Sun Java™ Desktop System Configuration Manager, Release 1 インストールガイド

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054,
U.S.A. 650-960-1300

Part No. 817-5591-10

2004 年 4 月 Revision A

著作権と商標について

目次

第 1 章

はじめに

Sun Java™ Desktop System Configuration Manager, Release 1 は、Sun Java™ Desktop System を実行するデスクトップホストの設定を一元化して行うことができます。組織構成のさまざまな要素に設定を割り当てることができるので、管理者が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。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。現在ログイン中のユーザのために設定データを取得し、それを Sun™ Java Desktop System に属するさまざまなアプリケーションが使用できるようにします。

この管理ツールは、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。

これらのデスクトップコンポーネントは、Sun Java™ Desktop System Configuration Agent を中心として編成されます。Sun Java Desktop System Configuration Agent は、ユーザーのために LDAP サーバーから設定データを取得し、それを多数の設定システムアダプタで利用できるようにします。これは、ポリシー設定によりローカル設定 (アプリケーションとユーザーの設定により提供されるデフォルトの設定) を補完するものです。現在サポートされている設定システムは、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します)、Mozilla™ Preferences、および StarRegistry (StarSuite の設定システム) です。

第 2 章

LDAP サーバー

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ます。エンティティは、LDAP リポジトリ内のエントリであり、企業の組織的な構成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: 属性を表します。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その他のエントリの関係は、リポジトリ内のエントリの物理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

Configuration Manager と共に既存の LDAP サーバーを使用するには、次のことを行う必要があります。

- Configuration Manager で設定データの格納時に使用されるカスタムのオブジェクトクラスおよび属性をサポートするようにサーバースキーマを拡張します。
- Configuration Manager でサポートされるエンティティに加えて、リポジトリ内のエントリのマッピング情報をカスタマイズし、サーバーに格納します。

配備ツール

Configuration Manager と共に既存の LDAP サーバーを使用するには、インストール CD に収録されている次の配備ツールが必要です。

- `88apoc-registry.ldif` — 設定データの保存に必要なオブジェクトの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 エンティティ間のマッピングについて記述するデフォルトの属性ファイルです。
- `UserProfileMapping` — LDAP のユーザーエントリの属性と Configuration Manager のユーザーエントリの属性間のマッピングについて記述するデフォルトの属性ファイルです。
- `createServiceTree` — LDAP リポジトリ内にマッピングファイルを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。

スキーマの拡張

設定データは、そのデータが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 サーバースキーマに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、提供を受けたスキーマ拡張ファイルでオブジェクトとクラスを Sun Java System Directory Server に追加するために LDIF 形式を使用する場合を考えます。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 エンティティ間のマッピングを指定するには、組織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で使用されるオブジェクトクラス
User/ObjectClass=inetorgperson
# リポジトリ内で固有の値をユーザーエントリ内に持つ属性
User/UniqueIdAttribute=uid
# ユーザーエントリの組織エントリ内のオプションコンテナ。不要な場合はこの行を削除してください。
User/Container=ou=People
# 管理アプリケーション内で表示する名前の形式
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。このリストでは、形式 <item1>,<item2>,...,<itemN> を使用し、整列することが必要です。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/ホストの役割の DN を含む物理属性 (フィルタ内で使用可能)。
Role/MemberAttribute=nsRoleDN
# この属性のクエリーをユーザーまたはホストに対して実行すると、そのユーザーまたはホストが属する役割の DN が返されます。
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
```

ドメインのエンティティは、組織のエンティティ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
```

ホストのエンティティは、ユーザーのエンティティ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で使用されるオブジェクトクラス
Host/ObjectClass=ipHost
# リポジトリ内で固有の値をホストエントリ内に持つ属性
Host/UniqueIdAttribute=cn
# ホストエントリのドメインエントリ内のオプションコンテナ。不要な場合はこの行を削除してください。
Host/Container=ou=Hosts
```

ユーザープロファイルのマッピング

LDAP のユーザーエントリの属性と Configuration Manager のユーザーエンティティの属性間のマッピングを指定するには、ユーザープロファイルマッピングファイルを編集する必要があります。各キーが Configuration Manager の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、ユーザープロファイル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
# 推奨対象外であるため、対応する LDAP 項目は存在しません。
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を LDAP リポジトリの状態に合わせてカスタマイズしたら、マッピングファイルを配備できます。必要なオブジェクトのクラスと属性がすでに LDAP サーバーのスキーマに含まれている場合は、スクリプト `createServiceTree` を直接実行できます。このオブジェクトのクラスと属性が含まれていない場合は、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 Servers との併用が想定されたスクリプトです。このスクリプトは、提供を受け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た後、LDAP サーバーを再起動してから `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ことが必要です。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 <ディレクトリサーバーのディレクトリ>
```

<ディレクトリサーバーのディレクトリ>パラメータは、ディレクトリサーバーインストールの `slapd-`<サーバー名>サブディレクトリへのパスであることが必要です。インストールでデフォルトディレクトリを使用し、サーバーの名前が `myserver.mydomain` である場合、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、または `deployApoc` スクリプトから起動された場合のいずれでも、LDAP サーバー (ホスト名、ポート番号、およびベース DN) の場所と管理アクセス権 (完全な DN とパスワード) を持つユーザーの定義をユーザーに入力するように求めます。その後、このスクリプトは LDAP サーバー内に bootstrap サービスツリーを作成し、そこにマッピングファイルを保存します。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。

- ホスト名 (デフォルト: localhost): LDAP サーバーのホスト名
- ポート番号 (デフォルト: 389): LDAP サーバーのポート番号
- ベース DN: LDAP リポジトリのベース DN
- ユーザー 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するための十分なアクセス権を持つユーザーの完全な DN
- パスワード: 上記ユーザーのパスワード

次の DN を持つエントリが作成され、

```
ou=ApocRegistry,ou=OrganizationConfig,ou=1.0,ou=ApocService,ou=services,
<ベース DN>
```

2 つのマッピングファイルの内容が入力されます。

前述したように、`deployApoc` スクリプトによって実行される処理では、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の手順が Sun Java System Directory Server と緊密に対応することを前提としています。その他のディレクトリでは、`createServiceTree` スクリプトを実行する前に、スキーマを手動で拡張する必要があります。OpenLDAP と Active Directory の使用についての詳細は、付録 C および付録 D ををそれぞれ参照してください。

作成されるツリーは、エンティティに関連する設定データを格納するツリーと一致しており、Sun Java System Identity Server のサービス管理で使用されるツリーの構成に対応しています。

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特定のユーザーまたはホストの識別子に関連する完全な DN を確認するために、読み取り/検索権を持つ LDAP サーバーへの接続を確立できる必要があります。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および検索の匿名アクセスを有効にする必要があります。Kerberos 認証による GSSAPI 方式を使用するには、『Sun Java System Directory Server Administration Guide』の章「Implementing Security」の記述に従って、LDAP サーバーを設定する必要があります。

第 3 章

Sun™ Web Console

Sun Web Console は、Sun Microsystems 製品用の一般的な 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トリポイント
- 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 回ログインした後、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、いくつかのブラウザと同様に数種類のクライアントおよびサーバーオペレーティングシステムをサポートします。

クライアント

- Solaris™ 8 以降で動作する 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 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
- Redhat 8 以降、または Redhat Enterprise Linux 2.1

- SuSE Linux 2.1 以降
- J2SE™ Version 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、このマニュアルの付録 A に記載されているパッケージの概要と既知の問題に関する説明をお読みください。

Sun Web Console for Solaris SPARC® (バージョン 8 以降) および Linux オペレーティングシステム用のインストールバイナリが 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

1. Java Desktop System Management Tools CD 上で、この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に対応する Sun Web Console ディレクトリに移動します。

   Linux システムでは `/linux/swc`、Solaris SPARC では `/solsparc/swc` に移動します。

2. 「`./setup`」と入力します

   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。名前「logfile」を持つインストールログを作成するには、「`./setup | tee logfile`」と入力します。

   注: Web コンソールの大半のインストール処理と設定処理は、`setup` の実行時に自動的に行われます。Sun Web Console の `setup` アプリケーションについて詳細は、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、各言語について 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`」と入力します。*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`」と入力します。*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	簡体字中国語版 Sun(TM) Web Console 2.0
SUNWdmcon, SUNWdmctg	ドイツ語版 Sun(TM) Web Console 2.0
SUNWemcon, SUNWemctg	スペイン語版 Sun(TM) Web Console 2.0
SUNWfmcon, SUNWfmctg	フランス語版 Sun(TM) Web Console 2.0
SUNWhmcon, SUNWhmctg	繁体字中国語版 Sun(TM) Web Console 2.0
SUNWimcon, SUNWimctg	イタリア語版 Sun(TM) Web Console 2.0
SUNWjmcon, SUNWjmctg	日本語版 Sun(TM) Web Console 2.0
SUNWkmcon, SUNWkmctg	韓国語版 Sun(TM) Web Console 2.0

パッケージ名	説明
SUNWsmcon, SUNWsmctg	スウェーデン語版 Sun(TM) Web Console 2.0

コンソールの実行

一般的に、新しいアプリケーションを登録する場合にのみ、Sun Web Console サーバーを停止、再開する必要があります。

Sun Web Console を最初に開始する前に、Configuration Manager のインストールが完了していることを確認してください。

- Sun Web Console を開始するには、「`smcwebserver start`」と入力します。
- Sun Web Console を中止するには、「`smcwebserver stop`」と入力します。
- Sun Web Console にアクセスするには、ブラウザに URL「`https://<hostname>.<domain-name>:6789`」を入力します。

Sun Web Console では、導入後すぐに 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) がサポートされます。ただし、LDAP 認証など、その他の認証機構を設定することもできます。

> 注: デフォルトのセッションタイムアウトは 15 分です。タイムアウト期間は `smreg` コマンドで設定できます。たとえば、タイムアウト期間を 5 分に設定するには、「`smreg add -p -c session.timeout.value=5`」と入力します。

Sun Web Console のコマンドについて詳細は、smcwebserver および 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関連サブディレクトリのいずれかでこのコマンドを実行しないでください。これらのディレクトリでは、`pkgrm` にエラーが発生します。

Sun Web Console のポート情報

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

- サービスを停止するための 8005
- http アクセスのための 6789

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` を使用して Sun Web Console を再起動してください。

第 4 章

Sun Java™ Desktop System Configuration Manager, Release 1

Configuration Manager には、Sun Web Console 上で動作する管理ツールが用意されています。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、Gnome、Mozilla、StarSuite、Evolution など各種のデスクトップアプリケーションに固有の設定を表示するために、数種類の設定テンプレートを使用します。

Configuration Manager のインストール

Configuration Manager をインストールするには、Sun Web Console を起動している必要があります。

1. Java Desktop System Management Tools CD 上の対応する Configuration Manager ディレクトリに移動します。

   Linux システムでは、`/linux/apoc` に移動します。Solaris SPARC では、`/solsparc/apoc` に移動します。

2. 「`./setup`」と入力します
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: 389)。
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、第 2 章「LDAP サーバー」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、第 2 章「LDAP サーバー」の「組織のマッピング」を参照してください。

8. LDAP サーバーに対してクエリーを実行するのに必要な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. DAP 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。

このインストールの処理中に、LDAP を通じてユーザーを認証できる追加ログインモジュールが Sun Web Console に付加されます。

インストール処理の終了時に、Sun Web Console が自動的に再開するので、Configuration Manager にアクセスできます。

注: `usr/share/webconsole/apoc/configure` スクリプトを使用することにより、Configuration Manager の以前の設定をいつでも変更できます。たとえば、このスクリプトを使って、Configuration Manager を再インストールすることなく、別の LDAP サーバーに変更できます。

Configuration Manager の実行

1. Configuration Manager にアクセスするには、ブラウザに次の URL を入力します。

   `https://<ホスト名>.<ドメイン名>:6789`

2. プロンプトで、既存の LDAP ユーザーのユーザー名 (uid) とパスワードを入力します。

   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**」をクリックします。

注: Sun Web Console の起動ページをスキップし、Configuration Manager を直接開くには、ブラウザに次の URL を入力します。

`https://<ホスト名>.<ドメイン名>:6789/apoc`

Configuration Manager のアンインストール

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

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

第 5 章

デスクトップコンポーネント

デスクトップクライアントでは、Configuration Manager から設定データにアクセスするために、Sun Java™ Desktop System Configuration Agent を必要とします。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ほか、特定の設定システムにデータを統合します。現在サポートされている設定システムは、GConf、Mozilla Preferences、および StarSuite Registry です。

これらのコンポーネントはすべて、Java Desktop System に付属し、その一部としてインストールされます。

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、ホストの名前や IP アドレスなど、ホストに関する情報を取得します。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 資格情報の取得処理をユーザーログイン処理に統合するには、Java Desktop System ホストに `pam_krb5` モジュールをインストールして設定することが必要です。pam モジュールの設定例が Java Desktop System CD の `/usr/share/doc/packages/pam_krb5/README.SuSE` ディレクトリに収録されています。また、gdm を使って Kerberos をユーザーログインに統合できます。たとえば、次の /etc/pam.d/gdm ファイルを使用できます。

```
#%PAM-1.0
auth    required     pam_unix2.so  nullok #set_secrpc
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
```

Configuration Agent

Configuration Agent は、apoc パッケージに含まれています。対応する RPM をインストールすると、この API で必要になるファイルがインストールされ inetd に登録されます。RPM は手動でインストールすることもでき、Java Desktop System のインストールと共にインストールすることもできます。

ブートストラップ情報

リモート設定データにアクセスするには、LDAP サーバーの場所を Configuration Agent に指定する必要があります。この場所を指定するには、YaST2 設定ツール autoYaST を使用するか、または /opt/apoc/lib ディレクトリ内の policymgr.properties 属性ファイルを手動で編集します。YaST2 では、Network/Advanced セクションにこのデータを追加できます。

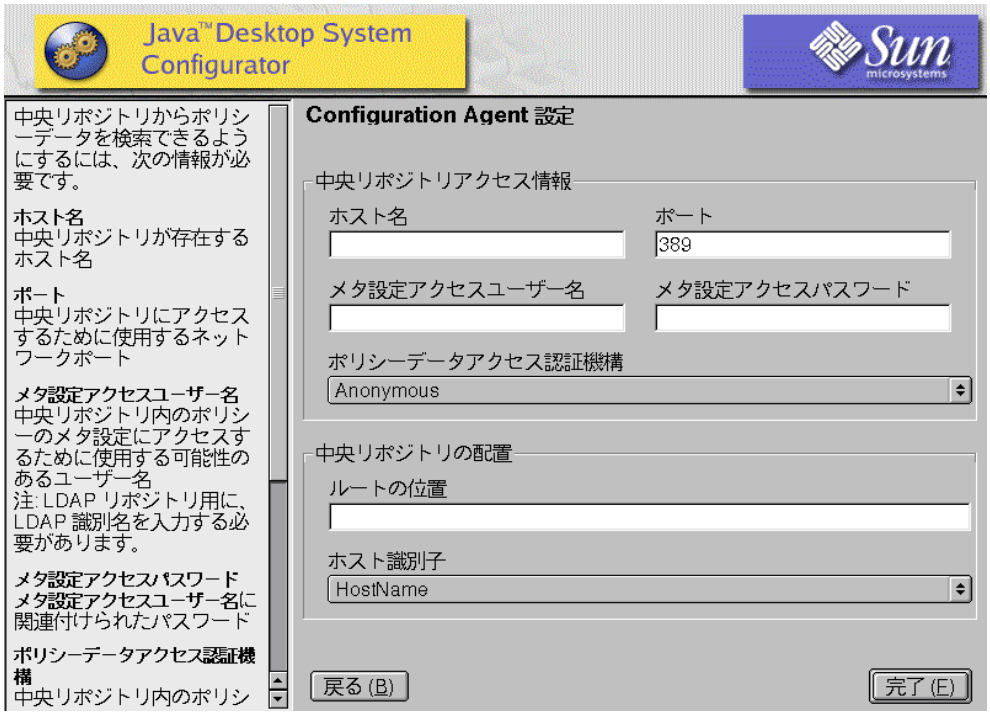

図 *1 YaST の Java Desktop System Configuration Agent*

Configuration Agent を実行するには、次の情報が必要です。

- 「ホスト名(host name)」(サーバー): LDAP サーバーのホスト名。
- 「ポート(Port)」(ポート): LDAP サーバーのポート番号。
- 「メタ設定アクセスユーザー名 (metadata access user name)」(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: ディレクトリ内で匿名アクセスを有効にする場合、この設定は空欄のまま残しておくことができます。
- 「メタ設定アクセスパスワード (metadata access password)」(パスワード): 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を有効にする場合、この設定は空欄のまま残しておくことができます。

- 「ポリシーデータアクセス認証機構 (policy data access authentication mechanism)」(AuthType): LDAP サーバーによるユーザーの認証方法に応じて、GSSAPI または匿名のいずれかを指定できます。
- 「ルートの位置 (root location)」(BaseDn): LDAP リポジトリのベース DN。
- 「ホスト識別子 (host identifier)」(HostIdentifier): HostName または IPAddress を指定できます。また、ホストの確認に使用される LDAP 属性の内容と一致する設定が必要です。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。
- 「Connect timeout (Connect Timeout)」(Connect Timeout): LDAP サーバーへの接続の試みがタイムアウトする秒数を示します。デフォルト値は 1 秒です。

注**:** この設定を変更する場合は常に、Configuration Agent を再起動する必要があります。

Sun Web Console 上で Configuration Agent を再起動するには、どの関連クライアントアプリケーションも実行していないことを確認してから、ルートとしてログインし、`/opt/apoc/bin/apocd restart` というコマンドを入力します。

操作設定

Configuration Agent の操作設定をローカルまたはリモートに指定できます。設定をローカルに指定するには、`/opt/apoc/lib` ディレクトリ内の `apocd.properties` ファイルを編集します。設定をリモートに指定するには、Configuration Manager 内で Configuration Agent ポリシーを使用します。この属性ファイル内で、次の設定を指定できます。

- DaemonPort: Configuration Agent がデスクトップ上のクライアントからの通信を聴取するポート。
- MaxClientThreads: 同時に処理できるクライアント要求の最大数
- MaxClientConnections: クライアント接続の最大数。
- MaxRequestSize: クライアント要求の最大サイズ。
- DaemonChangeDetectionInterval: この設定リストの変更検出周期 (分)。
- ChangeDetectionInterval: クライアント設定データの変更検出周期 (分)。
- 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)。
- 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)。
- LogLevel: エージェントのログファイルの詳細レベル

DaemonPort の設定はローカルでのみ変更できます。変更内容を有効にするには、エージェントを再起動する必要があります。その他の設定はすべて、エージェント設定の次回の変更検出周期で有効になります。LogLevel で指定するログレベルは、Java Logger レベルに一致する値であることが必要です。このレベルは、重要度の降順に、SEVERE、WARNING、INFO、CONFIG、FINE、FINER、および FINEST です。

設定データの変更の伝播

リモート設定データの変更内容をクライアント側アプリケーションに伝播する処理を調整するために、「操作設定」で説明した ChangeDetectionInterval 設定を使用できます。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。ChangeDetectionInterval により小さな値を指定すると、Configuration Agent および LDAP サーバーの動作がより活発になります。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。

Configuration Agent のポート情報

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

1. デーモンポート (デフォルトは 38900)。デーモン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。
2. デーモン管理ポート (デフォルトは 38901)。デーモン制御プログラム `apocdct1` がデーモンと通信するときに使用します。

デーモンポートの変更:

デーモンポートを変更するには、そのデーモンの `apocd.properties` ファイルの `DaemonPort` プロパティと、`/etc/services` と `/etc/inetd.conf` の `apocd` エントリを変更する必要があります。そのあと、デーモンを再起動して `inetd` を再読み込みします。

デーモン管理ポートの変更:

デーモン管理ポートを変更するには、そのデーモンの `apocd.properties` ファイルの `DaemonAdminPort` プロパティを変更する必要があります。そのあと、デーモンを再起動します。

GConf アダプタ

GConf アダプタは `apoc-adapter-gconf` パッケージに含まれています。このアダプタを対応する RPM からインストールすると、`/etc/gconf/2/path` 内の GConf データソースパスが Configuration Manager ソースを含むように更新されます。以前のパスのバックアップが `/etc/gconf/2/path.apocBackup` に保存されます。以前のパスがカスタムデータソースを参照している場合、デフォルトパスからの変更部分を新しくインストールされた Configuration Manager パスに結合することにより、パスを更新する必要があります。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。ユーザー設定の後、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後、ユーザー設定の前にこのデータソースを挿入します。

Mozilla Adapter

Mozilla アダプタは、`mozilla-apoc-integration` パッケージに含まれています。対応する 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

付録 A – Sun Web Console

既知の問題

セキュリティ

ユーザーの知識が不足している場合、ある種のユーザー操作によりセッションがアクティブ状態のまま放置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 を自動的にログアウトしません。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 のセッションを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

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

-h = 使用状況に関する情報を出力します。

-n = インストールの終了時にサーバーを開始しません。

-u = Sun Web Console をアンインストールします。

-f = setup アプリケーションでこれらのパッケージをインストールした場合、Tomcat と Java 1.4 がアンインストールされます。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。

使用可能な setup パラメータについてさらに詳細は、「`setup -h`」を実行してください。

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関連サブディレクトリのいずれかでこのコマンドを実行しないでください。これらのディレクトリでは、`pkgrm` にエラーが発生します。

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 タグライブラリ
SUNWmcon	Sun Web Console
SUNWmcos	Sun Web Console 向けの共通 Solaris サービス
SUNWmcosx	Sun Web Console 向けの Solaris リリース固有のサービス
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 タグライブラリ
SUNWmcon	Sun Web Console
SUNWmcos	Sun Web Console 向けの共通 Linux サービス
SUNWmcosx	Sun Web Console 向けの Linux リリース固有のサービス
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

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
SUNWapmgo	Gnome 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
apoc-evolution-templates	Evolution テンプレート
apoc-gnome-templates	Gnome 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

第 8 章

付録 C

Configuration Manager で OpenLDAP サーバーを使用する

Configuration Manager データのリポジトリとして OpenLDAP サーバーを使用するには、サーバーのスキーマを拡張して、設定データの格納に使用するオブジェクトクラスと属性を機能として使用できるようにする必要があります。apoc.schema というカスタムスキーマファイルは、Java Desktop System Management Tools CD にある Configuration Manager 配備ツールの openldap サブディレクトリにあります。

このファイルは OpenLDAP 設定ディレクトリ (`/etc/openldap`) の `schema` サブディレクトリにコピーする必要があり、また、そのディレクトリにある `slapd.conf` ファイルにそれを含むことにより OpenLDAP スキーマに追加する必要があります。これは、include `/etc/openldap/schema/apoc.schema` という 1 行をそのファイルにあるスキーマインクルードのシーケンスの最後に挿入することで実行できます。OpenLDAP サーバーのスキーマを拡張する場合の詳細は、サーバーのマニュアルを参照してください。

設定データを格納する OpenLDAP データベースを準備するには、Configuration Manager に提供されている配備ツールを使用する必要があります。そのスキーマを前出のインストール手順により拡張しておき、`createServiceTree` スクリプトだけを実行する必要があります。スクリプトは、`./createServiceTree` コマンドを使用して任意のユーザーとして配備ツールディレクトリから起動する必要があります。このスクリプトはこのマニュアルの配備ツールの節で説明されているように、OpenLDAP データベースに関する情報をユーザーに求めます。OpenLDAP に提供されている代表的なオブジェクトクラスと属性を使用するデフォルトのマップファイルは、配備ツールディレクトリの `openldap` サブディレクトリにあります。ファイル名は `OrganisationalMapping` で、`createServiceTree` を起動する前に、主配備ツールディレクトリにあるファイルと同じ名前でファイルからコピーすることにより、配備できます。

Configuration Manager Agent はデータを必要とするユーザーの DN を提供し、パスワードは提供せずに、匿名で OpenLDAP サーバーに接続しようとすることに注意してください。匿名による認証は、OpenLDAP サーバーの一部のリリースでは無効になります。ただし、OpenLDAP 設定ディレクトリ (`/etc/openldap`) にある `slapd.conf` ファイルに定義されている共通サーバーパラメータの allow `bind_anon_cred` という 1 行を追加することのよって有効にする必要があります。パラメータに関する詳細は、サーバーのマニュアルを参照してください。

Configuration Manager で Active Directory サーバーを使用する

Configuration Manager データのリポジトリとして Active Directory サーバーを使用するには、サーバーのスキーマを拡張して、設定データの格納に使用するオブジェクトクラスと属性を機能として使用できるようにする必要があります。`apoc-ad.ldf` というスキーマ拡張ファイルは、Management Tools CD にある Configuration Manager 配備ツールの `ad` サブディレクトリにあります。詳細は、配備ツールの節を参照してください。

`apoc-ad.ldf` ファイルは次の手順で Active Directory スキーマにインポートする必要があります。

1. スキーマ拡張を有効にします。操作の詳細は、Active Directory の文書を参照してください。
2. Execute the following from the command prompt: `ldifde -i -c "DC=Sun,DC=COM" <Base DN> -f apoc-ad-registry.ldf.`
3. コマンド行で `ldifde -i -c "DC=Sun,DC=COM" <Base DN> -f apoc-ad-registry.ldf` を実行します。

   注**:** <*Base DN*> には Active Directory のベース DN を指定します。

Active Directory サーバーを準備して設定データを格納するには、配備ツールを使用する必要があります。そのスキーマを前出のインストール手順により拡張しておき、`createServiceTree` スクリプトだけを実行する必要があります。スクリプトは、`./createServiceTree` コマンドを使用して任意のユーザーとして配備ツールディレクトリから起動する必要があります。このスクリプトは Active Directory データベースに関する情報をユーザーに求めます。Active Directory に提供されている代表的なオブジェクトクラスと属性を使用するデフォルトのマップファイルは、配備ツールディレクトリの `ad` サブディレクトリにあります。ファイル名は `OrganisationalMapping` で、`createServiceTree` を起動する前に、主配備ツールディレクトリにあるファイルと同じ名前でファイルからコピーすることにより、配備できます。

その位置から、Active Directory サーバーは Configuration Manager で使用できます。Configuration Manager をインストールするときに、ツリーへの読み取りアクセス権を持つユーザーの完全な DN とパスワードを提供します。この場合のユーザーは、他の目的には Active Directory を使用することができません。このようなユーザーを設定する方法については、Active Directory の文書を参照してください。さらに、Active Directory のドメイン名は、Configuration Manager を実行するマシンに認識されている必要があります。これを行うためには、そのドメイン名とともに Active Directory サーバーの IP アドレスをマシンの `/etc/hosts` ファイルに関連付ける 1 行を追加します。

Java Desktop System ホストから設定データを取り出すには、Active Directory のドメイン名がそのホストにも認識されている必要があります。Java Desktop System ユーザーの認証は、匿名か GSSAPI を使用するか、どちらかの方法で可能です。

- anonymous 接続を使用して認証するには、Active Directory サーバーは everyone ユーザーに読み取りアクセス権を与えるように設定されている必要があります。操作方法の詳細は、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 `/etc/krb5.conf` ファイルで、Active Directory レルムを定義し、Key Distribution Center (KDC) として Active Directory サーバーを指定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 がサポートする DES タイプ、つまり `des-cbc-crc` と `des-cbc-md5` を指定する必要があります。操作方法の詳細は、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 にログインしているユーザーの有効な資格を入手している必要があります。これは、手作業で `kinit` コマンドを実行し、Active Directory に定義されている

ユーザーパスワードを入力することで行えます。他のスキーマはログイン時にこれらの資格を生成します。詳細は、Java Desktop System の文書を参照してください。