



Java Desktop System Configuration Manager, Release 1.1 インストールガイド

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819-0960-10

Copyright 2004 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

本製品およびそれに関連する文書は著作権法により保護されており、その使用、複製、頒布および逆コンパイルを制限するライセンスのもとにおいて頒布されます。サン・マイクロシステムズ株式会社による事前の許可なく、本製品および関連する文書のいかなる部分も、いかなる方法によっても複製することが禁じられています。フォント技術を含む第三者のソフトウェアは、著作権により保護されており、提供者からライセンスを受けているものです。

本製品の一部は、カリフォルニア大学からライセンスされている Berkeley BSD システムに基づいていることがあります。UNIX は、X/Open Company, Ltd. が独占的にライセンスしている米国ならびに他の国における登録商標です。

Sun, Sun Microsystems, Sun のロゴ、docs.sun.com、AnswerBook、AnswerBook2、および Solaris は、米国およびその他の国における米国 Sun Microsystems, Inc. (以降、米国 Sun Microsystems 社とします) の商標もしくは登録商標です。すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャに基づくものです。

OPEN LOOK および Sun™ Graphical User Interface は、米国 Sun Microsystems 社が独自のユーザーおよびライセンス実施件者向けに開発しました。米国 Sun Microsystems 社は、コンピュータ産業用のビジュアルまたはグラフィカルユーザーインターフェースの概念の研究開発における米国 Xerox 社の先駆者としての成果を認めるものです。米国 Sun Microsystems 社は米国 Xerox 社から Xerox Graphical User Interface の非独占的ライセンスを取得しており、このライセンスは米国 Sun Microsystems 社のライセンス実施権者にも適用されます。

U.S. Government Rights – Commercial software. 政府関係のユーザーは Sun Microsystems 社の標準ライセンス契約および FAR の適用可能な規約条項と補足条項に従うものとします。

本書は、「現状のまま」をベースとして提供されており、商品性、特定目的に対する適合性または第三者の権利の非侵害の暗黙の保証を含み、それに限定されない、明示的であるか暗黙的であるかを問わず、なんらの保証も行われぬものとします。

原典: *Java Desktop System Configuration Manager Release 1.1 Installation Guide*

Part No: 817-7575-10

Revision A



041129@10536



目次

- 1 概要 9
 - 概要 9

- 2 **LDAP** サーバー 11
 - 概念 11
 - セットアップ 12
 - 展開ツール 12
 - スキーマ拡張 12
 - 組織のマッピング 13
 - ユーザープロファイルのマッピング 14
 - 配備 15
 - その他の考慮事項 16

- 3 **Sun Web Console** 17
 - システム要件 17
 - クライアント 18
 - サーバー 18
 - Sun Web Console のインストール 18
 - ▼ Sun Web Console をインストールする 18
 - コンソールの実行 19
 - Sun Web Console のアンインストール 20
 - Sun Web Console ポート情報 20

- 4 **Java Desktop System Configuration Manager, Release 1.1** 23
 - Configuration Manager のインストール 23

▼ Configuration Manager をインストールする	23
Configuration Manager の実行	24
▼ Configuration Manager を起動する	24
Configuration Manager のアンインストール	25
5 Linux および Solaris へのデスクトップコンポーネントのインストール	27
Configuration Agent	27
ブートストラップ情報	28
ポートの設定	31
変更検出間隔	31
操作設定	32
エージェント設定の適用	35
データアクセス/ユーザー認証	36
GConf アダプタ	37
Java Preferences アダプタ	37
Mozilla アダプタ	37
StarSuite アダプタ	37
6 Microsoft Windows へのデスクトップコンポーネントのインストール	39
Configuration Agent のインストール	39
「ようこそ」	40
使用許諾契約	41
インストールの種類	41
インストール準備完了	49
インストールの概要	49
Configuration Agent の設定の変更	50
Configuration Agent のアンインストール	51
A Sun Web Console のパッケージ	53
既知の問題点	53
セキュリティ	53
Setup スクリプトの使用法	53
Sun Web Console のパッケージ	54
Solaris のパッケージ	54
Linux RPM	54

B	Configuration Manager のパッケージ	57
	Configuration Manager のパッケージ	57
	Solaris のパッケージ	57
	Linux RPM	57
C	Configuration Manager による OpenLDAP と Active Directory の使用	59
	Configuration Manager による OpenLDAP サーバーの使用	59
	Configuration Manager による Active Directory サーバーの使用	60

目次

図 5-1	Configuration Agent、設定リポジトリ	28
図 5-2	Configuration Agent、認証機構	29
図 5-3	Configuration Agent、ポートの設定	31
図 5-4	Configuration Agent、データディレクトリ	32
図 5-5	Configuration Agent、要求の処理とロギング	33
図 5-6	Configuration Agent、設定の要約ページ	35
図 6-1	Configuration Agent、「ようこそ」ページ	40
図 6-2	Configuration Agent、「ソフトウェアライセンス条項」	41
図 6-3	Configuration Agent、インストールの種類	41
図 6-4	Configuration Agent、設定リポジトリ	43
図 6-5	Configuration Agent、認証機構	44
図 6-6	Configuration Agent、ポートの設定	45
図 6-7	Configuration Agent、データディレクトリ	46
図 6-8	Configuration Agent、要求の処理とロギング	47
図 6-9	Configuration Agent、設定の要約ページ	48
図 6-10	Configuration Agent、「インストール準備完了」ページ	49
図 6-11	Configuration Agent、「インストールの概要」ページ	49

第 1 章

概要

Java™ Desktop System Configuration Manager Release 1.1 について簡単に説明します。

概要

Java™ Desktop System Configuration Manager Release 1.1 の目的は、Java™ Desktop System が動作しているデスクトップホストを一括して設定することです。組織構造の様々な要素に合わせて設定を割り当てることができるため、管理者はユーザーまたはホストのグループを簡単に管理できます。主なコンポーネントは次のとおりです。

- 管理対象のユーザーおよびホストの組織的な構成を含む LDAP サーバー。この LDAP サーバーには設定データが格納されます。
- 管理者が組織的な構成の各要素に設定データを指定し、その割り当てを行うことができる Web ベースの管理ツール。
- クライアントホストにインストールされたデスクトップコンポーネント。これらのコンポーネントは、現在ログインしているユーザーの代わりに設定データを取得したり、その設定データを Java Desktop System を構成する様々なアプリケーションが利用できるようにします。

管理ツールは、Sun Web Console 内で動作する Web ベースのアプリケーションです。この管理ツールにより、管理者が LDAP サーバーの組織構成を参照し、その各要素にポリシーを割り当てることができます。このポリシーは、ポリシーテンプレートに従って表示、編集されます。ポリシーテンプレートは、管理ツールにより操作される設定を指定するものです。

デスクトップコンポーネントは Java™ Desktop System Configuration Agent を中心に編成されています。これらのコンポーネントはユーザーの代わりに LDAP サーバーから設定データを取得したり、その設定データを様々な設定システムアダプタが利用できるようにして、ローカルの設定 (アプリケーションやユーザー設定によるデフォルト)

トの設定) をポリシー設定で補完します。現在サポートされる設定システムは、GConf (Gnome デスクトップや Evolution などの Gnome アプリケーションの設定を処理する)、Mozilla™ Preferences、および StarRegistry (StarSuite の設定システム) です。

第 2 章

LDAP サーバー

この章では、Configuration Manager で使用するために LDAP サーバーを設定および配備する方法について説明します。

概念

Java Desktop System Configuration Manager フレームワーク内では、設定データはエンティティに関連付けられています。このエンティティとは、LDAP リポジトリ内のエントリのことであり、会社の組織構造の各要素に対応します。

認識されるエンティティは次のとおりです。

- 組織: 一般的には、階層全体の組織的 (部門、グループ、チーム) または地理的 (大陸、国、サイト) な単位を表します。
- ユーザー: 階層全体のリーフノードを表します。その名前が示すように、通常はユーザーに対応します。
- ドメイン: ネットワーク組織の論理編成単位を表します。
- ホスト: 階層全体のリーフノードを表しますが、ネットワーク上のマシンも参照します。
- 役割: プロパティを表します。通常は、ユーザーの集合に適用される職務 (管理者、サイト管理) により区別されます。

組織とユーザーのエンティティがユーザーツリーを指定するのに対し、ドメインとホストのエンティティはホストツリーを指定します。この 2 つのツリーは独立していますが、フレームワーク内では同様の方法で操作されます。

組織およびドメインのエンティティとほかのエンティティの関係は、リポジトリ内におけるエントリの物理的な位置によって指定されます。つまり、組織およびドメインのエンティティには、ツリー内でこれら 2 つのエンティティの下に位置する任意のエントリを含めることができます。ユーザーまたはホストとその役割の関係は、ユーザーおよびホストのエントリの属性によって指定されます。

エンティティに関連付けられた設定データは、フレームワークによって管理される特殊なエントリに格納されます。これらのエントリは、エントリに関連付けられたサービス名およびサービスコンテナによって識別されます。

セットアップ

既存の LDAP サーバーを Configuration Manager で使用するには、以下を行う必要があります。

- Configuration Manager が設定データを格納するときに使用する独自のオブジェクトクラスと属性をサポートするようにサーバーのスキーマを拡張します。
- Configuration Manager がサポートするエントリと同様に、リポジトリ内のエントリに対するマッピング情報をカスタマイズして、サーバーに格納します。

展開ツール

既存の LDAP サーバーを Configuration Manager で使用するには、インストール CD にある次の展開ツールが必要です。

- 88apoc-registry.ldif: 設定データを格納するために必要なオブジェクトクラスと属性を導入するスキーマファイルです。
- OrganizationMapping: LDAP エントリと Configuration Manager エントリ間のマッピングを記述するデフォルトのプロパティファイルです。
- UserProfileMapping: LDAP ユーザーエントリ属性と Configuration Manager ユーザープロファイル属性間のマッピングを記述するデフォルトのプロパティファイルです。
- createServiceTree: マッピングファイルを LDAP リポジトリに格納するスクリプトです。
- deployApoc: LDAP サーバーのスキーマを拡張し、マッピングファイルを LDAP リポジトリに格納するスクリプトです。

スキーマ拡張

設定データは、そのデータに関連するエントリに接続するエントリツリーに格納されます。エントリツリーによって使用されるオブジェクトのクラスと属性を LDAP サーバーに格納するには、LDAP サーバースキーマの該当するオブジェクトとクラスを追加する必要があります。たとえば、提供されているスキーマ拡張ファイルは Sun Java™ System Directory Server にオブジェクトとクラスを追加する際 LDIF フォーマットを使用します。オブジェクトとクラスをその他の LDAP サーバーに追加するには、そのサーバーで認識される形式を使用する必要があります。

組織のマッピング

LDAP エントリと Configuration Manager エントリ間のマッピングを指定するには、Organization マッピングファイルを編集する必要があります。さまざまなキーに、LDAP リポジトリのレイアウトに一致する値を指定する必要があります。

ユーザーエンティティは、すべてのエンティティで使用されるオブジェクトクラスと、リポジトリ全体で固有の値を持つ必要がある属性によって識別されます。管理アプリケーション内でのユーザーの表示方法に影響する名前の形式を指定できます。また、組織内のユーザーエントリでコンテナエントリを使用する場合に、オプションでそのコンテナエントリを指定できます。次に、キーの名前とその標準値を示します。

```
# すべてのユーザーエントリが使用するオブジェクトクラス
User/ObjectClass=inetorgperson
# ユーザーエントリの値がリポジトリ内で一意である属性
User/UniqueIdAttribute=uid
# ユーザーエントリの組織エントリ内にあるオプションのコンテナ
# 使用しない場合は、この行を削除すること
User/Container=ou=People
# 管理アプリケーションにおける表示名の形式
User/DisplayNameFormat=sn, givenname
```

役割のエンティティは、それが使用する可能性のあるオブジェクトクラスのリストと、対応する名前属性によって識別されます。これらのリストは、形式 `<item1>,<item2>,...,<itemN>` を使用して、整列する必要があります。つまり、リストでは命名属性と同数の項目を含め、 n 番目の命名属性で n 番目のオブジェクトクラスを使用する必要があります。2つのキーにより、役割とユーザーの関係、および役割とホストの関係が指定されます。*VirtualMemberAttribute* キーで指定する属性の値には、ユーザーまたはホストのエントリからクエリーできることが必要です。またこのキーには、エントリが属する役割の完全な DN を含める必要があります。*MemberAttribute* キーには、検索フィルタに適用するユーザーまたはホストのエントリの属性を指定する必要があります。またこのキーには、ユーザーまたはホストが属する役割の完全な DN を含める必要があります。*VirtualMemberAttribute* キーには Class Of Service 仮想属性を指定できるのに対し、*MemberAttribute* キーにはフィルタ内で使用可能な物理属性を指定する必要があります。次に、キーの名前とその標準値を示します。

```
# 役割のオブジェクトクラスのリスト
Role/ObjectClass=nsRoleDefinition
# 対応する命名属性の整列済みリスト
Role/NamingAttribute=cn
# ユーザーまたはホストの役割の DN を含む
# 物理的な属性 (フィルタで使用可能)
Role/MemberAttribute=nsRoleDN
# ユーザーまたはホストへのクエリーによって
# 自分が属する役割の DN が返される属性
Role/VirtualMemberAttribute=nsRole
```

組織のエンティティは、オブジェクトクラスから成る2つの整列済みリストと対応する命名属性により、役割に似た方法で識別されます。次に、キーの名前とその標準値を示します。

```
# 組織のオブジェクトクラスのリスト
Organization/ObjectClass=organization
# 対応する命名属性の整列済みリスト
Organization/NamingAttribute=o
```

ドメインのエントリは、組織のエントリと同様の方法で識別されます。次に、キーの名前とその標準値を示します。

```
# ドメインのオブジェクトクラスのリスト
Domain/ObjectClass=ipNetwork
# 対応する命名属性の整列済みリスト
Domain/NamingAttribute=cn
```

ホストのエントリは、ユーザーのエントリと同様の方法で識別されます。次に、キーの名前とその標準値を示します。

```
# すべてのホストエントリが使用するオブジェクトクラス
Host/ObjectClass=ipHost
# ホストエントリの値がリポジトリ内で一意である属性
Host/UniqueIdAttribute=cn
# ホストエントリのドメインエントリ内にあるオプションのコンテナ
# 使用しない場合は、この行を削除すること
Host/Container=ou=Hosts
```

ユーザープロファイルのマッピング

LDAP ユーザーエントリ属性と Configuration Manager ユーザーエントリ属性間のマッピングを指定するには、User Profile マッピングファイルを編集する必要があります。各キーは Configuration Manager ユーザー属性に対応します。ユーザーエントリ内の属性の名前に、組織のマッピングによって識別された値であるキーを割り当てることができます。User/DisplayNameFormat 設定で使用する属性は、User Profile マッピング内で割り当てることが必要です。次に、キーの名前とその標準値を示します。

```
# inetOrgPerson.givenName
org.openoffice.UserProfile/Data/givenname = givenname
# person.sn
org.openoffice.UserProfile/Data/sn = sn
# inetOrgPerson.initials
org.openoffice.UserProfile/Data/initials = initials
# organizationalPerson.street
org.openoffice.UserProfile/Data/street = street,postalAddress,streetAddress
# organizationalPerson.l (city)
org.openoffice.UserProfile/Data/l = l
# organizationalPerson.st (state)
org.openoffice.UserProfile/Data/st = st
# organizationalPerson.postalCode
org.openoffice.UserProfile/Data/postalcode = postalcode
# country.c (country)
org.openoffice.UserProfile/Data/c =
# organizationalPerson.o (company)
org.openoffice.UserProfile/Data/o = o,organizationName
```

```

# deprecated -- no LDAP corollary
org.openoffice.UserProfile/Data/position =
# organizationalPerson.title
org.openoffice.UserProfile/Data/title = title
# inetOrgPerson.homePhone
org.openoffice.UserProfile/Data/homephone = homephone
# organizationalPerson.telephoneNumber
org.openoffice.UserProfile/Data/telephonenumber = telephonenumber
# organizationalPerson.facsimileTelephoneNumber
org.openoffice.UserProfile/Data/facsimiletelephonenumber =
facsimiletelephonenumber,officeFax
# inetOrgPerson.mail
org.openoffice.UserProfile/Data/mail = mail

```

配備

マッピングファイルは、LDAP リポジトリの状態を反映するようにカスタマイズしたあと配備できます。LDAP サーバーのスキーマが必要なオブジェクトクラスと属性をすでに持っている場合、スクリプト `createServiceTree` を直接実行できます。それ以外の場合は、スクリプト `deployApoc` を実行する必要があります。

`deployApoc` スクリプトは、Sun Java™ System Directory Server で使用することを目的としています。このスクリプトは、提供されたスキーマ拡張ファイルを適切なディレクトリにコピーし、LDAP サーバーを動作させて、`createServiceTree` スクリプトを起動します。スキーマリポジトリ内にファイルをコピーしてサーバーを再起動するアクセス権を持つユーザーとしてこのスクリプトを実行する必要があります。また、このスクリプトの実行では、次の指定が必要になります。

```
./deployApoc <Directory Server Directory>
```

<Directory Server Directory> パラメータは、Directory Server インストールの `slapd-<server name>` サブディレクトリへのパスである必要があります。たとえば、デフォルトのディレクトリを使用して、サーバーの名前が `myserver.mydomain` であるインストールの場合、このディレクトリは `/var/Sun/mps/slapd-myserver.mydomain` になります。

`createServiceTree` スクリプトは、直接起動された場合も `deployApoc` スクリプトから起動された場合も、LDAP サーバーの位置 (ホスト名、ポート番号、およびベース DN) と管理権を持つユーザーの定義 (完全な DN とパスワード) をユーザーに確認します。次に、このスクリプトは LDAP サーバーにブートストラップサービスツリーを作成して、このツリーにマッピングファイルを格納します。このスクリプトは、任意のユーザーとして実行可能であり、次によって起動されます。

```
./createServiceTree
```

次に、以下の情報を入力するようにユーザーに求めます。

- **Host name** (デフォルト: `localhost`): LDAP サーバーのホスト名
- **Port number** (デフォルト: `389`): LDAP サーバーのポート番号

- **Base DN:** LDAP リポジトリのベース DN
- **User DN (デフォルト: `cn=Directory Manager`):** ベース DN の下に新しいエントリを作成できるアクセス権を持つユーザーの完全な DN
- **Password:** 当該ユーザーのパスワード

次の DN を持つエントリ:

```
ou=ApocRegistry,ou=default,ou=OrganizationConfig,ou=1.0,  
ou=ApocService,ou=services, <baseDN>
```

が 2 つのマッピングファイルの内容で作成されます。

上記のとおり、`deployApoc` スクリプトが実行する操作は、LDAP サーバーのインストールディレクトリ、レイアウト、およびスキーマ拡張プロシージャが Sun Java System Directory Server のものとほとんど一致すると仮定します。

`createServiceTree` スクリプトを実行するためには、そのほかのディレクトリにも手作業でスキーマ拡張を行う必要があります。OpenLDAP と ActiveDirectory を使用する方法の詳細については、付録 C を参照してください。

作成されたツリーは、エンティティに関連する設定データを格納するものと一致し、Sun Java System Identity Server のサービス管理に使用されるツリーの構造に整理されます。

その他の考慮事項

Configuration Manager フレームワークでは、デスクトップからのユーザーまたはホストの識別子に関連する完全な DN を識別するために、LDAP サーバーへの接続を(読み取り/検索権付きで)作成することが必要です。そのため、匿名の接続を許容するようにリポジトリを設定するか、あるいは読み取り/検索権を持つ専用のユーザーを作成する必要があります。

管理アプリケーションは、エンティティの設定データを格納するためのサービスツリーをそのエンティティにマッピングされるエントリの下に作成します。したがって、管理目的で使用されるユーザーエントリには、管理対象となるエントリの下にサブエントリを作成する権限が必要です。

デスクトップクライアントからのユーザーに対するフレームワークの認証は、Anonymous および GSSAPI の 2 通りの方式により実行できます。Anonymous メソッドでは、デスクトップクライアントが LDAP サーバーからのデータの取得を試みたときにアカウント情報を提供しないので、リポジトリ全体で読み取り/検索の匿名アクセスを有効にする必要があります。GSSAPI メソッド(認証に Kerberos を使用)を使用する場合は、LDAP サーバーを『Sun Java™ System Directory Server 5 2004Q2 管理ガイド』の「認証と暗号化の管理」の章で説明しているとおりに設定する必要があります。

第 3 章

Sun™ Web Console

Sun Web Console は、Sun Microsystems 製品用に共通の Web ベースのシステム管理ソリューションを提供する目的で設計されました。これは、ユーザーがシステム管理アプリケーションにアクセス可能な単一の場所として役立ちます。これらのシステム管理アプリケーションは、そのすべてが一貫したユーザーインターフェースを提供します。

このコンソールは、さまざまな理由から Web モデルに基づいています。しかし、その第一の理由はシステム管理者が Web ブラウザを使ってシステム管理アプリケーションにアクセスできるようにすることです。

Sun Web Console が提供する機能は次のとおりです。

- 共通の認証と承認
- 共通のログ
- すべてのシステム管理アプリケーションに対して、同じ HTTPS ベースのポートを経由する単一のエン트리ポイント
- 共通の外観と操作性

このコンソールの主な利点は、管理者が 1 回ログインしたあと、コンソール内でどのアプリケーションでも使用可能であることです。

システム要件

Sun Web Console は、複数のクライアントサーバー型オペレーティングシステムといくつかのブラウザをサポートします。

クライアント

- Solaris™ 8 以降で動作する Netscape™ 6.2x および 7.x
- Windows 98、98 SE、ME、2000、および XP で動作する Netscape 4.7x、6.2x、および 7.x
- Windows 98、98 SE、ME、2000、および XP で動作する Internet Explorer 5.x および 6.x
- Linux および Solaris 上で動作する Mozilla

サーバー

- Solaris 8 以降
- Red Hat 8 以降、または Red Hat Enterprise Linux 2.1
- SuSE Linux 2.1 以降
- J2SE™ バージョン 1.4.1_03 以降
セットアッププログラムによりサーバー上で J2SE 1.4.1 以前のバージョンが検出された場合、Java Desktop System Management Tools CD に収録されている J2SE バージョンを使ってインストールをアップグレードするように求められます。
- Tomcat: 4.0.3 以降
Tomcat は Java Desktop System Management Tools CD に収録されています。

Sun Web Console のインストール

Sun Web Console をインストールする前に、本書の付録 A にあるパッケージの要約と既知の問題点に関する節をお読みください。

Solaris SPARC (バージョン 8 以降) 用と Linux オペレーティングシステム用の Sun Web Console は、Java Desktop System Management Tools CD に収録されています。

▼ Sun Web Console をインストールする

- 手順
1. **Java Desktop System Management Tools CD** 上で、コンソールをインストールするオペレーティングシステム用の **Sun Web Console** ディレクトリ (**swc**) に移動します。
 2. `./setup` を入力します。

デフォルトでは、Sun Web Console はインストールログファイルを作成しません。「logfile」という名前のインストールログファイルを作成するには、`./setup | tee logfile` を入力します。

注 - セットアップを実行すると、Web コンソールのほとんどのインストールと設定は自動的に実行されます。Sun Web Console の設定アプリケーションについての詳細は、[付録 A](#) を参照してください。

3. **Sun Web Console** をローカライズする場合、言語ごとに 2 つの追加パッケージをインストールする必要があります。以下の表で言語に対応するパッケージ名を調べ、次のいずれかの操作を行います。

- Solaris では、`pkgadd -d path/pkgname.pkg pkgname` を入力します。ここで、*pkgname* は追加する言語パッケージの名前です。
- Linux では、`rpm -i path/pkgname<...>.rpm` を入力します。ここで、*pkgname* は追加するパッケージの名前です。

パッケージ名	説明
SUNWcmcon、SUNWcmctg	簡体字中国語の Sun™ Web Console 2.0
SUNWdmcon、SUNWdmctg	ドイツ語の Sun™ Web Console 2.0
SUNWemcon、SUNWemctg	スペイン語の Sun™ Web Console 2.0
SUNWfmcon、SUNWfmctg	フランス語の Sun™ Web Console 2.0
SUNWhmcon、SUNWhmctg	繁体字中国語の Sun™ Web Console 2.0
SUNWimcon、SUNWimctg	イタリア語の Sun™ Web Console 2.0
SUNWjmcon、SUNWjmctg	日本語の Sun™ Web Console 2.0
SUNWkmcon、SUNWkmctg	韓国語の Sun™ Web Console 2.0
SUNWsmcon、SUNWsmctg	スウェーデン語の Sun™ Web Console 2.0

コンソールの実行

新しいアプリケーションを登録する場合、通常、Sun Web Console サーバーを停止して再起動するだけです。



注意 – 初めて Sun Web Console を起動する前に、 Configuration Manager インストールが完了していることを確認してください。コンソールに最低 1 つのアプリケーションを配備するまでは、Sun Web Console は正常に実行されません。

- Sun Web Console を起動するには、 **smcwebserver start** を入力します。
- Sun Web Console を停止するには、 **smcwebserver stop** を入力します。
- Sun Web Console にアクセスするには、次の URL をブラウザに入力します。
`https://<hostname>.<domainname>:6789`

Sun Web Console は簡単に導入でき、UNIX ベースの認証と役割ベースのアクセス制御 (RBAC: Role Based Access Control) をサポートします。ただし、LDAP 認証など、ほかの認証機構も設定できます。

注 – デフォルトのセッションタイムアウトは 15 分です。 **smreg** コマンドを使用すれば、このタイムアウトを設定できます。たとえば、このタイムアウトを 5 分に設定するには、 **smreg add -p -c session.timeout.value=5** を入力します。

Sun Web Console のコマンドの詳細については、 **smcwebserver** と **smreg** のマニュアルページを参照してください。

Sun Web Console のアンインストール

Sun Web Console をアンインストールするには、 **/usr/lib/webconsole/setup -u** を実行します。

注 – **/usr/lib/webconsole** ディレクトリまたはそのサブディレクトリにいる場合は、このコマンドを実行しないでください。実行すると、 **pkgrm** は失敗します。

Sun Web Console ポート情報

Configuration Manager は Sun Web Console のポートを使用します。

- 8005 – サービスのシャットダウン用
- 6789 – https アクセス用

この2つのポートは、`/etc/opt/webconsole/server.xml` で変更できます。ポートを変更したあと、`/usr/sbin/smcwebserver restart` コマンドで Sun Web Console を再起動します。

第 4 章

Java™ Desktop System Configuration Manager, Release 1.1

Configuration Manager は、Sun Web Console 上で動作する管理ツールを提供します。この Web ベースのユーザーインターフェースにより、管理者が組織の階層をナビゲートしてデスクトップアプリケーションのポリシーを指定できます。このポリシーは、組織、役割、ユーザー、ドメイン、ホストなど、階層内の各項目に対して指定できます。Configuration Manager はいくつかの設定テンプレートを使用して、Gnome、Mozilla、StarSuite、および Evolution などの様々なデスクトップアプリケーションに固有な設定を表示します。

Configuration Manager のインストール

Configuration Manager をインストールする前に、Sun Web Console の作業用インストールが必要になります。

▼ Configuration Manager をインストールする

- 手順
1. **Java Desktop System Management Tools CD** 上で、対応する **Configuration Manager** ディレクトリ (**apoc**) に移動します。
 2. **./setup** を入力します。
 3. **LDAP** サーバーのホスト名を入力します。
デフォルト名は **localhost** です。
 4. **LDAP** サーバーのポート番号を入力します。デフォルトのポート番号は **389** です。
 5. **LDAP** リポジトリのベース **DN** を入力します。

6. ユーザーエンティティを識別するためのオブジェクトクラスの名前を入力します。デフォルトのオブジェクトクラスは **inetorgperson** です。
詳細については、「LDAP サーバー」の章の13 ページの「組織のマッピング」を参照してください。
7. **LDAP** レポジトリ全体で固有の属性の名前を入力します。デフォルトの属性は **uid** です。
詳細については、「LDAP サーバー」の章の13 ページの「組織のマッピング」を参照してください。
8. **LDAP** サーバーにクエリーを実行できるアクセス権を持つユーザーの完全な **DN** を入力します。
読み取り/検索権を持つ完全な **DN** を使用します。匿名アクセスを使用する場合は、このフィールドを空欄のままにしておきます。
9. **LDAP** のアクセス権を割り当てたユーザーのパスワードを入力します。
LDAP サーバーへの匿名アクセスをセットアップする場合は、この手順を無視してください。
インストール時に、LDAP 経由でユーザーを認証するためのログインモジュールが Sun Web Console に追加されます。
インストールの終了時、Sun Web Console は自動的に再起動して、Configuration Manager にアクセスできるようになります。

注 - /usr/share/webconsole/apoc/configure スクリプトを使用すれば、以前の Configuration Manager の設定はいつでも変更できます。たとえば、このスクリプトを使用すれば、Configuration Manager を再インストールしなくても、異なる LDAP サーバーに変更できます。

Configuration Manager の実行

▼ Configuration Manager を起動する

- 手順
1. 次の URL をブラウザに入力します。 **https://<hostname>.<domainname>:6789**
 2. 既存の **LDAP** ユーザーのユーザー名 (**uid**) とパスワードをプロンプトに入力します。
Sun Web Console が開きます。

3. コンソールウィンドウで「**Sun Java Desktop System Configuration Manager, Release 1.1**」をクリックします。

参考 Configuration Manager に直接アクセスする

Sun Web Console の起動ページを飛ばして、Configuration Manager に直接アクセスするには、次の URL をブラウザに入力します。

```
https://<hostname>.<domainname>:6789/apoc
```

Configuration Manager のアンインストール

Configuration Manager を Sun Web Console からアンインストールするには、Java Desktop System Management Tools CD で、対応する Configuration Manager ディレクトリに移動して、`./setup -u` を実行します。

注 – Configuration Manager をアンインストールすると、LDAP ログインモジュールも Sun Web Console から削除されます。

第 5 章

Linux および Solaris™ へのデスク トップコンポーネントのインストール

この章では、Solaris および Linux オペレーティングシステムに固有の情報を提供します。

Configuration Manager から設定データにアクセスするためには、デスクトップクライアントに Java™ Desktop System Configuration Agent が必要です。Configuration Agent は、リモートの設定データリポジトリおよびアダプタと通信すると同時に、データを特定の設定システムに統合します。現在サポートされている設定システムは、GConf、Java Preferences、Mozilla Preferences、および StarSuite Registry です。

Configuration Agent

Configuration Agent は、次の表に示す各種のパッケージに含まれています。

Solaris のパッケージ名	Linux の RPM 名	説明
SUNWapbas	apoc-base	設定共有ライブラリ
SUNWapmsc	apoc-misc	Configuration Agent に関する その他のファイル
SUNWapoc	apoc	Configuration Agent
SUNWapdc	apoc-config	Configuration Agent ウィザード

上記のパッケージをインストールすると、この API に必要なファイルもインストールされます。このパッケージは手作業でも Java Desktop System インストール経由でもインストールできます。インストール後は、自分のシステム上で Configuration Agent を設定および有効にする必要があります。

リモートの設定ファイルにアクセスするには、Configuration Agent に最小限のブートストラップ情報 (LDAP サーバーのホスト名やポートなど) が必要です。この情報は、`polycmgr.properties`、`apocd.properties`、`os.properties` などのプロパティファイルの集合として保守されます。これらの属性ファイルはローカルの `/etc/apoc` ディレクトリに格納されます。これらのプロパティファイルは手作業でも、Configuration Agent の設定ウィザードでも編集できます。

この設定ウィザードは、グラフィカルユーザーインターフェースを提供し、これに従って Configuration Agent に必要な設定ができます。このウィザードのすべてのページで、対応するヘルプ画面が利用できます。このウィザードを起動するには、スーパーユーザー (root) として `/usr/bin/apoc-config` スクリプトを実行します。また、対応するデスクトップメニューエントリは、設定/システムの設定/ネットワークの設定、あるいは、Nautilus ファイルマネージャの `system-settings:///Network Settings` より利用できます。

注 - ウィザードはグラフィカルインターフェースを起動しなくても起動できます。たとえば、`/usr/bin/apoc-config -nodisplay` を実行すると、ウィザードはコンソールモードで起動します。

ブートストラップ情報

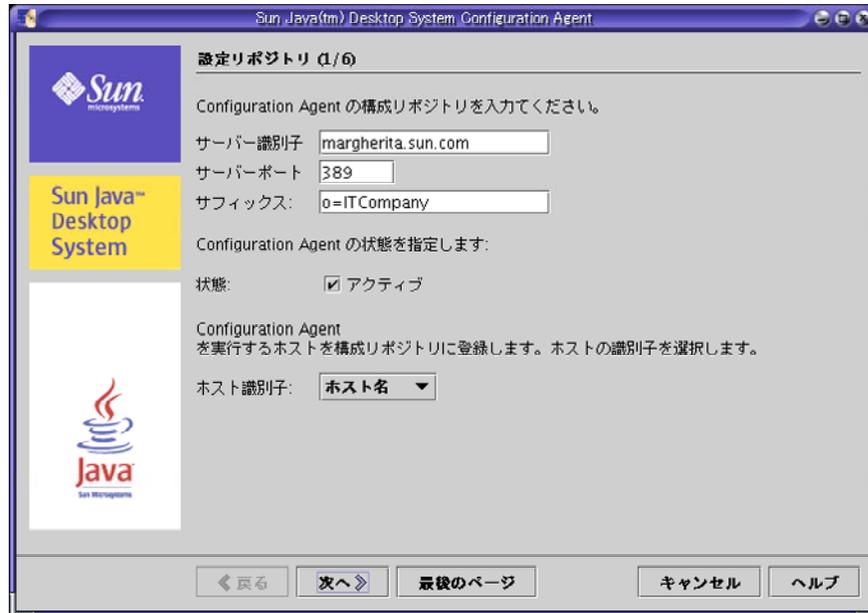


図 5-1 Configuration Agent、設定リポジトリ

注 - 対応するプロパティファイルキーを括弧内に示しています (適切な場合)。

- サーバー識別子 (Server): LDAP サーバーのホスト名。
- サーバーポート (Port): LDAP サーバーのポート番号。
- サフィックス (BaseDn): LDAP リポジトリのベース DN。
- 状態: Configuration Agent の状態。このチェックボックスを使用すると、Configuration Agent をアクティブまたは非アクティブにできます。設定リポジトリを使用するには、Configuration Agent をアクティブにしておく必要があります。アクティブにすると、inetd への必要な登録も自動的に行われます。

注 - Configuration Agent を手作業で有効または無効にするには、**root** としてログインして、それぞれ、コマンド `/usr/lib/apocd enable` または `/usr/lib/apocd disable` を入力します。

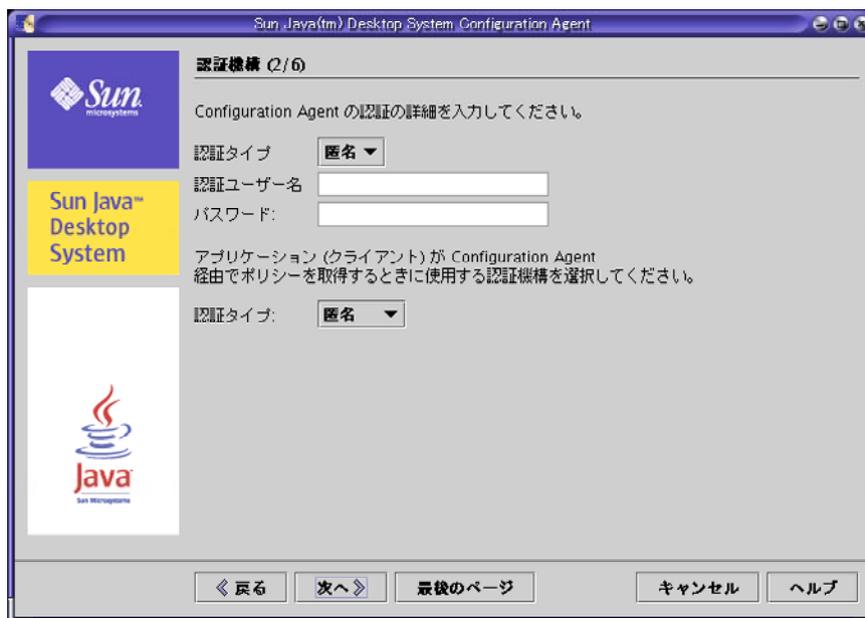


図 5-2 Configuration Agent、認証機構

- ホスト識別子 (HostIdentifier): 「HostName」または「IPAddress」のいずれかを指定できます。この識別子は、ホストを識別するときに使用する LDAP 属性の内容に一致するように設定する必要があります。この属性は、マッピングファイル内では Host/UniqueIdAttribute として指定されます。
- Configuration Agent の 認証タイプ: 「匿名」または「Simple」のいずれかを指定できます。「匿名」を選択した場合、「認証ユーザー名」と「パスワード」フィールドは自動的に無効になります。
- 認証ユーザー名 (AuthDn): リポジトリに対する読み取り権および検索権を持つユーザーの完全な DN。
- パスワード (Password): 登録された LDAP ユーザーのパスワード。

注-ディレクトリ内で匿名アクセスが有効である場合、「認証ユーザー名」と「パスワード」の設定は空白のまま構いません。

- アプリケーションの 認証タイプ (AuthType): LDAP サーバーによるユーザーの認証方法に応じて、「匿名」または「GSSAPI」のいずれかを指定できます。

注 - 詳細については、36 ページの「データアクセス/ユーザー認証」を参照してください。

ポートの設定

Configuration Agent は、次の 2 つのポートを使用します。

- エージェントポート (DaemonPort): エージェントがクライアントアプリケーションと通信するときに使用します。デフォルトは **38900** です。
- 管理ポート (DaemonAdminPort): エージェントコントローラプログラム `apocd` が、エージェントと通信するときに使用します。デフォルトは **38901** です。



図 5-3 Configuration Agent、ポートの設定

変更検出間隔

Configuration Agent は、次の 2 つの間隔を使用して、設定データに変更があるかどうかを定期的にチェックします。

- 一般的な検出間隔 (ChangeDetectionInterval): デスクトップアプリケーション (クライアント) の設定データの変更検出間隔 (分)。

注 --1 を指定すると、変更検出がオフになります。

- エージェント設定用の検出間隔 (DaemonChangeDetectionInterval): エージェントに固有な設定データの変更検出間隔 (分)。

注 --1 を指定すると、変更検出がオフになります。

汎用の検出間隔を使用すると、リモートの設定データの変更をクライアント側のアプリケーションに伝播する間隔を調整できます。この設定で指定する値は、リモートに加えられた変更の内容がクライアントアプリケーションに反映されるまでの最大期間 (分) です。

この値が小さくなるほど、Configuration Agent と LDAP サーバーの活動が増えます。したがって、この設定値を調整する場合は注意が必要です。たとえば、最初の配備段階でこの値を 1 分に設定するれば、クライアントアプリケーションに対するリモート設定の影響を簡単にテストできます。テストが完了したら、この設定を初期値に戻します。

操作設定

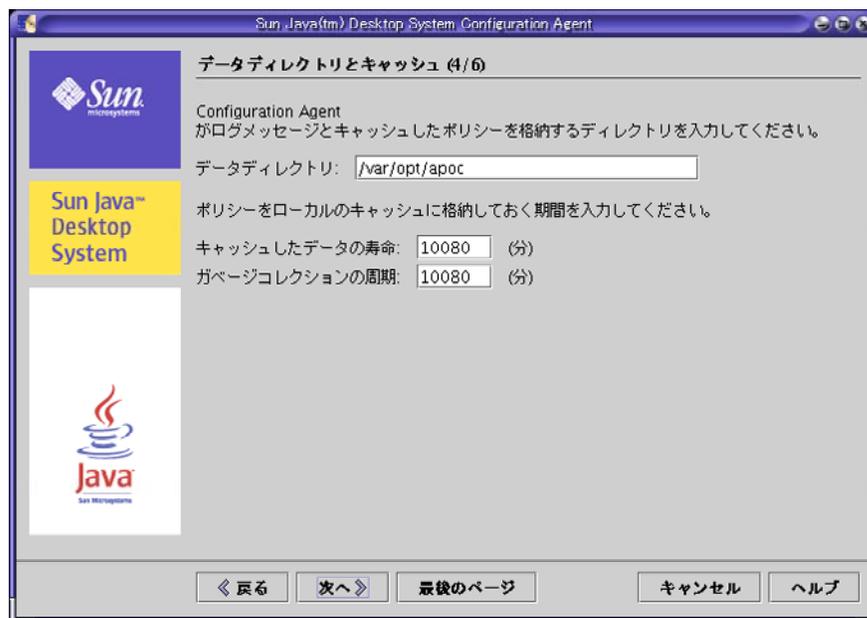


図 5-4 Configuration Agent、データディレクトリ

次の設定が設定できます。

- データディレクトリ (DataDir): 実行時データを格納するために使用されるディレクトリ。デフォルトは `/var/opt/apoc` です。
- キャッシュしたデータの寿命 (TimeToLive): ローカルデータベース内にオフラインでない設定データが留まる間隔 (分)。



図 5-5 Configuration Agent、要求の処理とロギング

- ガベージコレクションの周期 (GarbageCollectionInterval): ローカル設定データベースのガベージコレクション周期 (分)。
- クライアントスレッドの最大数 (MaxClientThreads): 同時に処理できるクライアント要求の最大数。
- クライアント接続の最大数 (MaxClientConnections): クライアント接続の最大数。
- 要求の最大サイズ (MaxRequestSize): クライアント要求の最大サイズ。
- 接続タイムアウト (ConnectTimeout): LDAP サーバーが接続要求に応答できる間隔 (秒)。デフォルトは 1 秒です。
- ログのレベル (LogLevel): エージェントのログファイルの詳細レベル。ロギングレベルは Java Logger のレベルに一致します。次に、これらのレベルを重要度の降順に示します。
 - SEVERE
 - WARNING
 - INFO
 - CONFIG
 - FINE
 - FINER
 - FINEST

注-ほとんどの操作設定(「データディレクトリ」と「接続タイムアウト」の設定を除く)は、LDAP サーバーに格納された対応するポリシー経由で集中的に保守できます。この機能を使用する場合は、対応する設定をウィザードで変更しないでください。代わりに、Configuration Manager 内の Configuration Agent ポリシーを使用して、操作設定を集中的に指定します。

エージェント設定の適用

Configuration Manager を使用して LDAP サーバーに格納した操作設定(「データディレクトリ」と「接続タイムアウト」の設定を除く)は、エージェント設定の次回の変更検出時に自動的に適用されます(DaemonChangeDetectionInterval を参照)。

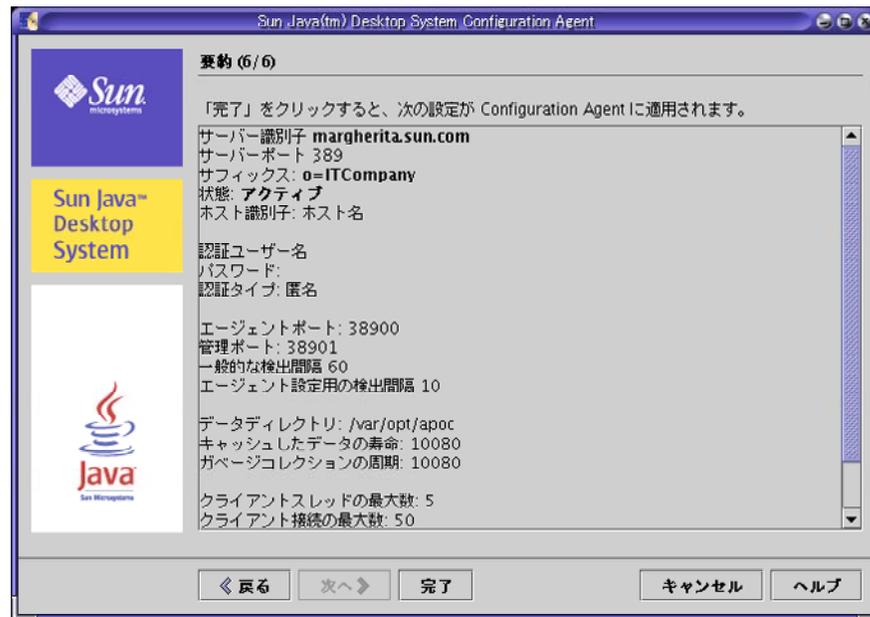


図 5-6 Configuration Agent、設定の要約ページ

ローカルで変更したその他の設定については、Configuration Agent を再読み込みまたは再起動する必要があります。設定ウィザードを使用する場合は、再読み込みまたは再起動は自動的に実行されます。

注 - Configuration Agent を手作業で再起動するには、関連するクライアントアプリケーションが動作していないことを確認し、root としてログインして、コマンド `/usr/lib/apoc/apocd restart` を入力します。

データアクセス/ユーザー認証

Configuration Agent は、デスクトップユーザーのログイン ID に基づいて LDAP サーバーから情報を取得します。組織のマッピングファイルの `User/UniqueIdAttribute` 設定により、ログイン ID と、LDAP サーバー内のユーザーエンティティがマッピングされます。Configuration Agent はまた、ホストについての情報 (ホストの名前と IP アドレスなど) を取得します。この情報は、組織のマッピングファイルの `Host/UniqueIdAttribute` 設定により、LDAP サーバー内のホストエンティティにマッピングされます。

LDAP サーバーにアクセスする場合、匿名または GSSAPI の 2 つの方法があります。匿名アクセスでは、デスクトップ上での処理は必要ありません。GSSAPI 方式では、デスクトップ上で Kerberos 資格情報を取得する必要があります。Kerberos 資格情報とユーザーログインの取得を統合するには、`pam_krb5` モジュールを Java Desktop System ホストにインストールおよび設定する必要があります。

`gdm` を使用して Kerberos とユーザーログインを統合できます。たとえば、次のように `/etc/pam.d/gdm` ファイルを使用します。

```
##PAM-1.0
auth    required    pam_unix2.so  nullok #set_secrcp
auth    optional    pam_krb5.so  use_first_pass missing_keytab_ok ccache=SAFE putenv_direct
account required    pam_unix2.so
password required    pam_unix2.so  #strict=false
session required    pam_unix2.so  # trace or none
session required    pam_devperm.so
session optional    pam_console.so
```

このように Kerberos をユーザーログインに統合する場合は、スクリーンセーバーの Kerberos サポートを有効にするとよいでしょう。たとえば、次のように `/etc/pam.d/xscreensaver` ファイルを使用します。

```
auth required pamkrb5.so use_first_pass missing_keytab_ok
ccache=SAFE putenv_direct
```

GConf アダプタ

GConf アダプタは、Solaris の場合は SUNWapoc-adapter-gconf パッケージの一部であり、Linux の場合は apoc-adapter-gconf RPM の一部です。該当するパッケージまたは RPM からアダプタをインストールすると、`/etc/gconf/2/path` にある GConf データソースパスは Configuration Manager ソースを含むように更新されます。アダプタから提供される 2 つのデータソースは、次のとおりです。

- "apoc:readonly:": ポリシーの非保護設定へのアクセスを可能にします。ユーザー設定のあと、ローカルのデフォルト設定の前にこのデータソースを挿入します。
- "apoc:readonly:mandatory@": ポリシーの保護設定へのアクセスを可能にします。ローカルの必須設定のあと、ユーザー設定の前にこのデータソースを挿入します。

Java Preferences アダプタ

Java Preferences アダプタは、Solaris の場合は SUNWapcj パッケージの一部であり、Linux の場合は apoc-adapter-java RPM の一部です。該当するパッケージまたは RPM からアダプタをインストールすると、Solaris の `/opt/SUNWapcj` ディレクトリまたは Linux の `/opt/apocjava` ディレクトリに必要なファイルが追加されます。

Mozilla アダプタ

Mozilla アダプタは、Solaris の場合は SUNWmozapoc-adapter パッケージの一部であり、Linux の場合は mozilla-apoc-integration RPM の一部です。該当するパッケージまたは RPM からアダプタをインストールすると、必要なファイルが Mozilla の既存インストールに追加され、自動的に登録されます。

StarSuite アダプタ

StarSuite アダプタは標準の StarSuite インストールに含まれています。これにより、特殊な変更を加えることなく、ポリシー設定データにアクセスできます。

第 6 章

Microsoft Windows へのデスクトップ コンポーネントのインストール

集中設定機能に必要なコンポーネントはすべて Java Desktop System に付属しており、Java Desktop System と一緒にインストールされます。一般的に、このような集中設定の機能は、ほかのシステムでも使用できます。ただしその場合、必要なインストール手順を実行する必要があります。Configuration Manager のインストールでは、Microsoft Windows システムで簡単にセットアップを行えるようにするために、インストール手順をわかりやすく説明する GUI ベースのインストーラが用意されています。

Configuration Agent のインストール

注 - Configuration Agent もインストーラも、Java™ Runtime Environment (JRE) 1.4 以上がインストールされている必要があります。本インストールには JRE は含まれていません。インストールされている Java のバージョンが分からない場合は、**java -version** を実行します。このコマンドによって、Java バージョンが適切であるか、システムに正常に設定されているかを確認できます。

このインストーラは、バッチファイル `setup.bat` によって起動されます。このバッチファイルは Java Desktop System Management Tools CD にあります。十分な管理特権を持つユーザー (**local administrator**) としてスクリプトを起動する必要があります。管理特権が十分でない場合は、新しいシステムサービスとして Configuration Agent をインストールおよび起動することはできません。

次のように入力してインストーラを起動します。

```
setup [-help]
      [-saveState filename] [-no]
      [-noconsole] [-state filename]
```

[-nodisplay]

次のオプションがインストーラによって認識されます。

-help	ヘルプテキストを表示する
-saveState	対話式インストールセッションからの入力を、指定したファイルに書き込む。デフォルトのファイル名は statefile.out
-state	指定した状態ファイルをサイレントインストールの入力として使用する。このオプションは -noconsole とともに使用する
-noconsole	ユーザーインターフェースなしでインストーラを起動する。-state とともに使用し、サイレントインストールを開始する
-nodisplay	グラフィカルユーザーインターフェースを起動せずにコマンド行モードでインストーラを起動する
-no	ソフトウェアをインストールせずにインストーラを実行する。このオプションは、サイレントインストールの準備のために -saveState とともに使用すると便利

「ようこそ」

Configuration Agent インストールウィザードが呼び出されると、「ようこそ」ページが表示されます。「ようこそ」ページには、インストールプロセスに関する情報が示されます。

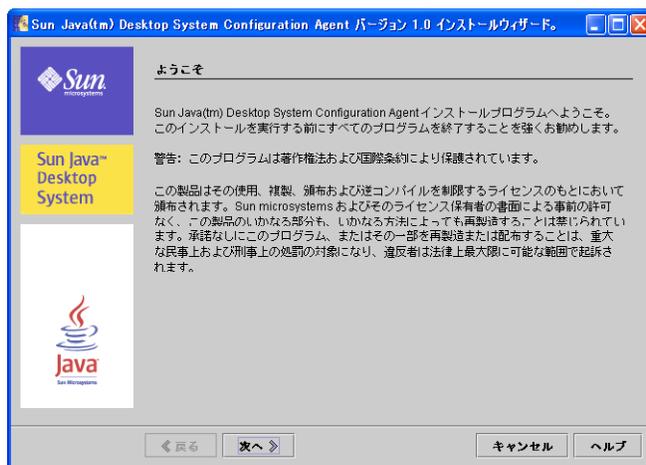


図 6-1 Configuration Agent、「ようこそ」ページ

「次へ」ボタンをクリックしてウィザードの次のページに進みます。

使用許諾契約

「ソフトウェアライセンス条項」ページには、Configuration Agent のライセンス情報が示されます。

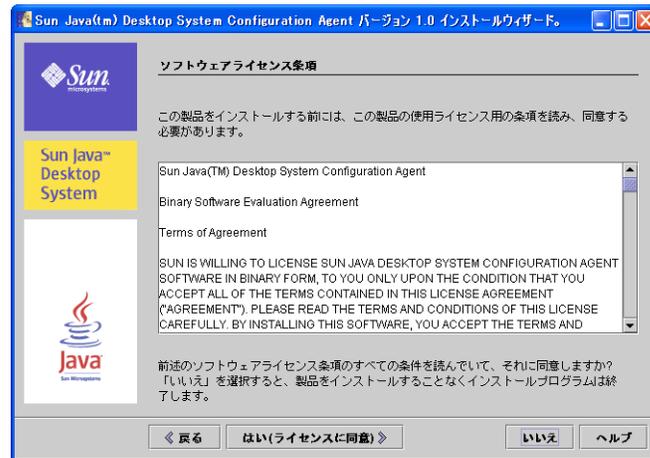


図 6-2 Configuration Agent、「ソフトウェアライセンス条項」

ライセンス合意書の内容を読み、「はい (ライセンスに同意)」ボタンをクリックしてインストールプロセスを続行します。

インストールの種類

インストーラには 3 種類のインストールが用意されています。これらは、インストールプロセス中に定義できる設定の数だけが異なります。



図 6-3 Configuration Agent、インストールの種類のパージ

- 高速: ほとんどのユーザーには「高速」をお勧めします。Configuration Agent を十分に機能させるために不可欠な設定だけが示されます。その他の設定は適切なデフォルト値に設定されます。
- カスタム: Configuration Agent の定義可能なすべての設定が表示されるので、エージェントの動作を詳細に調整できます。これは上級ユーザーのみにお勧めします。
- 最小: Configuration Agent はインストールされますが、指定するように求められる設定はありません。インストールの完了後に、設定ウィザードを起動して Configuration Agent を設定する必要があります。

インストールの種類を選択して「次へ」をクリックします。選択したインストールタイプに応じて、それぞれの設定のページが表示されます。次の節で各設定を説明します。

注 - 「高速」または「最小」のオプションを選択した場合、以降で説明する設定のページの中には該当しないものもあります。

注 - Configuration Agent を起動するには、ホスト名、LDAP サーバーのポートなどのブートストラップ情報が必要です。さらに、ロギングレベル、データディレクトリなどの詳細な設定も定義できます。これらのすべての設定は、インストールプロセス中に指定できます。この情報は、`polycmgr.properties`、`apocd.properties`、`os.properties` などのプロパティファイルの集合として保守されます。これらのファイルは、ローカルの `Program Files\Sun\Apoc\lib` ディレクトリに格納されます。これらのプロパティファイルは手作業でも、Configuration Agent の設定ウィザードでも編集できます。

Configuration Agent ウィザードと設定情報

この節では、Configuration Agent で使用できるすべての設定について具体的に説明します。

この設定ウィザードは、グラフィカルユーザーインターフェースを提供し、これに従って Configuration Agent に必要な設定ができます。このウィザードのすべてのページで、対応するヘルプ画面が利用できます。

ブートストラップ情報

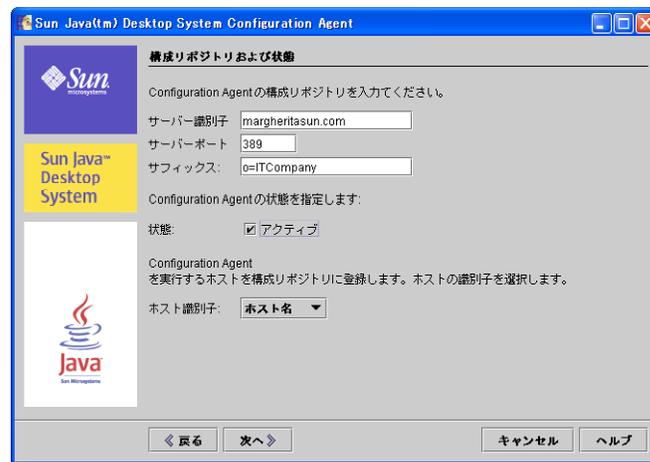


図 6-4 Configuration Agent、設定リポジトリ

注 – 対応するプロパティファイルキーを括弧内に示しています (適切な場合)。

- サーバー識別子 (Server): LDAP サーバーのホスト名。
- サーバーポート (Port): LDAP サーバーのポート番号。
- サフィックス (BaseDn): LDAP リポジトリのベース DN。
- 状態: Configuration Agent の状態。チェックボックスを使用して Configuration Agent をアクティブまたは非アクティブにできます。設定リポジトリを使用するには、Configuration Agent をアクティブにしておく必要があります。アクティブにしておくこと、サービス管理マネージャへの必要な登録が自動的に行われます。

注 - 手動で Configuration Agent を有効または無効にするには、管理者としてログインし、有効にする場合はコマンド `Program Files\Sun\Apoc\bin\apocd enable` を、無効にする場合はコマンド `Program Files\Sun\Apoc\bin\apocd disable` をそれぞれ入力します。



図 6-5 Configuration Agent、認証機構

- ホスト識別子 (HostIdentifier): 「HostName」または「IPAddress」のいずれかを指定できます。この識別子は、ホストを識別するときに使用する LDAP 属性の内容に一致するように設定する必要があります。この属性は、マッピングファイル内では Host/UniqueIdAttribute として指定されます。
- Configuration Agent の 認証タイプ: 「匿名」または「Simple」のいずれかを指定できます。「匿名」を選択した場合、「認証ユーザー名」と「パスワード」フィールドは自動的に無効になります。
- 認証ユーザー名 (AuthDn): リポジトリに対する読み取り権および検索権を持つユーザーの完全な DN。

注 - このディレクトリで匿名アクセスが有効である場合、この設定は空のままにしておくことができます。

- パスワード (Password): 登録された LDAP ユーザーのパスワード。

注- このディレクトリで匿名アクセスが有効である場合、この設定は空のままにしておくことができます。

- アプリケーションの認証タイプ (AuthType): LDAP サーバーによるユーザーの認証方法に応じて、「匿名」または「GSSAPI」のいずれかを指定できます。

ポートの設定

Configuration Agent は、次の 2 つのポートを使用します。

- エージェントポート (DaemonPort): エージェントがクライアントアプリケーションと通信するときに使用します。デフォルトは **38900** です。
- 管理ポート (DaemonAdminPort): エージェントコントローラプログラム `apocd.exe` が、エージェントと通信するときに使用します。デフォルトは **38901** です。

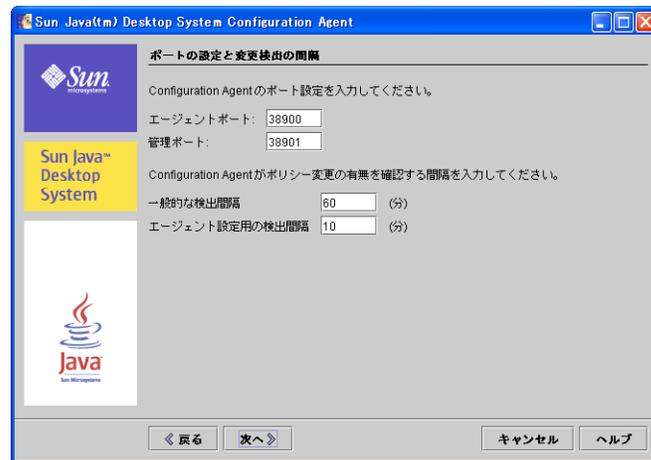


図 6-6 Configuration Agent、ポートの設定

変更検出間隔

Configuration Agent は、次の 2 つの間隔を使用して、設定データに変更があるかどうかを定期的にチェックします。

- 一般的な検出間隔 (ChangeDetectionInterval): デスクトップアプリケーション (クライアント) の設定データの変更検出間隔 (分)。

注 --1 を指定すると、変更検出がオフになります。

- エージェント設定用の検出間隔 (DaemonChangeDetectionInterval): エージェントに固有な設定データの変更検出間隔 (分)。

注 --1 を指定すると、変更検出がオフになります。

汎用の検出間隔を使用すると、リモートの設定データの変更をクライアント側のアプリケーションに伝播する間隔を調整できます。この設定で指定する値は、リモートに加えられた変更の内容がクライアントアプリケーションに反映されるまでの最大期間 (分) です。

この値が小さくなるほど、Configuration Agent と LDAP サーバーの活動が増えます。したがって、この設定値を調整する場合は注意が必要です。たとえば、最初の配備段階でこの値を1分に設定するれば、クライアントアプリケーションに対するリモート設定の影響を簡単にテストできます。テストが完了したら、この設定を初期値に戻します。

操作設定

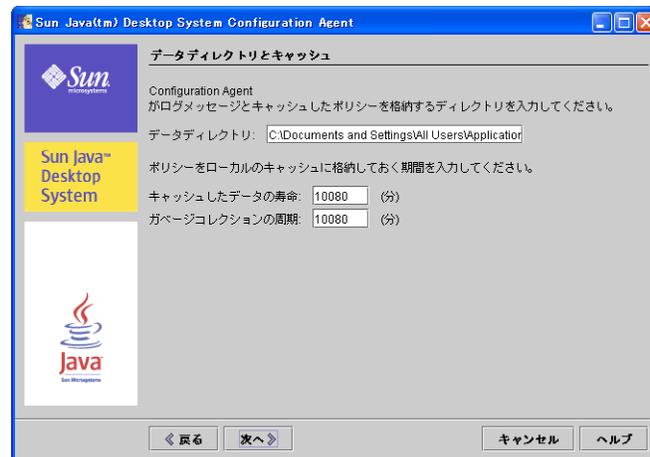


図 6-7 Configuration Agent、データディレクトリ

次の設定が設定できます。

- データディレクトリ (DataDir): 実行時データを格納するために使用されるディレクトリ。

- キャッシュしたデータの寿命 (TimeToLive): ローカルデータベース内に非オフラインの設定データが留まる間隔 (分)。

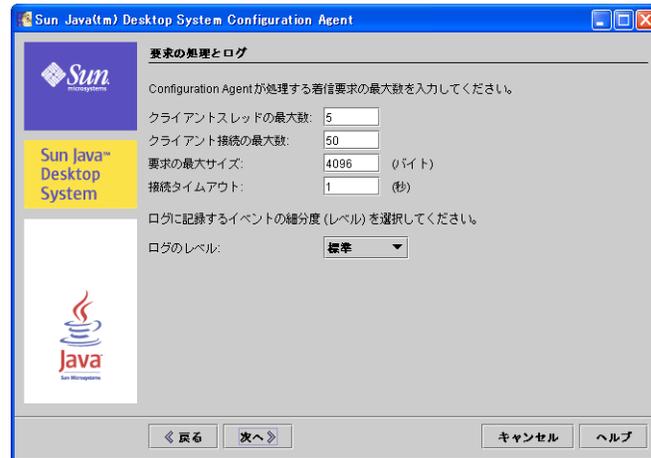


図 6-8 Configuration Agent、要求の処理とロギング

- ガベージコレクションの周期 (GarbageCollectionInterval): ローカル設定データベースのガベージコレクション周期 (分)。
- クライアントスレッドの最大数 (MaxClientThreads): 同時に処理できるクライアント要求の最大数。
- クライアント接続の最大数 (MaxClientConnections): クライアント接続の最大数。
- 要求の最大サイズ (MaxRequestSize): クライアント要求の最大サイズ。
- 接続タイムアウト (ConnectTimeout): LDAP サーバーが接続要求に応答できる間隔 (秒)。デフォルトは 1 秒です。
- ログのレベル (LogLevel): エージェントのログファイルの詳細レベル。ロギングレベルは Java Logger のレベルに一致します。次に、これらのレベルを重要度の降順に示します。
 - OFF
 - SEVERE
 - WARNING
 - INFO
 - CONFIG
 - FINE
 - FINER
 - FINEST
 - ALL

注 – ほとんどの操作設定 (「データディレクトリ」と「接続タイムアウト」の設定を除く) は、LDAP サーバーに格納された対応するポリシー経由で集中的に保守できます。この機能を使用する場合は、対応する設定をウィザードで変更しないでください。代わりに、Configuration Manager 内の Configuration Agent ポリシーを使用して、操作設定を集中的に指定します。

エージェント設定の要約

Configuration Manager を使用して LDAP サーバーに格納した操作設定 (「データディレクトリ」と「接続タイムアウト」の設定を除く) は、エージェント設定の次回の変更検出時に自動的に適用されます (DaemonChangeDetectionInterval を参照)。

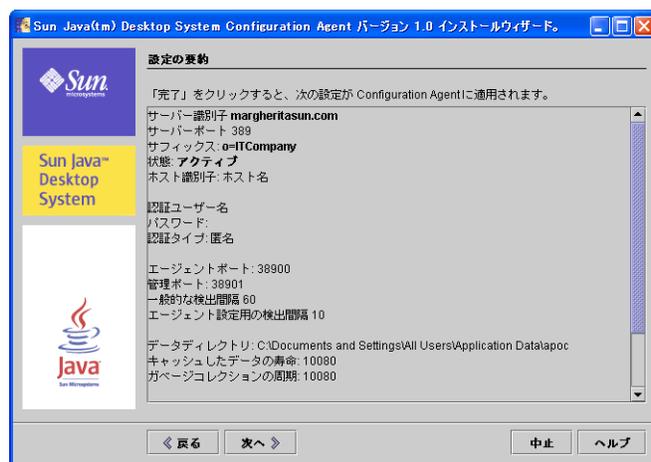


図 6-9 Configuration Agent、設定の要約ページ

ローカルで変更したその他の設定については、Configuration Agent を再読み込みまたは再起動する必要があります。設定ウィザードを使用する場合、再読み込みまたは再起動は自動的に実行されます。

注 – 手で Configuration Agent を再起動するには、関連するクライアントアプリケーションが実行されていないことを確認し、管理者としてログインしてコマンド `Program Files\Sun\Apoc\bin\apocd restart` を入力します。または、コントロールパネルの管理ツールにあるサービスを起動します。使用可能なサービスのリストから「**Configuration Agent**」を選択し、「サービスの再起動」をクリックします。

インストール準備完了

「インストール準備完了」ページには、インストールされる Configuration Agent の項目が表示されます。

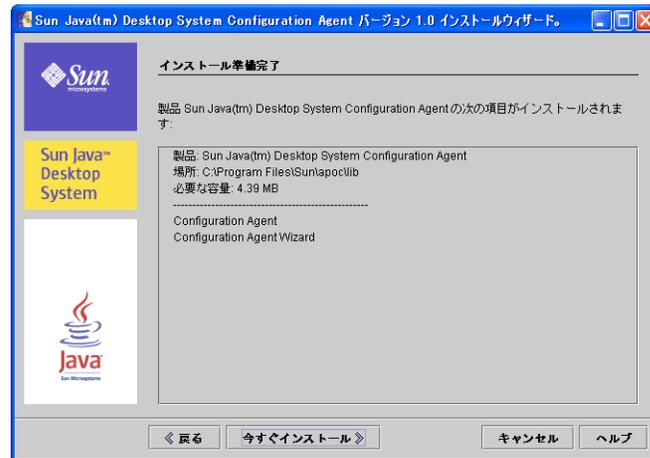


図 6-10 Configuration Agent、「インストール準備完了」ページ

この時点では、システムはまだ変更されていません。実際にインストールが開始されるのは、「今すぐインストール」ボタンをクリックしてからです。

インストールの概要

ウィザードの「インストールの概要」ページには、インストールされる項目の状態が表示されます。それぞれの項目の「詳細」ボタンをクリックすると、その項目の詳細情報が表示されます。

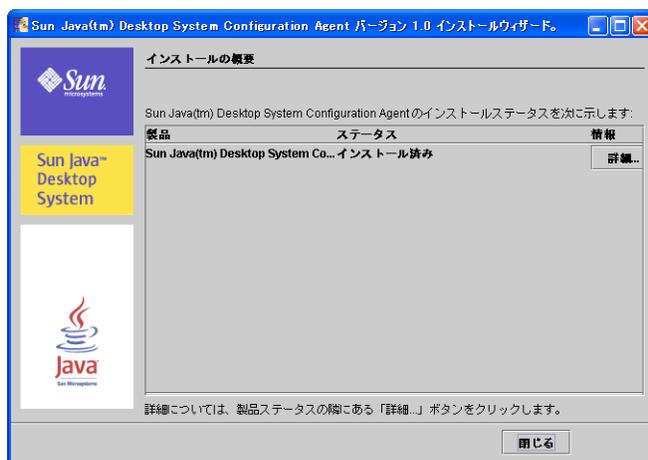


図 6-11 Configuration Agent、「インストールの概要」ページ

インストーラによって、Configuration Agent のバイナリがデフォルトのプログラムファイルディレクトリ (通常は C:\Program Files) 内の Sun\apoc に格納されます。さらに、「スタート」-「プログラム」-「Sun」メニューに Java Desktop System の新しいフォルダも自動的に作成されます。このフォルダには 2 つのショートカットが含まれていて、Configuration Agent をさらに詳細に設定したり、アンインストールすることができます。

Configuration Agent の設定の変更

前述のように、Configuration Agent の設定はインストール後にいつでも変更できます。Program Files\Sun\Apoc\bin\apoc-config.bat スクリプトを使用して、管理者としてウィザードを起動します。または、「スタート」メニューから「プログラム」-「Sun」-「Java Desktop System」の「**Configuration Agent**」エントリを選択することによっても、Configuration Agent ウィザードを起動できます。

注-ウィザードはグラフィカルインタフェースを起動しなくても起動できます。たとえば、コンソールモードでウィザードを起動するには、Program Files\Sun\Apoc\bin\apoc-config -nodisplay を実行します。

Configuration Agent のアンインストール

Configuration Agent をアンインストールするには、「スタート」メニューから「プログラム」 - 「Sun」 - 「Java Desktop System」 - 「**Configuration Agent**」 - 「Configuration Agentのアンインストール」を選択します。

Sun Web Console のパッケージ

既知の問題点

セキュリティ

ユーザーのアクションによって、ユーザーが知らずに、セッションがアクティブ状態のままにされる可能性があります。たとえば、ユーザーがブラウザウィンドウを閉じても、Sun Web Console からは自動的にログアウトしません。この場合、ユーザーはアプリケーションウィンドウを閉じる前に、Sun Web Console のセッションから明示的にログアウトする必要があります。

Setup スクリプトの使用法

要約: `setup [-h] | [-n] | [-d <var>,<arch>[,client1,client2,...]]`
`[-u [-f]]`

-h = 使用法を表示します。

-n = インストール終了時にサーバーを起動しません。

-u = Sun Web Console をアンインストールします。

-f = Tomcat と Java 1.4 がセットアップアプリケーションによってインストールされている場合は、これらのパッケージをアンインストールします。このパラメータを使用できるのは、-u パラメータを併用する場合に限られます。

利用できるセットアップパラメータの詳細については、`setup -h` を実行します。

Sun Web Console をアンインストールするには、`/usr/lib/webconsole/setup -u` を実行します。

注 - /usr/lib/webconsole ディレクトリまたはそのサブディレクトリにいる場合は、このコマンドを実行しないでください。実行すると、pkgrm は失敗します。

Sun Web Console のパッケージ

Solaris のパッケージ

パッケージ名	説明
SUNWmctag	Sun Web Console の UI Tag ライブラリ
SUNWmcon	Sun Web Console
SUNWmcos	Sun Web Console の Solaris 共通サービス
SUNWmcosx	Sun Web Console の Solaris リリース固有サービス
SUNWmconr	Sun Web Console のルート
SUNWjato	Sun One Application Framework ランタイム
SUNWtcatu	Tomcat

Linux RPM

パッケージ名	説明
SUNWmctag	Sun Web Console の UI Tag ライブラリ
SUNWmcon	Sun Web Console
SUNWmcos	Sun Web Console の Linux 共通サービス
SUNWmcosx	Sun Web Console の Linux リリース固有サービス

パッケージ名	説明
SUNWmconr	Sun Web Console のルート
SUNWjato	Sun One Application Framework ランタイム
tomcat4	Tomcat

付録 B

Configuration Manager のパッケージ

Configuration Manager のパッケージ

Solaris のパッケージ

パッケージ名	説明
SUNWapm	Configuration Manager
SUNWapmca	Configuration Agent テンプレート
SUNWapmev	Evolution テンプレート
SUNWapg26	Gnome 2.6 テンプレート
SUNWapmmo	Mozilla テンプレート
SUNWapmso	StarSuite テンプレート

Linux RPM

パッケージ名	説明
apoc-manager	Configuration Manager
apoc-agent-templates	Configuration Agent テンプレート

パッケージ名	説明
apoc-evolution-templates	Evolution テンプレート
apoc-gnome26-templates	Gnome 2.6 テンプレート
apoc-mozilla-templates	Mozilla テンプレート
apoc-staroffice-templates	StarSuite テンプレート

Configuration Manager による OpenLDAP と Active Directory の使用

Configuration Manager による OpenLDAP サーバーの使用

OpenLDAP サーバーを Configuration Manager データのリポジトリとして使用するには、設定データを格納するために使用するオブジェクトクラスと属性を提供するようにサーバーのスキーマを拡張する必要があります。Java Desktop System Management Tools CD で提供される Configuration Manager 展開ツールの `openldap` サブディレクトリには、`apoc.schema` という名前のカスタムスキーマファイルがあります。

このファイルを OpenLDAP 設定ディレクトリ (`/etc/openldap`) の `schema` サブディレクトリ内にコピーしたうえで、そのディレクトリにある `slapd.conf` ファイルにカスタムスキーマファイルを含めることにより、カスタムスキーマファイルを OpenLDAP に追加します。このためには、ファイル内に存在する一連のスキーマの末尾に `include /etc/openldap/schema/apoc.schema` という行を挿入します。OpenLDAP サーバーのスキーマの拡張について詳細は、サーバーのマニュアルを参照してください。

OpenLDAP データベースに設定データを格納できるように準備するには、Configuration Manager で提供される展開ツールを使用する必要があります。このスキーマは以前のインストール手順ですでに拡張されているので、`createServiceTree` スクリプトのみを実行する必要があります。任意のユーザーとして、このスクリプトを展開ツールのディレクトリからコマンド `./createServiceTree` を使って起動する必要があります。このスクリプトを実行すると、この文書の展開ツールに関する節に記載されている OpenLDAP データベースの情報を入力するように求められます。OpenLDAP 内の標準のオブジェクトクラス/属性を使用するデフォルトマッピングファイルは、展開ツールの `openldap` サブディレクトリに含まれています。このファイルの名前は `OrganisationalMapping` です。このファイルを配備するには、`createServiceTree` を起動する前に展開ツールのメインディレクトリに含まれている同名のファイルに上書きコピーします。

注 - Configuration Manager エージェントは、データが必要なユーザーの DN だけを
提供して、パスワードを提供せずに、OpenLDAP サーバーに匿名で接続しようとしま
す。OpenLDAP サーバーの一部のリリースでは、デフォルトでこのモードの匿名認証
が無効になる場合があります。この場合、OpenLDAP 設定ディレクトリ
(/etc/openldap) 内のファイル slapd.conf に定義される共通のサーバーパラメー
タに allow bind anon_cred という行を追加することにより、このモードを有効に
する必要があります。パラメータの詳細については、サーバーのマニュアルを参照し
てください。

Configuration Manager による Active Directory サーバーの使用

Active Directory サーバーを Configuration Manager データのリポジトリとして使用するには、設定データを格納するために使用するオブジェクトクラスと属性を提供するようにサーバーのスキーマを拡張する必要があります。Java Desktop System Management Tools CD で提供される Configuration Manager 展開ツールの ad サブディレクトリには、apoc-ad.ldf という名前のスキーマ拡張ファイルがあります。詳細については、展開ツールに関する節を参照してください。

次の手順に従って、apoc-ad.ldf ファイルを Active Directory スキーマにインポートする必要があります。

1. スキーマの拡張を有効にします。操作方法について詳細は、Active Directory の文書を参照してください。
2. コマンドプロンプトから `ldifde -i -c "DC=Sun,DC=COM" <BaseDN> -f apoc-ad-registry.ldf` を実行します。

注 - 「<BaseDN>」は Active Directory のベース DN で置き換えます。

Active Directory サーバーで設定データを格納するための準備を行うには、展開ツールを使用する必要があります。このスキーマは、以前のインストール手順ですでに拡張されているので、createServiceTree スクリプトのみを実行する必要があります。任意のユーザーとして、このスクリプトを展開ツールのディレクトリから ./createServiceTree によって起動する必要があります。このスクリプトを実行すると、Active Directory データベースの情報を入力するように求められます。Active Directory 内の標準のオブジェクトクラスおよび属性を使用するデフォルトマッピングファイルは、展開ツールのディレクトリの ad サブディレクトリに含まれています。このファイルの名前は OrganisationalMapping です。このファイルを配備するには、createServiceTree を起動する前に展開ツールのメインディレクトリに含まれている同名のファイルに上書きコピーします。

これ以降、Active Directory サーバーを Configuration Manager で使用できます。Configuration Manager をインストールするときには、ツリーの読み取り権を持ったユーザーの完全な DN とパスワードを指定します。このユーザーとして、他の目的で Active Directory を使用できないユーザーを選択できます。このようなユーザーを設定する方法についての詳細情報は、Active Directory のマニュアルを参照してください。さらに、Active Directory のドメイン名は Configuration Manager が動作しているマシンによって認識されている必要があります。このためには、Active Directory サーバーの IP アドレスをそのドメイン名にマッピングする行をコンピュータの `/etc/hosts` ファイルに追加します。

Java Desktop System ホストから設定データを取得するには、そのホストでも Active Directory のドメイン名が認識されていることが必要です。Java Desktop System ユーザーの認証は、匿名および GSSAPI の 2 通りの方法で行うことができます。

- 匿名接続を使用して認証するには、すべてのユーザーに読み取り権利を許可するように Active Directory サーバーを設定する必要があります。操作方法について詳細は、Active Directory の文書を参照してください。
- GSSAPI を使用して認証するには、Kerberos パラメータを指定するファイル `/etc/krb5.conf` を変更することにより、Active Directory 領域を定義したうえで、その Key Distribution Center (KDC) として Active Directory サーバーを参照する必要があります。また、デフォルトの暗号化タイプとして、Active Directory でサポートされる DES タイプ (`des-cbc-crc` および `des-cbc-md5`) を指定しなければなりません。操作方法について詳細は、Kerberos の文書を参照してください。設定データにアクセスする前に、Java Desktop System にログインしたユーザーの有効な資格情報を取得する必要があります。この操作は、`kinit` コマンドを実行したあと、Active Directory で定義されたユーザーパスワードを入力することにより手動で実行できます。ログイン時にこれらの資格情報を自動的に生成する方法もあります。詳細については、Java Desktop System のマニュアルを参照してください。

