <HelpText>Enter at least 1 character, but not more than 16 characters.</HelpText>
  <Label>Language</Label>
  <Parser id="TextParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
<MOAttribute id="email">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[null]</Default>
  <Width>64</Width>
  <HelpText>Enter an email address in the form text@text where text is a set of
characters with no spaces.</HelpText>
  <Label>Email</Label>
  <Parser id="EmailParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
<MOAttribute id="mlhg_id">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[null]</Default>
  <Width>16</Width>
  <HelpText>Enter at least 1 character, but not more than 16 characters.</HelpText>

```

```

<Label>MLHG ID</Label>
<Parser id="TextParser">
  <JavaScript>TBD</JavaScript>
  <RegExp>TBD</RegExp>
</Parser>
<Fk id="MLH_PREF_LIST_PK">
  <Noun>mlhg_pref_list</Noun>
  <Param>mlhg_id</Param>
  <Fk id="MLHG_PK">
    <Noun>mlhg</Noun>
    <Param>id</Param>
  </Fk>
</Fk>
</MOAttribute>
<MOAttribute id="tgn_id">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[null]</Default>
  <Width>8</Width>
  <HelpText>Enter a number from 0 to 99999999.</HelpText>
  <Label>Trunk Group Number ID</Label>
  <Parser id="DecimalParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
  <Fk id="TRUNK_GRP_PK">
    <Noun>trunk_grp</Noun>
    <Param>id</Param>
  </Fk>
</MOAttribute>
<MOAttribute id="mgw_id">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[null]</Default>
  <Width>32</Width>
  <HelpText>Enter at least 0 characters, but not more than 32 characters.</HelpText>
  <Label>Media Gateway ID</Label>
  <Parser id="TextParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
  <Fk id="TERMINATION_PK">
    <Noun>termination</Noun>
    <Param>mgw_id</Param>
    <Fk id="MGW_PK">
      <Noun>mgw</Noun>
      <Param>id</Param>
    </Fk>
  </Fk>
</MOAttribute>
<MOAttribute id="status">
  <Required>>false</Required>
  <Type>single</Type>
  <Default>[ACTIVE]</Default>
  <Width>17</Width>
  <HelpText>Enter one of the following values: [ACTIVE, TEMP_OOS, TEMP_DISCONNECTED,
TEMP_UNAVAILABLE]</HelpText>
  <Label>Status</Label>
  <Permitted>[ACTIVE, TEMP_OOS, TEMP_DISCONNECTED, TEMP_UNAVAILABLE]</Permitted>
</MOAttribute>
<MOAttribute id="term_id">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[null]</Default>
  <Width>32</Width>
  <HelpText>Enter at least 1 character, but not more than 32 characters.</HelpText>
  <Label>Termination ID</Label>
  <Parser id="TextParser">
    <JavaScript>TBD</JavaScript>

```

```

    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
<Fk id="TERMINATION_PK">
  <Noun>termination</Noun>
  <Param>id</Param>
</Fk>
</MOAttribute>
<MOAttribute id="usage_sens">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[Y]</Default>
  <Width>3</Width>
  <HelpText>Enter a boolean value of Y for yes or N for no.</HelpText>
  <Label>Usage Sens</Label>
  <Parser id="BooleanParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
<MOAttribute id="id">
  <Required>>true</Required>
  <Type>text</Type>
  <Default>[null]</Default>
  <Width>30</Width>
  <HelpText>Enter at least 1 character, but not more than 30 characters.</HelpText>
  <Label>ID</Label>
  <Parser id="TextParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
<MOAttribute id="grp">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[N]</Default>
  <Width>3</Width>
  <HelpText>Enter a boolean value of Y for yes or N for no.</HelpText>
  <Label>Grp</Label>
  <Parser id="BooleanParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
<MOAttribute id="sub_profile_id">
  <Required>>true</Required>
  <Type>text</Type>
  <Default>[null]</Default>
  <Width>16</Width>
  <HelpText>Enter at least 1 character, but not more than 16 characters.</HelpText>
  <Label>Sub Profile Id</Label>
  <Parser id="TextParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
  <Fk id="SUBSCRIBER_PROFILE_PK">
    <Noun>subscriber_profile</Noun>
    <Param>id</Param>
  </Fk>
</MOAttribute>
<MOAttribute id="country">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[USA]</Default>
  <Width>16</Width>
  <HelpText>Enter at least 1 character, but not more than 16 characters.</HelpText>
  <Label>Country</Label>
  <Parser id="TextParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>

```

```

    </Parser>
  </MOAttribute>
  <MOAttribute id="cos_restrict_id">
    <Required>>false</Required>
    <Type>text</Type>
    <Default>[null]</Default>
    <Width>16</Width>
    <HelpText>Enter at least 1 character, but not more than 16 characters.</HelpText>
    <Label>COS Restrict ID</Label>
    <Parser id="TextParser">
      <JavaScript>TBD</JavaScript>
      <RegExp>TBD</RegExp>
    </Parser>
    <Fk id="COST_RESTRICT_PK">
      <Noun>cos_restrict</Noun>
      <Param>id</Param>
    </Fk>
  </MOAttribute>
  <MOAttribute id="qos_id">
    <Required>>false</Required>
    <Type>text</Type>
    <Default>[null]</Default>
    <Width>16</Width>
    <HelpText>Enter at least 1 character, but not more than 16 characters.</HelpText>
    <Label>QOS ID</Label>
    <Parser id="TextParser">
      <JavaScript>TBD</JavaScript>
      <RegExp>TBD</RegExp>
    </Parser>
    <Fk id="QOS_PK">
      <Noun>qos</Noun>
      <Param>id</Param>
    </Fk>
  </MOAttribute>
  <MOAttribute id="term_type">
    <Required>>false</Required>
    <Type>single</Type>
    <Default>[TERM]</Default>
    <Width>5</Width>
    <HelpText>Enter one of the following values: [TERM, TG, ROUTE, RG]</HelpText>
    <Label>TERM TYPE</Label>
    <Permitted>[TERM, TG, ROUTE, RG]</Permitted>
  </MOAttribute>
  <MOAttribute id="ring_type_dn1">
    <Required>>false</Required>
    <Type>text</Type>
    <Default>[1]</Default>
    <Width>1</Width>
    <HelpText>Enter a number from 1 to 3.</HelpText>
    <Label>Ring Type Dn1</Label>
    <Parser id="DecimalParser">
      <JavaScript>TBD</JavaScript>
      <RegExp>TBD</RegExp>
    </Parser>
  </MOAttribute>
  <MOAttribute id="immediate_release">
    <Required>>false</Required>
    <Type>text</Type>
    <Default>[N]</Default>
    <Width>3</Width>
    <HelpText>Enter a boolean value of Y for yes or N for no.</HelpText>
    <Label>Immediate Release</Label>
    <Parser id="BooleanParser">
      <JavaScript>TBD</JavaScript>
      <RegExp>TBD</RegExp>
    </Parser>
  </MOAttribute>
  <MOAttribute id="sip_url">
    <Required>>false</Required>

```

```

<Type>text</Type>
<Default>[null]</Default>
<Width>32</Width>
<HelpText>Enter at least 1 character, but not more than 32 characters.</HelpText>
<Label>Sip Url</Label>
<Parser id="TextParser">
  <JavaScript>TBD</JavaScript>
  <RegExp>TBD</RegExp>
</Parser>
</MOAttribute>
<MOAttribute id="zipcode">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[null]</Default>
  <Width>10</Width>
  <HelpText>Enter at least 1 character, but not more than 10 characters.</HelpText>
  <Label>Zipcode</Label>
  <Parser id="TextParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
<MOAttribute id="pic3">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[null]</Default>
  <Width>4</Width>
  <HelpText>Enter a PIC value as four numeric characters, NPIC, or NONE.</HelpText>
  <Label>Pic3</Label>
  <Parser id="PicParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
<MOAttribute id="pic2">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[null]</Default>
  <Width>4</Width>
  <HelpText>Enter a PIC value as four numeric characters, NPIC, or NONE.</HelpText>
  <Label>Pic2</Label>
  <Parser id="PicParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
<MOAttribute id="privacy">
  <Required>>false</Required>
  <Type>single</Type>
  <Default>[NONE]</Default>
  <Width>4</Width>
  <HelpText>Enter one of the following values: [FULL, NAME, NONE]</HelpText>
  <Label>Privacy</Label>
  <Permitted>[FULL, NAME, NONE]</Permitted>
</MOAttribute>
<MOAttribute id="pic1">
  <Required>>false</Required>
  <Type>text</Type>
  <Default>[null]</Default>
  <Width>4</Width>
  <HelpText>Enter a PIC value as four numeric characters, NPIC, or NONE.</HelpText>
  <Label>Pic1</Label>
  <Parser id="PicParser">
    <JavaScript>TBD</JavaScript>
    <RegExp>TBD</RegExp>
  </Parser>
</MOAttribute>
<MOAttribute id="state">
  <Required>>false</Required>

```

```
<Type>text</Type>
<Default>[null]</Default>
<Width>16</Width>
<HelpText>Enter at least 1 character, but not more than 16 characters.</HelpText>
<Label>State</Label>
<Parser id="TextParser">
  <JavaScript>TBD</JavaScript>
  <RegExp>TBD</RegExp>
</Parser>
</MOAttribute>
</ManagedObject>
```







## XML テスト ドライバ

---

この付録では、XML テスト ドライバについて説明します。

### XML 要求バッチ ファイル

次のサンプル XML/CORBA インターフェイス テスト ドライバは、CLI スクリプトとして作成された XML 要求バッチ ファイルを実行します。これにより、システムのプロビジョニングに使用される古い CLI スクリプトとの互換性を維持できます。

```
package com.sswitch.oam.drv;

import java.lang.*;
import java.io.*;
import java.util.*;
import java.text.*;
// XML Stuff
import org.apache.ecs.xml.*;
import org.apache.ecs.*;
import org.w3c.dom.*;
import org.xml.sax.*;
import org.apache.xml.serialize.*;
// BTS Interface Code objects...
import com.sswitch.oam.cad.*;
import com.sswitch.oam.xml.*;
import com.sswitch.oam.util.*;
import com.sswitch.oam.ccc.*;

/**
 * XmlBatch.java
 * Copyright (c) 2002, 2003 by Cisco Systems, Inc.
 * --Test driver for the XML/CORBA interface...
 * This test driver executes a batch file built as a CLI script as XML
 * requests. This allows compatibility with the older CLI scripts used to
 * provision the system. Note that this example can be built with the
 * provided tool "oo-cc" this simple script creates the correct CLASSPATH
 * and invokes the compiler with the correct options. Also, the "oo-idl"
 * tool can be used to generate the correct IDL output.
 *
 * @author A. J. Blanchard
 * @version 3.0
 */
```

```

public class XmlBatch {

    /*
     * Class private data
     */
    private String []                objArgs;
    private CorbaXmlIntf            objBts;

    private File                    objFile;
    private RandomAccessFile        objFileHandle;

    /**
     * Generic Constructor for the test driver.
     */
    protected XmlBatch(String[] args)
    {
        // Initialize the ORB.
        objArgs = args;
        objBts = new CorbaXmlIntf(args);
        return;
    }

    /**
     * This is the main method for the application.
     */
    public static void main(String[] args)
    {

        //
        // Verify that the argument match what is expected...
        // oo-run XmlRequest <CLI Request File> \
        //     -ORBopenorb.config=../OpenORB.xml"
        //
        if(args.length < 2)
        {
            System.out.println("\nStart program as: oo-run XmlRequest <XML Request File>
-ORBopenorb.config=\\\"../OpenORB.xml\\\"\\n");
            System.exit(0);
        }
        XmlBatch me = new XmlBatch(args);
        me.go();
        return;
    }

    /**
     * This is the primary execution method for the object. It performs the
     * actual request and calls for the print of the reply.
     */
    protected void go()
    {
        //
        // Log into the target machine with generic optiuser
        //
        try {
            objBts.connect();
            System.out.println("BTS10200 Login successful...");
        }
        catch (Exception e) {
            System.out.println("Exception in login = " + e);
            System.exit(1);
        }
        //
        // Read in the file and send request...
        //
        try {
            String reply = "";

```

```

openCLI();

while(true)
{
    CommandParser parser = new CommandParser();
    String cmd = readCLI(); // Fetch the request file...
    if(cmd==null)
        break;

    if(cmd.startsWith("#") || cmd.length()==0)
        continue;

    else
    {
        // Issue request to BTS 10200
        reply = objBts.request(parser.toXML(cmd));
        System.out.println("RETURN VALUE: ");
        parser.prettyPrint(reply);
    }
} // end while(1)

closeCLI();
// Clean up and logout
objBts.disconnect();
}
catch (CadExceptions ce) {
    System.out.println("CIS Command Exception: CODE="+ce.error_code +
        "\n"+ce.error_string);
    ce.printStackTrace();
}
catch (Exception e) {
    System.out.println("Batch Command Exception = " + e);
    e.printStackTrace();
}
return;
} // end go()

/**
 * Open the input file for reading. This allows the read method to suck
 * a line at a time of the CLI style input.
 */
protected void openCLI()
{
    try {
        objFile=new File(objArgs[1]);
        objFileHandle=new RandomAccessFile(objFile,"r");
    }
    catch(Exception e) {
        // In the event of an error. just bail out of the program
        System.out.println("Error in processing file:\n"+e.toString());
        System.exit(1);
    }
}

/**
 * This is the method that closes and clean up after a file has been
 * processed.
 */
protected void closeCLI() throws java.io.IOException
{
    objFileHandle.close();
}

/**
 * Read in the file provided as the request. Just exit on errors. Don't
 * worry about throwing an error exception.
 */

```

```
protected String      readCLI()
{
    String data=null;

    try {
        if((data = objFileHandle.readLine())!=null)
            data.trim();
    }
    catch (Exception e) {
        System.out.println("Unable to read "+objFile.toString()+
            " with error:\n"+e.toString());
        System.exit(1);
    }
    // May return a valid string or a NULL object...
    return data;
}

//=====
// Tools and utilities...
//=====

} // end XmlRequest
```

## CLI から CORBA XML へのトランザクション

次のサンプル テスト ドライバは通常の CLI コマンドを実行しますが、コマンドを CORBA XML トランザクションとして処理します。

```
package com.sswitch.oam.drv;

import java.lang.*;
import java.io.*;
import java.util.*;
import java.text.*;
// XML Stuff
import org.apache.ecs.xml.*;
import org.apache.ecs.*;
import org.w3c.dom.*;
import org.xml.sax.*;
import org.apache.xml.serialize.*;
// BTS Utility Code objects...
import com.sswitch.oam.cad.*;
import com.sswitch.oam.xml.*;
import com.sswitch.oam.util.*;
import com.sswitch.oam.ccc.*;

/**
 * XmlCli.java
 * Copyright (c) 2002, 2003 by Cisco Systems, Inc.
 * --Test driver for the XML/CORBA interface...
 * This test driver executes a normal CLI command and processes the request
 * as a CORBA XML transaction. The reply is then displayed. I am no as
 * concerned with complex data show(s) as with the ability to issue
 * provisioning commands. Note that this exmample can be built with the
 * provided tool "oo-cc" this simple script creates the correct CLASSPATH
 * and invokes the compiler with the correct options. Also, the "oo-idl"
 * tool can be used to generate the correct IDL output.
 *
 * @author A. J. Blanchard
 * @version 3.0
 */

public class XmlCli {

    /**
     * Class private data
     */
    private String [] objArgs;
    private CorbaXmlIntf objBts;

    /**
     * Generic Constructor for the test driver.
     */
    protected XmlCli(String[] args)
    {
        // Initialize the BTS ORB interface object.
        objArgs = args;
        objBts = new CorbaXmlIntf(args);
        return;
    }

    /**
     * This is the main method for the application.
     */
    public static void main(String[] args)
    {
        //
        // Verify that the argument match what is expected...
    }
}
```

```

// oo-run XmlCli -ORBopenorb.config="./OpenORB.xml"
//
if(args.length < 1)
{
    System.out.println("\nStart program as: oo-run XmlCli
-ORBopenorb.config=\"./OpenORB.xml\" \n");
    System.exit(0);
}
XmlCli me = new XmlCli(args);
me.go();
return;
}

/**
 * This is the primary execution method for the object. It performs the
 * actual request and calls for the print of the reply.
 */
protected void go()
{
    //
    // Log into the target machine with generic optiuser
    //
    try {
        objBts.connect();
        System.out.println("BTS10200 Login successful...");
    }
    catch (Exception e) {
        System.out.println("Exception in login = " + e);
        System.exit(1);
    }
    //
    // Read in the file and send request...
    //
    try {

        while(true)
        {
            openCLI(); // Put out the prompt...
            CommandParser parser = new CommandParser();
            String cmd = readCLI().trim(); // Fetch the request file...
            String reply = "";
            if((cmd.equals(""))
                continue;
            if(cmd.equals("exit"))
                break;
            else
            {
                // Issue request to BTS 10200
                reply = objBts.request(parser.toXML(cmd));
                System.out.println("RETURN VALUE: ");
                parser.prettyPrint(reply);
            }

            } // end while(1)

        closeCLI();
        // Clean up and logout
        objBts.disconnect();
    }
    catch (CdbExceptions ce) {
        System.out.println("CIS Command Exception: CODE="+ce.error_code +
            "\n"+ce.error_string);
        ce.printStackTrace();
    }
    catch (Exception e) {
        System.out.println("CIS Command Exception: \n"+e.toString());
        //e.printStackTrace();
    }
}

```

```
    }
    return;
} // end go()

/**
 * Open the input file for reading. This allows the read method to suck
 * a line at a time of the CLI style input.
 */
protected void      openCLI()
{
    System.out.print("CORBA-CLI> ");
    return;
}

/**
 * This is the method that closes and clean up after a file has been
 * processed.
 */
protected void      closeCLI() throws java.io.IOException
{
    System.out.println("\n Bye...");
    return;
}

/**
 * Read in the file provided as the request. Just exit on errors. Don't
 * worry about throwing an error exception.
 */
protected String    readCLI() throws java.io.IOException
{
    int      temp=0;
    int      idx=0;
    byte []  buf= new byte[256];

    while(true)
    {
        temp = System.in.read();
        if(temp==10)          // <ENTER Key>
            break;
        buf[idx++]=(byte) temp;
    }
    return new String(buf);
}

//=====
// Tools and utilities...
//=====

} // end XmlCli
```







## CORBA クライアント パッケージ (BTSxsdk) のサンプル実装

この付録では、CORBA クライアント パッケージのサンプル実装 (BTSxsdk) について詳しく説明します。BTSxsdk は、CORBA アダプタを通じて Cisco BTS 10200 ソフトスイッチを接続するためのサンプル CORBA クライアントを実装した Software Development Kit (SDK; ソフトウェア開発キット) です。BTSxsdk パッケージは、CORBA アダプタ (BTScis パッケージ) をインストールすると Element Management System (EMS; 要素管理システム) に自動的にインストールされます。この SDK は、独自のクライアント プログラムを開発するための出発点として使用してください。この SDK が正常に動作すれば、CORBA アダプタが EMS に正しくインストールされていることの確認にもなります。

### BTSxsdk Java の機能

この項では、BTSxsdk の機能のいくつかについて説明します。

### SDK

このソフトウェア開発キット (SDK) には次の機能が含まれています。

- ポート 684 を使用して、Cognitronics Privacy Manager アプリケーションのためのセキュアおよびノンセキュアな CORBA インターフェイス アクセスを提供します。
- Cisco BTS 10200 ソフトスイッチ CORBA インターフェイスは、Cisco BTS 10200 ソフトスイッチ CORBA インターフェイスへのノンセキュア アクセスとセキュア アクセスの両方をサポートします。
- お客様の OSS アプリケーションへの単一の Cisco BTS 10200 ソフトスイッチ CORBA インターフェイスで、OSS からの要求を 2 つの冗長 Cisco BTS 10200 ソフトスイッチ システムにリダイレクトできます。
- Cisco BTS 10200 ソフトスイッチ システム内のアクティブな EMS への CORBA アクセスを提供します。Cisco BTS 10200 ソフトスイッチ CORBA インターフェイスは、アクティブな EMS とのみバインドします。
- Cisco BTS 10200 ソフトスイッチのインストール中に、事前定義の NIC インターフェイスは不要です。以前の Cisco BTS 10200 ソフトスイッチ インストールでは、デュアル NIC インターフェイス構成でオペレータが NIC アドレスを指定する必要がありました。Cisco BTS 10200 ソフトスイッチ CORBA インターフェイスは、デュアル NIC 構成の場合、VIP または NIC アドレスに動的にバインドするようになりました。
- すべての Cisco BTS 10200 ソフトスイッチ システムに対する単一の管理インターフェイスを提供します。

## Java

Java クラスは次の機能を提供します。

- 複数の BTS サイトへの CLI コードを使用したアクセスをサポートします。
- 冗長な EMS ノードを隠します (ACTIVE アクセスのみ)。
- 各 EMS ノード上の冗長 NIC (複数可) を隠します。
- 各 EMS 上の CIS アプリケーションの状態を追跡します。
- Cisco BTS 10200 ソフトスイッチ クライアント アプリケーションへの単一の IOR 参照セットを提供します。

これらの機能のほとんどは、Cisco BTS 10200 ソフトスイッチへのアクセスを抽象化するための一連のクラスで提供されます。これにより、オブジェクトの管理と Cisco BTS 10200 ソフトスイッチへの接続が単純化されます。

## CORBA インターフェイス サーバント

### 拡張機能

次の拡張機能が CORBA インターフェイス サーバント (CIS) アプリケーションに追加されました。これらのアプリケーションは、動作中の CIS アプリケーション エンジンで適用されます。

- 個々の IP アドレスを各 EMS 上のネーム サービス (複数可) にバインドします。
- EM01 がスタンバイの場合はアンバインドし、シャットダウンまたはフェールオーバーします。
- 要求がアクティブな EM01 内にはない場合は、要求の処理を停止します。

### デュアル モード動作

デュアル モード動作が適用されます。ノンセキュアおよびセキュア プロビジョニングは、アクティブな EMS に対してのみ作用します。障害または手動要求によるスイッチオーバーが発生すると、オブジェクトはアンバインドされ、現在の接続は切断されます。結果として、スタンバイ側へのクエリーが発生すると Java 例外が生成されます。

## 前提条件

BTSxsdk パッケージは、CORBA アダプタをインストールすると EMS に自動的にインストールされます。BTSxsdk を別のマシンにインストールする場合、BTSxsdk パッケージを実装するには次の前提条件があります。

- BTSxsdk パッケージは、CORBA アダプタがインストールされた EMS と同じイントラネット上にある UNIX マシンにインストールする。
- Java 開発キット (JDK; Java Development Kit) 1.4.1\_01 以降がインストールされている。このマニュアルは、JDK が /usr/java にインストールされることを前提としています。

ユーザは、Cisco BTS 10200 ソフトスイッチ コマンドライン インターフェイス (CLI) アダプタのコマンドに関する知識を持っている必要があります。

## OpenORB の設定

この OpenORB パッケージは、JDK に対する修正を必要とします。このパッケージは、JDK のプロパティを更新して、OpenORB の CORBA インターフェイス オブジェクトの実装をポイントできるようにします。JDK を OpenORB 用に修正するには、次の 2 つの方法があります。

1. Cisco BTS 10200 ソフトスイッチ パッケージ BTSoorb によって、インストール中に自動的に JDK が修正されます。
2. 次のコマンドをこの順序で入力します。
  - `cd /opt/BTSxsdk/oorb/lib`
  - `java -jar openorb-1.3.1.jar`

## BTSxsdk のビルド

BTSxsdk パッケージは、Java でプレビルドされています。サンプル ファイルを修正しない場合、ビルドは不要です。必要に応じて、次の手順を実行してサンプル クライアント アプリケーションをビルドします。

---

**ステップ 1** BTSxsdk ディレクトリに移動します。

```
cd /opt/BTSxsdk
```

**ステップ 2** エクスポートを実行します。

```
export PROJECTDIR=`pwd`
```

**ステップ 3** エクスポートを実行します。

```
export JAVA_HOME=/usr/java
```

**ステップ 4** `make all` を実行します。

```
./bin/make all
```

---

## BTSxsdk の実行

サンプル CLI クライアント アプリケーションを実行するには、次の手順を実行します。

---

**ステップ 1** bin ディレクトリに移動します。

```
cd /opt/BTSxsdk/bin
```

**ステップ 2** 次のコマンドを発行して BTSxsdk を実行します。

**a.** ノンセキュアな CORBA サーバに接続するには、次のコマンドを入力します。

```
./bts-cli <Active EMS IP> -n <username> -p <password>
```

「Active EMS IP」には、次のいずれかを指定できます。

- アクティブな EMS のいずれかの管理 IP アドレス
- VIP
- アクティブな EMS の DNS 名

**b.** セキュアな CORBA サーバに接続するには、次のコマンドを入力します。

```
./bts-cli-secure <Active EMS Mgmt IP> -n <username> -p <password>
```

「Active EMS Mgmt IP」は、アクティブな EMS のいずれかの管理 IP アドレスにする必要があります。

CLI に類似したインターフェイスが表示されます。show sub; limit=1 などのテスト CLI コマンドを発行します。画面に XML 応答が表示されます。

---



## INDEX

- A**
- Apache XML 2-1
- B**
- Bts10200
    - API 1-11
    - login 1-11
    - logout 1-11
    - オブジェクト 1-10
    - セキュリティ API 1-11
    - プロビジョニング API 1-12
      - getCommandDoc 1-12
      - request 1-12
  - Bts10200\_Security オブジェクト 1-10
  - BTSxsdk
    - OpenORB の設定 C-3
    - 実行 C-4
    - 前提条件 C-2
    - ビルド C-3
- C**
- CadException
    - オブジェクト 1-10
    - マクロ 1-10
  - cadexception
    - CIS Error 1-18, 5-1
    - CIS No Data 1-18, 5-1
    - Class Not Found Exception 1-19, 5-3
    - Exception In Intializer Error 1-19, 5-2
    - Illegal Access Exception 1-19, 5-3
    - Instantiation Exception 1-19, 5-3
    - Invalid Key Exception 1-20, 5-4
    - Invalid Noun Exception 1-19, 5-3
    - Invalid Request Exception 1-19, 5-3
    - Invalid Value Exception 1-20, 5-4
    - Invalid Verb Exception 1-20, 5-4
    - Linkage Error 1-19, 5-2
    - Missing Parameter Exception 1-20, 5-4
    - No Error 1-18, 5-1
    - Permission Error 1-18, 5-2
    - Security Exception 1-19, 5-3
    - SQL Exception 1-20, 5-4
    - User Security Error 1-18, 5-2
  - cadexception、基本的な 1-18, 5-1
  - CIS
    - CIS Error、cadexception 1-18, 5-1
    - CIS No Data、cadexception 1-18, 5-1
    - cis-install.sh 3-4
    - IDL インターフェイスの実装 1-16
    - SDK
      - Java クラス C-2
      - 拡張機能 C-2
      - デュアル モード動作 C-2
      - 機能 2-1
      - パッケージ 3-4
  - Cisco BTS 10200
    - CIS パッケージ 3-4
    - ソフトスイッチの IDL コード 2-7
  - Class Not Found Exception、cadexception 1-19, 5-3
  - CLI から CORBA XML へのトランザクション B-5
  - Col、Reply スキーマの定義 2-6
  - CORBA
    - SDK パッケージ BTSxsdk 3-4
    - Secure Sockets Layer の実装 3-3
    - アーキテクチャとアプリケーション プログラミング インターフェイス 1-1
    - アーキテクチャの図 1-2
    - インターフェイス サーバント アダプタ実装 2-7
    - インターフェイス サーバントと XML 2-1
    - ～と EPOM のトラブルシューティング手順 5-7
    - 特殊文字に対する～と EPOM の応答 5-9
    - ネットワーク設定の変更 5-5
  - CORBA インターフェイス サーバント、「CIS」を参照
  - CORBA クライアント パッケージ

- BTSxsdk の実行 C-4
- BTSxsdk のビルド C-3
- OpenORB の設定 C-3
- 前提条件 C-2
- CORBA ネットワーク設定の変更 5-5
  
- D
- DataTable、Reply スキーマの定義 2-6
- Default、スキーマの定義 2-5
  
- E
- ECS 2-1
- EM\_ACCESS=13 1-19, 5-3
- EM\_DATABASE=19 1-20, 5-4
- EM\_ERROR=1 1-18, 5-1
- EM\_INIT=11 1-19, 5-2
- EM\_INSTANCE=14 1-19, 5-3
- EM\_KEY=21 1-20, 5-4
- EM\_LINKAGE=10 1-19, 5-2
- EM\_NODATA=2 1-18, 5-1
- EM\_NONE=0 1-18, 5-1
- EM\_NOTFOUND=12 1-19, 5-3
- EM\_NOUN=17 1-19, 5-3
- EM\_PARAM=22 1-20, 5-4
- EM\_PERMISSION=6 1-18, 5-2
- EM\_REQUEST=16 1-19, 5-3
- EM\_SECURITY=15 1-19, 5-3
- EM\_USERSEC=5 1-18, 5-2
- EM\_VALUE=20 1-20, 5-4
- EM\_VERB=18 1-20, 5-4
- Entry 要素 2-6
- EPOM
  - 共存性 5-7
  - トラブルシューティング 5-7
- Exception In Intializer Error、cadexception 1-19, 5-2
- Extensible Markup Language の処理 2-1
  
- H
- HelpText、スキーマの定義 2-5
  
- I
- id、スキーマの説明 2-5
- IDL
  - インターフェイス定義 1-16
  - コンパイルおよび POA 検索に必要な Java コードのスクリプト例 1-4
- Illegal Access Exception、cadexception 1-19, 5-3
- Instantiation Exception、cadexception 1-19, 5-3
- Internet Inter-ORB Protocol ( IIOP ) over SSL ( SSLIOP ) 3-1
- Invalid Key Exception、cadexception 1-20, 5-4
- Invalid Noun Exception、cadexception 1-19, 5-3
- Invalid Request Exception、cadexception 1-19, 5-3
- Invalid Value Exception、cadexception 1-20, 5-4
- Invalid Verb Exception、cadexception 1-20, 5-4
  
- J
- Java の SSL 実装 3-3
- JavaScript、スキーマの定義 2-5
  
- K
- Key
  - Request スキーマの定義 2-6
  - マクロ コマンド 1-16
  
- L
- Linkage Error、cadexception 1-19, 5-2
- login 1-11
- logout 1-11
  
- M
- ManagedObject 2-5
- Missing Parameter Exception、cadexception 1-20, 5-4
- MOAttribute、スキーマの定義 2-5
  
- N
- NameService 1-9
- No Error、cadexception 1-18, 5-1
- Noun、Request スキーマの定義 2-6

- O**
- ORB の仕様 1-2
  - ORB の展開
    - クライアントのビルドとバインド セマンティック 1-3
    - 図 1-3
- P**
- Parser、スキーマの定義 2-5
  - Permission Error、cadexception 1-18, 5-2
  - Permitted、スキーマの定義 2-5
- R**
- Reason、Reply スキーマの定義 2-6
  - Reply 1-16, 2-6
  - Request 1-16, 2-6
  - Required、スキーマの定義 2-5
  - Row、Reply スキーマの定義 2-6
- S**
- SAS プロトコル 3-1
  - SDK パッケージ BTSxsdk 3-4
  - Secure Sockets Layer のサポート 3-1
  - Security Attribute Service ( SAS ) 3-1
  - Security Exception、cadexception 1-19, 5-3
  - show コマンド、CLI、データの一括取得 ( ページング ) のための viii
  - SQL Exception、cadexception 1-20, 5-4
  - SSLIOP 3-4
  - Status、Reply スキーマの定義 2-6
  - subscriber noun と add verb、XML 記述ドキュメント A-1
- T**
- Type、スキーマの定義 2-5
- U**
- User Security Error、cadexception 1-18, 5-2
- V**
- Value、Request スキーマの定義 2-6
  - Verb
    - Request スキーマの定義 2-6
    - スキーマの定義 2-5
- W**
- Width、スキーマの定義 2-5
- X**
- Xerces 2-1
  - XML
    - CLI から CORBA XML へのトランザクション B-5
    - CORBA インターフェイス サーバント 2-1
    - CORBA インターフェイス サーバント アダプタ実装 2-7
    - CORBA/XML インターフェイス 1-15
    - IDL コード 2-7
    - Request ドキュメント 1-16
    - subscriber noun と add verb A-1
    - インターフェイス viii
    - 外部キー リレーション A-8
    - 記述ドキュメント A-1
    - 基本概念およびコンポーネント 2-1
    - テスト ドライバ B-1
- い**
- 依存関係、セキュリティ拡張機能 3-3
  - 一括取得、データの ( ページング ) viii
  - インターフェイス viii
- か**
- 外部キー リレーション A-8
- き**
- キー**
- パスワード 3-4

- キー ストア
  - ディレクトリ構造 3-4
  - ～とトラスト ストアの命名 3-4
- 共存性、CORBA と EPOM 5-7
- く
- クライアント実装のスク립ト例 1-6
- こ
- このマニュアルについて vii
- コンパイラ ツール 1-3
- し
- システム セキュリティ拡張のシステム コンテキスト 3-3
- 縮小 Solaris イメージ 3-3
- 証明書
  - 公開鍵を使用した自己署名～ 3-4
  - ディレクトリ構造 3-4
  - 場所 3-4
- 証明書およびキー パスワード 3-4
- す
- スキーマ
  - ManagedObject 2-5
  - ManagedObject の～ 2-2
  - Reply 2-6
  - Reply の～ 2-4
  - Request 2-6
  - Request の～ 2-3
  - 定義
    - Col 2-6
    - DataTable 2-6
    - Default 2-5
    - HelpText 2-5
    - id 2-5
    - Key 2-6
    - MOAttribute 2-5
    - Noun 2-6
    - Parser 2-5
      - JavaScript 2-5
    - RegExp 2-5
    - Permitted 2-5
    - Reason 2-6
    - Required 2-5
    - Row 2-6
    - Status 2-6
    - Type 2-5
    - Value 2-6
    - Verb 2-5, 2-6
    - Width 2-5
- せ
- セキュリティ拡張機能
  - 依存関係 3-3
- セッション、接続とトランザクション ix
- 接続
  - XML インターフェイス viii
  - 許容される数 ix
  - セッション ix
  - 同時トランザクション ix
  - ～とトランザクション ix
- そ
- 操作、マクロ コマンド 1-17
- た
- 対象読者 viii
- て
- ディレクトリ構造 3-4
- データ、一括取得 (ページング) viii
- と
- 特殊文字
  - CORBA と EPOM の応答 5-9
- トラスト ストア
  - ディレクトリ構造 3-4
  - ～とキー ストアの命名 3-4
- トラブルシューティング 5-1
  - cadexception 5-1



- Cisco BTS 10200 ソフトスイッチの CORBA サービスの設定 5-7
  - CORBA が Cisco BTS 10200 ソフトスイッチに接続できない 5-6
  - CORBA と EPOM
    - 基本的な手順 5-7
  - CORBA ネットワーク設定の変更 5-5
  - トランザクション ix
  - トランザクション、接続とトランザクション ix
- れ
- 例外
    - login 1-11
    - logout 1-11
- ろ
- ログイン、接続とトランザクション ix
- ふ
- プロキシ 4-1
- へ
- ページング、データの一括取得 viii
- ま
- マクロ コマンド 1-13
    - CAD インターフェイス 1-16
      - Reply の形式 1-17
      - Request の形式 1-16
    - CORBA/XML インターフェイス 1-15
    - 外部インターフェイス 1-16
    - 操作 1-17
    - 動作とアトリビュート 1-13
    - パラメータ リスト 1-15
    - マクロ ID 1-14
    - マクロ コマンドの管理 1-13
    - マクロの定義 1-14
      - ルール 1-15
  - マクロ コマンド、ユーザ定義 1-17
  - マクロ コマンド定義 1-15
- も
- 戻り値
- login 1-11
  - logout 1-11
- ゆ
- ユーザ定義のマクロ コマンド 1-17