Sun Java™ System

Access Manager 6
管理ガイド

2005Q1

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

Part No: 819-1938

目 次

本書について

この『Sun Java™ System Access Manager 2005Q1 管理ガイド』では、ユーザーインタフェースとコマンド行インタフェースを使って Sun Java System Acceess Manager (旧称 Sun™ ONE Identity Manager) を管理する方法について説明します。

ここでは、次の項目について説明します。

対象読者

この『管理ガイド』は、IT 管理者、および Sun Java System のサーバーおよびソフトウェアを使用した統合アイデンティティ管理および Web アクセスプラットフォームを実装するソフトウェア開発者向けに書かれています。

本ガイドを読まれる方は、次の技術に精通していることが必要です。

- Sun Java System Directory Server
- Lightweight Directory Access Protocol (LDAP)
- Java™ テクノロジ
- JavaServer Pages™ (JSP) テクノロジ

- HyperText Transfer Protocol (HTTP)
- HTML (HyperText Markup Language)
- XML (eXtensible Markup Language)

お読みになる前に

Access Manager は Sun Java Enterprise System のコンポーネントです。これは、ネットワーク環境またはインターネット環境で配布されるエンタープライズアプリケーションをサポートするソフトウェアインフラストラクチャです。Sun Java Enterprise System のマニュアルを熟読してください。このマニュアルには、次の URL からオンラインでアクセスできます。

http://docs.sun.com/prod/entsys.05q1

Access Manager の配備では、Sun Java System Directory Server をデータストアとして使用するので、Directory Server のマニュアルを熟読する必要があります。このマニュアルには、次の URL からオンラインでアクセスできます。

http://docs.sun.com/coll/DirectoryServer_05q1

表記上の規則

この節の表では、本書で使用する表記規則について説明します。

表記上の規則

次の表では、本書で使用する表記規則について説明します。

表 1 表記上の規則

書体	意味	例
AaBbCc123 (モノスペース)	API とプログラミング言語の要素、HTML タグ、Web サイト URL、コマンド名、ファイル名、ディレクトリパス名、コンピュータの画面出力、サンプルコード。	`.login` ファイルを編集します。 `ls -a` を使用してすべてのファイルを一覧表示します。 `% You have mail.`

表 1　表記上の規則 (続き)

書体	意味	例
AaBbCc123 (太字のモノスペース)	ユーザーが入力する内容で、コンピュータの画面出力と対比するときに使用します。	**% su** Password:
AaBbCc123 (イタリック体)	本来の名前や値で置き換えるコマンドやパス名のプレースホルダ。	ファイルは、*install-dir*/bin ディレクトリに配置されています。

記号

次の表では、本書で使用する記号の表記規則について説明します。

表 2　記号の表記規則

記号	説明	例	意味
[ ]	任意指定のコマンドオプション。	ls [-l]	-l オプションは任意指定です。
{ | }	必須コマンドオプションの選択肢。	-d {y|n}	-d オプションでは、y 引数か n 引数を使用する必要があります。
-	複数のキーの同時操作を表します。	Control-A	Control キーを押した状態で A キーを押すことを表します。
+	複数のキーの連続操作を表します。	Ctrl+A+N	Control キーを押して離してから残りのキーを押すことを表します。
>	グラフィカルユーザーインタフェースにおけるメニュー項目選択を表します。	「ファイル」>「新規」>「テンプレート」	「ファイル」メニューから「新規」を選択します。「新規」サブメニューから「テンプレート」を選択します。

デフォルトパスとファイル名

次の表では、本書で使用するデフォルトパスとファイル名について説明します。

表 3　デフォルトパスとファイル名

用語	説明
AccessManager-base	Access Manager のベースインストールディレクトリを表します。Access Manager のデフォルトのベースインストールディレクトリおよびプロダクトディレクトリは、それぞれのプラットフォームによって異なります。 Solaris™ システム : `/opt/SUNWam` Linux システム : `/opt/sun/identity`
DirectoryServer-base	Sun Java System Directory Server のベースインストールディレクトリを示します。固有のパス名については、製品マニュアルを参照してください。
ApplicationServer-base	Sun Java System Application Server のベースインストールディレクトリを示します。固有のパス名については、製品マニュアルを参照してください。
WebServer-base	Sun Java System Web Server のベースインストールディレクトリを示します。固有のパス名については、製品マニュアルを参照してください。

シェルプロンプト

次の表では、本書で使用するシェルプロンプトについて説明します。

表 4　シェルプロンプト

シェル	プロンプト
UNIX または Linux の C シェル	*machine-name*%
UNIX または Linux の C シェルスーパーユーザー	*machine-name*#
UNIX または Linux の Bourne シェルと Korn シェル	$
UNIX または Linux の Bourne シェルと Korn シェルのスーパーユーザー	#
Microsoft Windows コマンドライン	C:¥

関連マニュアル

Sun の技術文書にオンラインでアクセスするには、http://docs.sun.com を参照してください。

マニュアルアーカイブを参照したり、書名、Part No.、テーマで検索したりすることができます。

このマニュアルセットの資料

表 5　Access Manager 6 2005Q1 のマニュアルセット

書名	説明
『Technical Overview』 http://docs.sun.com/doc/817-7643	Access Manager コンポーネントがどのように連動して、アイデンティティ管理を統合し、企業の資産と Web ベースのアプリケーションを防護するかについての高度な概要を説明します。Access Manager の基本概念と用語について説明します
『配備計画ガイド』 http://docs.sun.com/doc/819-1942?l=ja	既存の情報技術インフラストラクチャ内で配備を計画する方法について説明します。
『管理ガイド』(このガイド) http://docs.sun.com/doc/819-1938?l=ja	Access Manager コンソールの使用法、およびコマンド行でユーザーとサービスのデータを管理する方法について説明します。
『Migration Guide』 http://docs.sun.com/doc/817-7645	既存データと Sun Java System 製品配備を最新バージョンの Access Manager に移行する方法について説明します。Access Manager とその他の製品のインストールとアップグレードについては、『Sun Java Enterprise System 2005Q1 インストールガイド』を参照してください。
『Performance Tuning Guide』 http://docs.sun.com/doc/817-7646	Access Manager と関連コンポーネントの調整方法について説明します。
『Federation Management Guide』 http://docs.sun.com/doc/817-7648	Liberty Alliance Project に基づく連携管理について説明します。
『Developer's Guide』 http://docs.sun.com/doc/817-7649	Access Manager をカスタマイズし、組織の現在の技術インフラストラクチャにその機能を統合するカスタマイズ方法について説明します。製品とその API のプログラムに関連する情報も含まれています。
『Developer's Reference』 http://docs.sun.com/doc/817-7650	Access Manager パブリック C API を構成する、データ型、構造、および関数についてまとめてあります。

表 5　Access Manager 6 2005Q1 のマニュアルセット (続き)

書名	説明
『リリースノート』 http://docs.sun.com/doc/819-1946?l=ja	製品のリリース後、オンラインで利用できます。このリリースの最新情報、既知の問題、制限事項、インストールに関する注意事項、ソフトウェアまたはマニュアルに関する問題の報告方法など、最新情報を提供します。

Access Manager ポリシーエージェントのマニュアル

Access Manager ポリシーエージェントのマニュアルは、次のマニュアル Web サイトで利用できます。

http://docs.sun.com/coll/S1_IdServPolicyAgent_21

Access Manager のポリシーエージェントは、サーバー製品自体とは異なるスケジュールで提供されます。したがって、ポリシーエージェントのマニュアルは、Access Manager の主要マニュアルセットとは別に提供されます。このセットには、次のマニュアルが含まれています。

- 『Policy Agents For Web and Proxy Servers Guide』: 各種の Web サーバーやプロキシサーバーに Access Manager ポリシーエージェントをインストールし、設定する方法について説明します。また、トラブルシューティングや、各エージェントに固有の情報についても説明します。
- 『J2EE Policy Agents Guide』: ホストされている J2EE アプリケーションを保護する Access Manager ポリシーエージェントをインストールし、設定する方法について説明します。また、トラブルシューティングや、各エージェントに固有の情報についても説明します。
- 『リリースノート』は、エージェントセットのリリース後、オンラインでご利用になれます。『リリースノート』では、このリリースの最新情報、既知の問題、制限事項、インストールに関する注意事項、ソフトウェアまたはマニュアルに関する問題の報告方法について説明します。

その他のサーバーマニュアル

その他のサーバーマニュアルについては、次の URL を参照してください。

- Directory Server のマニュアル
  http://docs.sun.com/app/docs/coll/DirectoryServer_05q1_ja
- Web Server のマニュアル
  http://docs.sun.com/app/docs/coll/WebServer_05q1_ja
- Application Server のマニュアル
  http://docs.sun.com/app/docs/coll/ApplicationServer8_ee_04q4l
- Web Proxy Server のマニュアル
  http://docs.sun.com/prod/s1.webproxys#hic

Sun リソースへのオンラインアクセス

製品ダウンロード、プロフェッショナルサービス、パッチとサポート、その他の開発者用情報については、次の URL を参照してください。

ダウンロードセンター
http://wwws.sun.com/software/download/

プロフェッショナルサービス
http://www.sun.com/service/sunps/sunone/index.html

Sun エンタープライズサービスによる Solaris のパッチとサポート
http://sunsolve.sun.com/

開発者用情報
http://developers.sun.com/prodtech/index.html

Sun テクニカルサポートへの連絡方法

この製品に関する技術的な疑問があり、製品マニュアルを調べても解決しない場合は、次の URL を参照してください。

http://www.sun.com/service/contacting

関連するサードパーティの Web サイトの参照

このマニュアルに記載されたサードパーティの Web サイトの利用可能性について Sun は責任を負いません。これらのサイトや情報源を通して入手される内容、広告、製品、およびその他の資料について、Sun は保証することも、賠償責任などの責任を負うこともありません。これらのサイトや情報源を通して入手される内容、物品、およびサービスを使用または信用することにより発生する、または発生したと主張される、実際の、または申し立てられている損害や損失について、Sun は賠償責任などいかなる責任も負いません。

コメントの送付先

Sun では、マニュアルの改善のために、皆様からのコメントおよび提案をお待ちしております。

コメントを送るには、http://docs.sun.com にアクセスして「コメントの送信」をクリックしてください。オンラインフォームに、マニュアルのタイトルと Part No. を入力してください。Part No. は、マニュアルのタイトルページか先頭に記述されている 7 桁または 9 桁の番号です。

たとえば、このマニュアルのタイトルは『Sun Java System Access Manager 6 2005Q1 管理ガイド』で、Part No. は 819-1938 です。

第 I 部

Access Manager の設定

これは『Sun Java™ System Access Manager 6 2005Q1 管理ガイド』の第 1 部です。Access Manager をインストールしたあとに行う設定のオプションについて説明しています。次の章で構成されています。

第 1 章

Access Manager 2005Q1 の設定スクリプト

この章では、amconfig スクリプトとサイレントモード入力ファイルのサンプル (amsamplesilent) を使って Sun Java™ System Access Manager を設定および配備する方法について説明します。内容は次のとおりです。

Access Manager 2005Q1 インストール概要

新たにインストールする場合は、常に、Sun Java Enterprise System インストーラ を実行し、Access Manager 2005Q1 の最初のインスタンスをインストールします。インストーラを実行すると、Access Manager の設定オプションから次のうちどちらかを選択できます。

- 「今すぐ設定」オプションでは、Access Manager インストールパネル上で選択した内容 (またはデフォルトの内容) で、インストール中に最初のインスタンスを設定します。
- 「あとで設定」オプションでは、Access Manager 2005Q1 のコンポーネントをインストールしたあと、「Access Manager のインスタンスを再設定する」の説明のようにして、これらのコンポーネントを設定します。このオプションを選択すると、現在インストールしている製品はどれも設定されません。たとえば Access Manager と Application Server のインストールを選択し、「あとで設定」オプションを選択すると、どちらのアプリケーションも設定されません。

インストーラについての詳細は、『Sun Java Enterprise System 2005Q1 インストールガイド』(http://docs.sun.com/doc/819-0808?l=ja) を参照してください。

注 Access Manager 2005Q1 のバージョンを Solaris で確認するには、Access Manager パッチを検索し、インストールされている Access Manager のバージョンを確認します。次のコマンドを入力してください。

```
# showrev -p | grep SUNWam
```

Java Enterprise System インストーラは Access Manager 2005Q1 の `amconfig` スクリプトとサイレントモード入力ファイルのサンプル (`amsamplesilent`) を、Solaris システムの場合は *AccessManager-base*`/SUNWam/bin` ディレクトリに、Linux システムの場合は *AccessManager-base*`/identity/bin` ディレクトリに、それぞれインストールします。

AccessManager-base は、Access Manager のベースインストールディレクトリを示します。デフォルトのベースインストールディレクトリは、Solaris システムでは `/opt` であり、Linux システムでは `/opt/sun` となります。しかし、インストーラを実行する際、必要に応じて別のディレクトリを指定することもできます。

`amconfig` スクリプトは最上位レベルのスクリプトで、要求された処理を実行する際、必要に応じてほかのスクリプトを呼び出します。詳細は、「Access Manager の amconfig スクリプト」を参照してください。

サイレントモード入力ファイルのサンプル (`amsamplesilent`) は、`amconfig` スクリプトをサイレントモードで実行する際に指定しなければならない入力ファイルの一例です。

このサイレントモード入力ファイルのサンプルは ASCII テキストファイルで、Access Manager の設定変数が格納されます。amconfig スクリプトを実行する前に、amsamplesilent ファイルをコピーしてファイル内の変数を編集します。また、必要に応じてファイル名を変更します。設定変数は次のような構成になっています。

変数名 = 値

次に例を示します。

```
DEPLOY_LEVEL=1
NEW_INSTANCE=true
SERVER_HOST=ishost.example.com
```

設定スクリプト入力ファイルで設定できる変数のリストについては、「Access Manager の設定スクリプト入力ファイルのサンプル」を参照してください。

警告 amconfig スクリプトをサイレントモードで実行するときに使用するサイレントモード入力ファイルの書式は、Java Enterprise System のサイレントインストールの状態ファイルの書式には従わず、また必ずしも同じ変数名を使用するとは限りません。このファイルには、管理パスワードなどの機密データが含まれています。このファイルは、必要に応じてセキュリティ保護するか、削除してください。

Access Manager の amconfig スクリプト処理

Sun Java Enterprise System インストーラを使用して Access Manager の最初のインスタンスをインストールしたあと、amconfig スクリプトを実行して、サイレントモード入力ファイル内の変数の値により、次の処理を行うことができます。

- 同一ホストシステム上に Access Manager の追加インスタンスを配備し設定します。たとえば、Web コンテナの追加のインスタンスを設定したあと、その Web コンテナのインスタンス用の新しい Access Manager インスタンスの配備と設定ができます。
- Access Manager の最初のインスタンスと、すべての追加インスタンスの両方を再設定します。
- Access Manager SDK を配備し設定します。これは次の製品に対するサポートを可能にします。
  - BEA WebLogic Server 6.1 SP4 と SP5
  - BEA WebLogic Server 8.1 SP3
  - IBM WebSphere 5.1

- 特定の Access Manager コンポーネント、たとえばコンソールや連携管理モジュールなどを配備し設定します。
- amconfig スクリプトを使って配備した Access Manager のインスタンスやコンポーネントをアンインストールします。

Access Manager の設定スクリプト入力ファイルのサンプル

Java Enterprise System インストーラの実行後は、Access Manager の設定スクリプト入力ファイルのサンプル (amsamplesilent) が、Solaris システムでは *AccessManager-base*/SUNWam/bin のディレクトリに、Linux システムでは *AccessManager-base*/identity/bin のディレクトリにあります。

設定変数を設定するには、まず、amsamplesilent ファイルをコピーしてファイル名を変更します。次に、コピーしたファイル内の変数を、実行したい処理に合わせて変更します。

このサイレントモード入力ファイルのサンプルには、次のような設定変数があります。

- 配備モード変数
- Access Manager の設定変数
- Web コンテナの設定変数
- Directory Server の設定変数

配備モード変数

表 1-1 では必要な DEPLOY_LEVEL 変数の値を説明しています。この変数は、amconfig スクリプトが実行する処理を規定します。

表 1-1　Access Manager DEPLOY_LEVEL 変数

処理	DEPLOY_LEVEL 変数の値と説明
インストール	1 = 新しいインスタンスに対して、Access Manager を完全インストール (デフォルト)
	2 = Access Manager のコンソールのみをインストール
	3 = Access Manager SDK コンソールのみをインストール
	4 = SDK のみをインストールし、コンテナを設定
	5 = 連携管理モジュールのみをインストール
	6 = サーバーのみをインストール
アンインストール (設定解除)	11 = 完全にアンインストール
	12 = コンソールのみをアンインストール
	13 = SDK のみをアンインストール
	14 = SDK のみをアンインストールし、コンテナの設定を解除
	15 = 連携管理をアンインストール
	16 = サーバーのみをアンインストール
再インストール (再配備または再設定とも呼ぶ)	21 = すべての (コンソール、パスワード、サービス、共通) Web アプリケーションを再配備します。
	26 = すべての (コンソール、パスワード、サービス、共通) Web アプリケーションの配備を取り消します。

Access Manager の設定変数

表 1-2 で、Access Manager の設定変数について説明します。

表 1-2　Access Manager の設定変数

変数	説明
BASEDIR	Access Manager パッケージをインストールするベースディレクトリ。
	デフォルト : PLATFORM_DEFAULT
	Solaris システムでは PLATFORM_DEFAULT は `/opt`
	Linux システムでは PLATFORM_DEFAULT は `/opt/sun`

表 1-2　Access Manager の設定変数 (続き)

変数	説明
SERVER_HOST	Access Manager が実行中 (またはインストール予定) であるシステムの完全修飾ホスト名。 リモート SDK インストールの場合、この変数はリモートクライアントのホストではなく Access Manager がインストールされている (またはする予定の) ホスト。
SERVER_PORT	Access Manager ポート番号。デフォルト : `58080` リモート SDK インストールの場合、この変数はリモートクライアントのホストではなく Access Manager がインストールされている (またはする予定の) ホストのポート番号。
SERVER_PROTOCOL	サーバープロトコル : http または https。デフォルト : `http` リモート SDK インストールの場合、この変数はリモートクライアントのホストではなく Access Manager がインストールされている (またはする予定の) ホストのプロトコル。
CONSOLE_HOST	コンソールがインストールされたサーバーの完全修飾ホスト名。 デフォルト : Access Manager のホストに指定された値 (SERVER_HOST 変数)
CONSOLE_PORT	コンソールがインストールされ、接続待機している Web コンテナのポート。 デフォルト : Access Manager のポートに指定された値 (SERVER_PORT 変数)
CONSOLE_PROTOCOL	コンソールがインストールされた Web コンテナのプロトコル。 デフォルト : サーバープロトコル (SERVER_PROTOCOL 変数)
CONSOLE_REMOTE	Access Manager サービスのコンソールがリモートにある場合は、true に設定。そうでない場合は false に設定。デフォルト : `false`
DS_HOST	Directory Server の完全修飾ホスト名。
DS_PORT	Directory Server のポート。デフォルト : `389`
DS_DIRMGRDN	ディレクトリマネージャの DN: Directory Server への無制限のアクセスを持つユーザー。 デフォルト : `"cn=Directory Manager"`
DS_DIRMGRPASSWD	ディレクトリマネージャのパスワード (DS_DIRMGRDN 変数)。 ADMINPASSWD の表記に特殊文字を使用するには「注」を参照してください。

表 1-2　Access Manager の設定変数 (続き)

変数	説明
ROOT_SUFFIX	ディレクトリの初期またはルートサフィックス。使用中の Directory Server にこの値が必ず存在する必要があります。 ADMINPASSWD の表記に特殊文字を使用するには「注」を参照してください。
ADMINPASSWD	管理者 (amadmin) のパスワード。amldapuser のパスワードとは異なるパスワードにする必要があります。 **注 :** パスワード中に、スラッシュ (/) または円マーク (¥) などの特殊文字が含まれる場合には、これらの文字を引用符 (') で囲む必要があります。次に例を示します。 `ADMINPASSWD='¥¥¥¥¥####///'` ただし、実際のパスワードの文字に引用符を使うことはできません。
AMLDAPUSERPASSWD	amldapuser のパスワード。amadmin のパスワードとは異なるパスワードにする必要があります。 ADMINPASSWD の表記に特殊文字を使用するには「注」を参照してください。
CONSOLE_DEPLOY_URI	Access Manager の管理コンソールサブコンポーネントに関連した HTML ページ、クラス、および JAR ファイルにアクセスするための URI プレフィックス。 デフォルト : /amconsole
SERVER_DEPLOY_URI	アイデンティティ管理とポリシーサービスのコアサブコンポーネントに関連した HTML ページ、クラス、および JAR ファイルにアクセスするための URI プレフィックス。 デフォルト : /amserver
PASSWORD_DEPLOY_URI	Access Manager を実行中の Web コンテナが、入力された文字列と対応する配備アプリケーションとの間で行うマッピングを決める URI。 デフォルト : /ampassword
COMMON_DEPLOY_URI	Web コンテナ上の共通ドメインサービスにアクセスする URI プレフィックス。 デフォルト : /amcommon
COOKIE_DOMAIN	Access Manager がユーザーにセッション ID を付与する場合にブラウザに返す、信頼できる DNS ドメインの名前。少なくとも 1 つの値が存在する必要があります。形式は、通常、ピリオドのあとにサーバーのドメイン名を付けたものになります。 次に例を示します。 .example.com

表 1-2　Access Manager の設定変数 (続き)

変数	説明
JAVA_HOME	JDK インストールディレクトリのパス。デフォルト : /usr/jdk/entsys-j2se この変数は、コマンド行インタフェース (amadmin など) の実行可能ファイルによって使用される JDK を指定します。
AM_ENC_PWD	パスワードの暗号化鍵 : ユーザーパスワードを暗号化するのに Access Manager が使用する文字列です。デフォルト : none。値を「none」に設定した場合、ユーザーのパスワード暗号化鍵は amconfig によって生成されます。このため、ユーザーが指定したインストール、または amconfig によって作成されたインストールには、パスワードの暗号化が存在します。 **重要** : Access Manager またはリモート SDK の複数のインスタンスを配備する場合、すべてのインスタンスに対して同一のパスワード暗号化鍵を使用する必要があります。追加のインスタンスを配備するとき、最初のインスタンスの AMConfig.properties ファイル内の am.encryption.pwd プロパティの値をコピーして使用します。
PLATFORM_LOCALE	プラットフォームのロケール。デフォルト : en_US (米語)
NEW_OWNER	インストール後の Access Manager ファイルの新しい所有者。デフォルト : root
NEW_GROUP	インストール後の Access Manager ファイルの新しいグループ。デフォルト : other Linux へのインストールの場合は、NEW_GROUP の値に root を設定します。
XML_ENCODING	XML のエンコーディング。デフォルト : ISO-8859-1
NEW_INSTANCE	ユーザーが新たに作成した Web コンテナインスタンスに、設定スクリプトが Access Manager を配備するかどうかを指定します。 • true = Java Enterprise System インストーラが作成したインスタンスとは別に、ユーザーが新規に作成した Web コンテナインスタンスに Access Manager を配備します。 • false = インスタンスを再設定します。 デフォルト : false

Web コンテナの設定変数

Access Manager 用の Web コンテナを指定するには、サイレントモード入力ファイル内の WEB_CONTAINER 変数を表 1-3 のように設定します。

表 1-3　Access Manager WEB_CONTAINER 変数

値	Web コンテナ
WS6 (デフォルト)	Sun Java System Web Server 6.1 SP4
AS7	Sun Java System Application Server 7.0 Update 3 (前バージョンの Access Manager との互換性を保つために提供)
AS8	Sun Java System Application Server 8.1
WL6	BEA WebLogic Server 6.1 SP4 および SP5
WL8	BEA WebLogic Server 8.1
WAS4	IBM WebSphere 4.0.5 (前バージョンの Access Manager との互換性を保つために提供)
WAS5	IBM WebSphere 5.1

Sun Java System Web Server 6.1 SP4

表 1-4 では、サイレントモード入力ファイル内の Web Server 6.1 SP4 の設定変数について説明しています。

表 1-4　Web Server 6.1 SP4 設定変数

変数	説明
WS61_INSTANCE	Access Manager が配備される、または配備解除される Web Server インスタンス名。 デフォルト : `https-`*web-server-instance-name* ここで *web-server-instance-name* は Access Manager ホスト (SERVER_HOST 変数) を表します。
WS61_HOME	Web Server のベースインストールディレクトリ。 デフォルト : `/opt/SUNWwbsvr`
WS61_PROTOCOL	WS61_INSTANCE 変数によって設定され、Access Manager が配備される Web Server インスタンスが使用するプロトコル : http または https。 デフォルト : Access Manager プロトコル (SERVER_PROTOCOL 変数)

表 1-4　Web Server 6.1 SP4 設定変数 (続き)

変数	説明
WS61_HOST	Web Server インスタンス (WS61_INSTANCE 変数) 用の完全修飾ホスト名。 デフォルト : Access Manager ホストインスタンス (SERVER_HOST 変数)
WS61_PORT	Web Server が接続を待機しているポート。 デフォルト : Access Manager ポート番号 (SERVER_PORT 変数)
WS61_ADMINPORT	Web Server 管理サーバー が接続を待機しているポート。 デフォルト : `8888`
WS61_ADMIN	Web Server 管理者のユーザー ID。 デフォルト : `"admin"`
WS61_IS_SECURE	セキュリティ保護されたポートが有効になっているかどうかを指定します。 • true: セキュリティ保護されたポート (HTTPS プロトコル) が有効になっています。 • false: セキュリティ保護されたポート (HTTPS プロトコル) が無効になっています。 デフォルト : `false` (無効)

Sun Java System Application Server 7.0 Update 3

表 1-5 は、サイレントモード入力ファイル内の Application Server 7.0 Update 3 の設定変数について説明しています。

表 1-5　Application Server 7.0 Update 3 設定変数

変数	説明
AS70_HOME	Application Server 7.0 がインストールされているディレクトリへのパス。 デフォルト : `/opt/SUNWappserver7`
AS70_PROTOCOL	Application Server インスタンスによって使用されるプロトコル : http または https。 デフォルト : Access Manager プロトコル (SERVER_PROTOCOL 変数)
AS70_HOST	Application Server インスタンスが接続を待機している完全修飾ドメイン名 (FQDN)。 デフォルト : Access Manager ホスト (SERVER_HOST 変数)

表 1-5　Application Server 7.0 Update 3 設定変数 (続き)

変数	説明
AS70_PORT	Application Server インスタンスが接続を待機しているポート。 デフォルト : Access Manager ポート番号 (SERVER_PORT 変数)
AS70_ADMINPORT	Application Server 管理サーバー が接続を待機しているポート。 デフォルト : `4848`
AS70_ADMIN	Application Server が表示されているドメインでの Application Server 管理サーバーの管理者の名前。 デフォルト : `admin`
AS70_ADMINPASSWD	Application Server が表示されているドメインでの Application Server の管理者パスワード。 ADMINPASSWD の表記に特殊文字を使用するには「注」を参照してください。
AS70_INSTANCE	Access Manager を実行する Application Server インスタンスの名前。 デフォルト : `server1`
AS70_DOMAIN	この Access Manager インスタンスを配備したいドメインに対する Application Server ディレクトリへのパス。 デフォルト : `domain1`
AS70_INSTANCE_DIR	Application Server が、インスタンス用のファイルを保存するディレクトリへのパス。 デフォルト : `/var/opt/SUNWappserver7/domains/domain1/server1`
AS70_DOCS_DIR	Application Server がコンテンツ文書を保存するディレクトリ。 デフォルト : `/var/opt/SUNWappserver7/domains/domain1/server1/docroot`
AS70_IS_SECURE	セキュリティ保護されたポートが有効になっているかどうかを指定します。 • true: セキュリティ保護されたポート (HTTPS プロトコル) が有効になっています。 • false: セキュリティ保護されたポート (HTTPS プロトコル) が無効になっています。 デフォルト : `false` (無効) インストール中に、Application Server の admin ポートの SSL が有効になっていると、設定は失敗します。管理サーバーを `https` モードで使用しないでください。

Sun Java System Application Server 8.1

表 1-6 は、サイレントモード入力ファイル内の Application Server 8.1 の設定変数について説明しています。

表 1-6 Application Server 8.1 の設定変数

変数	説明
AS81_HOME	Application Server 8.1 がインストールされているディレクトリへのパス。 デフォルト : `/usr/appserver1`
AS81_PROTOCOL	Application Server インスタンスによって使用されるプロトコル : http または https。 デフォルト : Access Manager プロトコル (SERVER_PROTOCOL 変数)
AS81_HOST	Application Server インスタンスが接続を待機している完全修飾ドメイン名 (FQDN)。 デフォルト : Access Manager ホスト (SERVER_HOST 変数)
AS81_PORT	Application Server インスタンスが接続を待機しているポート。 デフォルト : Access Manager ポート番号 (SERVER_PORT 変数)
AS81_ADMINPORT	Application Server 管理サーバー が接続を待機しているポート。 デフォルト : `4849`
AS81_ADMIN	Application Server が表示されているドメインでの Application Server 管理サーバーの管理者の名前。 デフォルト : `admin`
AS81_ADMINPASSWD	Application Server が表示されているドメインでの Application Server の管理者パスワード。 ADMINPASSWD の表記に特殊文字を使用するには「注」を参照してください。
AS81_INSTANCE	Access Manager を実行する Application Server インスタンスの名前。 デフォルト : `server`
AS81_DOMAIN	この Access Manager インスタンスを配備したいドメインに対する Application Server ディレクトリへのパス。 デフォルト : `domain1`
AS81_INSTANCE_DIR	Application Server が、インスタンス用のファイルを保存するディレクトリへのパス。 デフォルト : `/var//appserver/domains/domain1`

表 1-6　Application Server 8.1 の設定変数 (続き)

変数	説明
AS81_DOCS_DIR	Application Server がコンテンツ文書を保存するディレクトリ。 デフォルト : /var/appserver/domains/domain1/docroot
AS81_IS_SECURE	セキュリティ保護されたポートが有効になっているかどうかを指定します。 • true: セキュリティ保護されたポート (HTTPS プロトコル) が有効になっています。 • false: セキュリティ保護されたポート (HTTPS プロトコル) が無効になっています。 デフォルト : false (無効) ampsamplesilent には、アプリケーションサーバーの管理ポートがセキュリティ保護されているかどうかを指定する追加の設定があります。 • true: アプリケーションサーバーの管理ポートはセキュリティ保護されています (HTTPS プロトコル)。 • false: アプリケーションサーバーの管理ポートはセキュリティ保護されていません (HTTP プロトコル)。 デフォルト : True (有効)。

BEA WebLogic Server 6.1 SP4 および SP5

表 1-7 は、サイレントモード入力ファイル内の BEA WebLogic Server 6.1 の設定変数について説明しています。

表 1-7　BEA WebLogic Server 6.1 SP4 および SP5 設定変数

変数	説明
WL61_HOME	WebLogic のホームディレクトリ。デフォルト : /export/bea61a
WL61_PROJECT_DIR	WebLogic プロジェクトディレクトリ。デフォルト : user_projects
WL61_DOMAIN	WebLogic ドメイン名。デフォルト : mydomain
WL61_SERVER	WebLogic サーバー名。デフォルト : myserver
WL61_INSTANCE	WebLogic インスタンス名。デフォルト : WS61_HOME/wlserver6.1
WL61_PROTOCOL	WebLogic プロトコル。デフォルト : http
WL61_HOST	WebLogic ホスト名。
WL61_PORT	WebLogic ポート。デフォルト : 7001

表 1-7　BEA WebLogic Server 6.1 SP4 および SP5 設定変数 (続き)

変数	説明
WL61_SSLPORT	WebLogic SSL ポート。デフォルト : 7002
WL61_ADMIN	WebLogic 管理者。デフォルト : "system"
WL61_PASSWORD	WebLogic 管理者のパスワード。 ADMINPASSWD の表記に特殊文字を使用するには「注」を参照してください。
WL61_JDK_HOME	WebLogic JDK のホームディレクトリ。デフォルト : WS61_HOME/jdk131

BEA WebLogic Server 8.1

表 1-8 は、サイレントモード入力ファイル内の BEA WebLogic Server 8.1 の設定変数について説明しています。

表 1-8　BEA WebLogic Server 8.1 設定変数

変数	説明
WL8_HOME	WebLogic のホームディレクトリ。デフォルト : /export/bea8
WL8_PROJECT_DIR	WebLogic プロジェクトディレクトリ。デフォルト : projects
WL8_DOMAIN	WebLogic ドメイン名。デフォルト : mydomain
WL8_SERVER	WebLogic サーバー名。デフォルト : myserver
WL8_INSTANCE	WebLogic インスタンス名。デフォルト : /export/bea8/weblogic81
WL8_PROTOCOL	WebLogic プロトコル。デフォルト : http
WL8_HOST	WebLogic ホスト名。デフォルト : none
WL8_PORT	WebLogic ポート。デフォルト : 7001
WL8_SSLPORT	WebLogic SSL ポート。デフォルト : 7002
WL8_ADMIN	WebLogic 管理者。デフォルト : "system"
WL8_PASSWORD	WebLogic 管理者のパスワード。 ADMINPASSWD の表記に特殊文字を使用するには「注」を参照してください。
WL8_JDK_HOME	WebLogic JDK のホームディレクトリ。デフォルト : WL8_HOME/jdk141_03
WL8_CONFIG_LOCATION	WebLogic 起動スクリプトの場所の親ディレクトリに設定する必要があります。

表 1-8　BEA WebLogic Server 8.1 設定変数 (続き)

変数	説明
WL8_IS_SECURE	セキュリティ保護されたポートが有効になっているかどうかを指定します。 • true: セキュリティ保護されたポート (HTTPS プロトコル) が有効になっています。 • false: セキュリティ保護されたポート (HTTPS プロトコル) が無効になっています。 デフォルト : `false` (無効)

IBM WebSphere 5.1

表 1-9 は、サイレントモード入力ファイル内の IBM WebSphere Server 5.1 の設定変数について説明しています。

表 1-9　IBM WebSphere 5.1 設定変数

変数	説明
WAS51_HOME	WebSphere のホームディレクトリ。デフォルト : `/opt/WebSphere/AppServer`
WAS51_JDK_HOME	WebSphere JDK のホームディレクトリ。デフォルト : `/opt/WebSphere/AppServer/java`
WAS51_CELL	WebSphere セル。デフォルト : `sample`
WAS51_DOMAIN	WebSphere ドメイン名。デフォルト : `mydomain`
WAS51_NODE	WebSphere ノード名。デフォルト : WebSphere がインストールされたサーバーのホスト名。デフォルト : `sample`
WAS51_INSTANCE	WebSphere インスタンス名。デフォルト : `server1`
WAS51_PROTOCOL	WebSphere のプロトコル。デフォルト : `http`
WAS51_HOST	WebSphere のホスト名。デフォルト : `sample`
WAS51_PORT	WebSphere のポート。デフォルト : `9080`
WAS51_SSLPORT	WebSphere の SSL ポート。デフォルト : `9081`
WAS51_ADMIN	WebSphere の管理者。デフォルト : `"admin"`
WAS51_ADMINPORT	WebSphere の管理者のポート。デフォルト : `9090`

表 1-9　IBM WebSphere 5.1 設定変数 (続き)

変数	説明
WAS51_IS_SECURE	セキュリティ保護されたポートが有効になっているかどうかを指定します。 • true: セキュリティ保護されたポート (HTTPS プロトコル) が有効になっています。 • false: セキュリティ保護されたポート (HTTPS プロトコル) が無効になっています。 デフォルト : `false` (無効)

Directory Server の設定変数

Access Manager 2005Q1 は Sun ONE Directory Server 5.1 と Sun Java System Directory Server 5 2005Q1 をサポートします。表 1-10 は、サイレントモード入力ファイル内の Directory Server の設定変数について説明します。

表 1-10　Directory Server の設定変数

変数	説明
DIRECTORY_MODE	Directory Server モード 1 = Directory Information Tree (DIT) の新規インストールに使用します。 2 = 既存の DIT に使用します。ネーミング属性とオブジェクトクラスは同一です。したがって、設定スクリプトは `installExisting.ldif` と `umsExisting.ldif` ファイルをロードします。 設定スクリプトは、設定作業中に入力された値 (たとえば、BASE_DIR, SERVER_HOST、ROOT_SUFFIX など) を使って、LDIF やプロパティファイルを更新します。 この更新は、「タグ交換 (Tag Swapping)」とも呼ばれますが、これは設定スクリプトがファイル内のダミーのタグを実際の設定値と入れ替えるためです。 3 = 手動でロードする時、既存の DIT に対して使用します。ネーミング属性とオブジェクトクラスは異なるので、設定スクリプトは `installExisting.ldif` と `umsExisting.ldif` ファイルをロードしません。スクリプトはタグ交換 (上記、モード 2 で説明) を行います。 LDIF ファイルを検査し、必要があれば修正して、LDIF ファイルとサービスを手動でロードします。 4 = 既存のマルチサーバーインストールに使用します。設定スクリプトは LDIF ファイルとサービスをロードしません。これは、この操作が既存の Access Manager のインストールに対するものだからです。スクリプトはタグ交換 (上記、モード 2 で説明) のみを行い、プラットフォームリストにサーバーエントリを追加します。 5 = 既存のアップグレードに使用します。スクリプトはタグ交換 (上記、モード 2 で説明) のみを行います。 デフォルト : `1`
USER_NAMING_ATTR	ユーザーネーミング属性 : 相対的な名前空間内のユーザーまたはリソースの固有識別子。デフォルト : `uid`
ORG_NAMING_ATTR	ユーザーの会社または組織のネーミング属性。デフォルト : `o`
ORG_OBJECT_CLASS	組織オブジェクトクラス。デフォルト : `sunManagedOrganization`
USER_OBJECT_CLASS	ユーザーオブジェクトクラス。デフォルト : `inetOrgPerson`
DEFAULT_ORGANIZATION	デフォルトの組織名。デフォルト : `none`

Access Manager の amconfig スクリプト

Java Enterprise System インストーラを実行後、amconfig スクリプトファイルが、Solaris システムでは *AccessManager-base*/SUNWam/bin のディレクトリに、Linux システムでは *AccessManager-base*/identity/bin のディレクトリにあります。

amconfig スクリプトは、サイレントインストール入力ファイルを読み取り、続いて必要があればサイレントモードでほかのスクリプトを呼び出し、要求された処理を実行します。

amconfig スクリプトを実行するには、次の構文を使います。

```
amconfig -s input-file
```

各表記の意味は次のとおりです。

-s は amconfig をサイレントモードで実行します。

input-file はサイレントインストール入力ファイルで、実行したい操作用の設定変数を含んでいます。詳細は、「Access Manager の設定スクリプト入力ファイルのサンプル」を参照してください。

注

Access Manager 2005Q1 リリースでは、以下のスクリプトをサポートしていません。

- 「create」引数を伴う amserver
- amserver.*instance*

また、デフォルトで、amserver start は、amsecuridd と amunixd ヘルパーの認証のみを開始します。amsecuridd 認証ヘルパーは、Solaris OS SPARC プラットフォームでのみ利用可能です。

Access Manager の配備シナリオ

Java Enterprise System インストーラを使って Access Manager の最初のインスタンスをインストールしたあと、サイレントモード入力ファイル内の設定変数を編集し、amconfig スクリプトを実行することにより、追加の Access Manager インスタンスを配備および設定することができます。

次のシナリオについて説明します。

- Access Manager の追加のインスタンスを配備する
- Access Manager のインスタンスを再設定する
- Access Manager インスタンスをアンインストールする
- すべての Access Manager インスタンスをアンインストールする

Access Manager の追加のインスタンスを配備する

Access Manager の追加のインスタンスを配備する前に、Web コンテナ用管理ツールを使用して Web コンテナのインスタンスを作成し、実行しておく必要があります。詳細は、Web コンテナについての個別のマニュアルを参照してください。

- Web Server 6.1 SP2 に関しては、次のものがあります。
  http://docs.sun.com/db/prod/entsys?l=ja
- Application Server 7.0 Update 3 に関しては、次のものがあります。
  http://docs.sun.com/db/prod/entsys?l=ja

追加の Access Manager インスタンスを配備する

1. そのインスタンスの Web コンテナに応じた、管理者としてログインします。たとえば、Web Server 6.1 が新しいインスタンスの Web コンテナになるのであれば、スーパーユーザー (root) として、または Web Server 管理サーバーのユーザーアカウントでログインします。
2. amsamplesilent ファイルを、書き込み可能なディレクトリにコピーし、そのディレクトリをカレントディレクトリにします。たとえば、/newinstances というディレクトリを作成します。

   ヒント : amsamplesilent ファイルのコピーを、配備する新しいインスタンスにふさわしい名前に変更します。たとえば、以降の手順では Web Server 6.1 に新しいインスタンスをインストールするのに、amnewws6instance という名前の入力ファイルを使用します。

3. 新しい amnewws6instance ファイルで次の変数を設定します。

   ```
   DEPLOY_LEVEL=1
   NEW_INSTANCE=true
   ```

   amnewws6instance ファイル内のほかの変数に対して、新しく作成するインスタンスで必要な設定をします。これらの変数の説明については、次節以降の表を参照してください。

   - Access Manager の設定変数
   - Web コンテナの設定変数
   - Directory Server の設定変数

   重要：Access Manager のインスタンスには、パスワードの暗号鍵はすべて同一のものを使わなければなりません。AM_ENC_PWD 変数を設定するには、最初のインスタンスの AMConfig.properties ファイル内の am.encryption.pwd プロパティの値をコピーします。

   将来、このインスタンスをアンインストールする場合のために、amnewws6instance ファイルを保存しておきます。

4. 新しい amnewws6instance ファイルを指定して、amconfig スクリプトを実行します。たとえば、Solaris システムでは、次のようになります。

   ```
   cd AccessManager-base/SUNWam/bin/
   ./amconfig -s /newinstances/amnewws6instance
   ```

   -s オプションは amconfig スクリプトをサイレントモードで実行します。

   amconfig スクリプトは、amnewws6instance ファイルの変数を使い、必要があればほかの設定スクリプトを呼び出して、新しいインスタンスを配備します。

Access Manager のインスタンスを再設定する

Java Enterprise System インストーラを使用してインストールした Access Manager の最初のインスタンスや、amconfig スクリプトを実行して配備した追加の Access Manager インスタンスを再設定できます。

たとえば、Access Manager の所有者とグループを変更するためにインスタンスを再設定してみます。

Access Manager のインスタンスを再設定する

1. そのインスタンスの Web コンテナに応じた、管理者としてログインします。たとえば、Web Server 6.1 が Web コンテナであれば、スーパーユーザー (root) 、または Web Server 管理サーバーのユーザーアカウントでログインします。

2. インスタンスを配備するのに使用したサイレントインストール入力ファイルを、書き込み可能なディレクトリにコピーし、そのディレクトリをカレントディレクトリにします。たとえば、Web Server 6.1 用のインスタンスを再設定するために、以降の手順では /reconfig ディレクトリ内の入力ファイル amnewinstanceforWS61 を使います。

3. amnewinstanceforWS61 ファイル内の、DEPLOY_LEVEL 変数の「再インストール」操作で説明した値のどれかに設定します。たとえば、完全インストールを再設定するには、DEPLOY_LEVEL=21 とします。

4. amnewinstanceforWS61 ファイルでは、NEW_INSTANCE 変数を false と設定します。

   ```
   NEW_INSTANCE=false
   ```

5. amnewinstanceforWS61 ファイル内のほかの変数も設定します。たとえば、インスタンスの所有者とグループを変更するには、NEW_OWNER と NEW_GROUP を新しい値に変更します。

   以下のその他の変数の説明については、次節以降の表を参照してください。

   - Access Manager の設定変数
   - Web コンテナの設定変数
   - Directory Server の設定変数

6. 編集した入力ファイルを指定して、amconfig スクリプトを実行します。たとえば、Solaris システムでは、次のようになります。

   ```
   cd AccessManager-base/SUNWam/bin/
   ./amconfig -s /reconfig/amnewinstanceforWS61
   ```

   -s オプションはスクリプトをサイレントモードで実行します。amconfig スクリプトは、amnewinstanceforWS61 ファイルの変数を使い、必要があればほかの設定スクリプトを呼び出して、インスタンスを再設定します。

Access Manager インスタンスをアンインストールする

amconfig スクリプトを実行してインストールした Access Manager のインスタンスをアンインストールできます。また、Access Manager インスタンスを一時的に設定解除できますが、Web コンテナインスタンスは削除しないかぎりそのまま残り、後日別の Access Manager インスタンスを再配備する際に使用できます。

Access Manager のインスタンスをアンインストールする

1. そのインスタンスの Web コンテナに応じた、管理者としてログインします。たとえば、Web Server 6.1 が Web コンテナであれば、スーパーユーザー (root) 、または Web Server 管理サーバーのユーザーアカウントでログインします。
2. インスタンスを配備するのに使用したサイレントインストール入力ファイルを、書き込み可能なディレクトリにコピーし、そのディレクトリをカレントディレクトリにします。たとえば、Web Server 6.1 用のインスタンスを設定解除するために、以降の手順では /unconfigure ディレクトリ内の入力ファイル amnewinstanceforWS61 を使います。
3. amnewinstanceforWS61 ファイル内の、DEPLOY_LEVEL 変数の「アンインストール (設定解除)」操作で説明した値のどれかに設定します。たとえば、完全インストールをアンインストール (または、設定解除) するには、DEPLOY_LEVEL=11 とします。
4. 編集した入力ファイルを指定して、amconfig スクリプトを実行します。たとえば、Solaris システムでは、次のようになります。

   ```
   cd AccessManager-base/SUNWam/bin/
   ./amconfig -s /unconfigure/aminstanceforWS61
   ```

   -s オプションはスクリプトをサイレントモードで実行します。amconfig スクリプトは amnewinstanceforWS61 ファイルを読み込み、インスタンスをアンインストールします。

Web コンテナインスタンスはそのまま残っているため、後日別の Access Manager インスタンスを再配備する際に使用できます。

すべての Access Manager インスタンスをアンインストールする

ここでは、すべての Access Manager 2005Q1 のインスタンスとパッケージをシステムから完全に削除します。

Access Manager 2005Q1 をシステムから完全に削除する

1. スーパーユーザー (root) としてログインするか、スーパーユーザー (root) になります。
2. インスタンスを配備するのに使用した入力ファイル中で、DEPLOY_LEVEL 変数の「アンインストール (設定解除)」操作で説明した値のうちの 1 つに設定します。たとえば、完全インストールをアンインストール (または、設定解除) するには、DEPLOY_LEVEL=11 とします。
3. 手順 2 で編集した入力ファイルを使用して、`amconfig` スクリプトを実行します。たとえば、Solaris システムでは、次のようになります。

   ```
   cd AccessManager-base/SUNWam/bin/
   ./amconfig -s /newinstances/amnewws6instance
   ```

   `amconfig` スクリプトはサイレントモードで動作し、インスタンスをアンインストールします。

   アンインストールしたいインスタンスすべてに対してこれらの手順を繰り返します。ただし、Java Enterprise System インストーラを使ってインストールした最初のインスタンスは除きます。
4. 最初のインスタンスをアンインストールし、すべての Access Manager パッケージをシステムから削除するには、Java Enterprise System アンインストーラを実行します。アンインストーラについての詳細は、『Sun Java Enterprise System インストールガイド』を参照してください。

第 2 章

Access Manager を SSL モードに設定する

SSL (Secure Socket Layer) を単純な認証で使用することで、機密性とデータの整合性とが保証されます。Access Manager を SSL モードにするには、通常は次のようにします。

1. Access Manager をセキュリティ保護された Web コンテナで設定する
2. Access Manager をセキュリティ保護された Directory Server に設定する

次の節でこれらの手順を説明します。

セキュリティ保護された Sun Java System Web Server で Access Manager を設定する

Sun Java System Web Server で実行する Access Manager を SSL モードに設定するには、次の手順を参照してください。

1. Access Manager コンソールで、サービス設定モジュールに移動し、「プラットフォーム」サービスを選択します。「サーバーリスト」属性で `http://` プロトコルを削除し、`https://` プロトコルを追加します。「保存」をクリックします。

注	必ず「保存」をクリックしてください。そうしないと、次の手順に進むことはできますが、設定の変更内容はすべて失われ、それを修正するために管理者としてログインすることもできなくなります。

手順 2 ～手順 25 では Sun Java System Web Server について説明します。

2. Web Server コンソールにログオンします。デフォルトのポート番号は、58888 です。
3. Access Manager を実行している Web Server インスタンスを選択し、「Manage」をクリックします。

   設定が変更されたことを知らせるポップアップウィンドウが表示されます。「了解」をクリックします。
4. 画面の右上部にある「Apply」ボタンをクリックします。
5. 「Apply Settings」をクリックします。

   Web Server が自動的に再起動されます。「OK」をクリックして先に進みます。
6. 選択した Web Server インスタンスを停止します。
7. 「Security」タブをクリックします。
8. 「Create Database」をクリックします。
9. 新しいデータベースのパスワードを入力し、「OK」をクリックします。

   あとで使用するために、このデータベースパスワードを書き留めておくようにしてください。
10. 証明書データベースが作成されたら、「Request a Certificate」をクリックします。
11. 画面に表示されるフィールドにデータを入力します。

    「Key Pair Field Password」フィールドは、手順 9 で入力した値と同じ値にします。場所のフィールドには、場所を完全名で入力する必要があります。「CA」などの省略形では動作しません。すべてのフィールドを定義する必要があります。「共通名」フィールドには、使用している Web Server のホスト名を入力します。
12. フォームを送信すると、次のようなメッセージが表示されます。

```
--BEGIN CERTIFICATE REQUEST---

afajsdllwqeroisdaoi234rlkqwelkasjlasnvdknbslajowijalsdkjfalsdflasdf

alsfjawoeirjoi2ejowdnlkswnvnwofijwoeijfwiepwerfoiqeroijeprwpfrwl

--END CERTIFICATE REQUEST--
```

13. このテキストをコピーし、証明書要求として送信します。

    ルート CA 証明書を取得するようにしてください。

14. 証明書の含まれた証明書応答が返されます。たとえば次のようになります。

```
--BEGIN CERTIFICATE---

afajsdllwqeroisdaoi234rlkqwelkasjlasnvdknbslajowijalsdkjfalsdflasdf

alsfjawoeirjoi2ejowdnlkswnvnwofijwoeijfwiepwerfoiqeroijeprwpfrwl

--END CERTIFICATE---
```

15. このテキストをクリップボードにコピーするか、ファイルに保存します。
16. Web Server コンソールで、「Install Certificate」をクリックします。
17. 「Certificate for this Server」をクリックします。
18. 「鍵ペアファイルパスワード」フィールドに、証明書データベースのパスワードを入力します。
19. 証明書を表示されたテキストフィールドに貼り付けます。またはラジオボタンをクリックし、テキストボックスにファイル名を入力します。「送信」をクリックします。

    ブラウザに証明書と、証明書を追加するボタンが表示されます。

20. 「Install Certificate」をクリックします。
21. 「Certificate for Trusted Certificate Authority」をクリックします。

22. 手順 16 ～手順 21 と同じ方法で、ルート CA 証明書をインストールします。
23. 両方の証明書をインストールしたら、Web Server コンソールで「Preferences」タブをクリックします。
24. 別のポートで SSL を有効にする場合は、「Add Listen Socket」を選択します。次に、「Edit Listen Socket」を選択します。
25. セキュリティ状態を「Disabled」から「Enabled」に変更し、「OK」をクリックして変更を実行します。

手順 26 ～手順 28 では、Access Manager について説明します。

26. `AMConfig.properties` ファイルを開きます。このファイルの場所は、デフォルトで `/opt/SUNWam/lib` です。
27. プロトコルで `http://` が出現する箇所をすべて `https://` に変更します。ただし Web Server インスタンスディレクトリの箇所は除きます。これは `AMConfig.properties` でも指定していますが、そのままにしておきます。
28. `AMConfig.properties` ファイルを保存します。
29. Web Server コンソールで、Web Server インスタンスをホスティングする Access Manager の「ON/OFF」ボタンをクリックします。

    Web Server で「Start/Stop」ページにテキストボックスが表示されます。
30. このテキストフィールドに、証明書データベースのパスワードを入力し、「Start」を選択します。

セキュリティ保護された Sun Java System Application Server で Access Manager を設定する

SSL が有効になっている Sun Java System Application Server 上で Access Manager を実行するには、次の 2 つの手順で設定します。まず、インストールされた Access Manager に対して Application Server のインスタンスをセキュリティ保護します。次に、Access Manager 自体を設定します。

Application Server 6.2 を SSL で設定する

Application Server インスタンスをセキュリティで保護するには、次の手順を実行します。

1. ブラウザに次のアドレスを入力して、Sun Java System Application Server コンソールに管理者としてログインします。

```
http://fullservername:port
```

デフォルトのポート番号は、4848 です。

2. インストール時に入力したユーザー名とパスワードを入力します。
3. Access Manager をインストールした (または、これからインストールする) Application Server インスタンスを選択します。設定が変更されたことが、右側のフレームに表示されます。
4. 「変更の適用」をクリックします。
5. 「再起動」をクリックします。Application Server が自動的に再起動されます。
6. 左側のフレームで、「セキュリティ」をクリックします。
7. 「データベースの管理」タブをクリックします。
8. 「データベースを作成」が選択されていない場合は、それをクリックします。
9. 新しいデータベースのパスワードを入力し、確認のパスワードを入力してから、「OK」をクリックします。あとで使用するために、このデータベースパスワードを書き留めておくようにしてください。
10. 証明書データベースが作成されたら、「証明書管理」タブをクリックします。
11. 「要求」リンクが選択されていない場合は、それをクリックします。
12. 証明書要求のデータを次のように入力します。
    a. 新規の証明書か、証明書の書き換えかを選択します。証明書の多くは、一定の期間が過ぎると期限切れになります。書き換え通知を自動的に送信する認証局 (CA) もあります。
    b. 証明書要求を送信する方法を指定します。

       要求を電子メールメッセージで受け取る CA の場合は、「CA 電子メールアドレス」を選択し、CA の電子メールアドレスを入力します。CA のリストを表示するには、「List of Available Certificate Authorities」をクリックします。

       Sun Java System Certificate Server を使用している内部 CA に証明書を要求する場合は、「証明書発行局 URL」をクリックし、Certificate Server の URL を入力します。この URL は、Certificate Server で証明書要求を処理するプログラムを指している必要があります。
    c. 鍵ペアファイルのパスワードを入力します。これは、手順 9 で指定したパスワードです。

d. 次の識別情報を入力します。

「共通名」: ポート番号も含む完全なサーバー名。

「要求者名」: 要求者の名前。

「電話番号」: 要求者の電話番号。

「共通名」: デジタル証明書のインストール先となる Sun Java System Application Server の完全修飾名。

「電子メールアドレス」: 管理者の電子メールアドレス。

「組織名」: 組織の名前。認証局によっては、この属性に入力されたホスト名が、この組織に登録済みのドメインに属していることが必要になります。

「組織単位名」: 部課名など、組織の運営単位の名前。

「地域」: 市区町村の名前。

「州または都道府県名」: 組織がアメリカ合衆国またはカナダで運営されている場合は、その州の名前。省略形は使用しないでください。

「国名」: 国を表す 2 文字の ISO コード。たとえば、アメリカ合衆国のコードは `US` です。

13. 「OK」ボタンをクリックします。次のようなメッセージが表示されます。

```
--BEGIN NEW CERTIFICATE REQUEST---

afajsdllwqeroisdaoi234rlkqwelkasjlasnvdknbslajowijalsdkjfalsdfla

alsfjawoeirjoi2ejowdnlkswnvnwofijwoeijfwiepwerfoiqeroijeprwpfrwl

--END NEW CERTIFICATE REQUEST--
```

14. このテキスト全体をファイルにコピーし、「OK」をクリックします。ルート CA 証明書を取得するようにしてください。
15. CA を選択し、その CA の Web サイトにある指示に従ってデジタル証明書を取得します。証明書は CMS、Verisign、または Entrust.net から取得できます。
16. 認証局からデジタル証明書を受け取ったら、そのテキストをクリップボードにコピーするか、ファイルに保存します。
17. Sun Java System Application Server コンソールに移動し、「インストール」リンクをクリックします。
18. 「証明書」の「このサーバー」を選択します。

19. 「鍵ペアファイルパスワード」フィールドに、証明書データベースのパスワードを入力します。これは、手順 9 で入力したパスワードです。
20. 「メッセージ」テキストフィールドに、証明書をヘッダーも含めて貼り付けるか、ファイル名を入力します。適切なラジオボタンをクリックします。
21. 「OK」ボタンをクリックします。ブラウザに証明書と、証明書を追加するボタンが表示されます。
22. 「サーバー証明書を追加」をクリックします。
23. 手順 10 ～手順 22 と同じ方法で、ルート CA 証明書をインストールします。ただし、手順 18 では、「証明書」の「信頼できる証明書発行局 (CA)」を選択します。
24. 両方の証明書をインストールしたら、左側のフレームで「HTTP サーバー」ノードを展開します。
25. 「HTTP サーバー」の下にある「HTTP リスナー」を選択します。
26. `http-listener-1` を選択します。ソケットの情報がブラウザに表示されます。
27. `http-listener-1` で使用するポートの値を、Application Server のインストール時に入力した値から、より適切な値 (443 など) に変更します。
28. 「SSL/TLS を有効」を選択します。
29. 「証明書のニックネーム」を選択します。
30. 「戻すサーバー名」を指定します。手順 12 で指定した「共通名」と同じにする必要があります。
31. 「保存」をクリックします。
32. Sun Java System Access Manager ソフトウェアをインストールする Application Server インスタンスを選択します。設定が変更されたことが、右側のフレームに表示されます。
33. 「変更の適用」をクリックします。
34. 「再起動」をクリックします。Application Server が自動的に再起動されます。

Application Server 8.1 を SSL で設定する

Application Server インスタンスをセキュリティで保護するには、次の手順を実行します。

1. Application Server インスタンスが停止していることを確認します。
2. asadmin>change-master-password コマンドを使用し、トークンパスワードを変更します。
3. Application Server コンソールに移動し、「設定」>「HTTP サービス」>「HTTP リスナー」を選択します。
4. 有効にするリスナーを右側区画でクリックし、「セキュリティ : 有効」を選択します。
5. certutil がインストールされているかどうかを確認します。
   a. /usr/sfw/bin に移動します。
   b. インストールされていない場合は、次のディレクトリから SUNWtlsu パッケージをインストールします。

      /share/builds/integration/security/SECURITY_3_9_3_03B4/packages/~platform~

   c. シェル環境変数 LD_LIBRARY_PATH

      LD_LIBRARY_PATH には、/usr/lib/mps/secv1 を含める必要があります。
6. certutil を使用し、certdb にインストールされている証明書を確認します。
   a. /var/opt/SUNWappserver/domains/domain1/config に移動します。
   b. certutil -L -d
   c. 次のように出力されます。

      /var/opt/SUNWappserver/domains/domain1/config/% certutil -L -d

   Application Server 8.1 では、インストール時に自己署名サーバー証明書 (ニックネーム s1as) がインストールされ、SSL 対応ポートである 4848、8181 で使用されます。
7. 証明書要求を生成します。構文は次のとおりです。

   ```
   certutil -R -s subj -o cert-request-file [-d certdir] [-P
   dbprefix] [-p phone] [-a]
   ```

   次に例を示します。

   ```
   certutil -R -s "CN=test.company1.com, O=company1.com, C=US" -o
   cert.req -d . -a
   ```
8. 次のコマンドを使用し、CA から証明書を取得します。

```
certutil -A -n cert-name -t trustargs [-d certdir] [-P dbprefix]
[-a] [-i input]
```

9. サーバー証明書をファイルに保存します。
10. 次のコマンド構文を使用し、信頼できる CA 証明書をインストールします。

```
certutil -A -n cert-name -t trustargs [-d certdir] [-P dbprefix]
[-a] [-i input]
```

`cacert.txt` などのファイルに、信頼できる CA 証明書を保存します。

11. `certdb` をリスト表示し、正常にインストールされたことを確認します。次のコマンドを入力してください。

```
/var/opt/SUNWappserver/domains/domain1/config/% certutil -L -d
```

12. Application Server 管理コンソールに移動し、「HTTP リスナー」を選択します。

「一般設定」において、新しいサーバー証明書で HTTP リスナーを設定します。

13. Application Server を再起動します。

Access Manager を SSL モードに設定する

Access Manager を SSL モードに設定する

1. Access Manager コンソールで、サービス設定モジュールに移動し、「プラットフォーム」サービスを選択します。「サーバーリスト」属性で、同じ URL を HTTPS プロトコルで追加し、SSL が有効になっているポート番号を追加します。「保存」をクリックします。

注 Access Manager の 1 つのインスタンスが、2 つのポート (HTTP と HTTPS) で待機しているとき、未処理のまま蓄積されたクッキーを使って Access Manager にアクセスしようとすると、Access Manager は応答しなくなります。この設定はサポートされていません。

---

2. `AMConfig.properties` ファイルを開きます。デフォルトでは次の場所にあります。

```
/etc/opt/SUNWam/config
```

3. プロトコルで `http://` が出現する箇所をすべて `https://` に変更します。また、ポート番号を、SSL が有効になっているポート番号に変更します。
4. `AMConfig.properties` ファイルを保存します。
5. Application Server を再起動します。

セキュリティ保護された BEA WebLogic Server による AMSDK の設定

BEA WebLogic Server は、最初にインストールして Web コンテナとして設定してから、SSL で AMSDK を使用して設定する必要があります。インストールの詳細については、BEA WebLogic Server のマニュアルを参照してください。Access Manager の Web コンテナとして WebLogic を設定するには、31 ページの第 1 章「Access Manager 2005Q1 の設定スクリプト」を参照してください。

セキュリティ保護された WebLogic インスタンスを設定するには、次を実行します。

1. クイックスタートメニューを使用してドメインを作成します。
2. WebLogic インストールディレクトリに移動し、証明書要求を生成します。
3. `vetri_csr.txt` CSR を使用し、サーバー証明書を CA に申請します。
4. 承認証明書を `approvedcert.txt` などのテキストファイルに保存します。
5. 次のコマンドを使用し、`cacerts` でルート CA をロードします。

   ```
   cd jdk141_03/jre/lib/security/
   ```

   ```
   jdk141_03/jre/bin/keytool -keystore cacerts  -keyalg RSA -import
   -trustcacerts -alias "Greenday CA" -storepass changeit -file
   /opt/bea81/cacert.txt
   ```

6. 次のコマンドを使用し、サーバー証明書をロードします。

   ```
   jdk141_03/jre/bin/keytool -import -keystore keystore -keyalg RSA
   -import -trustcacerts -file approvedcert.txt -alias "mykey"
   ```

7. ユーザー名とパスワードを使用し、WebLogic コンソールにログインします。
8. 次の場所を参照します。

   yourdomain> Servers> myserver> Configure Keystores

9. 「Custom Identity」、次に「Java Standard Trust」を選択します。
10. キーストアの場所を入力します。たとえば `/opt/bea81/keystore` のように入力します。
11. キーストアパスワードとキーストアパスフレーズを入力します。次に例を示します。

    キーストアパスワード : `JKS/Java Standard Trust` (WL 8.1 の場合は JKS のみ)

    キーストアパスフレーズ : `changeit`

12. SSL 非公開鍵の別名とパスワードを確認してください。

注　完全な SSL ライセンスを使用しないと、SSL が起動しません。

13. Access Manager では、`AmConfig.properties` の次のパラメータが、インストール中に自動的に設定されます。設定されていない場合は、適切に編集できます。

    ```
    com.sun.identity.jss.donotInstallAtHighestPriority=true [Access Manager 6.3 以上では不要 ]

    com.iplanet.security.SecureRandomFactoryImpl=com.iplanet.am.util.SecureRandomFactoryImpl

    com.iplanet.security.SSLSocketFactoryImpl=netscape.ldap.factory.JSSESocketFactory

    com.iplanet.security.encryptor=com.iplanet.services.util.JCEEncryption2
    ```

    JDK パスが次のようになっている場合は、keytool ユーティリティを使用し、証明書データベースにルート CA をインポートします。

    ```
    com.iplanet.am.jdk.path=/usr/jdk/entsys-j2se
    ```

    次に例を示します。

    ```
    /usr/jdk/entsys-j2se/jre/lib/security

    /usr/jdk/entsys-j2se/jre/bin/keytool -keystore cacerts  -keyalg RSA -import -trustcacerts -alias "machinename" -storepass changeit -file

    /opt/bea81/cacert.txt
    ```

    keytool ユーティリティは次のディレクトリにあります。

    ```
    /usr/jdk/entsys-j2se/jre/bin/keytool
    ```

14. Access Manager `amadmin` コマンド行ユーティリティから `-D"java.protocol.handler.pkgs=com.iplanet.services.comm"` を削除します。

15. Access Manager を SSL モードに設定します。詳細は、63 ページの「Access Manager を SSL モードに設定する」を参照してください。

セキュリティ保護された IBM WebSphere Application Server による AMSDK の設定

IBM WebShpere Server は、最初にインストールして Web コンテナとして設定してから、SSL で AMSDK を使用して設定する必要があります。インストール手順については、WebSphere Server のマニュアルを参照してください。Access Manager の Web コンテナとして WebLogic を設定するには、31 ページの第 1 章「Access Manager 2005Q1 の設定スクリプト」を参照してください。

セキュリティ保護された WebSphere インスタンスを設定するには、次を実行します。

1. Websphere の `/bin` ディレクトリの `ikeyman.sh` を起動します。
2. 「Signer」メニューから認証局 (CA) からの証明書をインポートします。
3. 「Personal Certs」メニューから CSR を生成します。
4. 前の手順で作成された証明書を取得します。
5. 「Personal Certificates」を選択し、サーバー証明書をインポートします。
6. WebSphere コンソールからデフォルト SSL 設定を変更し、暗号を選択します。
7. デフォルトの IBMJSSE SSL プロバイダを設定します。
8. 次のコマンドを入力し、作成したファイルからアプリケーションサーバー JVM キーストアに、ルート CA 証明書をインポートします。

   ```
   $ appserver_root-dir/java/bin/ keytool -import -trustcacerts -alias
   cmscacert -keystore ../jre/lib/security/cacerts -file
   /full_path_cacert_filename.txt
   ```

   app-server-root-dir はアプリケーションサーバーのルートディレクトリであり、*full_path_cacert_filename.txt* は、証明書を含むファイルのフルパスです。

9. Access Manager において、AmConfig.properties の次のパラメータを、JSSE を使用するように更新します。

```
com.sun.identity.jss.donotInstallAtHighestPriority=true

com.iplanet.security.SecureRandomFactoryImpl=com.iplanet.am.util.Se
cureRandomFactoryImpl

com.iplanet.security.SSLSocketFactorImpl=netscape.ldap.factory.JSSE
SocketFactory

com.iplanet.security.encyptor=com.iplanet.services.unil.JCEEncrypti
on
```

10. Access Manager を SSL モードに設定します。詳細は、63 ページの「Access Manager を SSL モードに設定する」を参照してください。

Access Manager を SSL モードの Directory Server に設定する

ネットワーク上でセキュリティ保護された通信を確保するため、Access Manager には LDAPS 通信プロトコルが含まれています。LDAPS は LDAP の標準プロトコルで、SSL (Secure Socket Layer) 上で実行されます。SSL 接続を有効にするためには、まず Directory Server を SSL モードにして、次に Access Manager を Directory Server に接続します。基本的な手順は次のとおりです。

1. Directory Server 用の証明書を入手してインストールし、Directory Server が認証局 (CA) からの証明書を信頼するように設定します。
2. SSL をオンにします。
3. SSL が有効化された Directory Service に接続するよう、認証、ポリシーおよびプラットフォームサービスを設定します。
4. セキュリティ保護された状態で Directory Server に接続できるよう Access Manager を設定します。

Directory Server を SSL モードに設定する

Directory Server を SSL モードに設定するには、サーバー証明書を入手してインストールし、CA からの証明書を信頼するように Directory Server を設定し、SSL をオンにしなければなりません。これらの作業をどのように行うかについての詳細は、『Directory Server 管理ガイド』の中の第 11 章「認証と暗号化の管理」を参照してください。このマニュアルは、次の場所にあります。

http://docs.sun.com/doc/817-7161?l=ja

また、PDF 形式のマニュアルを次の場所からダウンロードできます。

http://docs.sun.com/coll/DirectoryServer_04q2

Directory Server の SSL がすでに有効になっている場合は次の節に進んでください。そこで Access Manager を Directory Server に接続する方法の詳細について説明します。

SSL が有効化された Directory Server に Access Manager を接続する

Directory Server が SSL モードに設定されたら、Access Manager をセキュリティ保護された状態で Directory Server に接続する必要があります。そのためには、次の手順を実行します。

1. Access Manager コンソールで、「サービス設定」モジュールの LDAP 認証サービスに移動します。
   a. Directory Server ポートを SSL ポートに変更します。
   b. LDAP サーバーへの SSL アクセスを有効」属性を選択します。
2. 「サービス設定」モジュールのメンバーシップ認証サービスに移動します。
   a. Directory Server ポートを SSL ポートに変更します。
   b. 「LDAP サーバーへの SSL アクセスを有効」属性を選択します。
3. 「サービス設定」の「ポリシー設定」サービスに移動します。
   a. Directory Server ポートを SSL ポートに変更します。
   b. 「SSL の LDAP サーバーへのアクセスを有効」属性を選択します。
4. テキストエディタで `serverconfig.xml` を開きます。このファイルは、次の場所にあります。

   `/etc/opt/SUNWam/config`

   a. `<Server>` 要素で、次の値を変更します。

      `port` - Access Manager が待機するセキュリティ保護されたポート番号 (デフォルト値は 636) を指定します。

      `type` - SIMPLE を SSL に変更します。

   b. `serverconfig.xml` を保存して閉じます。
5. `AMConfig.properties` ファイルを開きます。デフォルトでは次の場所にあります。

   `AcessManager-base``/etc/opt/SUNWam/config`

   次のプロパティを変更します。

   a. `Directory Port = 636` (デフォルト値を使う場合)
   b. `ssl.enabed = true`
   c. `AMConfig.properties` を保存します。
6. サーバーを再起動します。

第 II 部

コンソールからの Access Manager の管理

これは『Sun Java™ System Access Manager 6 2005Q1 管理ガイド』の第 2 部です。Access Manager のグラフィカルユーザーインタフェースと、その使用方法について説明しています。次の章で構成されています。

第 3 章

アイデンティティ (識別情報) 管理

この章では、Sun Java™ System Access Manager 6 2005Q1 のアイデンティティ管理機能について説明します。アイデンティティ管理モジュールインタフェースでは、すべての Access Manager オブジェクトおよびアイデンティティを表示、管理、および設定する方法を提供します。この章は、次の節で構成されています。

Access Manager コンソール

Access Manager コンソールは、ロケーション区画、ナビゲーション区画、データ区画の 3 つの部分で構成されます。これら 3 つの区画をすべて活用することで、管理者はディレクトリを移動したり、ユーザーおよびサービスを設定したり、ポリシーを作成したりすることができます。

ヘッダー区画

ヘッダー区画はコンソールの上部にあります。ヘッダー区画にあるタブを使用すると、さまざまな管理モジュールの表示に切り換えることができます。

- 「アイデンティティ管理」モジュール - アイデンティティ関連のオブジェクトを作成および管理することができます。
- 「サービス設定」モジュール - Access Manager のデフォルトサービスを設定できます。
- 「現在のセッション」モジュール - 管理者は、現在のセッション情報を参照したり、任意のセッションを終了したりできます。

- 「連携管理」モジュール - Liberty Alliance Project で策定した連携ネットワークアイデンティティのオープンスタンダードを利用できます。

「場所」フィールドは、ディレクトリツリー内の管理者の位置までの経路です。このパスはナビゲーションのために使用します。

「ようこそ」フィールドは、現在コンソールを実行しているユーザーの名前を、ユーザープロファイルへのリンク付きで表示します。

「検索」リンクは、特定の Access Manager オブジェクトタイプのエントリを検索できるインタフェースを表示します。プルダウンメニューを使用してオブジェクトタイプを選択し、検索文字列を入力します。結果は検索テーブルに表示されます。ワイルドカードも使用できます。

「ヘルプ」リンクは、ブラウザのウィンドウを開きます。このウィンドウにはアイデンティティ管理、現在のセッション、連携管理、およびこのマニュアルの第 IV 部である「属性リファレンスガイド」についての情報があります。

「ログアウト」リンクは、ユーザーが Access Manager からログアウトできます。

ナビゲーション区画

ナビゲーション区画は、Access Manager コンソールの左部分の区画です。ディレクトリオブジェクト部分 (グレーのボックス内) には、現在開かれているディレクトリオブジェクトの名前と、そのプロパティへのリンクが表示されます。ナビゲーション区画に表示されるオブジェクトのほとんどには、対応するプロパティのリンクがあります。このリンクを選択すると、右側のデータ区画にそのエントリの属性が表示されます。「表示」メニューでは、選択したディレクトリオブジェクト配下のディレクトリが一覧表示されます。サブディレクトリ数によっては、ページ移動のメカニズムが用意されます。

データ区画

データ区画は、コンソールの右部分の区画です。すべてのオブジェクト属性とその値を表示および設定できるほか、それぞれのグループ、ロール、組織に対してエントリを選択できる場所です。

ヒント　「すべて選択」または「すべてを選択解除」アイコンをクリックすると、リスト内のすべての項目を選択または選択解除できます。

Access Manager グラフィカルユーザーインタフェースには、基本的なビューが 2 つあります。ログインしているユーザーのロールによって、アイデンティティ管理ビューまたはユーザープロファイルビューにアクセスできます。

アイデンティティ管理ビュー

管理者のロールを持つユーザーが Access Manager に認証されると、デフォルトのビューはアイデンティティ管理ビューになります。このビューでは、管理者は管理タスクを実行できます。管理者のロールに応じて実行できる管理タスクには、オブジェクト (ユーザー、組織、ポリシーなど) の作成、削除、管理、およびサービスの設定が含まれます。

ユーザープロファイルビュー

管理者のロールを割り当てられていないユーザーが Access Manager に認証されると、デフォルトのビューは各自のユーザープロファイルになります。このビューでは、各自の個人プロファイルに固有の属性値を修正できます。これには名前、ホームアドレス、パスワード以外にも、さまざまな属性が含まれます。ユーザープロファイルビューに表示される属性は拡張できます。オブジェクトおよびアイデンティティのカスタマイズした属性を追加するには、『Access Manager Developer's Guide』を参照してください。

プロパティ機能

エントリのプロパティを表示または修正するには、オブジェクト名の隣にある「プロパティ」の矢印をクリックします。属性とその値がデータ区画に表示されます。オブジェクトが異なると表示されるプロパティも異なります。

エントリのプロパティを拡張する詳細については、『Access Manager Developer's Guide』を参照してください。

アイデンティティ管理インタフェース

アイデンティティ管理インタフェースでは、アイデンティティ関連のオブジェクトを作成および管理することができます。Access Manager コンソールまたはコマンド行インタフェースを使用して、ユーザー、ロール、グループ、ポリシー、組織、サブ組織、およびコンテナの各オブジェクトを定義、修正、または削除できます。コンソールにはデフォルト管理者がいます。組織、グループ、コンテナ、ユーザー、サービス、ポリシーを作成し管理するための権限は、管理者によって異なります。ロールに基づいて、管理者を追加作成できます。管理者は Access Manager とともにインストールされると、Directory Server 内に定義されます。

Access Manager オブジェクトの管理

ユーザー管理インタフェースには、Access Manager オブジェクト (組織、グループ、ユーザー、サービス、ロール、ポリシー、コンテナ、エージェント) の表示および管理に必要なすべてのコンポーネントが含まれています。この節では、オブジェクトタイプと、それらを設定する方法の詳細について説明します。

ほとんどの Access Manager オブジェクトタイプで、「表示オプション」と「利用可能なアクション」を設定して Access Manager コンソールに表示する Web インタフェースを表示または非表示にすることができます。設定は、組織およびロールのレベルで行い、ユーザーは、所属する組織または割り当てられたロールがある組織から設定を継承します。設定については、この章の終わりの方で説明します。

組織

組織は、企業が部門とリソースの管理に使用する最上位レベルの階層構造を表します。インストール時に、Access Manager は最上位レベルの組織 (インストール時に定義) をダイナミックに作成して、Access Manager の企業構成を管理します。インストール後に組織を追加作成して、企業を個別に管理できます。作成した組織はすべて、最上位レベルの組織の下に入ります。

組織を作成する

1. アイデンティティ (識別情報) 管理モジュールの「表示」メニューから、「組織」を選択します。
2. ナビゲーション区画で「新規」をクリックします。
3. フィールドの値を入力します。「名前」だけが必須です。フィールドは次のとおりです。

   「名前」: 組織の名前の値を入力します。

「ドメイン名」: ドメインネームシステム (DNS) を使用している場合は、DNS の完全な名前を入力します。

「組織の状態」:「アクティブ」または「非アクティブ」の状態を選択します。

デフォルトは「アクティブ」です。これは、その組織の存続期間中であればいつでも、「プロパティ」アイコンを選択して変更できます。「非アクティブ」を選択すると、その組織にログインした場合、ユーザーアクセスが無効になります。

「組織のエイリアス」: このフィールドでは、組織のエイリアス名を指定します。URL ログインで、認証にエイリアスを使用できるようになります。たとえば exampleorg という組織があり、エイリアスとして 123 および abc を指定すると、次の URL を使用して組織にログインできます。

```
http://machine.example.com/amserver/UI/Login?org=exampleorg
```

```
http://machine.example.com/amserver/UI/Login?org=abc
```

```
http://machine.example.com/amserver/UI/Login?org=123
```

組織のエイリアス名は、組織全体で一意でなければなりません。「一意の属性リスト」を使用して一意性を実現できます。

「DNS エイリアス名」: 組織の DNS 名にエイリアス名を追加できます。この属性では、実際のドメインエイリアスだけを使用できます。ランダムな文字列は使用できません。たとえば example.com という DNS があり、exampleorg という組織のエイリアスとして example1.com および example2.com を指定すると、次の URL を使用して組織にログインできます。

```
http://machine.example.com/amserver/UI/Login?org=exampleorg
```

```
http://machine.example1.com/amserver/UI/Login?=org=exampleorg
```

```
http://machine.example2.com/amserver/UI/Login?org=exampleorg
```

「一意の属性リスト」: 組織内のユーザー用に一意の属性名リストを追加できます。たとえば、電子メールアドレスを指定する一意の属性名を追加した場合、同一の電子メールアドレスを持つ 2 人のユーザーを作成することができなくなります。このフィールドには、コンマ区切りのリストも指定できます。リスト内の属性名は、どれも一意性を定義します。たとえば、このフィールドに次の属性名リストが指定されたとします。

```
PreferredDomain, AssociatedDomain
```

また、特定のユーザーに対して、PreferredDomain は http://www.example.com と定義されています。この場合、コンマ区切りのリスト全体が、その URL に関して一意であると定義されます。

すべてのサブ組織で一意性が要求されます。

4. 「了解」をクリックします。

   新しい組織がナビゲーション区画に表示されます。組織の作成時に定義したプロパティを編集するには、編集対象の組織の「プロパティ」の矢印をクリックし、データ区画の「表示」メニューから「一般」を選択し、プロパティを編集して「了解」をクリックします。「表示オプション」表示および「利用可能なアクション」表示を使用して、Access Manager コンソールの外観をカスタマイズし、この組織に対して認証されるユーザーの動作を指定します。

組織を削除する

1. アイデンティティ管理で、「表示」メニューから「組織」を選択します。

   作成されたすべての組織が表示されます。特定の組織を表示するには、検索文字列を入力して「検索」をクリックします。

2. 削除する組織名の横にあるチェックボックスを選択します。

3. 「削除」をクリックします。

注	削除を実行するときに警告メッセージは表示されません。組織内のエントリがすべて削除されます。この操作を元に戻すことはできません。

ポリシーに組織を追加する

Access Manager オブジェクトは、ポリシーのサブジェクト定義を通じてポリシーに追加されます。ポリシーを作成または修正するときに、ポリシーの「サブジェクト」ページで、組織、ロール、グループ、ユーザーをサブジェクトとして定義できます。サブジェクトを定義すると、ポリシーがオブジェクトに適用されます。詳細は、119 ページの「ポリシーを管理する」を参照してください。

グループ

グループは、共通の機能、特徴、または関心事を持つユーザーの集まりを表します。通常、このグループには関連付けられた権限はありません。グループは、組織内および管理されているほかのグループ内という、2 つのレベルに存在できます。ほかのグループ内に存在するグループは、サブグループと呼ばれます。サブグループは、親グループ内に「物理的に」存在する子ノードです。

Access Manager は、入れ子グループもサポートします。入れ子グループは、1 つのグループに含まれる既存のグループを表します。サブグループとは対照的に、入れ子グループは DIT 内のどこにでも存在できます。入れ子グループは、多数のユーザーに対するアクセス権の設定を簡単にします。

グループを作成するときには、「加入によるメンバーシップ」(スタティックグループ) または「フィルタのメンバーシップ」(フィルタを適用したグループ) を使用するグループを作成できます。これは、ユーザーをグループに追加する方法を制御します。ユーザーはスタティックグループのみに追加できます。ダイナミックグループは、フィルタを使用してユーザーの追加を制御します。入れ子グループまたはサブグループは、両方に追加できます。

スタティックグループ (「加入によるメンバーシップ」)

加入によるグループメンバーシップを指定すると、指定した管理されているグループタイプを基に、スタティックなグループが作成されます。管理されているグループタイプの値が `static` (スタティック) の場合は、`groupOfNames` または `groupOfUniqueNames` オブジェクトクラスを使用して、グループメンバーがグループエントリに追加されます。管理されているグループタイプの値が `dynamic` (ダイナミック) の場合は、LDAP フィルタを使用して、`memberof` 属性を含むユーザーエントリだけを検索して返します。詳細は、245 ページの「管理されているグループタイプ」を参照してください。

注 管理されているグループタイプのデフォルトは dynamic です。このデフォルトは、管理サービス設定で変更できます。

フィルタを適用したグループ (フィルタのメンバーシップ)

フィルタを適用したグループは、LDAP フィルタを使用して作成したダイナミックグループです。エントリはすべてフィルタを通してまとめられ、グループにダイナミックに割り当てられます。フィルタはエントリの属性を検索して、その属性を含むエントリを返します。たとえば、建物番号に基づいてグループを作成する場合、フィルタを使用すると建物番号属性を含むすべてのユーザーのリストを返します。

注 Access Manager は、Directory Server とともに参照整合性プラグインを使用するように設定されている必要があります。参照整合性プラグインが有効になっているときは、削除操作や名前の変更操作の直後に、指定された属性について整合性更新が実行されます。これにより、関連するエントリどうしの関係がデータベース全体で維持されます。Directory Server では、データベースインデックスによって検索パフォーマンスが向上します。このプラグインを有効にする方法の詳細は、『Sun Java System Access Manager Migration Guide』を参照してください。

スタティックグループを作成する

1. グループを作成する組織、グループ、またはグループコンテナに移動します。
2. 「表示」メニューから「グループ」を選択します。

3. 「新規」をクリックします。
4. データ区画のグループタイプに「加入によるメンバーシップ」を選択します。
5. 「グループ名」フィールドにグループの名前を入力します。「次へ」をクリックします。
6. 「ユーザーのグループ加入を有効」属性を選択すると、ユーザーが自分でそのグループに加入できるようになります。
7. DIT に複数のグループコンテナを定義し、(管理サービスから)「グループコンテナを表示」属性を無効にしている場合は、スタティックグループが所属する親グループコンテナを選択できます。それ以外の場合は、このフィールドは表示されません。
8. 「終了」をクリックします。

   グループを作成すると、データ区画の「表示」メニューから「一般」を選択して「ユーザーのグループ加入を有効」属性を編集できます。

スタティックグループのメンバーを追加または削除する

1. メンバーを追加するグループの横にあるプロパティの矢印をクリックします。
2. データ区画で、「表示」メニューから「メンバー」を選択します。

   「アクションの選択」メニューで実行するアクションを選択します。実行できるアクションは次のとおりです。

   「新規ユーザー」: このアクションでは、新規ユーザーが作成され、ユーザー情報の保存時にユーザーがグループに自動的に追加されます。

   「ユーザーを追加」: このアクションでは、既存のユーザーがグループに追加されます。このアクションを選択する場合、追加するユーザーを指定する検索条件を作成します。条件の作成に使用するフィールドでは、「いずれか」または「すべて」演算子を使用します。「すべて」は、指定したすべてのフィールドに一致するユーザーを返します。「いずれか」は、指定したいずれか 1 つのフィールドに一致するユーザーを返します。フィールドを空白のままにすると、そのフィールドはその特定の属性に対して可能なすべてのエントリと一致します。

   検索条件を作成したら、「次へ」をクリックします。返されたユーザーのリストから、追加するユーザーを選択し、「終了」をクリックします。

ヒント ユーザーの完全な組織パスを表示するには、「パスを表示」ボタンをクリックします。

「グループを追加」: このアクションでは、入れ子グループが現在のグループに追加されます。このアクションを選択すると、検索範囲、グループの名前 ("*" ワイルドカードを使用可能) を含む検索条件を作成し、ユーザーがグループそのものに加入できるかどうかを指定できます。情報を入力したら、「次へ」をクリックします。返されたグループのリストから、追加するグループを選択し、「終了」をクリックします。

「メンバーを消去」: このアクションでは、グループからメンバー (ユーザーとグループを含む) が消去されますが、削除はされません。消去するメンバーを選択し、「利用可能なアクション」リストから「メンバーを消去」を選択します。

「メンバーを削除」: このアクションでは、選択したメンバーが永久に削除されます。削除するメンバーを選択し、「利用可能なアクション」リストから「メンバーを削除」を選択します。

フィルタを適用したグループを作成する

1. グループを作成する組織またはグループに移動します。
2. 「表示」メニューから「グループ」を選択します。
3. 「新規」をクリックします。
4. データ区画内からグループタイプに「フィルタのメンバーシップ」を選択します。
5. 「グループ名」フィールドにグループの名前を入力します。「次へ」をクリックします。
6. LDAP 検索フィルタを作成します。

   デフォルトでは、Access Manager は基本検索フィルタインタフェースを表示します。フィルタの作成に使用する基本フィールドでは、「いずれか」または「すべて」演算子を使用します。「すべて」は、指定したすべてのフィールドに一致するユーザーを返します。「いずれか」は、指定したいずれか 1 つのフィールドに一致するユーザーを返します。フィールドを空白のままにすると、そのフィールドはその特定の属性に対して可能なすべてのエントリと一致します。

   「高度」ボタンを選択すると、フィルタ属性自体を定義できます。例を示します。

   ```
   (&(uid=user1)(|(inetuserstatus=active)(!(inetuserstatus=*))))
   ```

   「終了」をクリックすると、検索条件に一致するすべてのユーザーが自動的にグループに追加されます。

フィルタを適用したグループのメンバーを追加または削除する

1. メンバーを追加するグループの横にあるプロパティの矢印をクリックします。
2. データ区画で、「表示」メニューから「メンバー」を選択します。

   「アクション」メニューで実行するアクションを選択します。実行できるアクションは次のとおりです。

「グループを追加」：このアクションでは、入れ子グループが現在のグループに追加されます。このアクションを選択すると、検索範囲、グループの名前 ("*" ワイルドカードを使用可能) を含む検索条件を作成し、ユーザーがグループそのものに加入できるかどうかを指定できます。情報を入力したら、「次へ」をクリックします。返されたグループのリストから、追加するグループを選択し、「終了」をクリックします。

「メンバーを消去」：このアクションでは、グループからメンバー (グループを含む) が消去されますが、削除はされません。消去するメンバーを選択し、「利用可能なアクション」リストから「メンバーを消去」を選択します。

「メンバーを削除」：このアクションでは、選択されたメンバーが永久に削除されます。削除するメンバーを選択し、「利用可能なアクション」リストから「メンバーを削除」を選択します。

ポリシーにグループを追加する

Access Manager オブジェクトは、ポリシーのサブジェクト定義を通じてポリシーに追加されます。ポリシーを作成または修正するときに、ポリシーの「サブジェクト」ページで、組織、ロール、グループ、ユーザーをサブジェクトとして定義できます。サブジェクトを定義すると、ポリシーがオブジェクトに適用されます。詳細は、119 ページの「ポリシーを管理する」を参照してください。

ユーザー

ユーザーは、個人のアイデンティティを表します。Access Manager のアイデンティティ管理モジュールを使用して、組織、コンテナ、およびグループに対するユーザーの作成と削除、ロールやグループに対するユーザーの追加と削除が可能です。サービスをユーザーに割り当てることもできます。

注 `amadmin` と同じユーザー ID でサブ組織のユーザーを作成すると、`amadmin` のログインが失敗します。このような問題が起こったら、管理者はディレクトリサーバーコンソールを使って、そのユーザーの ID を変更します。これで、管理者がデフォルトの組織にログインできるようになります。さらに、認証サービスの「ユーザー検索の開始 DN」をピープルコンテナ DN に設定し、ログイン処理で一意の一致が返されるようにもできます。

ユーザーを作成する

1. ユーザーを作成する組織、コンテナ、またはピープルコンテナに移動します。
2. 「表示」メニューから「ユーザー」を選択します。
3. 「新規」をクリックします。

「新規ユーザー」ページがデータ区画に表示されます。

4. ユーザーが使用できるサービスがある場合は、ユーザーが加入するサービスを「利用可能なサービス」ページから選択します。このページを省略する場合は、「次へ」をクリックします。
5. 次のデフォルトの必須値のデータを入力します。

   「ユーザー ID」: このフィールドは、ユーザーが Access Manager へログインするために使用する名前を取得します。このプロパティは、DN 以外の値になることがあります。

   「名」: このフィールドはユーザーの名 (ファーストネーム) を取得します。名 (ファーストネーム) の値と姓 (ラストネーム) の値によって、Access Manager コンソールの右上隅にあるログイン名を示すフィールドのユーザーが識別されます。これは必須の値ではありません。

   「姓」: このフィールドはユーザーの姓 (ラストネーム) を取得します。名 (ファーストネーム) の値と姓 (ラストネーム) の値によって、Access Manager コンソールの右上隅にあるログイン名を示すフィールドのユーザーが識別されます。

   「フルネーム」: このフィールドはユーザーのフルネームを取得します。

   「パスワード」: このフィールドには、「ユーザー ID」フィールドで指定した名前のパスワードを取得します。

   「パスワード (確認)」: パスワードを確認します。

   「ユーザー状態」: このオプションは、Access Manager による認証をユーザーに許可するかどうかを指定します。アクティブなユーザーだけが Access Manager を使用して認証を受けることができます。デフォルト値は「アクティブ」です。
6. 「終了」をクリックします。

ロールおよびグループにユーザーを追加する

1. ユーザーを修正する組織に移動します。
2. 「表示」メニューから「ユーザー」を選択します。
3. ナビゲーション区画で、修正するユーザーを選択し、「プロパティ」の矢印をクリックします。
4. データ区画の「表示」メニューから、「ロール」または「グループ」を選択します。ユーザーにすでに割り当てられているロールおよびグループだけが表示されます。「追加」をクリックし、選択可能な使用できるロールおよびグループのリストを表示します。
5. ユーザーを追加するロールまたはグループを選択し、「保存」をクリックします。

ユーザーにサービスを追加する

1. ユーザーを修正する組織に移動します。
2. ナビゲーション区画の「表示」メニューから「ユーザー」を選択します。
3. ナビゲーション区画で、修正するユーザーを選択し、「プロパティ」の矢印をクリックします。
4. データ区画の「表示」メニューから「サービス」を選択します。ユーザーが利用可能なサービスのリストが「サービスを追加」ページに表示されます。
5. ユーザーに割り当てるサービスを選択します。
6. 「了解」をクリックします。

   サービスの属性を編集するには、サービス名の横にある「編集」リンクをクリックします。編集可能なサービスの場合のみ「編集」リンクが表示されます。

ユーザーを消去する

1. データ区画の「表示」メニューから、「ロール」または「グループ」を選択します。
2. 「選択」リストからユーザーを消去するロールまたはグループを選択し、「消去」をクリックします。「すべて消去」をクリックして、すべての利用可能なロールおよびグループからユーザーを消去することもできます。
3. 「保存」をクリックして、ユーザーを消去します。

注	削除操作の前に警告メッセージは表示されず、削除操作を元に戻すことはできません。

ポリシーにユーザーを追加する

Access Manager オブジェクトは、ポリシーのサブジェクト定義を通じてポリシーに追加されます。ポリシーを作成または修正するときに、ポリシーの「サブジェクト」ページで、組織、ロール、グループ、ユーザーをサブジェクトとして定義できます。サブジェクトを定義すると、ポリシーがオブジェクトに適用されます。詳細は、119 ページの「ポリシーを管理する」を参照してください。

サービス

組織またはコンテナのサービスを有効にするには、2 つの処理が必要です。コンテナは組織と同様の働きをします。最初の手順で、サービスを組織に追加する必要があります。サービスを追加後、その組織専用に設定されたテンプレートを設定する必要があります。詳細は、第 IV 部「属性リファレンスガイド」を参照してください。

注	新しいサービスは、まずコマンド行の `amadmin` を使用して Access Manager にインポートする必要があります。サービスの XML スキーマのインポートについては、『Access Manager Developer's Guide』を参照してください。

サービスを追加する

1. サービスを追加する組織に移動します。
2. 「表示」メニューから「サービス」を選択します。
3. 「追加」をクリックします。

   この組織に追加可能なサービスのリストがデータ区画に表示されます。
4. 追加する各サービスの横にあるチェックボックスを選択します。
5. 「了解」をクリックします。追加されたサービスがナビゲーション区画に表示されます。

注	親組織に追加されているサービスだけがサブ組織レベルで表示されます。

サービス用のテンプレートを作成する

1. 追加したサービスがある組織またはロールに移動します。

   アイデンティティ管理モジュールで「表示」メニューから「組織」を選択し、ナビゲーション区画から組織を選択します。
2. 「表示」メニューから「サービス」を選択します。
3. 有効にするサービス名の横にあるプロパティアイコンをクリックします。

   「現在このサービスにはテンプレートが存在しません。新規に作成しますか ?」というメッセージがデータ区画に表示されます。
4. 「はい」をクリックします。

このサービス用のテンプレートが親の組織またはロール用に作成されます。このサービスのデフォルト属性と値がデータ区画に表示されます。デフォルトサービスの属性については、241 ページの「属性リファレンスガイド」で説明しています。

5. デフォルト値を受け入れるか、または変更して、「保存」をクリックします。

サービスを消去する

1. サービスを消去する組織に移動します。

   アイデンティティ管理モジュールで「表示」メニューから「組織」を選択し、ナビゲーション区画から組織を選択します。

2. 「表示」メニューから「サービス」を選択します。
3. 消去するサービスのチェックボックスを選択します。
4. 「消去」をクリックします。

注 サービスがサブ組織のレベルで登録されている場合は、親組織のレベルでそのサービスを消去することはできません。

ロール

ロールとは、グループの概念に似た、Directory Server の 1 つのエントリメカニズムです。グループにはメンバーがあるように、ロールにもメンバーがあります。ロールのメンバーは、ロールを持つ LDAP エントリです。ロール自体の基準は、LDAP エントリの属性で定義されます。このエントリは、エントリの識別名 (DN) 属性で特定されます。Directory Server にはさまざまなタイプのロールがありますが、Access Manager で管理できるのは、管理ロールだけです。

注 そのほかの Directory Server ロールタイプもディレクトリの配備で使用できますが、Access Manager コンソールで管理することはできません。ポリシーのサブジェクト定義にほかの Directory Server タイプを使用することもできます。ポリシーサブジェクトについての詳細は、116 ページの「ポリシーの作成」を参照してください。

ユーザーには 1 つ以上のロールを持たせることができます。たとえば、セッションサービスとパスワードリセットサービスの属性を持つ契約社員ロールを作成できます。管理者は契約社員エントリの別の属性を設定しなくても、新しい契約社員が開始すると、契約社員にこのロールを割り当てることができます。契約社員がエンジニアリング部門に属し、エンジニアリング従業員に適用可能なサービスとアクセス権が必要な場合は、管理者は契約社員にエンジニアリングロールおよび契約社員ロールを割り当てることができます。

Access Manager では、ロールを使用して、アクセス制御の命令を適用します。Access Manager をはじめてインストールしたときに、管理者アクセス権を定義するアクセス制御命令 (ACI) が定義されます。次にこれらの ACI をロール (組織管理者ロール、組織ヘルプデスク管理者ロールなど) に割り当てます。このロールをユーザーに割り当てると、ユーザーのアクセス権が定義されます。

ユーザーは、管理サービスで「ユーザーのロールを表示」属性が有効である場合だけ、割り当てられたロールを確認できます。詳細は、254 ページの「ユーザープロファイルページにロールを表示」を参照してください。

注 Access Manager は、Directory Server とともに参照整合性プラグインを使用するように設定されている必要があります。参照整合性プラグインが有効になっているときは、削除操作や名前の変更操作の直後に、指定された属性について整合性更新が実行されます。これにより、関連するエントリどうしの関係がデータベース全体で維持されます。Directory Server では、データベースインデックスによって検索パフォーマンスが向上します。このプラグインを有効にする方法の詳細は、『Sun Java System Access Manager Migration Guide』を参照してください。

グループ同様に、ロールもフィルタで作成することも、スタティックに作成することもできます。

「スタティックロール」: フィルタロールとは対照的に、スタティックロールはユーザーをロールの作成時に追加しなくても作成できます。これにより、特定のユーザーを指定されたロールに追加するときの制御がより細かくできます。

「フィルタロール」: フィルタロールは、LDAP フィルタを使用して作成したダイナミックロールです。ユーザーはすべてフィルタを通してまとめられ、ロールの作成時にそのロールに割り当てられます。フィルタはエントリの属性と値のペア (`ca=user*` など) を検索して、その属性を含むユーザーをロールに自動的に割り当てます。

スタティックロールを作成する

1. ナビゲーション区画で、ロールを作成する組織に移動します。
2. 「表示」メニューから「ロール」を選択します。

組織の構成時にデフォルトのロールが作成され、ナビゲーション区画に表示されます。デフォルトのロールは次のとおりです。

コンテナヘルプデスク管理者 (Container Help Desk Admin): コンテナのヘルプデスク管理者ロールは、組織単位のすべてのエントリに対する読み取りアクセス権、およびそのコンテナ単位だけにあるユーザーエントリの `userPassword` 属性に対する書き込みアクセス権を持っています。

組織のヘルプデスク管理者 (Organization Help Desk Admin): 組織のヘルプデスク管理者は、組織のすべてのエントリに対する読み取りアクセス権、および `userPassword` 属性に対する書き込みアクセス権を持っています。

注 サブ組織を作成するときは、管理者ロールは親組織ではなくサブ組織に作成してください。

コンテナ管理者 (Container Admin): コンテナ管理者ロールは、LDAP 組織単位のすべてのエントリに対する読み取りアクセス権と書き込みアクセス権を持っています。Access Manager では、LDAP 組織単位をコンテナと呼ぶことがあります。

組織ポリシー管理者 (Organization Policy Admin): 組織のポリシー管理者は、組織のすべてのポリシーに対する読み取りアクセス権と書き込みアクセス権を持っており、組織内のすべてのポリシーについて作成、割り当て、修正、および削除ができます。

ピープルコンテナ管理者 (People Container Admin): デフォルトで、新規に作成した組織のユーザーエントリはその組織のピープルコンテナのメンバーです。ピープルコンテナ管理者は、組織のピープルコンテナのすべてのユーザーエントリに対する読み取りアクセス権と書き込みアクセス権を持っています。なお、このロールは、ロールおよびグループ DN を含む属性に対する読み取りアクセス権と書き込みアクセス権を持っていないため、ロールまたはグループの属性を変更したり、ロールまたはグループからユーザーを消去したりすることができません。

注 ほかのコンテナは、Access Manager とともに設定して、ユーザーエントリ、グループエントリ、またはほかのコンテナを保持することができます。組織を構成したあとで、作成されたコンテナに管理者ロールを適用するには、デフォルトのコンテナ管理者ロールまたはコンテナヘルプデスク管理者を使用します。

グループ管理者 (Group Admin): グループ管理者は、特定グループのすべてのメンバーに対する読み取りアクセス権および書き込みアクセス権を持っており、新しいユーザーの作成、管理しているグループへのユーザーの割り当て、および作成したユーザーの削除を行うことができます。

グループを作成すると、そのグループを管理するのに必要な権限を持つグループ管理者ロールが自動的に作成されます。このロールはグループのメンバーに自動的には割り当てられません。グループの作成者、またはグループ管理者ロールへのアクセス権を持つ人が割り当てる必要があります。

最上位レベル管理者 (Top-level Admin): 最上位レベル管理者は、最上位レベル組織のすべてのエントリに対する読み取りアクセス権と書き込みアクセス権を持っています。言い換えれば、最上位レベル管理者ロールには、Access Manager アプリケーション内のすべての設定主体に対する権限があります。

組織管理者 (Organization Admin): 組織管理者は、組織のすべてのエントリに対する読み取りアクセス権と書き込みアクセス権を持っています。組織を作成すると、その組織を管理するのに必要な権限を持つ組織管理者ロールが自動的に作成されます。

3. ナビゲーション区画で「新規」をクリックします。「新規ロール」テンプレートがデータ区画に表示されます。
4. 「スタティックロール」を選択し、名前を入力します。「次へ」をクリックします。
5. ロールの詳細を入力します。
6. 「タイプ」メニューからロールのタイプを選択します。

   ロールは、管理者ロールまたはサービスロールにすることができます。ロールのタイプは、Access Manager コンソールでどこからユーザーを開始するかをコンソールで決定するために使用します。管理者ロールは、ロールの所有者が管理者権限を持っていることをコンソールに通知します。サービスロールは、その所有者がエンドユーザーであることをコンソールに通知します。

7. 「アクセス権」メニューから、ロールに適用する権限のデフォルトセットを選択します。これは、組織内のエントリにアクセスする権限です。ここで示すデフォルトの権限は順不同です。権限は次のとおりです。

   アクセス権なし (No permission): ロールにアクセス権が設定されません。

   組織管理者 (Organization Admin): 組織管理者は設定済み組織のすべてのエントリに対する読み取りアクセス権と書き込みアクセス権を持っています。

   組織のヘルプデスク管理者 (Organization Help Desk Admin): 組織のヘルプデスク管理者は、設定済み組織のすべてのエントリに対する読み取りアクセス権、および `userPassword` 属性に対する書き込みアクセス権を持っています。

   組織ポリシー管理者 (Organization Policy Admin): 組織のポリシー管理者は、組織のすべてのポリシーに対する読み取りアクセス権と書き込みアクセス権を持っています。組織のポリシー管理者は、ピア組織に対する参照ポリシーを作成できません。

   一般に、「アクセス権なし」ACI をサービスロールに割り当て、管理者ロールにはデフォルト ACI のいずれかを割り当てます。

8. 「終了」をクリックします。

作成されたロールがナビゲーション区画に表示され、ロールのステータス情報がデータ区画に表示されます。

「表示」メニューで「表示オプション」と「利用可能なアクション」を選択して設定することもできます。詳細は、この章の終わりの方にある「表示オプション」および「利用可能なアクション」を参照してください。

スタティックロールにユーザーを追加する

1. 修正するロールを選択し、「プロパティ」の矢印をクリックします。
2. データ区画の「表示」メニューから「ユーザー」を選択します。
3. 「追加」をクリックします。
4. 検索条件を入力します。表示される 1 つ以上のフィールドを基に、ユーザーの検索方法を選択できます。フィールドは次のとおりです。

   「一致」: 演算子を含めるフィルタのフィールドに、演算子を含めることができます。「すべて」は、指定したすべてのフィールドに一致するユーザーを返します。「いずれか」は、指定したいずれか 1 つのフィールドに一致するユーザーを返します。

   「名」: 名 (ファーストネーム) でユーザーを検索します。

   「ユーザー状態」: ユーザーの状態 (アクティブまたは非アクティブ) でユーザーを検索します。

   「ユーザー ID」: ユーザー ID でユーザーを検索します。

   「姓」: 姓 (ラストネーム) でユーザーを検索します。

   「フルネーム」: フルネームでユーザーを検索します。
5. 「次へ」をクリックすると、検索が始まります。検索結果が表示されます。
6. ユーザー名の横にあるチェックボックスを選択して、返された名前の中からユーザーを選択します。
7. 「終了」をクリックします。

   これで、ユーザーがロールに割り当てられます。

フィルタロールを作成する

1. ナビゲーション区画で、ロールを作成する組織に移動します。
2. 「表示」メニューから「ロール」を選択します。

   組織の構成時にデフォルトのロールが作成され、ナビゲーション区画に表示されます。デフォルトのロールは次のとおりです。

コンテナヘルプデスク管理者 (Container Help Desk Admin): コンテナのヘルプデスク管理者ロールは、組織単位のすべてのエントリに対する読み取りアクセス権、およびそのコンテナ単位だけにあるユーザーエントリの `userPassword` 属性に対する書き込みアクセス権を持っています。

組織のヘルプデスク管理者 (Organization Help Desk Admin): 組織のヘルプデスク管理者は、組織のすべてのエントリに対する読み取りアクセス権、および `userPassword` 属性に対する書き込みアクセス権を持っています。

注 サブ組織を作成するときは、管理者ロールは親組織ではなくサブ組織に作成してください。

コンテナ管理者 (Container Admin): コンテナ管理者ロールは、LDAP 組織単位のすべてのエントリに対する読み取りアクセス権と書き込みアクセス権を持っています。Access Manager では、LDAP 組織単位をコンテナと呼ぶことがあります。

組織ポリシー管理者 (Organization Policy Admin): 組織のポリシー管理者は、組織のすべてのポリシーに対する読み取りアクセス権と書き込みアクセス権を持っており、組織内のすべてのポリシーについて作成、割り当て、修正、および削除ができます。

ピープルコンテナ管理者 (People Container Admin): デフォルトで、新規に作成した組織のユーザーエントリはその組織のピープルコンテナのメンバーです。ピープルコンテナ管理者は、組織のピープルコンテナのすべてのユーザーエントリに対する読み取りアクセス権と書き込みアクセス権を持っています。なお、このロールは、ロールおよびグループ DN を含む属性に対する読み取りアクセス権と書き込みアクセス権を持っていないため、ロールまたはグループの属性を変更したり、ロールまたはグループからユーザーを消去したりすることができません。

注 ほかのコンテナは、Access Manager とともに設定して、ユーザーエントリ、グループエントリ、またはほかのコンテナを保持することができます。組織を構成したあとで、作成されたコンテナに管理者ロールを適用するには、デフォルトのコンテナ管理者ロールまたはコンテナヘルプデスク管理者を使用します。

グループ管理者 (Group Admin): グループ管理者は、特定グループのすべてのメンバーに対する読み取りアクセス権および書き込みアクセス権を持っており、新しいユーザーの作成、管理しているグループへのユーザーの割り当て、および作成したユーザーの削除を行うことができます。

グループを作成すると、そのグループを管理するのに必要な権限を持つグループ管理者ロールが自動的に作成されます。このロールはグループのメンバーに自動的には割り当てられません。グループの作成者、またはグループ管理者ロールへのアクセス権を持つ人が割り当てる必要があります。

最上位レベル管理者 (Top-level Admin): 最上位レベル管理者は、最上位レベル組織のすべてのエントリに対する読み取りアクセス権と書き込みアクセス権を持っています。言い換えれば、最上位レベル管理者ロールには、Access Manager アプリケーション内のすべての設定主体に対する権限があります。

組織管理者 (Organization Admin): 組織管理者は、組織のすべてのエントリに対する読み取りアクセス権と書き込みアクセス権を持っています。組織を作成すると、その組織を管理するのに必要な権限を持つ組織管理者ロールが自動的に作成されます。

3. ナビゲーション区画で「新規」をクリックします。「新規ロール」テンプレートがデータ区画に表示されます。
4. 「フィルタロール」を選択し、名前を入力します。「次へ」をクリックします。
5. ロールの詳細を入力します。
6. 「タイプ」メニューからロールのタイプを選択します。

   ロールは、管理者ロールまたはサービスロールにすることができます。ロールのタイプは、Access Manager コンソールでどこからユーザーを開始するかをコンソールで決定するために使用します。管理者ロールは、ロールの所有者が管理者権限を持っていることをコンソールに通知します。サービスロールは、その所有者がエンドユーザーであることをコンソールに通知します。
7. 「アクセス権」メニューから、ロールに適用する権限のデフォルトセットを選択します。
8. これは、組織内のエントリにアクセスする権限です。ここで示すデフォルトの権限は順不同です。権限は次のとおりです。

   アクセス権なし (No permission): ロールにアクセス権が設定されません。

   組織管理者 (Organization Admin): 組織管理者は設定済み組織のすべてのエントリに対する読み取りアクセス権と書き込みアクセス権を持っています。

   組織のヘルプデスク管理者 (Organization Help Desk Admin): 組織のヘルプデスク管理者は、設定済み組織のすべてのエントリに対する読み取りアクセス権、および `userPassword` 属性に対する書き込みアクセス権を持っています。

   組織ポリシー管理者 (Organization Policy Admin): 組織のポリシー管理者は、組織のすべてのポリシーに対する読み取りアクセス権と書き込みアクセス権を持っています。組織のポリシー管理者は、ピア組織に対する参照ポリシーを作成できません。

   一般に、「アクセス権なし」ACI をサービスロールに割り当て、管理者ロールにはデフォルト ACI のいずれかを割り当てます。
9. 検索条件を入力します。フィールドは次のとおりです。

「一致」: 演算子を含めるフィルタのフィールドに、演算子を含めることができます。「すべて」は、指定したすべてのフィールドに一致するユーザーを返します。「いずれか」は、指定したいずれか 1 つのフィールドに一致するユーザーを返します。

「名」: 名 (ファーストネーム) でユーザーを検索します。

「ユーザー状態」: ユーザーの状態 (アクティブまたは非アクティブ) でユーザーを検索します。

「ユーザー ID」: ユーザー ID でユーザーを検索します。

「姓」: 姓 (ラストネーム) でユーザーを検索します。

「フルネーム」: フルネームでユーザーを検索します。

「高度」ボタンを選択すると、フィルタ属性自体を定義できます。例を示します。

```
(&(uid=user1)(|(inetuserstatus=active)(!(inetuserstatus=*))))
```

フィルタを空白のままにすると、次のロールが作成されます。

```
(objectclass = inetorgperson)
```

ロール作成プロセスを取り消すには、「取消し」をクリックします。

10. 「終了」をクリックして、フィルタ条件を基に、検索を開始します。そのフィルタ条件で定義されたユーザーがロールに自動的に割り当てられます。

    「表示」メニューで「表示オプション」と「利用可能なアクション」を選択して設定することもできます。詳細は、この章の終わりの方にある「表示オプション」および「利用可能なアクション」を参照してください。

注 ロールプロファイルページやユーザープロファイルページを使用して、ユーザーをスタティックロールに追加することもできます。

ロールからユーザーを消去する

1. 変更するロールを含む組織に移動します。

   アイデンティティ管理モジュールで「表示」メニューから「組織」を選択し、ナビゲーション区画から組織を選択します。

2. 「表示」メニューから「ロール」を選択します。
3. 変更するロールを選択します。
4. 「表示」メニューから「ユーザー」を選択します。
5. 消去する各ユーザーの横にあるチェックボックスを選択します。
6. 「消去」をクリックします。

これで、ロールからユーザーが消去されます。

ポリシーにロールを追加する

Access Manager オブジェクトは、ポリシーのサブジェクト定義を通じてポリシーに追加されます。ポリシーを作成または修正するときに、ポリシーの「サブジェクト」ページで、組織、ロール、グループ、ユーザーをサブジェクトとして定義できます。サブジェクトを定義すると、ポリシーがオブジェクトに適用されます。詳細は、119 ページの「ポリシーを管理する」を参照してください。

ロールへのサービスをカスタマイズする

ロールで利用可能なサービス、およびそのサービス属性に対するアクセスレベルをロール単位でカスタマイズできます。ロール固有の値を属性に設定して、利用可能な各サービスを 1 つのロール用にカスタマイズできます。個々のサービスおよびサービスの個々の属性に対するアクセスを許可することもできます。特定のタイプのユーザー (たとえば、部長) にだけアクセスを許可するサービスが必要な場合があります。これを実現するには、すべてのユーザーにサービスを割り当てますが、ロールに属する部長タイプにだけ特定のサービスへのアクセスを許可します。

同じ考え方がサービス属性にも当てはまります。ユーザーのアカウントは多くの属性から構成され、たとえばアカウントの有効期限など、それらの属性の一部にユーザーがアクセスを許可されていない場合があります。アカウントの管理者にはこの属性のアクセスが許可されますが、ユーザー (アカウントの所有者) には許可されません。サービスおよび属性へのアクセスのカスタマイズは、ナビゲーション区画でロールの「サービス」表示を使用して行います。

サービスを表示するには、サービスを組織レベルで先に追加する必要があります。ロールに追加されたユーザーは、ロールのサービス属性を継承します。

サービスを設定する

1. ロールの「サービス」表示で、「このロールのサービス設定」というラベルが付いたセクションに進みます。
2. サービス名の横にある「編集」リンクをクリックして、そのロールで利用可能にするサービスを選択します。

   サービステンプレートを作成していない場合は、作成するように要求されます。「はい」をクリックします。
3. サービス属性を変更します。特定のサービスの属性の詳細については、このマニュアルの第 3 部「属性リファレンスガイド」を参照してください。
4. 「保存」をクリックします。

注	サービスへのアクセスを拒否すると (選択されていない場合)、サービスはこのロールを持つユーザーの Access Manager コンソールに表示されません。さらに、ユーザーの登録または登録解除、ユーザーへのサービスの割り当て、またはサービステンプレートの作成、削除、表示、修正ができなくなります。

属性へのアクセスをカスタマイズする

1. ロールの「サービス」表示で、「このロールのサービスアクセス」というラベルが付いたセクションに進みます。
2. 変更対象のサービスに対して有効または無効状態を選択します。有効は、アクセスの変更を許可します。無効は、アクセスの変更を許可しません。
3. 「アクセスを変更」リンクをクリックします。
4. 「読み取り／書き込み」または「読み取り専用」チェックボックスを選択し、その属性へのアクセスレベルを割り当てます。
5. 「了解」をクリックしてから、「保存」をクリックします。

特定のサービスの属性の詳細については、このマニュアルの第 4 部「属性リファレンス」を参照してください。

ポリシーにロールを追加する

Access Manager オブジェクトは、ポリシーのサブジェクト定義を通じてポリシーに追加されます。ポリシーを作成または修正するときに、ポリシーの「サブジェクト」ページで、組織、ロール、グループ、ユーザーをサブジェクトとして定義できます。サブジェクトを定義すると、ポリシーがオブジェクトに適用されます。詳細は、119 ページの「ポリシーを管理する」を参照してください。

ロールを削除する

1. 削除するロールを含む組織に移動します。
2. アイデンティティ管理で「表示」メニューから「組織」を選択し、ナビゲーション区画から組織を選択します。デフォルトの最上位組織と選択した組織がロケーションパスに表示されます。
3. 「表示」メニューから「ロール」を選択します。
4. ロール名の横にあるチェックボックスを選択します。
5. 「削除」をクリックします。

ポリシー

ポリシーでは、組織の Web リソースを保護するためのルールを定義します。ポリシーの作成、修正、および削除はアイデンティティ管理モジュールを使用して実行しますが、これらの手順は 116 ページの「ポリシーの作成」で説明します。

エージェント

Access Manager ポリシーエージェントは、Web サーバーおよび Web プロキシサーバー上のコンテンツを無許可の侵入から保護します。管理者が設定したポリシーに基づいてサービスおよび Web リソースへのアクセスを制御します。

エージェントオブジェクトは、ポリシーエージェントプロファイルを定義します。また、Access Manager が Access Manager リソースを保護する特定のエージェントの認証情報およびその他のプロファイル情報を保存することを可能にします。Access Manager コンソールを使用して、管理者はエージェントプロファイルを表示、作成、変更、および削除できます。

エージェントを作成する

1. 作成するエージェントを含む組織に移動します。
2. 「表示」メニューから「エージェント」を選択します。
3. 「新規」をクリックします。
4. フィールドの値を入力します。「名前」だけが必須です。フィールドは次のとおりです。

   「名前」: エージェントの名前またはアイデンティティを入力します。この名前を使用してエージェントは Access Manager にログインします。1 バイト文字による名前のみ受け付けます。

   「パスワード」: エージェントのパスワードを入力します。このパスワードは、LDAP 認証時にエージェントが使用するパスワードと一致する必要があります。

   「パスワードの確認」: パスワードを確認します。

   「説明」: エージェントの簡単な説明を入力します。たとえば、エージェントインスタンスの名前またはエージェントが保護するアプリケーションの名前を入力します。

   「エージェントキー値」: キーと値のペアでエージェントのプロパティを設定します。Access Manager はこのプロパティを使用して、ユーザーに関する資格情報アサーションへのエージェントの要求を受け取ります。現在、1 つのプロパティだけが有効であり、その他のプロパティはすべて無視されます。次の形式を使用します。

agentRootURL=http://*server_name:port*/

「デバイスの状態」: エージェントのデバイスの状態を入力します。「アクティブ」に設定すると、エージェントは Access Manager に対して認証を行い、通信できます。「非アクティブ」に設定すると、エージェントは Access Manager に対して認証できません。

5. 「了解」をクリックします。

エージェントを削除する

1. 削除するエージェントを含む組織に移動します。
2. 「表示」メニューから「エージェント」を選択します。
3. エージェントの名前の横にあるチェックボックスを選択します。
4. 「削除」をクリックします。

コンテナ

コンテナエントリは、オブジェクトクラスおよび属性が異なるために組織エントリが使用できない場合に使用します。Access Manager コンテナエントリと Access Manager 組織エントリは、必ずしも LDAP オブジェクトクラス `organizationalUnit` および `organization` と同等とはかぎらないことに留意してください。これらは抽象アイデンティティエントリです。可能であれば、コンテナエントリではなく組織エントリを使用します。

注 コンテナの表示は必要に応じて行います。コンテナを表示するには、サービス設定モジュールで「表示メニューにコンテナを表示」を選択します。詳細は、245 ページの「表示メニューにコンテナを表示」を参照してください。

コンテナを作成する

1. コンテナを作成する組織またはコンテナに移動します。

   「表示」メニューから「コンテナ」を選択します。

2. 「新規」をクリックします。

   コンテナのテンプレートがデータ区画に表示されます。

3. 作成するコンテナの名前を入力します。
4. 「了解」をクリックします。

「表示」メニューで「表示オプション」と「利用可能なアクション」を選択して設定することもできます。詳細は、この章の終わりの方にある「表示オプション」および「利用可能なアクション」を参照してください。

コンテナを削除する

1. 削除対象のコンテナを含む組織またはコンテナに移動します。
2. 「表示」メニューから「コンテナ」を選択します。
3. 削除するコンテナ名の横にあるチェックボックスを選択します。
4. 「削除」をクリックします。

注 コンテナを削除すると、そのコンテナに含まれるオブジェクトがすべて削除されます。すべてのオブジェクトとサブコンテナが対象になります。

ピープルコンテナ

ピープルコンテナはデフォルトの LDAP 組織単位です。ユーザーはすべて、組織内で作成されるときにその組織単位に割り当てられます。ピープルコンテナは組織レベルにあり、サブピープルコンテナとしてピープルコンテナレベルにあります。ピープルコンテナにはほかのピープルコンテナとユーザーだけを含めることができます。必要に応じて、ピープルコンテナを組織に追加することができます。

注 ピープルコンテナの表示は必要に応じて行います。ピープルコンテナを表示するには、サービス設定モジュールで「ピープルコンテナを表示」を選択します。詳細は、244 ページの「ピープルコンテナを表示」を参照してください。

ピープルコンテナの作成

1. ピープルコンテナを作成する組織またはピープルコンテナに移動します。

   「表示」メニューから「ピープルコンテナ」を選択します。
2. 「新規」をクリックします。

   ピープルコンテナのテンプレートがデータ区画に表示されます。
3. 作成するピープルコンテナの名前を入力します。
4. 「了解」をクリックします。

ピープルコンテナの削除

1. 削除対象のピープルコンテナを含む組織またはピープルコンテナに移動します。
2. 「表示」メニューから「ピープルコンテナ」を選択します。
3. 削除するピープルコンテナ名の横にあるチェックボックスを選択します。
4. 「削除」をクリックします。

注 ピープルコンテナを削除すると、そのピープルコンテナに含まれるオブジェクトがすべて削除されます。すべてのユーザーとサブピープルコンテナが対象になります。

グループコンテナ

グループコンテナを使用してグループを管理します。グループコンテナにはグループとほかのグループコンテナだけを含めることができます。グループコンテナの「グループ」は、すべての管理されているグループの親エントリとしてダイナミックに割り当てられます。必要に応じて、グループコンテナを追加することができます。

注 グループコンテナの表示は必要に応じて行います。グループコンテナを表示するには、サービス設定モジュールで「グループコンテナを表示」を選択します。詳細は、245 ページの「グループコンテナを表示」を参照してください。

グループコンテナを作成する

1. 作成対象のグループコンテナを含む組織またはグループコンテナに移動します。
2. 「表示」メニューから「グループコンテナ」を選択します。

   デフォルトの「グループ」は組織の作成時に作成されています。
3. 「新規」をクリックします。
4. 「名前」フィールドに値を入力して、「了解」をクリックします。新しいグループコンテナがナビゲーション区画に表示されます。

グループコンテナを削除する

1. 削除対象のグループコンテナを含む組織に移動します。
2. 「表示」メニューから「グループコンテナ」を選択します。

   デフォルトの「グループ」と、作成したすべてのグループコンテナがナビゲーション区画に表示されます。

3. 削除するグループコンテナの横にあるチェックボックスを選択します。
4. 「削除」をクリックします。

表示オプション

組織、ロール、およびコンテナの場合、「表示オプション」表示を使用して、Access Manager コンソールに Access Manager オブジェクトを表示する方法をカスタマイズできます。すべてのオブジェクトタイプにすべての表示オプションを使用できるわけではありません。

表示オプションを変更する

1. 表示オプションを変更する組織の「プロパティ」の矢印をクリックします。
2. データ区画の「表示」メニューから「表示オプション」を選択します。
3. 「一般」セクションでプロパティを編集します。プロパティは次のとおりです。

   「フルネーム属性を生成」: この属性を選択すると、Access Manager が常にユーザーのフルネームを生成するようになります。フルネームは、ユーザーのプロファイルの名と姓の値から形成されます。

   「常に最初のエントリを選択」: 検索にこの属性を選択すると、ナビゲーション区画で指定されたアイデンティティオブジェクトタイプの最初の項目が自動的に選択され、データ区画にはその項目が表示されます。

   「ユーザープロファイルのページタイトル」: ユーザープロファイルページのタイトルに使用する属性をこのプルダウンメニューから選択します。

   「初期検索を無効」: この値は、1 つ以上のアイデンティティオブジェクトタイプに対する Access Manager の初期検索を無効にします。初期検索を無効にすると、パフォーマンスが向上し、タイムアウトエラーの発生が少なくなります。

4. 「Access Manager オブジェクトの設定を表示」セクションで表示オプションを変更します。このセクションでは、Access Manager のコンテナおよびオブジェクトの表示方法をカスタマイズできます。「Access Manager コンテナ」オプションを使用すると、ナビゲーション区画の「表示」メニューに表示するオブジェクトビューを指定できます。「Access Manager オブジェクト」フィールドを使用すると、データ区画の「表示」メニューに表示するオブジェクトビューを指定できます。
5. 「保存」をクリックします。

利用可能なアクション

特定の Access Manager オブジェクトタイプの場合、「利用可能なアクション」表示を使用してユーザーのアクセス権を定義できます。

ユーザーに対して利用可能なアクションを設定する

1. 利用可能なアクションを設定するアイデンティティオブジェクトの「プロパティ」の矢印をクリックします。
2. データ区画の「表示」メニューから「利用可能なアクション」を選択します。
3. Access Manager オブジェクトに利用可能なアクションタイプを選択します。アクションタイプは、各オブジェクトに対するユーザーのアクセスの可否を定義します。アクションタイプは次のとおりです。

   「アクセス権なし」: ユーザーは、このオブジェクトにアクセスできません。

   「表示」: ユーザーは、このオブジェクトに対して読み取りアクセス権のみを持ちます。

   「変更」: ユーザーは、このオブジェクトを変更および表示できます。

   「削除」: ユーザーは、このオブジェクトを変更、表示、および削除できます。

   「完全アクセス」: ユーザーは、このオブジェクトを作成、変更、表示、および削除できます。
4. 「保存」をクリックします。前に保存した状態に値を戻すには、「リセット」をクリックします。

第 4 章

現在のセッション

この章では、Sun Java™ System Access Manager 6 2005Q1 のセッション管理機能について説明します。セッション管理モジュールでは、ユーザーセッションの情報を確認したり、ユーザーセッションを管理したりする手段を用意しています。さまざまなセッションの時間を追跡するほかに、管理者がセッションを終了することができます。システム管理者は、プラットフォームサーバーリストに表示されたロードバランササーバーを無視してください。

現在のセッションのインタフェース

「現在のセッション」モジュールインタフェースを使用すると、適切な権限を持った管理者は、Access Manager にログインしている任意のユーザーのセッション情報を参照できます。

セッション管理フレーム

セッション管理フレームには、現在管理されている Access Manager の名前が表示されます。

セッション情報ウィンドウ

セッション情報ウィンドウには、Access Manager に現在ログイン中のすべてのユーザーと、各ユーザーのセッション時間が表示されます。表示フィールドは次のとおりです。

「ユーザー ID」: 現在ログイン中のユーザーのユーザー ID が表示されます。

「残り時間」: ユーザーの再認証までの、セッションの残り時間 (分単位) が表示されます。

「最大セッション時間」: ユーザーがログインした状態でいられる最大時間 (分単位) が表示されます。この時間が経過すると、セッションが期限切れになり、ユーザーはアクセスするために再度認証を受ける必要があります。

「アイドル時間」: ユーザーがアイドル状態になっている時間 (分単位) が表示されます。

「最大アイドル時間」: ユーザーの再認証が必要になるまでの残りの最大アイドル時間 (分単位) が表示されます。

時間の制限値は、管理者がセッション管理サービスに定義します。詳細は、387 ページの「セッションサービス属性」を参照してください。

「ユーザー ID」フィールドに入力して「フィルタ」をクリックすれば、特定のユーザーのセッションや特定の範囲のセッションを表示できます。ワイルドカードも使用できます。

「更新」ボタンをクリックすれば、セッションの表示が更新されます。

セッションの終了

適切な権限を持った管理者は、ユーザーのセッションをいつでも終了させることができます。そのためには、次の手順を実行します。

1. 終了させるユーザーのセッションを選択します。
2. 「セッションを終了」をクリックします。

第 5 章

ポリシー管理

この章では、Sun Java™ System Access Manager 6 2005Q1 のポリシー管理機能について説明します。Access Manager のポリシー管理機能では、最上位レベル管理者または最上位レベルポリシー管理者が、すべての組織で使用できる特定サービスのポリシーの表示、作成、削除、修正を行うことができるようになります。組織またはサブ組織管理者あるいはポリシー管理者が、その組織で特定の目的に使用するポリシーを表示、作成、削除、修正することもできます。

この章は、次の節で構成されています。

概要

ポリシーは、組織の保護されたリソースに対するアクセス権限を指定するルールを定義します。ビジネスには、保護、管理、監視しなければならない、リソース、アプリケーション、およびサービスがあります。ポリシーはこれらのリソースに対するアクセス権と使用を管理し、ユーザーがあるリソースに対して、いつ、どのようにアクションを実行できるかを定義します。ポリシーがオブジェクトに適用されると、特定のオブジェクトからアクセスできるリソースが定義されます。

注	オブジェクトとは主体です。主体には、個人、企業、ロール、グループなど、アイデンティティを持つことができるすべてのものが該当します。詳細については『Java™ 2 Platform Standard Edition Javadocs』を参照してください。

1 個のポリシーでは、二者択一または任意設定の決定を定義できます。二者択一の決定は、「はい」/「いいえ」、「真」/「偽」または「許可する」/「許可しない」の形式です。任意設定の決定では、属性の値を表します。たとえば、メールサービスは、ユーザーごとの最大保存容量の値セットを持つ `mailboxQuota` 属性を含んでいます。一般的に、ポリシーはオブジェクトが、どのような条件でどのリソースに何をできるかを定義するように設定されています。

ポリシー管理機能

ポリシー管理機能には、ポリシーの作成および管理を行うポリシーサービスが備わっています。ポリシーサービスにより、管理者は Access Manager 配備の中でリソースを保護するために、アクセス権を定義、変更、付与、無効化、および削除することができます。通常、ポリシーサービスには、データストアと、ポリシーの作成、管理、評価ができるインターフェースライブラリ、およびポリシーエンフォーサ (ポリシーエージェント) が含まれています。Access Manager は Sun Java System Directory Server をデータ保存に使い、ポリシーの評価とポリシーサービスのカスタマイズのために Java と C の API を提供します。詳細は『Access Manager Developer's Guide』を参照してください。また、管理者は Access Manager コンソールを使用してポリシー管理を行うこともできます。Access Manager は、ダウンロード可能なポリシーエージェントを使用してポリシーを適用する、URL ポリシーエージェントサービスを提供します。

URL ポリシーエージェントサービス

Access Manager では、初期状態で、ポリシーを適用するための URL ポリシーエージェントサービスが利用できます。このサービスにより、管理者はポリシーエンフォーサ、つまりポリシーエージェントにより、ポリシーの作成および管理を行えます。

ポリシーエージェント

ポリシーエージェントは企業のリソースが保存されているサーバーへのポリシー適用ポイント (Policy Enforcement Point、PEP) です。ポリシーエージェントは Access Manager とは別に Web サーバー上にインストールされており、ユーザーが保護された Web サーバー 上のリソースを要求すると、追加の認証ステップとして働きます。この認証は、リソースが実行するあらゆるユーザー認証要求に追加されます。ポリシーエージェントは Web サーバーを保護し、一方、認証プラグインはリソースを保護します。

たとえば、リモートインストールされた Access Manager で保護されている人事部の Web サーバーにエージェントがインストールされているとします。このエージェントにより、適切なポリシーを持っていない担当者には機密の給与情報またはその他の秘密情報は表示されません。このポリシーは Access Manager の管理者が定義し、Access Manager の配備に保存され、リモート Web サーバーのコンテンツにユーザーがアクセスするのをポリシーエージェントが許可または拒否するのに使います。

最新の Sun Java System Access Manager ポリシーエージェントは Sun Microsystems Download Center からダウンロードできます。

ポリシーエージェントのインストールおよび管理についての詳細は『Sun Java System Access Manager J2EE Policy Agents Guide』または『Web Policy Agents Guide』を参照してください。

注 ポリシーの評価に特定の順序はありませんが、評価の途中であるアクションの値が「許可しない」となったときには、ポリシー設定サービスにより「拒否決定で評価を続行」属性が有効になっていないかぎり、以降のポリシーの評価は中止されます。詳細は、369 ページの「ポリシー設定サービス属性」を参照してください。

ポリシーエージェントが決定を適用するのは Web URL (`http://...`) だけですが、Java と C の Policy Evaluation API を使いエージェントをプログラミングすれば、ほかのリソースにもポリシーを適用可能です。

さらに、場合によってはポリシー設定サービスの「リソースコンパレータ」属性をデフォルト設定から次のように変更する必要があります。

`serviceType=`*`Name_of_LDAPService`*`|class=com.sun.identity.policy.plugins.SuffixResourceName|wildcard=*|delimiter=,|caseSensitive=false`

もしくは、`LDAPResourceName` などの実装により、`com.sun.identity.policy.interfaces.ResourceName` を実装して、その上で「リソースコンパレータ」を適切に設定するという方法もあります。

注 「リソースコンパレータ」属性のフィールドについての説明は、369 ページの「ポリシー設定サービス属性」を参照してください。

ポリシーエージェントプロセス

Web ブラウザがポリシーエージェントによって保護されたサーバー上の URL を要求すると、保護された Web リソースに対するプロセスが始まります。このサーバーにインストールされたポリシーエージェントはこの要求を傍受し、既存の認証クレデンシャルをチェックします (セッショントークン)。

エージェントが要求を傍受し、既存のセッショントークンを検証したら、プロセスは以下のように続きます。

1. セッショントークンが有効であれば、ユーザーのアクセスは許可または拒否されます。ユーザーのトークンが無効であれば、以下の手順にあるようにユーザーを認証サービスにリダイレクトします。
2. 認証サービスはクレデンシャルが有効かどうかを検証し、トークンを発行します。
3. ユーザーのクレデンシャルが適切に認証されると、エージェントは Access Manager の内部サービスのアクセスに使う URL を定義するネーミングサービスに要求を出します。
4. ネーミングサービスはポリシーサービスのロケータを返し、エージェントはユーザーに適用されるポリシー決定を取得するためにポリシーサービスに要求を送信します。
5. アクセスされるリソースに関してのポリシー決定に基づいて、ユーザーのアクセスが許可または拒否されます。ポリシー決定へのアドバイスが異なる認証レベルまたは認証メカニズムを示している場合、エージェントはすべての基準が検証されるまで要求を認証サービスにリダイレクトします。

既存のセッショントークンが存在しない要求をエージェントが傍受した場合は、そのリソースが異なる認証方法で保護されているときでも、エージェントはユーザーをデフォルトのログインページにリダイレクトします。

注 ポリシーベースのリソース認証と、ユーザー認証は異なったタイプの認証です。これに関する詳細情報は 128 ページの「ポリシーベースのリソース管理」を参照してください。

ポリシータイプ

Access Manager を使用して設定できるポリシーは、標準ポリシーか参照ポリシーの 2 種類です。標準ポリシーは、複数のルール、サブジェクト、および条件から構成されます。参照ポリシーは、複数のルール、および組織への参照から構成されます。

標準ポリシー

Access Manager では、アクセス許可を定義するポリシーを標準ポリシーと呼びます。標準ポリシーは、複数のルール、サブジェクト、および条件から構成されます。

ルール

ルールは 1 つのリソース、1 つ以上のアクション、および 1 つの値から成ります。基本的に、ルールがポリシーを定義します。

- リソースは保護される特定のオブジェクトを指します。たとえば、人事サービスを使ってアクセスする HTML ページまたはユーザーの給与情報です。
- アクションはリソースに対して実行される操作の名前です。Web サーバーアクションの例としては POST または GET があります。たとえば、人事サービスに対して `canChangeHomeTelephone` といったアクションを許可する場合があります。
- 値は各アクションの可否を示します。たとえば、allow (許可する) または deny (許可しない) です。

注 リソースなしでアクションを定義することも可能です。

サブジェクト

サブジェクトはポリシーが影響を与えるユーザーまたはユーザーの集合 (たとえば、グループ、または特定のロールを持つ複数のユーザー) を定義します。サブジェクトはポリシーに割り当てられます。サブジェクトの一般則は、ユーザーがポリシー中の少なくとも 1 つのサブジェクトのメンバーである場合にのみポリシーが適用される、というものです。デフォルトのサブジェクトは次のとおりです。

- 認証済みユーザー
- Access Manager ロール
- LDAP グループ
- LDAP ロール
- LDAP ユーザー

- 組織
- Web サービスクライアント

Access Manager ロールと LDAP ロールの比較

Access Manager ロールは Access Manager を使用して作成します。これらのロールは Access Manager が規定するオブジェクトクラスを持ちます。LDAP ロールは、Directory Server ロール機能を使用するロール定義です。これらのロールは Directory Server のロール定義が規定するオブジェクトクラスを持ちます。すべての Access Manager ロールは Directory Server のロールとして使用できます。しかし、すべての Directory Server ロールが必ずしも Access Manager ロールというわけではありません。LDAP ロールは、「ポリシー設定サービス」を設定することにより、既存のディレクトリから利用できます。Access Manager ロールは、ホスティングする Access Manager ポリシーサービスを通してのみアクセスできます。Access Manager の SDK とキャッシュにアクセスするので、Access Manager ロールのメンバーシップを評価する方が高速です。ポリシー設定サービスで LDAP ロール検索フィルタを変更して、検索範囲を絞り込みパフォーマンスを向上させることができます。

入れ子ロール

入れ子ロールはポリシー定義のサブジェクトの LDAP ロールとして正しく評価できます。

条件

条件によって、ポリシーに制約を定義できます。たとえば、給与アプリケーション用のポリシーを定義する場合、アプリケーションへのアクセスを特定の時間帯だけに制限するようにアクションに対して条件を定義することができます。また、所定の IP アドレスまたは企業のイントラネットからの要求に対してのみアクションを許可するように条件を定義することもできます。

条件は、同じドメインの別の URI で別のポリシーを設定するために、補助的に使用されます。たとえば、`http://org.example.com/hr/*jsp` は `org.example.net` で午前 9 時～午後 5 時だけアクセスできますが、`http://org.example.com/finance/*.jsp` は `org.example2.net` で午前 5 時～午後 11 時にアクセスできます。これは IP 条件と時間条件を使用して実現します。またルールのリソースを `http://org.example.com/hr/*.jsp` に指定することで、ポリシーは `http://org.example.com/hr` 以下、サブディレクトリ内を含むすべての JSP に適用されるようになります。

注　参照、ルール、リソース、サブジェクト、条件、アクション、値の各用語は、`policy.dtd` 内の *Referral*、*Rule*、*ResourceName*、*Subject*、*Condition*、*Attribute*、*Value* の各要素に対応しています。

ポリシーアドバイス

条件で決定したようにポリシーを適用できない場合は、ポリシーを要求に適用できなかった理由を示すアドバイスメッセージを条件によって作成できます。このアドバイスメッセージは、ポリシー決定でポリシー適用ポイントに伝わります。ポリシー適用ポイントでは、このアドバイスを取得し、認証メカニズムにユーザーを戻してより高いレベルに認証するなど、適切なアクションを実行しようとします。アドバイスの適切なアクションを実行したあとでポリシーが適用可能になると、ユーザーはより高いレベルの認証を要求され、リソースにアクセスできるようになることがあります。

詳細については、次のクラスを参照してください。

`com.sun.identity.policy.ConditionDecision.getAdvices()`

条件が満たされない場合、アドバイスを提供するのは、`AuthLevelCondiiton` と `AuthSchemeCondition` のみです。

`AuthLevelCondition` アドバイスは、次のキーに関連します。

`com.sun.identity.policy.plugin.AuthLevelCondition.AUTH_LEVEL_CONDITION_ADVICE`

`AuthSchemeCondition` アドバイスは、次のキーに関連します。

`com.sun.identity.policy.plugin.AuthLevelCondition.AUTH_SCHEME_CONDITION_ADVICE`

カスタム条件でもアドバイスを作成できます。ただし、Access Manager ポリシーエージェントは、認証レベルアドバイスと認証方式アドバイスのみに応答します。カスタムエージェントを作成してより多くのアドバイスを理解させて応答させたり、既存の Access Manager エージェントを拡張してより多くのアドバイスを理解させて応答させたりすることができます。詳細については、次の場所にあるポリシーエージェントのマニュアルを参照してください。

http://docs.sun.com/app/docs/coll/S1_IdServPolicyAgent_21

参照ポリシー

管理者は、ある組織のポリシーの定義と決定を別の組織に委任することが必要になる場合があります。または、あるリソースに対するポリシー決定を別のポリシー製品に委任することもできます。参照ポリシーは、ポリシーの作成と評価の両方に対するポリシーの委任を管理します。1 つ以上のルールと、1 つ以上の参照で構成されます。

ルール

ルールは、ポリシーの定義と評価が参照されるリソースを定義します。

参照

参照は、ポリシーの評価をどの組織に対して参照するかを定義します。デフォルトでは、2 種類の参照があります。ピア組織とサブ組織です。それぞれ、同じレベルの組織、下位レベルの組織を表します。詳細は、118 ページの「ピア組織およびサブ組織のポリシーの作成」を参照してください。

注 参照先の組織では、その組織をすでに参照済みのリソース (またはサブリソース) のポリシーを定義または評価できます。ただし、この制約はルート組織には適用されません。

ポリシー DTD

作成して設定したポリシーは、Directory Server に XML 形式で保存されます。Directory Server では、XML でエンコードされたデータは 1 か所に保管されます。ポリシーは `amadmin.dtd` (またはコンソール) を使って定義され設定されますが、実際には `policy.dtd` に基づく XML として、Directory Server に保存されます。`policy.dtd` は、`amAdmin.dtd` から抽出されたポリシー要素タグ (ポリシー作成タグを除く) を含んでいます。したがって、ポリシーサービスが Directory Server からポリシーをロードすると、`policy.dtd` に基づいて XML をパースします。ポリシーをコマンド行から作成するときには、`amAdmin.dtd` のみが使われます。この節では、`policy.dtd` の構造について説明します。`policy.dtd` は次の場所にあります。

AccessManager-base`/SUNWam/dtd (Solairs)`

AccessManager-base`/identity,dtd (Linux)`

注 本章ではこれ以降、ディレクトリ情報は Solaris についてのみ記します。Linux のディレクトリ構造は異なっていることに注意してください。詳細は、21 ページの「本書について」を参照してください。

Policy 要素

Policy はアクセス権、つまりポリシーのルールとだれにどのルールを適用するか、またはサブジェクトを定義するルート要素です。また、ポリシーが参照 (委任) ポリシーかどうか、なんらかの制限 (または条件) があるかどうかも定義します。この要素には、ルール、条件、サブジェクト、参照というサブ要素のうち 1 つ以上が含まれることがあります。必須の XML 属性は `name` で、これはポリシーの名前を指定します。`referralPolicy` 属性は、ポリシーが参照ポリシーであるかどうかを識別します。指定がなければ標準ポリシーとして扱われます。オプションの XML 属性には *name* と *description* があります。

注 ポリシーに *referral* タグを付けると、サブジェクトと条件はポリシー評価の際、無視されます。逆に、ポリシーに *normal* タグを付けると、Referrals は無視されます。

Rule 要素

Rule 要素では、ポリシーの詳細を定義します。*ServiceName*、*ResourceName*、*AttributeValuePair* という 3 つのサブ要素を持つことができます。この要素は、リソース名やそこで実行されるアクションだけではなく、ポリシーが作成されたサービスやアプリケーションのタイプを定義します。アクションを持たないルールも定義できます。たとえば、参照ポリシールールにはアクションはありません。

注 *ResourceName* 要素を含まないポリシーの定義も可能です。

ServiceName 要素

ServiceName 要素は、ポリシーを適用するサービスの名前を定義します。この要素は、サービスのタイプを表しています。ほかの要素は含みません。値は、そのサービスの XML ファイルで (`sms.dtd` に基づいて) 定義されているとおりです。*ServiceName* 要素の XML サービス属性はサービスの名前 (値は文字列) です。

ResourceName 要素

ResourceName 要素は対象となるオブジェクトを定義します。ポリシーは、このオブジェクトを保護するように設定されています。ほかの要素は含みません。*ResourceName* 要素の XML サービス属性は、オブジェクトの名前です。*ResourceName* の例としては、Web サーバー上の `http://www.sunone.com:8080/images`、または

ディレクトリサーバー上の `ldap://sunone.com:389/dc=example,dc=com` があります。リソースの具体的な例としては、`salary://uid=jsmith,ou=people,dc=example,dc=com` で、対象となるオブジェクトは John Smith の給与情報です。

AttributeValuePair 要素

AttributeValuePair 要素はアクションとその値を定義します。この要素は Subject 要素、Referral 要素、および Condition 要素のサブ要素として扱われます。これは *Attribute* と *Value* 要素を含み、XML サービス属性は含んでいません。

Attribute 要素

Attribute 要素はアクションの名前を定義します。アクションとは処理、つまりリソースに対して行われるイベントのことです。POST や GET は Web サーバーリソースに対して行われるアクションであり、READ や SEARCH はディレクトリサーバーリソースに対して行われるアクションです。*Attribute* 要素は *Value* 要素とペアにする必要があります。*Attribute* 要素自体は、ほかの要素を含みません。*Attribute* 要素に対する XML サービス属性は、アクションの名前です。

Value 要素

Value 要素はアクションの値を定義します。Allow (許可する)/Deny (許可しない)、Yes (はい) /No (いいえ) はアクションの値の例です。ほかのアクションの値は、ブール型、数値型、または文字列型が可能です。値は、そのサービスの XML ファイルの中で (`sms.dtd` に基づいて) 定義されています。*Value* 要素はほかの要素を含まず、また XML サービス属性も含みません。

警告 許可しないルールは許可するルールより優先度が高くなります。たとえば、あるポリシーはアクセスを拒否し、別のポリシーが許可すると、(両方のポリシーのほかの条件がすべて一致する場合) 結果は拒否となります。許可しないポリシーを使用すると、ポリシー間で潜在的に衝突が生じるおそれがあるため、十分に注意して拒否ポリシーを使用することをお勧めします。明示的な許可しないルールを使用すると、さまざまなサブジェクト (ロールやグループメンバーシップなど) を通じてユーザーに割り当てられたポリシーが、アクセスを拒否する可能性があります。通常は、ポリシー定義プロセスでは、許可するルールのみを使うべきです。デフォルトで「許可しない」というのは、ほかに適用するポリシーがない場合に使用します。

Subjects 要素

Subjects サブ要素はポリシーが適用されるオブジェクトの集合を特定します。この包括的な集合はグループのメンバーシップ、ロールの所有権、または個別ユーザーに基づいて選択されます。この要素は、*Subject* というサブ要素を持ちます。定義できる XML 属性は以下のとおりです。

name: 集合の名前を定義します。

description: サブジェクトの説明を定義します。

includeType: 現在は使われていません。

Subject 要素

Subject サブ要素はポリシーを適用するオブジェクトの集合を特定します。この集合は、Subjects 要素が定義する集合をさらに絞り込んだものです。メンバーシップは、ロール、グループメンバーシップ、または単なる個別ユーザーのリストに基づきます。この要素は、AttributeValuePair 要素というサブ要素を含みます。必須の XML 属性は `type` で、特定の定義済みサブジェクトを持つ、オブジェクトの一般的な集合を特定します。ほかの XLM 属性には、集合の名前を定義する `name`、サブジェクトのメンバーでないユーザーに対してポリシーを適用するかどうかに関して、集合が定義されたとおりになっているかどうかを定義する `includeType` があります。

注	複数のサブジェクトを定義した場合、ポリシーを適用するためには少なくとも 1 つのサブジェクトをユーザーに適用しなければなりません。`includeType` を false にしてサブジェクトを定義するときは、ユーザーがそのサブジェクトのメンバーではないことが必要です。

Referrals 要素

Referrals サブ要素はポリシー参照の集合を特定します。この要素は、*Referral* というサブ要素を持ちます。ともに定義できる XML 属性には、集合の名前を定義する `name`、説明を含む `description` があります。

Referral 要素

Referral サブ要素は個別のポリシー参照を特定します。この要素は、サブ要素として AttributeValuePair 要素を持ちます。必須の XML 属性は `type` で、特定の定義済み参照を持つ、割り当ての一般的な集合を特定します。また、集合の名前を定義する `name` 属性を持つこともできます。

Conditions 要素

Conditions サブ要素はポリシーの制限 (時間範囲、認証レベル、その他) の集合を特定します。この要素は複数の *Condition* サブ要素を含んでいなければなりません。ともに定義できる XML 属性には、集合の名前を定義する `name`、説明を含む `description` があります。

注 conditions 要素は、ポリシーの中ではオプションの要素です。

Condition 要素

Condition サブ要素は特定のポリシーの制限 (時間範囲、認証レベルなど) を特定します。この要素は、サブ要素として AttributeValuePair 要素を持ちます。必須の XML 属性は `type` で、特定の定義済みサブジェクトを持つ、オブジェクトの一般的な集合を特定します。また、集合の名前を定義する `name` 属性を持つこともできます。

ポリシーサービスの追加

デフォルトで、Access Manager は URL ポリシーエージェントサービス (`iPlanetAMWebAgentService`) を提供します。このサービスは、XML ファイル形式で定義され、次のディレクトリにあります。

```
/etc/opt/SUNWam/config/xml
```

Access Manager にさらにポリシーサービスを追加することもできます。ポリシーサービスを作成したら、`amadmin` コマンド行ユーティリティを使って、これを Access Manager に追加します。

新しいポリシーサービスを追加する

1. 新しいポリシーサービスを sms.dtd に基づいて XML ファイルで作成します。新しいポリシーサービスファイルの雛型にできるポリシーサービス XML ファイルが 2 つ、Access Manager から提供されます。

   amWebAgent.xml はデフォルトの URL ポリシーエージェントサービスのための XML ファイルです。これは /etc/opt/SUNWam/config/xml/ にあります。

   SampleWebService.xml は、ポリシーサービスファイルのサンプルであり、/etc/opt/SUNWam/samples/policy にあります。

2. 新しいポリシーサービスをロードするディレクトリに XML ファイルを保存します。次に例を示します。

   /etc/opt/SUNWam/config/xml/newPolicyService.xml

3. 新しいポリシーサービスを amadmin コマンド行ユーティリティを使ってロードします。次に例を示します。

   AccessManager-base/SUNWam/bin/amadmin

   --runasdn "uid=amAdmin,ou=People,*default_org*,*root_suffix*"

   --password パスワード

   --schema etc/opt/SUNWam/config/xml/newPolicyService.xml

4. 新しいポリシーサービスをロードしたあと、Access Manager コンソールから作業を行うか、または、amadmin で新しいポリシーをロードすることにより、ポリシー定義のルールを定義できます。

ポリシーの作成

Policy API と Access Manager コンソールを使用してポリシーを作成、変更、および削除でき、amadmin コマンド行ツールを使用してポリシーを作成および削除できます。この節では、amadmin コマンド行ユーティリティと Access Manager コンソールを使用してポリシーを作成することを中心に説明します。Policy API についての詳細は『Access Manager Developer's Guide』を参照してください。

通常ポリシーは XML ファイルで作成し、amadmin コマンド行ユーティリティを使って Access Manager に追加し、Access Manager のコンソールを使って管理します (ただし、ポリシーをコンソールで作成することもできる)。これは、ポリシーが amadmin を使って直接変更できないからです。ポリシーを修正するには、そのポリシーを Access Manager から削除し、修正したポリシーを amadmin を使用して追加します。

一般に、ポリシーは組織ツリー全体で使用するために、組織またはサブ組織レベルで作成します。

amadmin でのポリシーの作成

1. ポリシーの XML ファイルを policy.dtd に基づいて作成します。このファイルは次のディレクトリにあります。

   AccessManager-base/SUNWam/dtd

2. ポリシーの XML ファイルを作成したら、次のコマンドを使用してロードできます。

AccessManager-base/SUNWam/bin/amadmin

--runasdn "uid=amAdmin,ou=People,*default_org*,*root_suffix*"

--password パスワード

--data *policy.xml*

複数のポリシーを同時に追加するには、各 XML ファイルにポリシーを 1 つずつ置くのではなく、1 つの XML ファイルにすべてのポリシーを置きます。複数の XML ファイルでポリシーを次々とロードすると、内部ポリシーインデックスが破損したり、ポリシーの評価に参加できないポリシーが生じたりするおそれがあります。

ポリシーを amadmin で作成するときは、認証スキーム条件を作成中に組織に認証モジュールを登録することの確認、組織、LDAP グループ、LDAP ロール、および LDAP ユーザーのサブジェクトを作成中に、対応する LDAP オブジェクト (組織、グループ、ロール、およびユーザー) が存在することの確認、IdentityServerRoles サブジェクトを作成中に、Access Manager ロールが存在することの確認、そしてサブ組織またはピア組織の参照を作成中に、関連がある組織が存在することの確認を行ってください。

SubOrgReferral、PeerOrgReferral、Organization サブジェクト、IdentityServerRoles サブジェクト、LDAPGroups サブジェクト、LDAPRoles サブジェクト、および LDAPUsers サブジェクトの Value 要素のテキストには、完全な DN を指定する必要があります。

Access Manager コンソールでのポリシーの作成

1. アイデンティティ管理インタフェースに移動します。
2. ポリシーを作成する組織を選択します。

   組織のポリシー管理ウィンドウの位置が正しいことを確認します。
3. 「表示」メニューから「ポリシー」を選択します。

   デフォルトでは、組織は「表示」メニューに表示されます。サブ組織がある場合は、すべてその下に表示されます。サブ組織のポリシーを作成する場合は、サブ組織を選択して、「表示」メニューから「ポリシー」を選択します。
4. ナビゲーションフレームで「新規」をクリックします。「ポリシーの作成」ウィンドウを開きます。
5. 作成するポリシーのタイプを、標準ポリシーまたは参照ポリシーのどちらかから選択します。

   サブ組織を参照する参照ポリシーが存在しない場合、そのサブ組織のポリシーを作成することはできません。

   この時点では、標準または参照ポリシーのフィールドすべてを定義する必要はありません。ポリシー作成後、ルール、サブジェクト、参照などを追加できます。
6. ポリシーの名前を入力して、「保存」をクリックします。
7. デフォルトでは、「一般」表示となっています。

   「一般」にはポリシー名が表示され、作成するポリシーの説明を入力できます。
8. 「保存」をクリックして、ポリシーの設定を完了します。

ピア組織およびサブ組織のポリシーの作成

ピア組織またはサブ組織のポリシーを作成するには、まず親組織または別のピア組織で参照ポリシーを作成する必要があります。サブ組織でポリシー設定サービスを登録し、テンプレートを作成することも必要です。参照ポリシーのルールの定義には、サブ組織が管理するリソースプレフィックスを含める必要があります。親組織または別のピア組織で参照ポリシーを作成すれば、サブ組織またはピア組織で標準ポリシーを作成できます。

この例では、o=isp が親組織、o=example.com はサブ組織で http://www.example.com のリソースおよびサブリソースを管理します。

サブ組織のポリシーを作成する

1. o=isp で参照ポリシーを作成します。参照ポリシーについては、125 ページの「参照ポリシーの修正」の手順を参照してください。

   参照ポリシーは、http://www.example.com をリソースとしてルールに定義し、参照内で example.com を SubOrgReferral の値として持つ必要があります。

2. 「組織」表示で example.com というサブ組織に移動します。

3. ポリシー設定サービスが example.com というサブ組織レベルに登録されていることを確認します。詳細は、127 ページの「ポリシー設定サービスの追加」を参照してください。

4. これで isp によってリソースが example.com の管理に委ねられたので、http://www.example.com というリソース、または http://www.example.com から始まる任意のリソースに対して標準ポリシーを作成できます。

   標準ポリシーの作成については、119 ページの「標準ポリシーの修正」の手順を参照してください。

   example.com で管理する別のリソースのポリシーを定義するには、追加の参照ポリシーを o=isp に作成する必要があります。

ポリシーを管理する

標準または参照ポリシーを作成し、Access Manager に追加したあとは、ポリシーの管理は、Access Manager のコンソールを使用して、ルール、サブジェクト、条件と参照を変更することにより行えます。

標準ポリシーの修正

アイデンティティ管理インタフェースでは、アクセス許可を定義するポリシーを作成できます。このようなポリシーを標準ポリシーと呼びます。標準ポリシーは、複数のルール、サブジェクト、および条件から構成できます。ここでは、標準ポリシー作成時に指定できるデフォルトのフィールドについて説明します。

ルールを変更する

1. アイデンティティ管理インタフェースで、「表示」メニューから「ポリシー」を選択します。

   その組織用に作成されたポリシーが表示されます。

2. 修正したいポリシーを選択し、「プロパティ」の矢印をクリックします。データフレームでポリシーの「編集」ウィンドウが開きます。

   デフォルトでは、「一般」表示となっています。「一般」表示内の属性については、116 ページの「ポリシーの作成」で説明しています。

3. 「表示」メニューから「ルール」を選び「新規」をクリックします。

   複数のポリシーサービスが存在する場合は、データ区画に一覧表示されます。ポリシーを作成したいサービスを選択して、「次へ」をクリックします。「新規ルール」ウィンドウが表示されます。

4. 「ルール」フィールドに、ポリシーのリソース、アクション、およびアクション値を定義します。フィールドは次のとおりです。

   「タイプ」: ポリシーを作成するサービスが表示されます。デフォルトは URL ポリシーエージェントです。

   「ルール名」: ルールの名前を入力します。

   「リソース名」: リソースの名前を入力します。次に例を示します。

   ```
   http://www.example.com
   ```

   現在、ポリシーエージェントでサポートされているリソースは `http://` と `https://` だけです。また、ホスト名の代わりに IP アドレスを使用することはできません。

   リソース名、ポート番号、およびプロトコルにはワイルドカードを使用できます。次に例を示します。

   ```
   http*://*:*/*.html
   ```

URL ポリシーエージェントサービスでは、ポート番号が入力されていない場合のデフォルトのポート番号は、http:// では 80、https:// では 443 となります。

リソースを http://host*:* として定義して、特定のマシンにインストールされたすべてのサーバーに対してリソースの管理を許可できます。また、次のリソースを定義して、組織のすべてのサービスに対する特定の組織権限を管理者に与えることができます。

http://*.*subdomain.domain.topleveldomain*

「選択、アクション」: URL ポリシーエージェントサービスでは、デフォルトとして次のアクションの両方または一方を選択できます。

- GET
- POST

「値」: URL ポリシーエージェントサービスでは、次のアクション値の 1 つを選択できます。

- 許可 - ルールに定義されたリソースに一致するリソースへのアクセスを許可
- 拒否 - ルールに定義されたリソースに一致するリソースへのアクセスを拒否

ポリシーでは、拒否ルールが許可ルールよりも優先されます。たとえばあるリソースに 2 つのポリシーがあり、1 つはアクセス拒否でもう 1 つはアクセス許可の場合、その結果はアクセスの拒否になります (両方のポリシーの条件が一致する場合)。拒否ポリシーを使用すると、ポリシー間で潜在的に衝突が生じるおそれがあるため、十分に注意して拒否ポリシーを使用することをお勧めします。通常は、ポリシー定義プロセスでは許可ルールだけを使用し、拒否の場合を実現するのに適用するポリシーがない場合にデフォルトの拒否を使用してください。

拒否ルールを明示的に使用すると、1 つ以上のポリシーでアクセスが許可される場合でも、異なるサブジェクト (ロールやグループのメンバーシップ) を通じてユーザーに割り当てられたポリシーによって、リソースへのアクセスを拒否されるおそれがあります。たとえば、1 つのリソースについて、Employee ロールに適用される拒否ポリシーと、Manager ロールに適用される許可ポリシーがあるとします。この場合、Employee ロールと Manager ロールの両方を割り当てられているユーザーへのポリシーの決定は拒否されます。

このような問題を解決する 1 つの方法は、条件プラグインを使ってポリシーを設計することです。上記の例では、Employee ロールに認証されたユーザーには拒否ポリシーを適用し、Manager ロールに認証されたユーザーには許可ポリシーを適用するという " ロール条件 " を利用することで、2 つのポリシーを区別できます。Manager ロールにはより高い認証レベルが与えられることから、認証レベル条件を使用する方法もあります。詳細は、123 ページの「「条件」を追加または変更する」を参照してください。

注 アクションにリソース定義が不要となるようサービスが定義されている場合、リソースフィールドは表示されません。リソースを要求するアクションと要求しないアクションの両方がサービスに含まれている場合、選択肢が表示され、リソースを要求しないアクションを伴うルールまたはリソースを要求するアクションを伴うルールのどちらかを選択できます。

5. 「終了」をクリックしてルールを保存します。これは、設定をメモリに保存するだけです。手順 7 に従ってプロセスを完了します。
6. 手順 1 から 5 を繰り返して、追加のルールを作成します。
7. ポリシーに対して作成されたすべてのルールが、「ルール」の表に表示されます。「保存」をクリックしてポリシーにルールを追加します。

   ポリシーからルールを削除するには、ルールを選択して「削除」をクリックします。

   ルール名の横にある「編集」リンクをクリックすれば、ルールの定義を編集できます。

サブジェクトを変更する

1. ポリシーのサブジェクトを定義するには、「表示」メニューから「サブジェクト」を選択し「新規」をクリックします。
2. デフォルトのサブジェクトタイプを次の中から選択します。

   「認証済みユーザー」：このサブジェクトタイプは、有効な SSOToken を持つユーザーすべてがこのサブジェクトのメンバーであることを示します。

   すべての認証済みユーザーは、ポリシーが定義されている組織とは別の組織に認証しても、このサブジェクトのメンバーになります。リソース所有者が、別の組織のユーザー用に管理されているリソースにアクセスできるようにする場合は、これが便利です。特定組織のメンバーに保護されているリソースに対するアクセスを制限する場合は、組織サブジェクトを使用してください。

   「Access Manager ロール」：このサブジェクトタイプは、Access Manager ロールのメンバーすべてがこのサブジェクトのメンバーであることを示します。Access Manager ロールは、Access Manager を使用して作成されます。これらのロールは Access Manager が規定するオブジェクトクラスを持ちます。Access Manager のロールには、ホスティングする Access Manager のポリシーサービス経由でのみアクセスできます。

   「LDAP グループ」：このサブジェクトタイプは、LDAP グループのメンバーすべてがこのサブジェクトのメンバーであることを示します。

「LDAP ロール」: このサブジェクトタイプは、LDAP ロールのメンバーすべてがこのサブジェクトのメンバーであることを示します。LDAP ロールは、Directory Server ロール機能を使用するロール定義です。LDAP ロールは、Directory Server ロール定義が規定するオブジェクトクラスを持ちます。ポリシー設定サービスで LDAP ロール検索フィルタを変更して、検索範囲を絞り込みパフォーマンスを向上させることができます。

「LDAP ユーザー」: このサブジェクトタイプは、LDAP ユーザーすべてがこのサブジェクトのメンバーであることを示します。

「組織」: このサブジェクトタイプは、組織のメンバーすべてがサブジェクトのメンバーであることを示します。

「Web サービスクライアント」: このサブジェクトタイプは、SSOToken に含まれる主体の DN がこのサブジェクトの選択された値のどれかに一致する場合、SSOToken が特定する Web サービスクライアント (WSC) がこのサブジェクトのメンバーであることを示します。有効な値は、信頼できる WSC の証明書に対応する、ローカル JKS キーストア内の信頼できる証明書の DN です。このサブジェクトは、Liberty Web サービスフレームワークに依存し、Liberty サービスプロバイダが WSC を承認するためにのみ使用する必要があります。

このサブジェクトをポリシーに追加する前に、キーストアが作成されていることを確認します。キーストアの設定に関する説明は、次の場所にあります。

AcessManager-base/SUNWam/samples/saml/xmlsig/keytool.html

「次へ」をクリックして先に進みます。

3. サブジェクトの名前を入力します。
4. 「排他的」フィールドを選択または選択解除します。

   このフィールドが選択されていないと (デフォルト)、ポリシーは、サブジェクトのメンバーであるアイデンティティに適用されます。このフィールドが選択されていると、ポリシーは、サブジェクトのメンバーではないアイデンティティに適用されます。

   ポリシーに複数のサブジェクトが存在する場合は、指定されたアイデンティティにポリシーが適用されることが少なくとも 1 つのサブジェクトで示されている場合に、そのポリシーがアイデンティティに適用されます。

5. 検索を実行して、サブジェクトに追加するアイデンティティを表示します。この手順は、「認証済みユーザー」サブジェクトや「Web サービスクライアント」サブジェクトには適用できません。

   デフォルト (*) の検索パターンでは、該当するすべてのエントリが表示されます。

6. サブジェクトに追加する個々のアイデンティティを選択するか、または「すべて追加」を選択して一度にすべてのアイデンティティを追加します。「追加」をクリックしてアイデンティティを「「選択」リストボックス」に移動します。この手順は、「認証済みユーザー」サブジェクトや「Web サービスクライアント」サブジェクトには適用できません。
7. 「終了」をクリックします。
8. サブジェクトの名前、タイプ、および排他の状況が、「サブジェクト」の表に表示されます。「保存」をクリックします。

   ポリシーからサブジェクトを削除するには、サブジェクトを選択して「削除」をクリックし、「保存」をクリックします。

   サブジェクト名の横にある「編集」リンクをクリックすれば、サブジェクトの定義を編集できます。

「条件」を追加または変更する

1. 「表示」メニューから「条件」を選択します。「新規」をクリックして新しい条件を追加するか、または「編集」リンクをクリックして既存の条件を編集します。
2. デフォルトの条件を次の中から選択します。
   - 認証レベル
   - 認証方式
   - IP アドレス
   - LE 認証レベル
   - セッション
   - 時間

   「認証レベル」の場合、ユーザーの認証レベルが、条件に設定された認証レベル以上である場合に、ポリシーが適用されます。「LE 認証レベル」の場合、ユーザーの認証レベルが、条件に設定された認証レベル以下である場合に、ポリシーが適用されます。
3. 「次へ」をクリックします。
4. 指定した条件に対する値を定義します。フィールドは次のとおりです。

   「名前」：条件の名前を入力します。

   認証レベル

   「認証レベル」：認証の信頼レベルを指定します。利用可能な認証レベルのリストは、認証レベルと認証モジュールの表に表示されます。

   認証レベルの条件を使用して、その組織に登録された認証モジュールレベル以外のレベルを指定できます。これは、別の組織から認証を受けたユーザーにポリシーを適用する場合に役立ちます。

認証方式

「認証方式」: プルダウンメニューから条件の認証方式を選択します。これらの認証方式は、組織認証モジュールのコア認証サービステンプレートから取得されます。

IP アドレス

「IP アドレス 開始 / 終了」: IP アドレスの範囲を指定します。

「DNS 名」: DNS 名を指定します。このフィールドには、完全修飾ホスト名または次の形式の文字列を指定できます。

domainname

**.domainname*

時間

「日付 開始 / 終了」: 日付の範囲を指定します。

「時刻」: 1 日での時間の範囲を指定します。

「曜日」: 曜日を指定します。

「タイムゾーン」: タイムゾーンを標準またはカスタムで指定します。カスタムのタイムゾーンとして指定できるのは、Java で認識されるタイムゾーン ID だけです (PST など)。値を指定しない場合は、デフォルト値は Access Manager JVM に設定されたタイムゾーンになります。

セッション

「最大セッション時間」: ポリシーを適用する間の最大ユーザーセッション時間を指定します。

「セッションを終了」: 選択すると、「最大セッション時間」フィールドで定義した許可される最大値をセッション時間が超えた場合に、ユーザーセッションを終了します。

5. 条件を定義したら、「終了」をクリックします。

   ポリシーに対して作成されたすべての条件が、「条件」の表に表示されます。

6. 「保存」をクリックします。

   ポリシーから条件を削除するには、条件を選択して「削除」をクリックします。

   条件名の横にある「編集」リンクをクリックすれば、条件の定義を編集できます。

参照ポリシーの修正

アイデンティティ管理インタフェースでは、ある組織のポリシーの定義や判断を、別の組織に委任できます。また、あるリソースに対するポリシーの判断を、別のポリシー製品に委任することもできます。参照ポリシーは、ポリシーの作成と評価の両方に対するポリシーの委任を管理します。参照ポリシーは、ルールおよび参照自体から構成されます。

ルールを変更する

1. 「表示」メニューから「ルール」を選択します。「新規」をクリックして新しいルールを追加するか、または「編集」リンクをクリックして既存のルールを編集します。
2. 「サービスタイプ」を選択します。新しいルールを作成している場合は、「次へ」をクリックします。
3. 「ルール」フィールドにリソースを定義します。フィールドは次のとおりです。

   「タイプ」: 作成するポリシーのポリシーサービスが表示されます。

   「ルール名」: ルールの名前を入力します。

   「リソース名」: リソースの名前を入力します。次に例を示します。

   `http://www.sunone.com`

   現在、ポリシーエージェントでサポートされているリソースは `http://` と `https://` だけです。また、ホスト名の代わりに IP アドレスを使用することはできません。

   リソース名、ポート番号、およびプロトコルにはワイルドカードを使用できます。

   URL ポリシーエージェントサービスでは、ポート番号が入力されていない場合のデフォルトのポート番号は、`http://` では 80、`https://` では 443 となります。

   リソースを `http://host*:*` として定義して、特定のマシンにインストールされたすべてのサーバーに対してリソースの管理を許可できます。また、次のリソースを定義して、組織のすべてのサービスに対する特定の組織権限を管理者に与えることができます。

   `http://*.`*`subdomain.domain.topleveldomain`*

4. 「終了」をクリックします。
5. 手順 1 から 4 を繰り返して、追加のルールを作成します。

   ポリシーに対して作成されたすべてのルールが、「ルール」の表に表示されます。
6. 「保存」をクリックします。

   ポリシーからルールを削除するには、ルールを選択して「削除」をクリックします。

   ルール名の横にある「編集」リンクをクリックすれば、ルールの定義を編集できます。

参照を追加する

1. 「表示」メニューから「参照」を選択します。「新規」をクリックして新しい参照を追加するか、または「編集」リンクをクリックして既存の参照を編集します。
2. 「ルール」フィールドにリソースを定義します。フィールドは次のとおりです。

   「参照」: 現在の参照タイプが表示されます。

   「名前」: 参照の名前を入力します。

   「含む」:「値」フィールドに表示する組織名を絞り込むためのフィルタを指定します。デフォルトでは、すべての組織名が表示されます。

   「値」: 参照の組織名を選択します。
3. 「了解」をクリックし、「保存」をクリックします。

   ポリシーから参照を削除するには、参照を選択して「削除」をクリックします。

   参照名の横にある「編集」リンクをクリックすれば、参照の定義を編集できます。

ポリシー設定サービス

各組織のポリシーに関連する属性の設定を Access Manager コンソールから行うにはポリシー設定サービスを使用します。Access Manager 認証サービスで使用するリソース名の実装および Directory Server データストアを定義することもできます。

サブジェクト評価のキャッシュ

ポリシー評価のパフォーマンスを上げるため、ポリシー設定サービスの「サブオブジェクト結果の有効時間」属性で定義されるとおり、サブジェクト評価を何分間かキャッシュします。これらのキャッシュされたポリシー決定は「サブオブジェクト結果の有効時間」属性で定義された時間が経つまで参照されます。この時間が経過したあとは、次にポリシーが評価されると、その決定はユーザーの状態変化 (該当する場合。ユーザーがグループから削除されるなど) を反映したものになります。

amldapuser の定義

amldapuser はインストール時に作成されたユーザーで、LADP およびメンバーシップ認証の間、Directory Server のバインドと検索に使われます。また、これはポリシー設定サービスでも使用されます。LADP、メンバーシップ、またはポリシー設定サービスが組織に登録されると、このユーザーの (インストール時に設定された) パスワードを入力しなければなりません。詳細は、『Sun Java System Access Manager Migration Guide』を参照してください。

ポリシー設定サービスの追加

ポリシー設定サービスの追加はサービスのタイプの追加と同じで、アイデンティティ管理インタフェース内で行います。デフォルトでは、ポリシー設定サービスは自動的に最上位レベルの組織に追加されます。作成するポリシーサービスは、すべての組織に追加する必要があります。ポリシー設定サービスを追加するときは、LDAP バインドパスワードをテンプレートに入力する必要があります。

ポリシー設定サービスを追加する

1. アイデンティティ管理インタフェースに移動します。

   コンソールが開くときのデフォルトのインタフェースはアイデンティティ管理です。

2. ポリシーを作成する組織を選択します。

   最上位レベル管理者としてログインした場合は、アイデンティティ管理モジュールがすべての設定済み組織が表示される最上位レベルの組織であることを確認します。デフォルトの最上位レベル組織は、インストール時に定義されます。

3. 「表示」メニューから「サービス」を選択します。

   その組織にすでにサービスが登録されている場合は、ナビゲーションフレームにそのサービスが表示されます。

4. ナビゲーションフレームで「追加」をクリックします。

   この組織にまだ登録されていないサービスのリストが、データフレームに表示されます。

5. データフレームに表示された「サービスを追加」ウィンドウから、「ポリシー設定」を選択して「了解」をクリックします。

   ポリシー設定サービスがナビゲーションフレームにあるサービスのリストに追加されます。

6. 「プロパティ」の矢印をクリックして、ポリシーサービスを設定します。
   a. ポリシーテンプレートが設定されていない場合、新しく登録されたポリシーサービス用にサービステンプレートを作成する必要があります。
   b. ポリシーサービスを設定するには、「はい」をクリックします。
   c. ポリシー設定属性を修正します。これらの属性については、369 ページの「ポリシー設定サービス属性」を参照してください。
7. 「保存」をクリックします。

   これで、選択した組織にポリシー設定サービスが追加されます。

注 サブ組織は、その親組織とは別にポリシーサービスを登録する必要があります。それは、サブ組織 `o=suborg,dc=sun,dc=com` は親の `dc=sun,dc=com` からポリシー設定サービスを継承しないためです。

ポリシーベースのリソース管理

組織によっては高度な認証シナリオが必要です。この場合、ユーザー認証は、アクセスしようとするリソースごとに特定のモジュールで行われます。ポリシーベースのリソース管理は、Access Manager の機能であり、そこではユーザーは Web リソースにアクセスするのにデフォルトの認証モジュールを経る必要はありません。

制限

ポリシーベースのリソース管理には次のような制限があります。

1. リソースに適用されるポリシーはすべて同じ認証方式か同じ認証レベルを持たなければなりません。たとえば、`abc.html` が LDAP 認証モジュールのポリシーで定義されていると、証明書に基づく認証モジュールのポリシーでは定義できません。
2. このポリシーについて定義できる条件はレベルと方式だけです。
3. この機能は、異なる DNS ドメイン間では働きません。

ポリシーベースのリソース管理の設定

Access Manager とポリシーエージェントがインストールされたら、ポリシーベースのリソース管理を設定できます。そのためには、Access Manager が Gateway サーブレットを指す必要があります。

1. `AMAgent.properties` を開きます。

   `AMAgent.properties` は Solaris 環境では `/etc/opt/SUNWam/agents/config/` にあります。

2. 次の行をコメントアウトします。

   ```
   #com.sun.am.policy.am.loginURL =
   http://identity_server_host.domain_name:port/amserver/UI/Login
   ```

3. ファイルに次の行を追加します。

   ```
   com.sun.am.policy.am.loginURL =
   http://identity_server_host.domain_name:port/amserver/gateway
   ```

4. サーバーを再起動します。

第 6 章

認証の管理

認証サービスは、Access Manager 配備にインストールされたすべての初期状態の認証モジュールへの、Web ベースのユーザーインタフェースを提供します。このインタフェースは、アクセスを要求するユーザーに対して、呼び出される認証モジュールに基づいたログイン条件画面を表示して認証クレデンシャルを収集する動的かつカスタマイズ可能な手段を提供します。このインタフェースは、開発者が実用的な Web アプリケーションを作成するのに役立つ Java 2 Enterprise Edition (J2EE) プレゼンテーションフレームワークである Sun Java System™ Application Framework (*JATO* と呼ばれることもある) を使用して作成されています。

ユーザーインタフェースのログイン URL

認証サービスユーザーインタフェースには、Web ブラウザの場所ツールバーにログイン URL を入力してアクセスします。この URL は次のとおりです。

`http://`*identity_server_host*. ドメイン名 : ポート / サービス配備 `_URI/UI/Login`

注	インストール時に、サービス配備 _URI は `amserver` として設定されます。このマニュアル全体にわたり、このデフォルトのサービス配備 URI が使用されています。

特定の認証方法や、成功、失敗した認証の URL のリダイレクトを定義するために、ユーザーインタフェースのログイン URL にログイン URL パラメータを付加することもできます。URL のリダイレクトの詳細情報は138 ページの「認証タイプ」を参照してください。

ログイン URL パラメータ

URL パラメータは、URL の終わりに付加される名前と値のペアです。このパラメータは疑問符 (?) で始まり、名前 = 値 の形式をとります。たとえば、次のようにいくつかのパラメータを 1 つのログイン URL に結合できます。

```
http:// サーバー名 . ドメイン名 : ポート
/amserver/UI/Login?module=LDAP&locale=ja&goto=http://www.sun.com
```

複数のパラメータを指定する場合は、アンパサンド (&) で区切ります。複数のパラメータを指定する場合は、次のガイドラインに従う必要があります。

- 各パラメータは、1 つの URL に 1 回のみ指定できます。たとえば、`module=LDAP&module=NT` は受け入れられません。
- `org` パラメータと `domain` パラメータは両方ともログイン組織を決定します。この場合、ログイン URL にはこの 2 つのパラメータどちらかを使用する必要があります。両方を使用する場合に優先順位を指定しないと、1 つのみが有効になります。
- `user`、`role`、`service`、`module`、および `authlevel` は、それぞれの基準に基づいて認証モジュールを定義するためのパラメータです。このため、ログイン URL にはこれらのパラメータのいずれか 1 つのみを使用する必要があります。複数のパラメータを使用する場合に優先順位を指定しないと、1 つのみが有効になります。

次の節では、ユーザーインタフェースのログイン URL に付加され、Web ブラウザの場所ツールバーに入力されたときに、さまざまな認証機能を実現するパラメータについて説明します。

ヒント 組織全体に配布する認証 URL およびパラメータを単純なものにするには、管理者は単純な URL を持つ HTML ページを作成し、そのページにすべての設定された認証方法に対するより複雑なログイン URL へのリンクを含めることができます。

goto パラメータ

`goto=`*successful_authentication_URL* パラメータは、認証設定サービスの「ログイン成功 URL」に定義された値を置き換えます。このパラメータは、認証が成功すると指定された URL にリンクします。`goto=logout_URL` パラメータも、ユーザーのログアウト時に指定された URL にリンクするのに使用できます。次に認証成功 URL の例を示します。

```
http:// サーバー名 . ドメイン名 : ポート
/amserver/UI/Login?goto=http://www.sun.com/homepage.html
```

次に `goto` ログアウト URL の例を示します。

```
http:// サーバー名 . ドメイン名 : ポート
/amserver/UI/Logout?goto=http://www.sun.com/logout.html
```

注 Access Manager が認証成功リダイレクト URL を確認する優先順位が定められています。リダイレクト URL とそれらの順番は認証方法に基づいているので、この順番および関連情報については、138 ページの「認証タイプ」で詳しく説明します。

gotoOnFail パラメータ

`gotoOnFail=failed_authentication_URL` パラメータは、認証設定サービスの「失敗ログイン URL」に定義された値を置き換えます。ユーザーが認証に失敗すると、指定された URL にリンクします。次に `gotoOnFail` URL の例を示します。`http://` サーバー名 . ドメイン名 : ポート
`/amserver/UI/Login?gotoOnFail=http://www.sun.com/auth_fail.html`

注 Access Manager が認証失敗リダイレクト URL を確認する優先順位が定められています。リダイレクト URL とそれらの順番は認証方法に基づいているので、この順番および関連情報については、138 ページの「認証タイプ」で詳しく説明します。

org パラメータ

org=*orgName* パラメータを使用すると、指定された組織のユーザーとしてユーザーを認証することができます。

ヒント　指定された組織のメンバーになっていないユーザーが、org パラメータで認証を試みると、エラーメッセージを受け取ります。ただし、次の条件をすべて満たせば、Directory Server に動的にユーザープロファイルを作成できます。

- コア認証サービスの「ユーザープロファイル」属性に、「ダイナミック」または「ユーザーエイリアスを使用してダイナミックに」が設定されている。
- ユーザーが、必要なモジュールに対する認証に成功している。
- ユーザーのプロファイルは、まだ Directory Server にない。

このパラメータから、組織およびそのロケールの設定に基づいて、正しいログインページが表示されます。このパラメータが設定されない場合は、デフォルトは最上位の組織になります。たとえば、org URL は次のようになります。

```
http:// サーバー名 . ドメイン名 : ポート /amserver/UI/Login?org=sun
```

user パラメータ

user=*userName* パラメータは、ユーザーのプロファイルの「ユーザー認証設定」属性に設定されたモジュールに基づいて、認証を強制します。たとえば、あるユーザーのプロファイルを、証明書モジュールを使用して認証し、別のユーザーを LDAP モジュールを使用して認証するように設定できます。このパラメータを追加すると、ユーザーはユーザーの組織に設定された認証方法ではなく、ユーザー用に設定された認証プロセスに送られます。次に例を示します。

```
http:// サーバー名 . ドメイン名 : ポート /amserver/UI/Login?user=jsmith
```

role パラメータ

role=*roleName* パラメータは、ユーザーを指定されたロール用に設定された認証プロセスに送ります。指定されたロールのメンバーになっていないユーザーが、このパラメータで認証を試みると、エラーメッセージを受け取ります。次に例を示します。

```
http:// サーバー名 . ドメイン名 : ポート /amserver/UI/Login?role=manager
```

locale パラメータ

Access Manager は、認証プロセスとコンソール自身について、ローカライズされた (英語以外の言語に翻訳された) 画面を表示することができます。`locale=`*localeName* パラメータは、指定されたロケールをその他の定義されたロケールよりも優先させることができます。ログインのロケールは、次の順序で次の場所から設定を検索したあとに、クライアントに表示されます。

1. ログイン URL のロケールパラメータの値

   `locale=`*localeName* パラメータの値は、定義されたその他のすべてのロケールよりも優先されます。

2. ユーザーのプロファイルに定義されたロケール

   URL パラメータがない場合は、ロケールはユーザープロファイルの「ユーザー設定言語」属性に設定された値に基づいて表示されます。

3. HTTP ヘッダーに定義されたロケール

   このロケールは、Web ブラウザによって設定されます。

4. コア認証サービスに定義されたロケール

   これは、コア認証モジュールの「デフォルト認証ロケール」属性の値です。

5. プラットフォームサービスに定義されたロケール

   これは、プラットフォームサービスの「プラットフォームロケール」属性の値です。

6. オペレーティングシステムのロケール

この優先順位から導き出されるロケールは、ユーザーのセッショントークンに格納され、Access Manager が、ローカライズされた認証モジュールのロードだけに使用します。認証に成功すると、ユーザーのプロファイルの「ユーザー設定言語」属性に定義されたロケールが使用されます。ロケールが設定されていない場合は、認証に使用されたロケールが引き続き使用されます。次に例を示します。

`http://` サーバー名 `.` ドメイン名 `:` ポート `/amserver/UI/Login?locale=ja`

注 画面のテキストおよびエラーメッセージのローカライズの方法については、『Access Manager Developer's Guide』を参照してください。

module パラメータ

`module=`*moduleName* パラメータを使用すると、指定した認証モジュールによって認証を行うことができます。どのモジュールでも指定できますが、まずユーザーが所属する組織に登録し、コア認証モジュールでその組織の認証モジュールの 1 つとして選択する必要があります。次に例を示します。

```
http:// サーバー名 . ドメイン名 : ポート /amserver/UI/Login?module=Unix
```

注 認証モジュール名は、URL パラメータで使用する場合には大文字と小文字が区別されます。

service パラメータ

`service=`*serviceName* パラメータを使用すると、サービスの設定された認証スキームによってユーザーを認証できます。認証設定サービスを使用して、異なるサービスに異なる認証スキームを設定できます。たとえば、オンラインの給料支払いアプリケーションにはより安全な証明書認証モジュールを使用した認証が必要になり、組織の従業員のディレクトリアプリケーションには LDAP 認証モジュールのみが必要になるなどです。認証スキームを、それらの各サービスに設定および指定できます。次に例を示します。

```
http:// サーバー名 . ドメイン名 : ポート /amserver/UI/Login?service=sv1
```

注 認証設定サービスを使用して、サービスに基づく認証のスキームを定義します。

arg パラメータ

`arg=`*newsession* パラメータを使用して、ユーザーの現在のセッションを終了し、新しいセッションを開始します。認証サービスは、1 回の要求でユーザーの既存のセッショントークンを破棄し、新しいログインを実行します。このオプションは、通常匿名の認証モジュールで使用されます。ユーザーは、まず匿名セッションで認証を受けてから、登録リンクまたはログインリンクをヒットします。次に例を示します。

```
http:// サーバー名 . ドメイン名 : ポート /amserver/UI/Login?arg=newsession
```

authlevel パラメータ

`authlevel=`*value* パラメータは、指定された認証レベル値以上の認証レベルのモジュールを呼び出すように認証サービスに指示します。各認証モジュールは、固定整数の認証レベルで定義されます。次に例を示します。

```
http:// サーバー名 . ドメイン名 : ポート /amserver/UI/Login?authlevel=1
```

注 認証レベルは、各モジュールの特定のプロファイルに設定されます。このモジュールについては、『Sun Java System Access Manager 管理ガイド』を参照してください。

domain パラメータ

このパラメータを使用すると、指定されたドメインとして識別される組織にユーザーがログインできます。指定するドメインは、組織のプロファイルの「ドメイン名」属性に定義された値に一致する必要があります。次に例を示します。

```
http:// サーバー名 . ドメイン名 : ポート /amserver/UI/Login?domain=sun.com
```

ヒント 指定されたドメイン、つまり組織のメンバーになっていないユーザーが、org パラメータで認証を試みると、エラーメッセージを受け取ります。ただし、次の条件をすべて満たせば、Directory Server に動的にユーザープロファイルを作成できます。

- コア認証サービスの「ユーザープロファイル」属性に、「ダイナミック」または「ユーザーエイリアスを使用してダイナミックに」が設定されている。
- ユーザーが、必要なモジュールに対する認証に成功している。
- ユーザーのプロファイルは、まだ Directory Server にない。

iPSPCookie パラメータ

iPSPCookie=yes パラメータを使用すると、ユーザーは持続 Cookie でログインできます。持続 Cookie とは、ブラウザウィンドウが閉じられたあとも存在し続ける Cookie のことです。このパラメータを使用するには、ユーザーがログインする組織のコア認証モジュールで「持続 Cookie」が有効になっている必要があります。ユーザーが認証されブラウザを閉じると、ユーザーは新しいブラウザセッションでログインすることが可能であり、再認証する必要なくコンソールにダイレクトされます。これは、コアサービスに指定された「Cookie の最大持続時間」属性の値まで有効です。次に例を示します。

```
http:// サーバー名 . ドメイン名 : ポート
/amserver/UI/Login?org=example&iPSPCookie=yes
```

IDTokenN パラメータ

このパラメータオプションを使用すると、ユーザーは URL または HTML 形式で認証クレデンシャルを渡すことができます。IDTokenN=*value* パラメータを使用すると、ユーザーは認証サービスユーザーインタフェースにアクセスせずに認証を受けることができます。この処理は、ゼロページログインと呼ばれます。ゼロページログインは、1 つのログインページを使用する認証モジュールの場合にのみ機能します。IDToken0、IDToken1、...、IDTokenN の値は、認証モジュールのログインページのフィールドにマッピングされます。たとえば、LDAP 認証モジュールは、userID 情報に IDToken1 を、パスワード情報に IDToken2 を使用できます。この場合、LDAP モジュールの IDTokenN URL は次のようになります。

```
http:// サーバー名 . ドメイン名 : ポート
/amserver/UI/Login?module=LDAP&IDToken1=userID&IDToken2=password
```

LDAP がデフォルトの認証モジュールである場合は、`module=LDAP` を省略できます。

匿名認証の場合は、ログイン URL パラメータは次のようになります。

```
http:// サーバー名 . ドメイン名 : ポート
/amserver/UI/Login?module=Anonymous&IDToken1=anonymousUserID
```

注 以前のリリースのトークン名 `Login.Token0`、`Login.Token1`、...、`Login.TokenN` は、現在はサポートされていますが今後のリリースではサポートされません。新しい `IDTokenN` パラメータを使用することをお勧めします。

認証タイプ

認証サービスは、さまざまな認証適用方法を提供します。それらの異なる認証方法には、ログイン URL パラメータを指定して、または認証プログラミングインタフェースを介してアクセスできます。認証モジュールを設定する前に、特定の認証モジュール名を含むように、コア認証サービス属性の組織認証モジュールを修正する必要があります。

認証設定サービスは、次の認証タイプ用の認証モジュールを定義するために使用します。

- 141 ページの「組織に基づく認証」
- 143 ページの「ロールに基づく認証」
- 147 ページの「サービスに基づく認証」
- 150 ページの「ユーザーに基づく認証」
- 152 ページの「認証レベルに基づく認証」
- 155 ページの「モジュールに基づく認証」

これらの認証タイプのいずれかで認証モジュールを定義すると、認証プロセスの成功または失敗に基づいて、リダイレクト URL、およびポストプロセス Java クラス仕様を提供するように設定できます。

認証タイプによってアクセスが決定される方法

各認証方法では、ユーザーは認証に成功するか失敗します。成功または失敗の決定後、各方法では次の手順を実行します。手順 1 〜 3 は認証が成功した場合に実行され、手順 4 は成功した認証と失敗した認証の両方で実行されます。

1. Access Manager によって、認証されたユーザーが Directory Server データストアに定義されているかどうか、またプロファイルが有効であるかどうかが確認されます。

   コア認証モジュールの「ユーザープロファイル」属性は、「必須」、「ダイナミック」、「ユーザーエイリアスを使用してダイナミックに」、または「無視」として定義できます。認証に成功すると、Access Manager によって、認証されたユーザーが Directory Server データストアに定義されているかどうかが確認され、「ユーザープロファイル」の値が「必須」である場合は、プロファイルが有効であるとみなされます。これはデフォルトの場合です。「ユーザープロファイル」が「ダイナミックに設定」である場合、認証サービスはユーザープロファイルを Directory Server データストアに作成します。「ユーザープロファイル」が「無視」に設定されている場合は、ユーザーの検証は行われません。

2. 認証ポストプロセス SPI が実行されます。

   コア認証モジュールには、値として認証ポストプロセスクラス名を含む「認証ポストプロセスクラス」属性が含まれています。`AMPostAuthProcessInterface` は、ポストプロセスインタフェースです。このインタフェースは、認証の成功または失敗時、またはログアウト時に実行できます。

3. セッショントークンで次のプロパティが追加または更新され、ユーザーのセッションがアクティブになります。

   Organization: これは、ユーザーが所属する組織の DN です。

   Principal: ユーザーの DN です。

   Principals: ユーザーが認証を受けた名前のリストです。このプロパティは、パイプで区切られたリストとして定義された複数の値を持つことができます。

   UserId: モジュールが返すユーザーの DN であるか、LDAP またはメンバーシップ以外のモジュールの場合はユーザー名です。すべての Principal は、同じユーザーにマッピングされる必要があります。ユーザー ID は、ユーザーがマッピングされるユーザー DN です。

注 このプロパティは、DN 以外の値になることがあります。

   UserToken: ユーザー名です。すべての Principal は、同じユーザーにマッピングされる必要があります。UserToken は、ユーザーがマッピングされるユーザー DN です。

Host: クライアント用のホスト名または IP アドレスです。

authLevel: ユーザーが認証を受けた最高のレベルです。

AuthType: ユーザーが認証を受けた認証モジュールのパイプで区切られたリストです (例、`module1|module2|module3`)。

clientType: クライアントブラウザのデバイスタイプです。

Locale: クライアントのロケールです。

CharSet: クライアント用に定めらた文字セットです。

Role: ロールに基づく認証にのみ適用可能であり、ユーザーが属すロールです。

Service: サービスに基づく認証にのみ適用可能であり、ユーザーが属すサービスです。

loginURL: クライアントのログイン URL です。

4. Access Manager は、成功または失敗した認証のあとにユーザーをリダイレクトする場所についての情報を検索します。

   URL のリダイレクトは、Access Manager のページまたは URL のどちらかにすることができます。リダイレクトは、Access Manager が認証方法に基づいて、また認証が成功したか失敗したかによってリダイレクトを検索する優先順位のもとに行われます。この順序については、次の認証方法についての節の、URL のリダイレクトの部分で詳しく説明します。

URL のリダイレクト

認証設定サービスでは、成功または失敗した認証に対する URL のリダイレクトを割り当てることができます。その URL 自体は、認証設定サービスの「ログイン成功 URL」および「ログイン失敗 URL」属性で定義します。URL のリダイレクトを有効にするために、ロール、組織、またはユーザー用に設定するように、認証設定サービスを組織に追加し、利用可能にする必要があります。認証設定サービスの追加時は、LDAP で必須、というように認証モジュールを追加するようにしてください。詳細は、157 ページの「認証設定」を参照してください。

組織に基づく認証

この認証方法では、ユーザーが組織またはサブ組織に対する認証を受けることができます。これは、Access Manager のデフォルトの認証方法です。組織の認証方法は、コア認証モジュールを組織に登録し、「組織認証設定」属性を定義することによって設定します。

組織に基づく認証のログイン URL

認証の組織は、ユーザーインタフェースのログイン URL に org パラメータまたは domain パラメータを定義して指定できます。認証の要求の組織は、次の優先順位で判断されます。

1. domain パラメータ。
2. org パラメータ。
3. 管理サービスの「DNS エイリアス名」(組織のエイリアス名) 属性の値。

正しい組織を呼び出した後、ユーザーが認証を受ける認証モジュールは、コア認証サービスの「組織認証設定」属性から取得されます。組織に基づく認証を指定し、開始するログイン URL を次に示します。

http:// サーバー名 . ドメイン名 : ポート /amserver/UI/Login

http:// サーバー名 . ドメイン名 : ポート /amserver/UI/Login?domain= ドメイン名

http:// サーバー名 . ドメイン名 : ポート /amserver/UI/Login?org= 組織名

定義されたパラメータがない場合、組織はログイン URL に指定されたサーバーホストとドメインから判断されます。

組織に基づく認証のリダイレクト URL

組織に基づく認証が成功または失敗した時に、Access Manager はユーザーのリダイレクト先の情報を検索します。この情報をアプリケーションが検索する優先順位を次に示します。

組織に基づく認証が成功した場合のリダイレクト URL

組織に基づく認証が成功した場合のリダイレクト URL は、次の場所を次の優先順位で確認することによって判断されます。

1. 認証モジュールで設定された URL。
2. goto ログイン URL パラメータで設定された URL。
3. ユーザーのプロファイル (amUser.xml) の iplanet-am-user-success-url 属性用に clientType カスタムファイルに設定された URL。

4. ユーザーのロールエントリの iplanet-am-auth-login-success-url 属性用に clientType カスタムファイルに設定された URL。
5. ユーザーの組織エントリの iplanet-am-auth-login-success-url 属性用に clientType カスタムファイルに設定された URL。
6. グローバルデフォルトとして iplanet-am-auth-login-success-url 属性用に clientType カスタムファイルに設定された URL。
7. ユーザーのプロファイル (amUser.xml) の iplanet-am-user-success-url 属性に設定された URL。
8. ユーザーのロールエントリの iplanet-am-auth-login-success-url 属性に設定された URL。
9. ユーザーの組織エントリの iplanet-am-auth-login-success-url 属性に設定された URL。
10. グローバルデフォルトとして iplanet-am-auth-login-success-url 属性に設定された URL。

組織に基づく認証に失敗した場合のリダイレクト URL

組織に基づく認証が失敗した場合のリダイレクト URL は、次の場所を次の順序で確認することによって判断されます。

1. 認証モジュールで設定された URL。
2. gotoOnFail ログイン URL パラメータで設定された URL。
3. ユーザーのエントリ (amUser.xml) の iplanet-am-user-failure-url 属性用に clientType カスタムファイルに設定された URL。
4. ユーザーのロールエントリの iplanet-am-auth-login-failure-url 属性用に clientType カスタムファイルに設定された URL。
5. ユーザーの組織エントリの iplanet-am-auth-login-failure-url 属性用に clientType カスタムファイルに設定された URL。
6. グローバルデフォルトとして iplanet-am-auth-login-failure-url 属性用に clientType カスタムファイルに設定された URL。
7. ユーザーのエントリ (amUser.xml) の iplanet-am-user-failure-url 属性に設定された URL。
8. ユーザーのロールエントリの iplanet-am-auth-login-failure-url 属性に設定された URL。
9. ユーザーの組織エントリの iplanet-am-auth-login-failure-url 属性に設定された URL。
10. グローバルデフォルトとして iplanet-am-auth-login-failure-url 属性に設定された URL。

組織に基づく認証を設定する

認証モジュールは、最初にコア認証サービスを組織に追加することで、組織用に設定できます。

組織の認証属性を設定するには、次の手順を実行します。

1. 認証属性を設定する組織に移動します。
2. 「表示」メニューから「サービス」を選択します。
3. サービスのリスト表示で、「コア」をクリックします。

   コア認証属性がデータ区画に表示されます。
4. 「管理者認証」属性の隣にある「編集」リンクをクリックします。これにより、管理者専用に認証サービスを定義できます。この属性は、管理者とエンドユーザーの認証モジュールを別々のものにする必要がある場合に使用できます。デフォルトの認証モジュールは LDAP です。

   認証サービスを定義したら、「保存」をクリックして変更内容を保存します。次に「閉じる」をクリックし、組織の「コア認証」属性に戻ります。
5. 「組織認証設定」属性の隣にある「編集」リンクをクリックします。これにより、組織内の全ユーザー用に認証モジュールを定義できます。デフォルトの認証モジュールは LDAP です。
6. 認証サービスを定義したら、「保存」をクリックして変更内容を保存します。次に「閉じる」をクリックし、組織の「コア認証」属性に戻ります。

ロールに基づく認証

この認証方法では、ユーザーは組織またはサブ組織内の (スタティックまたはフィルタリングされた) ロールに対する認証を受けることができます。

注	認証設定サービスは、組織に登録してからでなければロールにインスタンスとして登録できません。

認証を成功させるには、ユーザーはロールに属し、そのロールに設定された認証設定サービスインスタンスに定義された各モジュールに対する認証を受ける必要があります。ロールに基づく認証の各インスタンスに対して、次の属性を指定できます。

「競合の解決レベル」: 同一のユーザーが含まれることがある 2 つの異なるロールに定義された認証設定サービスインスタンスに対する優先レベルを設定します。たとえば、User 1 が Role 1 および Role 2 に割り当てられている場合を想定します。ユーザーが認証を試みたときに、認証の成功または失敗時のリダイレクトや認証後プロセスに対して Role 1 の優先順位がより高くなるように、Role1 により高い競合解決レベルを設定することができます。

「認証設定」: ロールの認証プロセスに設定された認証モジュールを定義します。

「ログイン成功 URL」: 認証が成功した場合にユーザーがリダイレクトされる URL を定義します。

「ログイン失敗 URL」: 認証が失敗した場合にユーザーがリダイレクトされる URL を定義します。

「認証ポストプロセスクラス」: 認証後インタフェースを定義します。

ロールに基づく認証のログイン URL

ロールに基づく認証は、ユーザーインタフェースのログイン URL にロールパラメータを定義して指定できます。正しいロールを呼び出した後、ユーザーが認証を受ける認証モジュールは、そのロールに定義された認証設定サービスインスタンスから取得されます。

このロールに基づく認証を指定し開始するログイン URL を次に示します。

`http://` サーバー名 . ドメイン名 : ポート `/amserver/UI/Login?role=` ロール名

`http://` サーバー名 . ドメイン名 : ポート `/amserver/UI/Login?org=` 組織名 `&role=` ロール名

org パラメータが設定されていない場合、ロールが属す組織はログイン URL そのものに指定されたサーバーホストおよびドメインから判断されます。

ロールに基づく認証のリダイレクト URL

ロールに基づく認証が成功または失敗した時に、Access Manager はユーザーのリダイレクト先の情報を検索します。この情報をアプリケーションが検索する優先順位を次に示します。

ロールに基づく認証が成功した場合のリダイレクト URL

ロールに基づく認証が成功した場合のリダイレクト URL は、次の場所を次の順序で確認することによって判断されます。

1. 認証モジュールで設定された URL。
2. `goto` ログイン URL パラメータで設定された URL。

3. ユーザーのプロファイル (amUser.xml) の iplanet-am-user-success-url 属性用に clientType カスタムファイルに設定された URL。
4. ユーザーが認証を受けたロールの iplanet-am-auth-login-success-url 属性用に clientType カスタムファイルに設定された URL。
5. 認証されたユーザーの別のロールエントリの iplanet-am-auth-login-success-url 属性用に clientType カスタムファイルに設定された URL。このオプションは、前のリダイレクト URL が失敗した場合の代替リダイレクト URL です。
6. ユーザーの組織エントリの iplanet-am-auth-login-success-url 属性用に clientType カスタムファイルに設定された URL。
7. グローバルデフォルトとして iplanet-am-auth-login-success-url 属性用に clientType カスタムファイルに設定された URL。
8. ユーザーのプロファイル (amUser.xml) の iplanet-am-user-success-url 属性に設定された URL。
9. ユーザーが認証されたロールの iplanet-am-auth-login-success-url 属性に設定された URL。
10. 認証されたユーザーの別のロールエントリの iplanet-am-auth-login-success-url 属性に設定された URL。このオプションは、前のリダイレクト URL が失敗した場合の代替リダイレクト URL です。
11. ユーザーの組織エントリの iplanet-am-auth-login-success-url 属性に設定された URL。
12. グローバルデフォルトとして iplanet-am-auth-login-success-url 属性に設定された URL。

ロールに基づく認証が失敗した場合のリダイレクト URL

ロールに基づく認証が失敗した場合のリダイレクト URL は、次の場所を次の順序で確認することによって判断されます。

1. 認証モジュールで設定された URL。
2. goto ログイン URL パラメータで設定された URL。
3. ユーザーのプロファイル (amUser.xml) の iplanet-am-user-failure-url 属性用に clientType カスタムファイルに設定された URL。
4. ユーザーが認証を受けたロールの iplanet-am-auth-login-failure-url 属性用に clientType カスタムファイルに設定された URL。
5. 認証されたユーザーの別のロールエントリの iplanet-am-auth-login-failure-url 属性用に clientType カスタムファイルに設定された URL。このオプションは、前のリダイレクト URL が失敗した場合の代替リダイレクト URL です。

6. ユーザーの組織エントリの iplanet-am-auth-login-failure-url 属性用に clientType カスタムファイルに設定された URL。
7. グローバルデフォルトとして iplanet-am-auth-login-failure-url 属性用に clientType カスタムファイルに設定された URL。
8. ユーザーのプロファイル (amUser.xml) の iplanet-am-user-failure-url 属性に設定された URL。
9. ユーザーが認証されたロールの iplanet-am-auth-login-failure-url 属性に設定された URL。
10. 認証されたユーザーの別のロールエントリの iplanet-am-auth-login-failure-url 属性に設定された URL。このオプションは、前のリダイレクト URL が失敗した場合の代替リダイレクト URL です。
11. ユーザーの組織エントリの iplanet-am-auth-login-failure-url 属性に設定された URL。
12. グローバルデフォルトとして iplanet-am-auth-login-failure-url 属性に設定された URL。

ロールに基づく認証を設定する

認証モジュールは、認証設定サービスをロールレベルで追加すると、ロール用に設定されます。

1. 認証属性を設定する組織に移動します。
2. 「表示」メニューから「ロール」を選択します。
3. 認証設定を設定するロールを選択し、「プロパティ」の矢印をクリックします。

   ロールのプロパティがデータ区画に表示されます。
4. データ区画の「表示」メニューから「サービス」を選択します。
5. 必要に応じて「認証設定」属性を修正します。これらの属性の説明については、第 33 章「認証設定サービス属性」を参照するか、またはコンソール右上の「ヘルプ」リンクをクリックしてください。
6. 「保存」をクリックします。

注 新しいロールを作成している場合、そのロールに認証設定サービスは自動的に割り当てられません。ロールを作成する前に、ロールプロファイルページの上部で認証設定サービスを選択していることを確認してください。

ロールベースの認証を有効にするときは、LDAP 認証モジュールはデフォルトのままにでき、メンバーシップを設定する必要はありません。

サービスに基づく認証

この認証方法では、ユーザーは組織またはサブ組織に登録された特定のサービスまたはアプリケーションに対する認証を受けることできます。このサービスは、認証設定サービス内でサービスインスタンスとして設定され、インスタンス名が関連付けられます。認証を成功させるには、ユーザーはサービスに設定された認証設定サービスインスタンスに定義された各モジュールに対して認証を受ける必要があります。サービスに基づく認証の各インスタンスに対して、次の属性を指定できます。

「認証設定」: サービスの認証プロセスに設定された認証モジュールを定義します。

「ログイン成功 URL」: 認証が成功した場合にユーザーがリダイレクトされる URL を定義します。

「ログイン失敗 URL」: 認証が失敗した場合にユーザーがリダイレクトされる URL を定義します。

「認証ポストプロセスクラス」: 認証後インタフェースを定義します。

サービスに基づく認証のログイン URL

サービスに基づく認証は、ユーザーインタフェースのログイン URL にサービスパラメータを定義して指定できます。サービスを呼び出した後、ユーザーが認証を受ける認証モジュールは、そのサービスに定義された認証設定サービスインスタンスから取得されます。

このサービスに基づく認証を指定し開始するログイン URL を次に示します。

`http://` サーバー名 `.` ドメイン名 `:` ポート `/amserver/UI/Login?service=` サービス名

および

`http://` サーバー名 `.` ドメイン名 `:` ポート `/amserver/UI/Login?org=` 組織名 `&service=` サービス名

`org` パラメータが設定されていない場合、組織はログイン URL そのものに指定されたサーバーホストとドメインから判断されます。

サービスに基づく認証のリダイレクト URL

サービスに基づく認証が成功または失敗した時に、Access Manager はユーザーのリダイレクト先の情報を検索します。この情報をアプリケーションが検索する優先順位を次に示します。

サービスに基づく認証が成功した場合のリダイレクト URL

サービスに基づく認証が成功した場合のリダイレクト URL は、次の場所を次の順序で確認することによって判断されます。

1. 認証モジュールで設定された URL。
2. goto ログイン URL パラメータで設定された URL。
3. ユーザーのプロファイル (amUser.xml) の iplanet-am-user-success-url 属性用に clientType カスタムファイルに設定された URL。
4. ユーザーが認証を受けたサービスの iplanet-am-auth-login-success-url 属性用に clientType カスタムファイルに設定された URL。
5. ユーザーのロールエントリの iplanet-am-auth-login-sucess-url 属性用に clientType カスタムファイルに設定された URL。
6. ユーザーの組織エントリの iplanet-am-auth-login-success-url 属性用に clientType カスタムファイルに設定された URL。
7. グローバルデフォルトとして iplanet-am-auth-login-success-url 属性用に clientType カスタムファイルに設定された URL。
8. ユーザーのプロファイル (amUser.xml) の iplanet-am-user-success-url 属性に設定された URL。
9. ユーザーが認証されたサービスの iplanet-am-auth-login-success-url 属性に設定された URL。
10. ユーザーのロールエントリの iplanet-am-auth-login-success-url 属性に設定された URL。
11. ユーザーの組織エントリの iplanet-am-auth-login-success-url 属性に設定された URL。
12. グローバルデフォルトとして iplanet-am-auth-login-success-url 属性に設定された URL。

サービスに基づく認証が失敗した場合のリダイレクト URL

サービスに基づく認証が失敗した場合のリダイレクト URL は、次の場所を次の順序で確認することによって判断されます。

1. 認証モジュールで設定された URL。
2. goto ログイン URL パラメータで設定された URL。
3. ユーザーのプロファイル (amUser.xml) の iplanet-am-user-failure-url 属性用に clientType カスタムファイルに設定された URL。
4. ユーザーが認証を受けたサービスの iplanet-am-auth-login-failure-url 属性用に clientType カスタムファイルに設定された URL。
5. ユーザーのロールエントリの iplanet-am-auth-login-failure-url 属性用に clientType カスタムファイルに設定された URL。
6. ユーザーの組織エントリの iplanet-am-auth-login-failure-url 属性用に clientType カスタムファイルに設定された URL。

7. グローバルデフォルトとして iplanet-am-auth-login-failure-url 属性用に clientType カスタムファイルに設定された URL。
8. ユーザーのプロファイル (amUser.xml) の iplanet-am-user-failure-url 属性に設定された URL。
9. ユーザーが認証されたサービスの iplanet-am-auth-login-failure-url 属性に設定された URL。
10. ユーザーのロールエントリの iplanet-am-auth-login-failure-url 属性に設定された URL。
11. ユーザーの組織エントリの iplanet-am-auth-login-failure-url 属性に設定された URL。
12. グローバルデフォルトとして iplanet-am-auth-login-failure-url 属性に設定された URL。

サービスに基づく認証を設定する

認証モジュールは、認証設定サービスを追加すると、サービス用に設定されます。そのためには、次の手順を実行します。

1. アイデンティティ (識別情報) 管理モジュールの「表示」メニューから、「サービス」を選択します。

   追加済みのサービスのリストが表示されます。認証設定サービスを追加していない場合は、次の手順に進みます。サービスを追加済みの場合は、手順 4 に進みます。
2. ナビゲーション区画で「追加」をクリックします。

   利用可能なサービスのリストがデータ区画に表示されます。
3. 「認証設定」のチェックボックスを選択し、「了解」をクリックします。

   認証設定サービスがナビゲーション区画に表示され、追加されたことが管理者に示されます。
4. 「認証設定」の矢印をクリックします。

   「サービスインスタンスリスト」がデータ区画に表示されます。
5. 認証モジュールを設定するサービスインスタンスをクリックします。
6. 「認証設定」属性を修正し、「保存」をクリックします。これらの属性の説明については、第 33 章「認証設定サービス属性」を参照するか、またはコンソール右上の「ヘルプ」リンクをクリックしてください。

ユーザーに基づく認証

この認証方法では、ユーザーはユーザー専用に設定された認証プロセスに対する認証を受けることができます。このプロセスは、ユーザーのプロファイルの「ユーザー認証設定」属性の値として設定されます。認証を成功させるには、ユーザーは定義された各モジュールに対して認証する必要があります。

ユーザーに基づく認証のログイン URL

ユーザーに基づく認証は、ユーザーインタフェースのログイン URL にユーザーパラメータを定義して指定できます。正しいユーザーを呼び出した後、ユーザーが認証を受ける認証モジュールは、そのユーザーに定義されたユーザー認証インスタンスから取得されます。

このロールに基づく認証を指定し開始するログイン URL を次に示します。

`http://` サーバー名 `.` ドメイン名 `:` ポート `/amserver/UI/Login?user=` ユーザー名

`http://` サーバー名 `.` ドメイン名 `:` ポート `/amserver/UI/Login?org=` 組織名 `&user=` ユーザー名

org パラメータが設定されていない場合、ロールが属す組織はログイン URL そのものに指定されたサーバーホストおよびドメインから判断されます。

ユーザーエイリアスリスト属性

ユーザーに基づく認証の要求を受け取ると、認証サービスはまずユーザーが有効なユーザーであることを確認してから、ユーザーの認証設定データを取得します。ユーザーログイン URL パラメータの値に複数の有効なユーザープロファイルが関連付けられている場合は、すべてのプロファイルを指定されたユーザーにマップする必要があります。ユーザープロファイルのユーザーエイリアス属性 (`iplanet-am-user-alias-list`) には、ユーザーに属すその他のプロファイルを定義できます。マッピングが失敗すると、ユーザーは有効なセッションを拒否されます。ユーザーの 1 人がユーザーのマッピングの検証が行われない最上位の管理者であり、そのユーザーにスーパー管理者権限が与えられている場合は、例外です。

ユーザーに基づく認証のリダイレクト URL

ユーザーに基づく認証が成功または失敗した時に、Access Manager はユーザーのリダイレクト先の情報を検索します。この情報をアプリケーションが検索する優先順位を次に示します。

ユーザーに基づく認証が成功した場合のリダイレクト URL

ユーザーに基づく認証が成功した場合のリダイレクト URL は、次の場所を次の優先順位で確認することによって判断されます。

1. 認証モジュールで設定された URL。

2. goto ログイン URL パラメータで設定された URL。
3. ユーザーのプロファイル (amUser.xml) の iplanet-am-user-success-url 属性用に clientType カスタムファイルに設定された URL。
4. ユーザーのロールエントリの iplanet-am-auth-login-success-url 属性用に clientType カスタムファイルに設定された URL。
5. ユーザーの組織エントリの iplanet-am-auth-login-success-url 属性用に clientType カスタムファイルに設定された URL。
6. グローバルデフォルトとして iplanet-am-auth-login-success-url 属性用に clientType カスタムファイルに設定された URL。
7. ユーザーのプロファイル (amUser.xml) の iplanet-am-user-success-url 属性に設定された URL。
8. ユーザーのロールエントリの iplanet-am-auth-login-success-url 属性に設定された URL。
9. ユーザーの組織エントリの iplanet-am-auth-login-success-url 属性に設定された URL。
10. グローバルデフォルトとして iplanet-am-auth-login-success-url 属性に設定された URL。

ユーザーに基づく認証に失敗した場合のリダイレクト URL

ユーザーに基づく認証が失敗した場合のリダイレクト URL は、次の場所を次の順序で確認することによって判断されます。

1. 認証モジュールで設定された URL。
2. gotoOnFail ログイン URL パラメータで設定された URL。
3. ユーザーのエントリ (amUser.xml) の iplanet-am-user-failure-url 属性用に clientType カスタムファイルに設定された URL。
4. ユーザーのロールエントリの iplanet-am-auth-login-failure-url 属性用に clientType カスタムファイルに設定された URL。
5. ユーザーの組織エントリの iplanet-am-auth-login-failure-url 属性用に clientType カスタムファイルに設定された URL。
6. グローバルデフォルトとして iplanet-am-auth-login-failure-url 属性用に clientType カスタムファイルに設定された URL。
7. ユーザーのエントリ (amUser.xml) の iplanet-am-user-failure-url 属性に設定された URL。
8. ユーザーのロールエントリの iplanet-am-auth-login-failure-url 属性に設定された URL。

9. ユーザーの組織エントリの iplanet-am-auth-login-failure-url 属性に設定された URL。
10. グローバルデフォルトとして iplanet-am-auth-login-failure-url 属性に設定された URL。

ユーザーに基づく認証を設定する

1. アイデンティティ (識別情報) 管理モジュールの「表示」メニューから、「ユーザー」を選択します。

   ユーザーのリストがナビゲーション区画に表示されます。

2. 修正したいユーザーを選択し、「プロパティ」の矢印をクリックします。

   ユーザープロファイルがデータ区画に表示されます。

注 新しいユーザーを作成している場合、そのユーザーに認証設定サービスは自動的に割り当てられません。ユーザーを作成する前に、ユーザープロファイルページの上部で認証設定サービスを選択していることを確認してください。このオプションを選択しないと、ユーザーはロールに定義された認証設定を継承しません。

3. 認証設定サービスがユーザーに割り当てられていることを確認するには、「表示」メニューで「サービス」を選択します。割り当てられている場合は、認証設定サービスが割り当て済みサービスとして表示されます。
4. データ区画の「表示」メニューから「ユーザー」を選択します。
5. 「ユーザー認証設定」属性の隣にある「編集」リンクをクリックして、ユーザー用の認証モジュールを定義します。
6. 「保存」をクリックします。

認証レベルに基づく認証

それぞれの認証モジュールは、その認証レベルに整数値が関連付けられています。認証レベルを割り当てるには、「サービス設定」で認証モジュールの「プロパティ」矢印をクリックし、モジュールの認証レベル属性で対応する値を変更します。認証レベルが高いということは、1 つ以上の認証モジュールで認証を受けたそのユーザーの信頼性のレベルが高いということです。

ユーザーがそのモジュールに対する認証に成功すると、認証レベルがユーザーの SSO トークンに設定されます。複数の認証モジュールに対して認証を受ける必要があり、認証に成功した場合は、最高の認証レベルの値がユーザーの SSO トークンに設定されます。

ユーザーがサービスへのアクセスを試みる場合、サービスでは、そのユーザーの SSO トークンの認証レベルを確認することで、ユーザーがアクセスを許可されているかどうかを判別できます。次に、設定された認証レベルで認証モジュールにパスするように、ユーザーをリダイレクトします。

ユーザーは特定の認証レベルで認証モジュールにアクセスすることもできます。たとえばユーザーが次の構文でログインします。

```
http:// ホスト名 : ポート / 配備 _URI/UI/Login?authlevel= 認証レベル値
```

認証レベルが認証レベル値以上であるすべてのモジュールが、ユーザーが選択するための認証メニューとして表示されます。一致するモジュールが 1 つしかなかった場合は、その認証モジュールのログインページが直接表示されます。

この認証の方法では、ID が認証を受けられるモジュールのセキュリティレベルを、管理者が指定できます。各認証モジュールには、それぞれの「認証レベル」属性があり、この属性の値は任意の有効な整数として定義できます。認証レベルに基づく認証では、認証サービスは、ログイン URL パラメータに指定された値以上の認証レベルを持つ認証モジュールを含むメニューを持つモジュールログインページを表示します。ユーザーは、提示されたリストからモジュールを選択します。ユーザーがモジュールを選択すると、以降のプロセスはモジュールに基づく認証に基づきます。

認証レベルに基づく認証のログイン URL

認証レベルに基づく認証は、ユーザーインタフェースのログイン URL に authlevel パラメータを定義して指定できます。関連するモジュールのリストを示すログイン画面を呼び出した後、ユーザーは認証を受けるモジュールを選択する必要があります。認証レベルに基づく認証を指定し開始するログイン URL を次に示します。

```
http:// サーバー名 . ドメイン名 : ポート /amserver/UI/Login?authlevel= 認証レベル
```

および

```
http:// サーバー名 . ドメイン名 : ポート /amserver/UI/Login?org= 組織名
&authlevel= 認証レベル
```

org パラメータが設定されていない場合、ユーザーが属す組織はログイン URL そのものに指定されたサーバーホストおよびドメインから判断されます。

認証レベルに基づく認証のリダイレクト URL

認証レベルに基づく認証が成功または失敗した時に、Access Manager はユーザーのリダイレクト先の情報を検索します。この情報をアプリケーションが検索する優先順位を次に示します。

認証レベルに基づく認証が成功した場合のリダイレクト URL

認証レベルに基づく認証が成功した場合のリダイレクト URL は、次の場所を次の優先順位で確認することによって判断されます。

1. 認証モジュールで設定された URL。
2. goto ログイン URL パラメータで設定された URL。
3. ユーザーのプロファイル (amUser.xml) の iplanet-am-user-success-url 属性用に clientType カスタムファイルに設定された URL。
4. ユーザーのロールエントリの iplanet-am-auth-login-sucess-url 属性用に clientType カスタムファイルに設定された URL。
5. ユーザーの組織エントリの iplanet-am-auth-login-success-url 属性用に clientType カスタムファイルに設定された URL。
6. グローバルデフォルトとして iplanet-am-auth-login-success-url 属性用に clientType カスタムファイルに設定された URL。
7. ユーザーのプロファイル (amUser.xml) の iplanet-am-user-success-url 属性に設定された URL。
8. ユーザーのロールエントリの iplanet-am-auth-login-success-url 属性に設定された URL。
9. ユーザーの組織エントリの iplanet-am-auth-login-success-url 属性に設定された URL。
10. グローバルデフォルトとして iplanet-am-auth-login-success-url 属性に設定された URL。

認証レベルに基づく認証が失敗した場合のリダイレクト URL

認証レベルに基づく認証が失敗した場合のリダイレクト URL は、次の場所を次の順序で確認することによって判断されます。

1. 認証モジュールで設定された URL。
2. gotoOnFail ログイン URL パラメータで設定された URL。
3. ユーザーのエントリ (amUser.xml) の iplanet-am-user-failure-url 属性用に clientType カスタムファイルに設定された URL。
4. ユーザーのロールエントリの iplanet-am-auth-login-failure-url 属性用に clientType カスタムファイルに設定された URL。
5. ユーザーの組織エントリの iplanet-am-auth-login-failure-url 属性用に clientType カスタムファイルに設定された URL。
6. グローバルデフォルトとして iplanet-am-auth-login-failure-url 属性用に clientType カスタムファイルに設定された URL。

7. ユーザーのエントリ (amUser.xml) の iplanet-am-user-failure-url 属性に設定された URL。
8. ユーザーのロールエントリの iplanet-am-auth-login-failure-url 属性に設定された URL。
9. ユーザーの組織エントリの iplanet-am-auth-login-failure-url 属性に設定された URL。
10. グローバルデフォルトとして iplanet-am-auth-login-failure-url 属性に設定された URL。

モジュールに基づく認証

ユーザーは次の構文を使用して、特定の認証モジュールにアクセスできます。

http:// ホスト名 : ポート / 配備 _URI/UI/Login?module= モジュール名

認証モジュールにアクセスする前に、その認証モジュール名を含むように、コア認証サービス属性の組織認証モジュールを修正する必要があります。認証モジュール名がこの属性に含まれていない場合、ユーザーが認証を試みると「認証モジュールが拒否されました」ページが表示されます。

この認証の方法では、ユーザーは認証を受けるモジュールを指定できます。指定するモジュールは、ユーザーがアクセスする組織またはサブ組織に登録する必要があります。これは、組織のコア認証サービスの「組織認証モジュール」属性に設定されます。モジュールに基づく認証の要求を受け取ると、認証サービスは、モジュールが指定されたように正しく設定されていることを確認し、モジュールが定義されていない場合は、ユーザーはアクセスを拒否されます。

注 Access Manager コンソールを使用して認証モジュールを登録する方法については、第 7 章「認証オプション」を参照してください。

モジュールに基づく認証のログイン URL

モジュールパラメータを定義して、ユーザーインタフェースのログイン URL にモジュールに基づく認証を指定できます。モジュールに基づく認証を指定し開始するログイン URL を次に示します。

http:// サーバー名 . ドメイン名 : ポート /amserver/UI/Login?module= 認証モジュール名

http:// サーバー名 . ドメイン名 : ポート /amserver/UI/Login?org= 組織名 &module= 認証モジュール名

org パラメータが設定されていない場合、ユーザーが属す組織はログイン URL そのものに指定されたサーバーホストおよびドメインから判断されます。

モジュールに基づく認証のリダイレクト URL

モジュールに基づく認証が成功または失敗した時に、Access Manager はユーザーのリダイレクト先の情報を検索します。この情報をアプリケーションが検索する優先順位を次に示します。

モジュールに基づく認証が成功した場合のリダイレクト URL

モジュールに基づく認証が成功した場合のリダイレクト URL は、次の場所を次の優先順位で確認することによって判断されます。

1. 認証モジュールで設定された URL。
2. `goto` ログイン URL パラメータで設定された URL。
3. ユーザーのプロファイル (`amUser.xml`) の `iplanet-am-user-success-url` 属性用に `clientType` カスタムファイルに設定された URL。
4. ユーザーのロールエントリの `iplanet-am-auth-login-success-url` 属性用に `clientType` カスタムファイルに設定された URL。
5. ユーザーの組織エントリの `iplanet-am-auth-login-success-url` 属性用に `clientType` カスタムファイルに設定された URL。
6. グローバルデフォルトとして `iplanet-am-auth-login-success-url` 属性用に `clientType` カスタムファイルに設定された URL。
7. ユーザーのプロファイル (`amUser.xml`) の `iplanet-am-user-success-url` 属性に設定された URL。
8. ユーザーのロールエントリの `iplanet-am-auth-login-success-url` 属性に設定された URL。
9. ユーザーの組織エントリの `iplanet-am-auth-login-success-url` 属性に設定された URL。
10. グローバルデフォルトとして `iplanet-am-auth-login-success-url` 属性に設定された URL。

モジュールに基づく認証に失敗した場合のリダイレクト URL

モジュールに基づく認証が失敗した場合のリダイレクト URL は、次の場所を次の順序で確認することによって判断されます。

1. 認証モジュールで設定された URL。
2. `gotoOnFail` ログイン URL パラメータで設定された URL。
3. ユーザーのエントリ (`amUser.xml`) の `iplanet-am-user-failure-url` 属性用に `clientType` カスタムファイルに設定された URL。
4. ユーザーのロールエントリの `iplanet-am-auth-login-failure-url` 属性用に `clientType` カスタムファイルに設定された URL。

5. ユーザーの組織エントリの iplanet-am-auth-login-failure-url 属性用に clientType カスタムファイルに設定された URL。
6. グローバルデフォルトとして iplanet-am-auth-login-failure-url 属性用に clientType カスタムファイルに設定された URL。
7. ユーザーのエントリ (amUser.xml) の iplanet-am-user-failure-url 属性に設定された URL。
8. ユーザーのロールエントリの iplanet-am-auth-login-failure-url 属性に設定された URL。
9. ユーザーの組織エントリの iplanet-am-auth-login-failure-url 属性に設定された URL。
10. グローバルデフォルトとして iplanet-am-auth-login-failure-url 属性に設定された URL。

認証設定

認証設定サービスは、次の認証タイプ用の認証モジュールを定義するために使用します。

- 組織
- ロール
- サービス
- ユーザー

これらの認証タイプのいずれかで認証モジュールを定義すると、認証プロセスの成功または失敗に基づいて、リダイレクト URL、およびポストプロセス Java クラス仕様を提供するように設定できます。

認証モジュールを設定する前に、特定の認証モジュール名を含むように、コア認証サービス属性の組織認証モジュールを修正する必要があります。

認証設定のユーザーインタフェース

認証設定サービスでは、ユーザーがコンソール、または Access Manager 内でセキュリティ保護されたリソースにアクセスできるようになる前にパスしなければならない 1 つ以上の認証サービス (モジュール) を定義できます。組織、ロール、サービス、およびユーザーを基にした認証では、共通のユーザーインタフェースを使用して、認証モジュールを定義します。特定のオブジェクトタイプの認証設定インタフェースにアクセスする手順は、後続の節で説明します。

1. オブジェクトの「認証設定」属性の隣にある「編集」リンクをクリックして、「モジュールリスト」ウィンドウを表示します。
2. このウィンドウには、そのオブジェクトに割り当ててある認証モジュールのリストが表示されます。モジュールが存在しない場合は、「追加」をクリックして「モジュールを追加」ウィンドウを表示します。

   「モジュールを追加」ウィンドウには、定義するファイルが 3 つあります。

   モジュール名 : このプルダウンリストでは、コア認証モジュールの「組織認証モジュール」属性で有効になっている認証モジュール (追加されている可能性があるカスタムモジュールも含む) を選択できます。

   フラグ : プルダウンメニューで認証モジュールの要件を次のいずれかに指定できます。必須 - 認証には認証モジュールが必要です。

   - 必須 - 認証には認証モジュールが必要です。認証に成功または失敗すると、認証モジュールリストの次のモジュールへと認証が進行します。
   - 必要 - 認証には認証モジュールが必要です。認証に成功すると、認証モジュールリストの次のモジュールへと認証が進行します。認証に失敗すると、制御がアプリケーションに返されます。認証モジュールリストの次のモジュールには認証が進行しません。
   - 十分 - 認証に認証モジュールは不要です。認証に成功するとすぐに、制御がアプリケーションに返されます。この場合、認証モジュールリストの次のモジュールには認証が進行しません。認証に失敗すると、リストの次のモジュールへと認証が進行します。
   - オプション - 認証に認証モジュールは不要です。認証に成功または失敗しても、認証モジュールリストの次のモジュールへと認証が進行します。

   以上のフラグによって、認証モジュールの適用条件が確立されます。適用条件には上下関係があり、「必須」がもっとも高く、「オプション」がもっとも低くなります。

   たとえば、管理者が LDAP モジュールに「必須」フラグを設定している場合、ユーザーが特定のリソースにアクセスするためには、ユーザーの資格情報が LDAP の認証条件にパスすることが必要です。

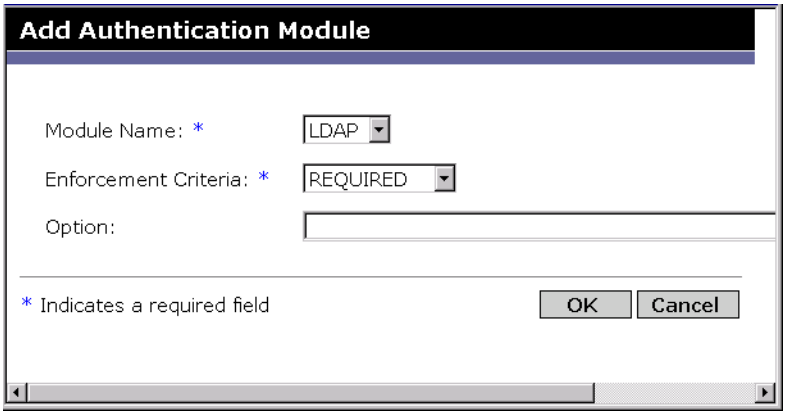
複数の認証モジュールを追加して各モジュールのフラグを「必須」に設定した場合、ユーザーがアクセスするためにはすべての認証条件にパスする必要があります。

フラグの定義の詳細については、次のサイトの JAAS (Java Authentication and Authorization Service) を参照してください。

`http://java.sun.com/security/jaas/doc/module.html`

オプション: モジュールの追加オプションをキー = 値のペアとして指定できます。複数のオプションを指定するときは、スペースで区切ります。

図 6-1 ユーザー用の「モジュールを追加」ウィンドウ



3. フィールドを選択したら、「OK」をクリックして「モジュールリスト」ウィンドウに戻ります。定義した認証モジュールがこのウィンドウに表示されます。「保存」をクリックします。

   このリストには必要なだけ多くの認証モジュールを追加できます。複数の認証モジュールを追加することを認証連鎖と言います。認証モジュールを連鎖している場合は、リストに表示される順番で、適用される階層の順序が決まります。認証連鎖については、160 ページの「認証モジュール連鎖」を参照してください。

   認証モジュールの順番を変更するには、次の手順を実行します。

   a. 「並べ換え」ボタンをクリックします。

   b. 並べ換えるモジュールを選択します。

   c. 「上」および「下」のボタンを使用して、希望する位置に移動します。

4. リストから認証モジュールを削除するには、認証モジュールの隣にあるチェックボックスを選択して「削除」をクリックします。

注	連鎖内の任意のモジュールで amadmin クレデンシャルを入力した場合は、amadmin プロファイルを受信します。この場合は、認証でエイリアスのマッピングや連鎖内のモジュールは確認されません。

認証モジュール連鎖

1 つ以上の認証モジュールが設定できるので、ユーザーは認証クレデンシャルをすべての認証モジュールに渡す必要があります。これは、認証連鎖と呼ばれます。Access Manager での認証連鎖は、認証サービスに統合された JAAS フレームワークを使用して実現されます。モジュール連鎖は、認証設定サービスの下に設定されます。登録された各モジュールには、次の 4 つの値の 1 つが割り当てられます。

- 必須
- 必要
- 十分
- オプション

フラグによって定義されている、連鎖のモジュールの認証が成功すると、JAAS フレームワークから認証サービスに制御が戻り、認証サービスでは認証に使用したすべてのユーザー ID を検証し、それらの ID を 1 人のユーザーにマッピングします。マッピングは、ユーザーのプロファイルで「ユーザーエイリアスリスト」属性を設定して行います。すべてのマップが正しい場合は有効なセッショントークンがユーザーに発行され、そうでない場合は、ユーザーは有効なセッショントークンを拒否されます。次のプロパティは、ほかのユーザーがこのユーザーに対してエイリアス化される 1 人の認証済みユーザーを表します。

- Principal (ユーザーに DN がある場合、そのユーザーの DN が含まれる)
- UserToken
- UserId

すべてのユーザー ID が同一ユーザーにマップされず、ユーザー ID の 1 つがローカルディレクトリサーバーに存在する場合は、プロファイルの動的な作成を有効にすると、その他のユーザー ID が既存のユーザーの「ユーザーエイリアスリスト」属性に追加されます。

注

- 認証連鎖で、すべてのユーザー ID が同一ユーザーにマップされない場合、失敗のリダイレクト URL は、最後に失敗した認証モジュールから選択されるか、すべての個々のモジュールが (異なるユーザー ID で) 成功する場合はなしになります。ユーザーに基づく認証の場合、認証ページでどのユーザー ID が与えられようと、失敗のリダイレクト URL は常にログイン URL のユーザーパラメータから選択されます。
- すべてのユーザー ID が同一ユーザーにマップされず、ユーザー ID の 1 つがローカルディレクトリサーバーに存在する場合は、プロファイルの動的な作成を有効にすると、その他のユーザー ID が既存のユーザーのユーザーエイリアスリスト属性に追加されます。

組織用の認証設定

認証モジュールは、最初にコア認証サービスを組織に追加することで、組織用に設定できます。

組織の認証属性を設定するには、次の手順を実行します。

1. 認証属性を設定する組織に移動します。
2. 「表示」メニューから「サービス」を選択します。
3. サービスのリスト表示で、「コア」をクリックします。

   コア認証属性がデータ区画に表示されます。
4. 管理者認証属性の隣にある「編集」リンクをクリックします。これにより、管理者専用に認証サービスを定義できます。この属性は、管理者とエンドユーザーの認証モジュールを別々のものにする必要がある場合に使用できます。デフォルトの認証モジュールは LDAP です。

   認証サービスを定義したら、「保存」をクリックして変更内容を保存します。次に「閉じる」をクリックし、組織の「コア認証」属性に戻ります。
5. 「組織認証設定」属性の隣にある「編集」リンクをクリックします。これにより、組織内の全ユーザー用に認証モジュールを定義できます。デフォルトの認証モジュールは LDAP です。
6. 認証サービスを定義したら、「保存」をクリックして変更内容を保存します。次に「閉じる」をクリックし、組織の「コア認証」属性に戻ります。

ロール用の認証設定

認証モジュールは、認証設定サービスをロールレベルで追加すると、ロール用に設定されます。

1. 認証属性を設定する組織に移動します。
2. 「表示」メニューから「ロール」を選択します。
3. 認証設定を設定するロールを選択し、「プロパティ」の矢印をクリックします。

   ロールのプロパティがデータ区画に表示されます。
4. データ区画の「表示」メニューから「サービス」を選択します。
5. 必要に応じて「認証設定」属性を修正します。これらの属性の説明については、第 33 章「認証設定サービス属性」を参照するか、またはコンソール右上の「ヘルプ」リンクをクリックしてください。
6. 「保存」をクリックします。

注

新しいロールを作成している場合、そのロールに認証設定サービスは自動的に割り当てられません。ロールを作成する前に、ロールプロファイルページの上部で認証設定サービスを選択していることを確認してください。

ロールベースの認証を有効にするときは、LDAP 認証モジュールはデフォルトのままにでき、メンバーシップを設定する必要はありません。

サービス用の認証設定

認証モジュールは、認証設定サービスを追加すると、サービス用に設定されます。そのためには、次の手順を実行します。

1. アイデンティティ (識別情報) 管理モジュールの「表示」メニューから、「サービス」を選択します。

   追加済みのサービスのリストが表示されます。認証設定サービスを追加していない場合は、次の手順に進みます。サービスを追加済みの場合は、手順 4 に進みます。
2. ナビゲーション区画で「追加」をクリックします。

   利用可能なサービスのリストがデータ区画に表示されます。
3. 「認証設定」のチェックボックスを選択し、「了解」をクリックします。

   認証設定サービスがナビゲーション区画に表示され、追加されたことが管理者に示されます。

4. 「認証設定」の矢印をクリックします。

   「サービスインスタンスリスト」がデータ区画に表示されます。

5. 認証モジュールを設定するサービスインスタンスをクリックします。

6. 「認証設定」属性を修正し、「保存」をクリックします。これらの属性の説明については、第 33 章「認証設定サービス属性」を参照するか、またはコンソール右上の「ヘルプ」リンクをクリックしてください。

ユーザー用の認証設定

1. アイデンティティ (識別情報) 管理モジュールの「表示」メニューから、「ユーザー」を選択します。

   ユーザーのリストがナビゲーション区画に表示されます。

2. 修正したいユーザーを選択し、「プロパティ」の矢印をクリックします。

   ユーザープロファイルがデータ区画に表示されます。

注 新しいユーザーを作成している場合、そのユーザーに認証設定サービスは自動的に割り当てられません。ユーザーを作成する前に、ユーザープロファイルページの上部で認証設定サービスを選択していることを確認してください。このオプションを選択しないと、ユーザーはロールに定義された認証設定を継承しません。

3. 認証設定サービスがユーザーに割り当てられていることを確認するには、「表示」メニューで「サービス」を選択します。割り当てられている場合は、認証設定サービスが割り当て済みサービスとして表示されます。

4. データ区画の「表示」メニューから「ユーザー」を選択します。

5. 「ユーザー認証設定」属性の隣にある「編集」リンクをクリックして、ユーザー用の認証モジュールを定義します。

6. 「保存」をクリックします。

アカウントのロック

認証サービスには、*n* 回失敗すると、ユーザーが認証からロックアウトされる機能があります。この機能はデフォルトではオフになっていますが、Access Manager コンソールを使用して有効にできます。

注	無効なパスワード例外をスローするモジュールのみが、アカウントロック機能を利用できます。

コア認証サービスには、この機能を有効化およびカスタマイズするための次の属性 (ただし、これらに限定されない) が含まれています。

- **「ログイン失敗時のロックアウトモードを有効」**は、アカウントロックを有効にします。
- **「ログイン失敗時のロックアウト回数」**は、ロックアウトまでのユーザーが認証を試みる回数を定義します。この回数は、ユーザー ID ごとにのみ有効であり、同一ユーザー ID が指定された回数だけ認証に失敗するとロックアウトされます。
- **「ログイン失敗時のロックアウト間隔」**は、ユーザーがロックアウトされるまでの「ログイン失敗時のロックアウト回数」値を完了する必要のある時間を分単位で定義します。
- **「ロックアウト通知の送信先電子メールアドレス」**は、ユーザーロックアウト通知が送信される電子メールアドレスを指定します。
- **「ユーザーに警告を出すまでの失敗回数」**は、警告メッセージがユーザーに表示されるまでの、認証の失敗回数を指定します。ロックアウトの発生が迫っていることを知らせる警告をユーザーが受けたあとに、さらに実行できるログインの試みを、管理者が設定できます。
- **「ログイン失敗時のロックアウト持続時間」**は、ロックアウト後に認証を再度試みるまでに、ユーザーが待つ必要がある時間を分単位で定義します。
- **「ロックアウト属性名」**は、物理ロックのためにユーザーのプロファイルのどの LDAP 属性を「非アクティブ」に設定するかを定義します。
- **「ロックアウト属性値」**は、**「ロックアウト属性名」**に指定された LDAP 属性を「非アクティブ」または「アクティブ」のどちらに設定するかを定義します。

アカウントのロックアウトに関する電子メールの通知が、管理者に送信されます。アカウントロックのアクティビティはログにも記録されます。アカウントロックの属性については、第 20 章「コア認証属性」を参照してください。

注 Microsoft® Windows 2000 オペレーティングシステムでこの機能を使用する場合の特別な指示については、『Access Manager Developer's Guide』の付録 A、「AMConfig.properties File」の「Simple Mail Transfer Protocol (SMTP)」を参照してください。

Access Manager では、物理ロックとメモリロックの 2 つのタイプのアカウントロックがサポートされます。次の節では、この 2 つについて説明します。

物理ロック

物理ロックは、Access Manager のデフォルトのロック動作です。ロックは、ユーザーのプロファイルの LDAP 属性の状態を非アクティブに変更することによって開始されます。「ロックアウト属性名」属性は、ロックの目的で使用する LDAP 属性を定義します。物理ロックの設定については、『Sun Java System Access Manager 管理ガイド』を参照してください。

注 エイリアス化されたユーザーとは、LDAP プロファイルで「ユーザーエイリアスリスト」属性 (`amUser.xml` の `iplanet-am-user-alias-list`) を設定して既存の LDAP ユーザープロファイルにマッピングされるユーザーのことです。エイリアス化されたユーザーは、コア認証サービスの「エイリアス検索属性名」フィールドに `iplanet-am-user-alias-list` を追加することによって確認できます。つまり、エイリアス化されたユーザーがロックアウトされると、ユーザーがエイリアス化された実際の LDAP プロファイルがロックされます。これは、LDAP およびメンバーシップ以外の認証モジュールの物理ロックアウトにのみ関係します。

メモリロック

メモリロックは、「ログイン失敗時のロックアウト持続時間」属性の値を 0 よりも大きな値に変更すると有効になります。有効にすると、ユーザーのアカウントは指定された時間メモリにロックされます。指定された期間が過ぎると、このアカウントはロック解除されます。メモリロック機能を使用する場合の考慮事項を次に示します。

- Access Manager が再起動されると、メモリにロックされたすべてのアカウントはロック解除されます。
- ユーザーのアカウントがメモリにロックされ、管理者がロックアウト持続時間を 0 に設定してアカウントロックメカニズムを物理ロックに変更すると、ユーザーのアカウントはメモリでロック解除され、ロックカウントがリセットされます。

- メモリのロックアウト後、LDAP およびメンバーシップ以外の認証モジュールを使用するときに、ユーザーが正しいパスワードでログインを試みると、「ユーザーがアクティブではありません」ではなく、「ユーザーのプロファイルがこの組織にありません。」エラーが返されます。

注 ユーザーのプロファイルに「失敗 URL」属性を設定する場合、ロックアウト警告メッセージもアカウントがロックされたことを示すメッセージも表示されず、ユーザーは定義された URL にリダイレクトされます。

認証サービスのフェイルオーバー

プライマリサーバーにハードウェアまたはソフトウェア上の問題で障害が発生した場合、または一時的にシャットダウンした場合には、認証サービスのフェイルオーバーにより、認証要求はセカンダリサーバーへ自動的にリダイレクトされます。

認証コンテキストは認証サービスが使用可能な Access Manager のインスタンス上でまず作成されなければなりません。Access Manager のこのインスタンスが使用できない場合は、認証フェイルオーバーメカニズムにより Access Manager の別のインスタンス上に認証コンテキストが作成されます。認証コンテキストは次のような順序でサーバーが使用可能かどうか確認します。

1. 認証サービス URL を AutoContext API に送ります。次に例を示します。

   ```
   AuthContext(orgName, url)
   ```

   この API を使う場合は、URL で参照されたサーバーのみを使用します。この場合、そのサーバー上で認証サービスが使用可能であっても、フェイルオーバーは起きません。

2. 認証コンテキストが `AMConfig.properties` ファイルの `com.iplanet.am.server*` 属性に定義されたサーバーをチェックします。

3. 手順 2 で失敗すると、認証コンテキストはネーミングサービスが利用可能なサーバーからのプラットフォームリストを照会します。ディレクトリサーバーの 1 つのインスタンスを共有する複数のインスタンスが、(主にフェイルオーバーを目的として) Access Manager 上にインストールされたときに、このプラットフォームリストが自動的に作成されます。

   たとえば、プラットフォームリストに `Server1`、`Server2`、および `Server3` の URL が含まれていると、認証コンテキストは `Server1`、`Server2`、および `Server3` のいずれかで認証が成功するまでループします。

プラットフォームリストは、ネーミングサービスの有無に依存しているので、常に同一のサーバーから得られるわけではありません。さらに、ネーミングサービスのフェイルオーバーが最初に起こります。複数ネーミングサービス URL は AMConfing.properties の com.iplanet.am.naming.url プロパティに定義されます。利用可能な最初のネーミングサービス URL は、認証フェイルオーバーが発生する (プラットフォームサーバーリスト中の) サーバーのリストを持つサーバーを特定するのに使われます。

完全修飾ドメイン名のマッピング

完全修飾ドメイン名 (FQDN) のマッピングは、ユーザーが誤った URL を入力した (保護されたリソースにアクセスするために部分的なホスト名または IP アドレスを指定したなど) 場合に認証サービスが訂正を行うことができるようにします。FQDN のマッピングは、AMConfig.properties ファイルで com.sun.identity.server.fqdnMap 属性を変更することによって可能になります。このプロパティは次の形式で指定します。

com.sun.identity.server.fqdnMap[*invalid-name*]=*valid-name*

invalid-name の値はユーザーが入力する可能性がある無効な FQDN ホスト名であり、*valid-name* はフィルタがユーザーをリダイレクトする実際のホスト名です。定められた要件に準拠するかぎり、コード例 1-1 に示すようにいくつでもマッピングを指定できます。このプロパティを設定しない場合は、ユーザーは com.iplanet.am.server.host=*server_name* プロパティに設定されたデフォルトのサーバー名に送信されます。このプロパティも AMConfig.properties ファイルにあります。

コード例 6-1 AMConfig.properties の FQDN マッピング属性

```
com.sun.identity.server.fqdnMap[isserver]=isserver.mydomain.com
com.sun.identity.server.fqdnMap[isserver.mydomain]=isserver.mydomain.com
com.sun.identity.server.fqdnMap[IP address]=isserver.mydomain.com
```

FQDN のマッピングの使用例

このプロパティは、サーバーにホストされたアプリケーションが複数のホスト名でアクセス可能な場合に、複数のホスト名のマッピングを作成するために使用できます。このプロパティを使用して、特定の URL について Access Manager が訂正を行わないように設定することもできます。たとえば、IP アドレスを使用してアプリケーションにアクセスするユーザーにリダイレクトが必要ない場合は、この機能は次のようなマップエントリを指定して実現できます。

```
com.sun.identity.server.fqdnMap[IP アドレス]=IP アドレス
```

警告 複数のマッピングを定義する場合は、無効な FQDN 名で値が重複しないようにします。そのようにしないと、アプリケーショにアクセスできなくなる場合があります。

持続 Cookie

持続 Cookie とは、Web ブラウザを閉じたあとも存在する Cookie のことであり、ユーザーが再認証なしに新しいブラウザセッションにログインすることを可能にします。Cookie の名前は、`AMConfig.properties` の `com.iplanet.am.pcookie.name` プロパティに定義されます。デフォルト値は、`DProPCookie` です。Cookie の値は、3DES で暗号化された文字列であり、この文字列には、ユーザー DN、組織名、認証モジュール名、最大セッション時間、アイドル時間、およびキャッシュ時間が含まれます。持続 Cookie を有効にするには、次のようにします。

1. コア認証モジュールで「持続 `Cookie` モード」をオンにします。
2. コア認証モジュールで「`Cookie` の最大持続時間」属性の時間値を設定します。
3. ユーザーインタフェースのログイン URL に `yes` の値で iPSPCookie パラメータを付加します。

   ユーザーがこの URL を使用して認証を受けると、ブラウザを閉じた場合、新しいブラウザウィンドウを開くことができ、再認証なしにコンソールにリダイレクトされます。手順 2 で定義された時間が経過するまでこのようになります。

持続 Cookie モードは、次の認証 SPI メソッドを使用してオンにできます。

```
AMLoginModule.setPersistentCookieOn()
```

複数 LDAP 認証モジュールの設定

フェイルオーバーの形式として、あるいは、Access Manager コンソールで値フィールドが 1 つだけ提供されている場合に、属性に複数の値を設定するために、管理者は 1 つの組織に複数の LDAP 認証モジュール設定を定義できます。これら追加の設定はコンソールに表示されませんが、要求を行っているユーザーの承認が初期検索で見つからない場合に、主設定とともに機能します。たとえば、1 つの組織で 2 つの異なるドメインでの認証に LDAP サーバーを介した検索を定義したり、あるいは 1 つのドメインに複数のユーザーネーミング属性を設定できます。後者の場合、コンソールにはテキストフィールドが 1 つのみあり、第 1 の検索基準でユーザーが見つからない場合は、LDAP モジュールは第 2 の検索範囲で検索します。追加の LDAP 構成を設定する手順を次に示します。

追加の LDAP 構成を設定する

1. 2 番目 (または 3 番目の) LDAP 認証設定に必要な属性および新しい値の完全なセットを含めた XML ファイルを作成します。

   利用可能な属性は、`etc/opt/SUNWam/config/xml` にある `amAuthLDAP.xml` で参照できます。この手順で作成された XML ファイルは、`amAuthLDAP.xml` とは異なり、`amadmin.dtd` の構造に基づいています。このファイルには、任意のまたはすべての属性を定義できます。コード例 6-2 は、LDAP 認証設定に利用できるすべての属性の値が含まれる副設定ファイルの例です。

コード例 6-2 LDAP の副設定を追加するための XML ファイルの例

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
  Copyright (c) 2002 Sun Microsystems, Inc. All rights reserved.
  Use is subject to license terms.
-->
<!DOCTYPE Requests
    PUBLIC "-//iPlanet//Sun ONE Identity Server 6.0 Admin CLI DTD//EN"
    "jar://com/iplanet/am/admin/cli/amAdmin.dtd"
>
<!--
  Before adding subConfiguration load the schema with
GlobalConfiguration defined and replace corresponding
 serviceName and subConfigID in this sample file OR load
 serviceConfigurationRequests.xml before loading this sample
-->
<Requests>
<OrganizationRequests DN="dc=iplanet,dc=com">
    <AddSubConfiguration subConfigName = "ssc"
        subConfigId = "serverconfig"
        priority = "0" serviceName="iPlanetAMAuthLDAPService">

        <AttributeValuePair>
            <Attribute name="iplanet-am-auth-ldap-server"/>
```

コード例 6-2　　LDAP の副設定を追加するための XML ファイルの例 (続き)

```
            <Value>newvalue</Value>
        </AttributeValuePair>
        <AttributeValuePair>
            <Attribute name="iplanet-am-auth-ldap-server"/>
            <Value>vbrao.red.iplanet.com:389</Value>
        </AttributeValuePair>
        <AttributeValuePair>
            <Attribute name="iplanet-am-auth-ldap-base-dn"/>
            <Value>dc=iplanet,dc=com</Value>
        </AttributeValuePair>
        <AttributeValuePair>
            <Attribute name="planet-am-auth-ldap-bind-dn"/>
            <Value>cn=amldapuser,ou=DSAME Users,dc=iplanet,dc=com</Value>
        </AttributeValuePair>
        <AttributeValuePair>
            <Attribute name="iplanet-am-auth-ldap-bind-passwd"/>
            <Value> プレーンテキストのパスワード </Value>
        </AttributeValuePair>
        <AttributeValuePair>
            <Attribute name="iplanet-am-auth-ldap-user-naming-attribute"/>
            <Value>uid</Value>
        </AttributeValuePair>
        <AttributeValuePair>
            <Attribute name="iplanet-am-auth-ldap-user-search-attributes"/>
            <Value>uid</Value>
        </AttributeValuePair>
        <AttributeValuePair>
            <Attribute name="iplanet-am-auth-ldap-search-scope"/>
            <Value>SUBTREE</Value>
        </AttributeValuePair>
        <AttributeValuePair>
            <Attribute name="iplanet-am-auth-ldap-ssl-enabled"/>
            <Value>false</Value>
        </AttributeValuePair>
        <AttributeValuePair>
            <Attribute name="iplanet-am-auth-ldap-return-user-dn"/>
            <Value>true</Value>
        </AttributeValuePair>
        <AttributeValuePair>
            <Attribute name="iplanet-am-auth-ldap-auth-level"/>
            <Value>0</Value>
        </AttributeValuePair>
        <AttributeValuePair>
            <Attribute name="iplanet-am-auth-ldap-server-check"/>
            <Value>15</Value>
        </AttributeValuePair>

    </AddSubConfiguration>

</OrganizationRequests>
</Requests>
```

2. 手順 1 で作成した XML ファイルで iplanet-am-auth-ldap-bind-passwd の値としてプレーンテキストのパスワードをコピーします。

   この属性の値は、41 ページのコード例 1-2 に太字で示されています。

3. amadmin コマンド行ツールを使用して、XML ファイルをロードします。

```
./amadmin -u amadmin -w administrator_password -v -t
name_of_XML_file.
```

この 2 番目の LDAP 設定は、Access Manager コンソールを使用して表示も変更もできません。

ヒント 複数 LDAP 設定に利用できるサンプルが用意されています。/*AcessManager-base*/SUNWam/samples/admin/cli/bulk-ops/ にある serviceAddMultipleLDAPConfigurationRequests.xml コマンド行テンプレートを参照してください。手順については、/*AcessManager-base*/SUNWam/samples/admin/cli/ にある Readme.html を参照してください。

セッションのアップグレード

認証サービスでは、1 つの組織に対して同一ユーザーが実行した 2 回目の成功した認証に基づいて有効なセッショントークンのアップグレードを行うことができます。有効なセッショントークンを持つユーザーが、現在の組織によってセキュリティ保護されているリソースに対して認証を試み、この 2 回目の認証が成功すると、セッションは新しい認証に基づく新しいプロパティで更新されます。認証が失敗すると、ユーザーの現在のセッションがアップグレードなしに戻されます。有効なセッショントークンを持つユーザーが別の組織によってセキュリティ保護されているリソースに対して認証を試みると、新しい組織の認証を受けるどうかを尋ねるメッセージを受け取ります。この時点では、ユーザーは、現在のセッションを維持するか、または新しい組織の認証を受けることができます。認証が成功すると、古いセッションは破棄され、新しいセッションが作成されます。

セッションのアップグレード時に、ログインページの時間切れになると、元の成功 URL へのリダイレクトが発生します。タイムアウト値は、次の条件に基づいて判断されます。

- 各モジュールに設定されたページタイムアウト値 (デフォルトは 1 分)
- AMConfig.properties の com.iplanet.am.invalidMaxSessionTime プロパティ (デフォルトは 10 分)
- iplanet-am-max-session-time (デフォルトは 120 分)

com.iplanet.am.invalidMaxSessionTimeout と iplanet-am-max-session-time の値は、ページタイムアウト値よりも大きい必要があり、そうでない場合はセッションアップグレード時に有効なセッション情報が失われ、前回の成功 URL へのリダイレクトは失敗します。

検証プラグインインタフェース

管理者は、組織に適したユーザー名またはパスワード検証ロジックを作成し、そのロジックを認証サービスにプラグインできます。この機能は、LDAP およびメンバーシップの認証モジュールのみでサポートされます。ユーザーを認証したりパスワードを変更する前に、Access Manager はこのプラグインを呼び出します。検証が成功すると、認証が継続され、失敗すると、認証失敗ページがスローされます。プラグインは、サービス管理 SDK の一部である com.iplanet.am.sdk.AMUserPasswordValidation クラスを拡張します。SDK についての情報は、Access Manager Javadocs の com.iplanet.am.sdk パッケージを参照してください。次の手順は、Access Manager 用の検証プラグインを作成および設定する方法を示しています。

1. 新しいプラグインクラスは、com.iplanet.am.sdk.AMUserPasswordValidation クラスを拡張し、validateUserID() および validatePassword() メソッドを実装します。検証が失敗した場合は、AMException がスローされます。
2. プラグインクラスをコンパイルし、.class ファイルを必要な場所に配置します。実行時に Access Manager がプラグインにアクセスできるように、クラスパスを更新します。
3. 最上位の管理者として Access Manager コンソールにログインします。「サービス管理」タブをクリックし、管理サービスの属性にアクセスします。「ユーザー ID とパスワードの検証プラグインクラス」フィールドにパッケージ名を含むプラグインクラスの名前を入力します。
4. ログアウトし、ログインし直します。

JAAS 共有状態

JAAS 共有状態は、認証モジュール間でユーザー ID とパスワードの両方の共有を実現します。各認証モジュールに対して次のオプションを定義します。

- 組織
- ユーザー
- サービス
- ロール

失敗した場合、モジュールは必要なクレデンシャルを要求します。認証の失敗後、モジュールが停止するか、ログアウト共有状態がクリアされます。

JAAS 共有状態の有効化

JAAS 共有状態を設定するには、次のようにします。

- `iplanet-am-auth-sharedstate-enabled` オプションを使用します。
- 共有状態オプションの使用法を次に示します。
  `iplanet-am-auth-shared-state-enabled=true`
- このオプションのデフォルトは、true です。

失敗すると、認証モジュールは、JAAS の仕様で提案される `tryFirstPass` オプションの動作ごとに必要なクレデンシャルを要求します。

JAAS 共有状態ストアオプション

「JAAS 共有状態ストア」オプションを設定するには、次のようにします。

- iplanet-amauth-store-shared-state-enabled オプションを使用します。
- 「共有状態」オプションの使用法を次に示します。
  `iplanet-am-auth-shared-state-enabled=true`
- このオプションのデフォルトは、false です。

コミット、中断、またはログアウト後に、共有状態はクリアされます。

第 7 章

認証オプション

Sun Java™ System Access Manager 6 2005Q1 では、認証のフレームワークを備えています。認証フレームワークは、エンタープライズのアプリケーションにアクセスするユーザーのアイデンティティを確認するプロセスです。認証とは、エンタープライズ内のアプリケーションにアクセスするユーザーのアイデンティティを確認するためのプロセスです。ユーザーは、Access Manager コンソールまたは Access Manager で保護されたリソースにアクセスする前に、認証プロセスにパスする必要があります。認証は、ユーザーのアイデンティティを検証するプラグインによって実装されます。このプラグインアーキテクチャの詳細については、『Access Manager Developer's Guide』で説明します。

デフォルト値の設定、認証モジュールの追加、認証テンプレートの作成、および関連認証モジュールの有効化を行うには、Access Manager コンソールを使用します。この章では、認証モジュールの概要と追加手順について説明します。この章は、次の節で構成されています。

コア認証

Access Manager では、コア認証モジュールばかりではなく、デフォルトで 15 種類の認証モジュールを提供しています。コア認証モジュールでは、認証モジュールの全体的な設定を行います。Active Directory 認証、匿名認証、証明書に基づく認証、HTTP 基本認証、JDBC 認証、LDAP 認証、任意の認証のモジュールを追加して有効にする前に、コア認証モジュールの追加と有効化を行う必要があります。コア認証モジュールおよび LDAP 認証モジュールは両方とも、デフォルトの組織に対して自動的に有効になります。第 20 章「コア認証属性」に、コア属性の詳細なリストを示します。

コアサービスを追加し、有効にする

1. コアモジュールを追加する組織に移動します。
2. 「表示」メニューから「サービス」を選択します。
3. ナビゲーション区画で「追加」をクリックします。

   利用可能なモジュールのリストがデータ区画に表示されます。
4. 「コア認証」のチェックボックスを選択し、「追加」をクリックします。

   コア認証モジュールがナビゲーション区画に表示され、追加されたことが管理者に示されます。
5. 「コア」認証のプロパティの矢印をクリックします。

   「現在このサービスにはテンプレートが存在しません。新規に作成しますか ?」というメッセージがデータ区画に表示されます。
6. 「作成」をクリックします。

   コア属性がデータ区画に表示されます。必要に応じて属性を修正します。コア属性の説明については、第 20 章「コア認証属性」を参照するか、またはコンソール右上の「ヘルプ」リンクをクリックしてください。

Active Directory 認証

Active Directory 認証モジュールでは、LDAP ディレクトリ認証モジュールと同様の認証が実行されますが、LDAP 認証モジュールの場合の Directory Server ではなく、Microsoft の Active Directory™ サーバーが使用されます。LDAP 認証モジュールを Active Directory サーバー用に設定できますが、このモジュールでは、LDAP と Active Directory 認証の両方を同一組織下に存在させることができます。

注 このリリースの Active Directory 認証モジュールでは、ユーザー認証のみがサポートされます。パスワードポリシーは LDAP 認証モジュールのみでサポートされます。

Active Directory 認証を追加し、有効にする

組織管理者または最上位レベル管理者として、Access Manager にログインする必要があります。

1. メンバーシップ認証を追加する組織に移動します。
2. 「表示」メニューから「サービス」を選択します。

   コアモジュールが追加済みの場合は、ナビゲーション区画に表示されます。追加済みでない場合は、Active Directory 認証モジュールとともに追加できます。
3. ナビゲーション区画で「追加」をクリックします。

   利用可能なモジュールのリストがデータ区画に表示されます。
4. 「Active Directory 認証」のチェックボックスを選択し、「追加」をクリックします。

   Active Directory 認証モジュールがナビゲーション区画に表示され、追加されたことが管理者に示されます。
5. 「Active Directory」認証のプロパティの矢印をクリックします。

   「現在このモジュールにはテンプレートが存在しません。新規に作成しますか ?」というメッセージがデータ区画に表示されます。
6. 「作成」をクリックします。

   Active Directory 認証属性がデータ区画に表示されます。必要に応じて属性を修正します。
7. 「保存」をクリックします。

   Active Directory 認証モジュールが有効になりました。

Active Directory 認証を使用してログインする

Active Directory 認証を使用してログインするには、282 ページの「組織認証モジュール」のコア認証モジュール属性を修正し、Active Directory 認証を有効にして選択する必要があります。これにより、ユーザーが `http://` ホスト名 : ポート / 配備 `_URI/UI/Login?module=AD` (大文字と小文字を区別) を使用してログインするときに、Active Directory 認証のログインウィンドウが表示されます。使用している認証タイプ (サービス、ロール、ユーザー、組織など) によっては、認証モジュールをデフォルトとして設定する場合に、URL でモジュール名を指定する必要がありません。

匿名認証

デフォルトでは、このモジュールを有効にすると、ユーザーは *anonymous* ユーザーとして Access Manager にログインできるようになります。「有効な匿名ユーザーリスト」属性を設定して、このモジュールに匿名ユーザーのリストを定義することもできます。匿名アクセスを許可するということは、パスワードなしでアクセスさせるということです。匿名アクセスは、特定の種類のアクセス (読み取りのためのアクセスや検索のためのアクセスなど)、特定のサブツリー、またはディレクトリ内の個別のエントリに制限されます。

匿名認証を追加し、有効にする

組織管理者または最上位レベル管理者として、Access Manager にログインする必要があります。

1. 匿名認証を追加する組織に移動します。
2. 「表示」メニューから「サービス」を選択します。

   コアモジュールが追加済みの場合は、ナビゲーション区画に表示されます。追加済みでない場合は、匿名認証モジュールとともに追加できます。
3. ナビゲーション区画で「追加」をクリックします。

   利用可能なモジュールのリストがデータ区画に表示されます。
4. 「匿名認証」のチェックボックスを選択し、「追加」をクリックします。

   匿名認証モジュールがナビゲーション区画に表示され、追加されたことが管理者に示されます。
5. 「匿名」認証のプロパティの矢印をクリックします。

「現在このサービスにはテンプレートが存在しません。新規に作成しますか?」というメッセージがデータ区画に表示されます。

6. 「作成」をクリックします。

   匿名認証属性がデータ区画に表示されます。必要に応じて属性を修正します。これらの属性の説明については、第 18 章「匿名認証属性」を参照するか、またはコンソール右上の「ヘルプ」リンクをクリックしてください。

7. 「保存」をクリックします。

   匿名認証モジュールが有効になります。

匿名認証を使用してログインする

匿名認証を使用してログインするには、282 ページの「組織認証モジュール」のコア認証モジュール属性を修正し、匿名認証を有効にして選択する必要があります。これにより、ユーザーが `http(s)://` ホスト名 `:` ポート `/` サーバー `_配備_URI//UI/Login?module=Anonymous&org=`*org_name* を使用してログインするときに、匿名認証のログインウィンドウが表示されます。匿名認証のログインウィンドウを表示せずにログインするには、次の構文を使用します。

`http(s)://` ホスト名 `:` ポート `/` サーバー `_配備_URI//UI/Login?module=Anonymous&org=`*org_name*`&Login.Token1=user_id`

使用している認証タイプ (サービス、ロール、ユーザー、組織など) によっては、認証モジュールをデフォルトとして設定する場合に、URL でモジュール名を指定する必要がありません。

注 匿名認証モジュールにおけるデフォルトの匿名ユーザー名属性値は `anonymous` です。ユーザーがログインするときは、この名前が使用されます。デフォルトの匿名ユーザーを組織内に作成する必要があります。そのユーザー ID は、匿名認証属性で指定されているユーザー名と同一にする必要があります。大文字と小文字を区別するようにもできます。

証明書に基づく認証

証明書に基づく認証では、個人用デジタル証明書 (PDC) を使用してユーザーを特定し、認証します。Directory Server に格納された PDC に一致すること、また証明書失効リスト (CRL) で確認されていることを求めるように、PDC を設定できます。

証明書に基づく認証モジュールを組織に追加する前に、行う必要のある作業があります。まず、Access Manager とともにインストールした Web コンテナを保護し、証明書に基づく認証で使用できるように設定する必要があります。証明書に基づくモジュールを有効にする前に、Web Server に対するこれらの初期設定手順について、『Sun ONE Web Server 6.1 管理者ガイド』の第 6 章「証明書と鍵の使用」を参照してください。このマニュアルは、次の場所にあります。

```
http://docs.sun.com/db/prod/s1websrv#hic
```

または、次の場所にある『Sun ONE Application Sever Administrator's Guide to Security』を参照してください。

```
http://docs.sun.com/db/prod/s1appsrv#hic
```

注 証明書に基づくモジュールを使用して認証されるユーザーは、ブラウザ用に PDC を要求する必要があります。使用しているブラウザによって、手順が異なります。詳細は、お使いのブラウザのマニュアルを参照してください。

証明書に基づく認証を追加し、有効にする

組織管理者として、Access Manager にログインする必要があります。

1. 証明書に基づく認証を追加する組織に移動します。
2. 「表示」メニューから「サービス」を選択します。

   コアモジュールが追加済みの場合は、ナビゲーション区画に表示されます。追加済みでない場合は、証明書に基づく認証モジュールとともに追加できます。
3. ナビゲーション区画で「追加」をクリックします。

   利用可能なモジュールのリストがデータ区画に表示されます。
4. 「証明書」認証のチェックボックスを選択し、「追加」をクリックします。

   証明書に基づく認証モジュールがナビゲーション区画に表示され、追加されたことが管理者に示されます。
5. 「証明書」に基づく認証のプロパティの矢印をクリックします。

「現在このサービスにはテンプレートが存在しません。新規に作成しますか?」というメッセージがデータ区画に表示されます。

6. 「作成」をクリックします。

   証明書に基づく認証属性がデータ区画に表示されます。必要に応じて属性を修正します。これらの属性の説明については、第 19 章「証明書認証属性」を参照するか、またはコンソール右上の「ヘルプ」リンクをクリックしてください。

7. 「保存」をクリックします。

証明書に基づく認証のプラットフォームサーバーリストにサーバー URL を追加する

このモジュールを追加するためには、Access Manager に組織管理者としてログインし、Access Manager と Web コンテナに対して SSL を設定し、クライアント認証を有効化しておく必要があります。詳細は、55 ページの「Access Manager を SSL モードに設定する」を参照してください。

証明書に基づく認証を使用してログインする

証明書に基づく認証をデフォルトの認証方法として設定するには、コア認証モジュール属性である「組織認証モジュール」(282 ページ参照) を修正する必要があります。これにより、ユーザーが `https://` ホスト名 : ポート / 配備 _URI`/UI/Login?module=Cert` を使用してログインするときに、証明書に基づく認証のログインウィンドウが表示されます。使用している認証タイプ (ロール、ユーザー、組織など) によっては、認証モジュールをデフォルトとして設定する場合に、URL でモジュール名を指定する必要がありません。

HTTP 基本認証

HTTP プロトコルのビルトイン認証サポートである基本認証を使用します。Web サーバーはユーザー名とパスワードを求めるクライアント要求を発行し、その情報を認証済み要求の一部としてサーバーに返します。Access Manager ではユーザー名とパスワードを取得し、LDAP 認証モジュールに対してユーザーを内部的に認証します。HTTP 基本認証が正常に機能するために、LDAP 認証モジュールを追加する必要があります (HTTP 基本モジュールを単独で追加しても機能しない)。詳細は、185 ページの「LDAP 認証を追加し、有効にする」を参照してください。いったん認証に成功したユーザーには、以降の認証でユーザー名とパスワードの入力は要求されません。

HTTP 基本認証を追加し、有効にする

組織管理者または最上位レベル管理者として Access Manager にログインし、LDAP 認証モジュールを前もって登録しておきます。

1. HTTP 基本認証を追加する組織に移動します。
2. 「表示」メニューから「サービス」を選択します。

   コアモジュールが追加済みの場合は、ナビゲーション区画に表示されます。追加済みでない場合は、HTTP 基本認証モジュールとともに追加できます。
3. ナビゲーション区画で「追加」をクリックします。

   利用可能なモジュールのリストがデータ区画に表示されます。
4. 「HTTP 基本認証」のチェックボックスを選択し、「追加」をクリックします。

   HTTP 基本認証モジュールがナビゲーション区画に表示され、追加されたことが管理者に示されます。
5. 「HTTP 基本」認証のプロパティの矢印をクリックします。

   「現在このサービスにはテンプレートが存在しません。新規に作成しますか ?」というメッセージがデータ区画に表示されます。
6. 「作成」をクリックします。

   HTTP 基本認証属性がデータ区画に表示されます。必要に応じて属性を修正します。これらの属性の説明については、第 21 章「HTTP 基本認証属性」を参照するか、またはコンソール右上の「ヘルプ」リンクをクリックしてください。
7. 「保存」をクリックします。

   HTTP 基本認証モジュールが有効になります。

HTTP 基本認証を使用してログインする

LDAP 認証を使用してログインするには、282 ページの「組織認証モジュール」のコア認証モジュール属性を修正し、HTTP 基本認証を有効にして選択する必要があります。これにより、ユーザーが `http://` ホスト名 : ポート / サーバー _ 配備 _URI/`UI/Login?module=HTTPBasic` を使用してログインするときに、HTTP 基本認証のログインウィンドウが表示されます。使用している認証タイプ (サービス、ロール、ユーザー、組織など) によっては、認証モジュールをデフォルトとして設定する場合に、URL でモジュール名を指定する必要がありません。認証に失敗した場合、ユーザーは新しいインスタンスを開いてログインし直す必要があります。HTTP 基本認証を使用したあとに完全にログアウトするには、存在するすべてのブラウザインスタンスを閉じて、新しいブラウザインスタンスを開始する必要があります。

JDBC 認証

JDBC (Java Database Connectivity) 認証モジュールでは、JDBC 技術に対応したドライバを提供する SQL データベースを通して Access Manager でユーザーを認証できるようにするメカニズムが提供されます。SQL データベースへの接続は、JDBC ドライバを通して直接行うか、JNDI 接続プールで行います。

注 このモジュールは、MySQL4.0 と Oracle 8i でテストされています。

JDBC 認証を追加し、有効にする

組織管理者または最上位レベル管理者として、Identity Server にログインする必要があります。

1. JDBC 認証を追加する組織に移動します。
2. 「表示」メニューから「サービス」を選択します。

   コアモジュールが追加済みの場合は、ナビゲーション区画に表示されます。追加済みでない場合は、JDBC 認証モジュールとともに追加できます。
3. ナビゲーション区画で「追加」をクリックします。

   利用可能なモジュールのリストがデータ区画に表示されます。
4. 「JDBC 認証」のチェックボックスを選択し、「追加」をクリックします。

   JDBC 認証モジュールがナビゲーション区画に表示され、追加されたことが管理者に示されます。

5. 「JDBC」認証のプロパティの矢印をクリックします。

   「現在このサービスにはテンプレートが存在しません。新規に作成しますか ?」というメッセージがデータ区画に表示されます。

6. 「作成」をクリックします。

   JDBC 認証属性がデータ区画に表示されます。必要に応じて属性を修正します。

7. 「保存」をクリックします。

   JDBC 認証モジュールが有効になります。

JDBC 認証を使用してログインする

JDBC 認証を使用してログインするには、282 ページの「組織認証モジュール」のコア認証モジュール属性を修正し、JDBC 認証を有効にして選択する必要があります。これにより、ユーザーが `http://` ホスト名 : ポート / 配備 `_URI/UI/Login?module=JDBC` (大文字と小文字を区別) を使用してログインするときに、JDBC 認証のログインウィンドウが表示されます。使用している認証タイプ (サービス、ロール、ユーザー、組織など) によっては、認証モジュールをデフォルトとして設定する場合に、URL でモジュール名を指定する必要がありません。

LDAP ディレクトリ認証

LDAP 認証モジュールを使用すると、ユーザーがログインするときに、特定のユーザー DN およびパスワードを使用して、LDAP Directory Server にバインドする必要があります。すべての組織ベースの認証では、デフォルトの認証モジュールです。ユーザーが Directory Server に存在するユーザー ID およびパスワードを指定すると、ユーザーは有効な Access Manager セッションへのアクセスが許可され、セットアップされます。コア認証モジュールおよび LDAP 認証モジュールは両方とも、デフォルトの組織に対して自動的に有効になります。このモジュールが無効である場合の手順を次に示します。

LDAP 認証を追加し、有効にする

組織管理者または最上位レベル管理者として、Access Manager にログインする必要があります。

1. LDAP 認証を追加する組織に移動します。
2. 「表示」メニューから「サービス」を選択します。

   コアモジュールが追加済みの場合は、ナビゲーション区画に表示されます。追加済みでない場合は、LDAP 認証モジュールとともに追加できます。
3. ナビゲーション区画で「追加」をクリックします。

   利用可能なモジュールのリストがデータ区画に表示されます。
4. 「LDAP 認証」のチェックボックスを選択し、「追加」をクリックします。

   LDAP 認証モジュールがナビゲーション区画に表示され、追加されたことが管理者に示されます。
5. 「LDAP」認証のプロパティの矢印をクリックします。

   「現在このサービスにはテンプレートが存在しません。新規に作成しますか ?」というメッセージがデータ区画に表示されます。
6. 「作成」をクリックします。

   LDAP 認証属性がデータ区画に表示されます。必要に応じて属性を修正します。これらの属性の説明については、第 23 章「LDAP 認証属性」を参照するか、またはコンソール右上の「ヘルプ」リンクをクリックしてください。
7. パスワードを「root ユーザーバインドパスワード」属性に入力します。デフォルトで、インストール中に入力した `amldapuser` パスワードが、バインドユーザーとして使用されます。匿名アクセスでのユーザーエントリ読み出しを Directory Server が許可する場合は、この手順はスキップできます。

   別のバインドユーザーを使用するには、「root ユーザーバインド DN」属性でユーザーの DN を変更し、そのユーザーのパスワードを「root ユーザーバインドパスワード」属性に入力します。
8. 「保存」をクリックします。

   LDAP 認証モジュールが有効になります。

LDAP 認証を使用してログインする

LDAP 認証を使用してログインするには、282 ページの「組織認証モジュール」のコア認証モジュール属性を修正し、LDAP 認証を有効にして選択する必要があります。これにより、ユーザーが `http://` ホスト名 : ポート `/server_` 配備 `_URI/UI/Login?module=LDAP` を使用してログインするときに、LDAP 認証のログインウィンドウが表示されます。使用している認証タイプ (サービス、ロール、ユーザー、組織など) によっては、認証モジュールをデフォルトとして設定する場合に、URL でモジュール名を指定する必要がありません。

LDAP 認証のフェイルオーバーを有効にする

LDAP 認証属性には、プライマリとセカンダリ両方の Directory Server の値フィールドがあります。Access Manager では、プライマリサーバーが利用できなくなると、認証を行うためにセカンダリサーバーを使用します。詳細は、300 ページの「プライマリ LDAP サーバー」および300 ページの「セカンダリ LDAP サーバー」の LDAP 属性を参照してください。

複数の LDAP 設定

フェイルオーバーの形式として、あるいは、Access Manager コンソールで値フィールドが 1 つだけ提供されている場合に、属性に複数の値を設定するために、管理者は 1 つの組織に複数の LDAP 設定を定義できます。これら追加の設定はコンソールに表示されませんが、要求を行っているユーザーの承認が初期検索で見つからない場合に、主設定とともに機能します。複数の LDAP 設定については、『Access Manager Developer's Guide』の「Multi LDAP Configuration」を参照してください。

メンバーシップ認証

メンバーシップ認証は、my.site.com または mysun.sun.com のように、パーソナライズされたサイトのように実装されます。モジュールが有効なときに、ユーザーは管理者の支援なしでアカウントを作成し、パーソナライズします。ユーザーは作成したアカウントを使用し、追加済みユーザーとしてアクセスできます。また、ユーザーはビューアのインタフェースにアクセスできます。ビューアのインタフェースは、認証データおよびユーザー設定として、ユーザープロファイルデータベースに保存されています。

メンバーシップ認証を追加し、有効にする

組織管理者または最上位レベル管理者として、Access Manager にログインする必要があります。

1. メンバーシップ認証を追加する組織に移動します。
2. 「表示」メニューから「サービス」を選択します。

   コアモジュールが追加済みの場合は、ナビゲーション区画に表示されます。追加済みでない場合は、メンバーシップ認証モジュールとともに追加できます。
3. ナビゲーション区画で「追加」をクリックします。

   利用可能なモジュールのリストがデータ区画に表示されます。
4. 「メンバーシップ認証」のチェックボックスを選択し、「追加」をクリックします。

   メンバーシップ認証モジュールがナビゲーション区画に表示され、追加されたことが管理者に示されます。
5. 「メンバーシップ」認証のプロパティの矢印をクリックします。

   「現在このサービスにはテンプレートが存在しません。新規に作成しますか?」というメッセージがデータ区画に表示されます。
6. 「作成」をクリックします。

   メンバーシップ認証属性がデータ区画に表示されます。必要に応じて属性を修正します。これらの属性の説明については、第 24 章「メンバーシップ認証属性」を参照するか、またはコンソール右上の「ヘルプ」リンクをクリックしてください。
7. パスワードを「root ユーザーバインドパスワード」に入力します。デフォルトで、インストール中に入力した amldapuser パスワードが、バインドユーザーとして使用されます。

   別のバインドユーザーを使用するには、「root ユーザーバインド DN」属性でユーザーの DN を変更し、そのユーザーのパスワードを「root ユーザーバインドパスワード」属性に入力します。

8. 「保存」をクリックします。

   メンバーシップ認証モジュールが有効になります。

メンバーシップ認証を使用してログインする

メンバーシップ認証を使用してログインするには、282 ページの「組織認証モジュール」のコア認証モジュール属性を修正し、メンバーシップ認証を有効にして選択する必要があります。これにより、ユーザーが `http://` ホスト名 : ポート / 配備 _URI/`UI/Login?module=Membership` を使用してログインするときに (大文字と小文字の区別に注意)、メンバーシップ認証 (自己登録) のログインウィンドウが表示されます。使用している認証タイプ (モジュール、ロール、ユーザー、組織など) によっては、認証モジュールをデフォルトとして設定する場合に、URL でモジュール名を指定する必要がありません。

MSISDN 認証

MSISDN (Mobile Station Integrated Services Digital Network) 認証モジュールでは、携帯電話などのデバイスに関連するモバイル加入者 ISDN を使用して認証できます。これは対話型モジュールではありません。このモジュールでは加入者 ISDN が取得されて Directory Server で確認され、番号が一致するユーザーが検索されます。

MSISDN 認証を追加し、有効にする

組織管理者または最上位レベル管理者として、Identity Server にログインする必要があります。

1. MSISDN 認証を追加する組織に移動します。
2. 「表示」メニューから「サービス」を選択します。

   コアモジュールが追加済みの場合は、ナビゲーション区画に表示されます。追加済みでない場合は、MSISDN 認証モジュールとともに追加できます。
3. ナビゲーション区画で「追加」をクリックします。

   利用可能なモジュールのリストがデータ区画に表示されます。
4. 「MSISDN 認証」のチェックボックスを選択し、「追加」をクリックします。

   MSISDN 認証モジュールがナビゲーション区画に表示され、追加されたことが管理者に示されます。

5. 「MSISDN」認証のプロパティの矢印をクリックします。

   「現在このサービスにはテンプレートが存在しません。新規に作成しますか?」というメッセージがデータ区画に表示されます。

6. 「作成」をクリックします。

   MSISDN 認証属性がデータ区画に表示されます。必要に応じて属性を修正します。

7. 「保存」をクリックします。

   MSISDN 認証モジュールが有効になります。

MSISDN 認証を使用してログインする

MSISDN 認証を使用してログインするには、282 ページの「組織認証モジュール」のコア認証モジュール属性を修正し、MSISDN 認証を有効にして選択する必要があります。これにより、ユーザーが `http://` ホスト名 : ポート / 配備 `_URI/UI/Login?module=MSISDN` (大文字と小文字を区別) を使用してログインするときに、MSISDN 認証のログインウィンドウが表示されます。使用している認証タイプ (サービス、ロール、ユーザー、組織など) によっては、認証モジュールをデフォルトとして設定する場合に、URL でモジュール名を指定する必要がありません。

Microsoft Windows NT 認証

Access Manager は、すでにインストールされている Microsoft Windows NT または Microsoft Windows 2000 サーバーで使用できるように設定できます。Access Manager では、NT 認証のクライアント部分を担当します。

1. NT サーバーを設定します。詳しい手順については、Microsoft Windows NT サーバーのマニュアルを参照してください。
2. Microsoft Windows NT 認証モジュールを追加し、有効にする前に、Samba クライアントを入手してインストールし、Solaris システム上の Access Manager と通信できるようにする必要があります。詳細は、315 ページの「Microsoft Windows NT 認証属性」を参照してください。
3. Microsoft Windows NT 認証モジュールの追加と有効化を行います。

Samba クライアントのインストール

Microsoft Windows NT 認証モジュールをアクティブにするには、Samba Client 2.2.2 をダウンロードして次のディレクトリにインストールする必要があります。

```
AccessManager-base/SUNWam/bin
```

Samba Client は、Microsoft Windows マシンと UNIX マシンを共存させるためのファイルサーバー兼プリントサーバーで、専用の Microsoft Windows NT/2000 Server を必要としません。詳細とダウンロードについては、http://wwws.sun.com/software/download/products/3e3af224.html を参照してください。

Red Hat Linux とともに出荷される Samba クライアントは、次のディレクトリに置かれています。

```
/usr/bin
```

Linux 用 Microsoft Windows NT 認証モジュールを使って認証を行うためには、クライアントのバイナリを Access Manager の次のディレクトリにコピーします。

```
AccessManager-base/sun/identity/bin
```

注 複数のインタフェースがある場合には、追加の設定が必要です。複数のインタフェースは `smb.conf` ファイルで設定し、それを `mbclient` へ伝えることにより、設定できます。

Microsoft Windows NT 認証を追加し、有効にする

組織管理者または最上位レベル管理者として、Access Manager にログインする必要があります。

1. Microsoft Windows NT 認証を追加する組織に移動します。
2. 「表示」メニューから「サービス」を選択します。

   コアモジュールが追加済みの場合は、ナビゲーション区画に表示されます。追加済みでない場合は、Microsoft Windows NT 認証モジュールとともに追加できます。
3. ナビゲーション区画で「追加」をクリックします。

   利用可能なモジュールのリストがデータ区画に表示されます。
4. 「Microsoft Windows NT 認証」のチェックボックスを選択し、「追加」をクリックします。

Microsoft Windows NT 認証モジュールがナビゲーション区画に表示され、追加されたことが管理者に示されます。

5. 「Microsoft Windows NT」認証のプロパティの矢印をクリックします。

   「現在このサービスにはテンプレートが存在しません。新規に作成しますか ?」というメッセージがデータ区画に表示されます。

6. 「作成」をクリックします。

   Microsoft Windows NT 認証属性がデータ区画に表示されます。必要に応じて属性を修正します。これらの属性の説明については、第 26 章「Microsoft Windows NT 認証属性」を参照するか、またはコンソール右上の「ヘルプ」リンクをクリックしてください。

7. 「保存」をクリックします。

   Microsoft Windows NT 認証モジュールが有効になります。

Microsoft Windows NT 認証を使用してログインする

Microsoft Windows NT 認証を使用してログインするには、282 ページの「組織認証モジュール」のコア認証モジュール属性を修正し、Microsoft Windows NT 認証を有効にして選択する必要があります。これにより、ユーザーが `http://` ホスト名 : ポート / 配備 _URI/`UI/Login?module=NT` を使用してログインするときに、Microsoft Windows NT 認証のログインウィンドウが表示されます。使用している認証タイプ (サービス、ロール、ユーザー、組織など) によっては、認証モジュールをデフォルトとして設定する場合に、URL でモジュール名を指定する必要がありません。

RADIUS サーバー認証

Access Manager は、すでにインストールされている RADIUS サーバーで使用できるように設定できます。エンタープライズで認証のためにレガシーの RADIUS サーバーを使用している場合に便利です。RADIUS 認証モジュールを有効にするには、次の 2 つの手順を行います。

1. RADIUS サーバーを設定します。

   詳しい手順については、RADIUS サーバーのマニュアルを参照してください。

2. RADIUS 認証モジュールを登録し、有効にします。

RADIUS 認証を追加し、有効にする

組織管理者として、Access Manager にログインする必要があります。

1. RADIUS 認証を追加する組織に移動します。
2. 「表示」メニューから「サービス」を選択します。

   コアモジュールが追加済みの場合は、ナビゲーション区画に表示されます。追加済みでない場合は、RADIUS 認証モジュールとともに追加できます。
3. ナビゲーション区画で「追加」をクリックします。

   利用可能なモジュールのリストがデータ区画に表示されます。
4. 「RADIUS 認証」のチェックボックスを選択し、「追加」をクリックします。

   RADIUS 認証モジュールがナビゲーション区画に表示され、追加されたことが管理者に示されます。
5. 「RADIUS」認証のプロパティの矢印をクリックします。

   「現在このサービスにはテンプレートが存在しません。新規に作成しますか ?」というメッセージがデータ区画に表示されます。
6. 「作成」をクリックします。

   RADIUS 認証属性がデータ区画に表示されます。必要に応じて属性を修正します。これらの属性の説明については、第 27 章「RADIUS 認証属性」を参照するか、またはコンソール右上の「ヘルプ」リンクをクリックしてください。
7. 「保存」をクリックします。

   RADIUS 認証モジュールが有効になります。

RADIUS 認証を使用してログインする

RADIUS 認証を使用してログインするには、282 ページの「組織認証モジュール」のコア認証モジュール属性を修正し、RADIUS 認証を有効にして選択する必要があります。これにより、ユーザーが `http://` ホスト名 : ポート / 配備 `_URI/UI/Login?module=RADIUS` を使用してログインするときに、RADIUS 認証のログインウィンドウが表示されます。使用している認証タイプ (サービス、ロール、ユーザー、組織など) によっては、認証モジュールをデフォルトとして設定する場合に、URL でモジュール名を指定する必要がありません。

Sun ONE Application Server で RADUIS を設定する

RADUIS クライアントがそのサーバーに対してソケット接続を作成するとき、デフォルトでは、Application Server の `server.policy` ファイルで SocketPermission の connect アクセス権だけが与えられています。RADUIS 認証を正常に機能させるには、次のアクションを許可する必要があります。

- accept (受け入れ)
- connect (接続)
- listen (待機)
- resolve (解決)

ソケット接続のアクセス権を与えるには、Application Server の `server.policy` ファイルにエントリを追加します。SocketPermission は、ホストの指定と、そのホストへの接続方法を指定する一連のアクションとで構成されます。ホストは次のように指定されます。

`host = hostname | IPaddress:portrange:portrange = portnumber | -portnumberportnumber-portnumber`

ホストは、DNS 名または IP アドレスの数値で表されるか、ローカルマシンの場合は localhost と表されます。DNS 名でホストを指定する場合は、ワイルドカード "*" を 1 つだけ使用できます。ワイルドカードを使用する場合は、`*.example.com` のように、左端に置く必要があります。

ポート (portrange) は省略可能です。`N-` という形式のポート指定は、`N` またはそれ以上の番号を持つすべてのポートを表します。ここで、`N` はポート番号です。`-N` という形式のポート指定は、`N` またはそれ以下の番号を持つすべてのポートを表します。

`listen` アクションは、ローカルホストで使用される場合のみ有効です。`resolve` (ホスト /IP 解決のネームサービスルックアップ) アクションは、ほかのアクションで暗黙的に使用されます。

たとえば、SocketPermission を作成するときに次のアクセス権をコードに与えると、そのコードは `machine1.example.com` のポート `1645` に接続することと、そのポート上で接続を受け入れることができます。

```
permission java.net.SocketPermission machine1.example.com:1645,
"connect,accept";
```

同様に、次のアクセス権を与えられたコードは、ローカルホストのポート 1024 ～ 65535 に接続することと、これらのポートで接続受け入れおよび待機を行うことができます。

```
permission java.net.SocketPermission "machine1.example.com:1645",
"connect,accept";

permission java.net.SocketPermission "localhost:1024-",
"accept,connect,listen";
```

注	リモートホストに対する接続受け入れや接続作成のアクセス権をコードに与えると、悪意のあるコードによって、本来アクセス権を持たない第三者に機密データが転送されたり共有されたりしやすくなるので、問題が発生することがあります。適切なアクセス権だけを与えるために、ポート番号を範囲で指定するのではなく、正確なポート番号を指定してください。

SafeWord 認証

Secure Computing の SafeWord™ または SafeWord PremierAccess™ 認証サーバーに対して送られる SafeWord 認証要求を処理するように、Access Manager を設定できます。Access Manager は、SafeWord 認証のクライアント部分を担当します。SafeWord サーバーは、Access Manager のインストールされているシステムにも、別のシステムにも置くことができます。

SafeWord 認証を追加し、有効にする

組織管理者または最上位レベル管理者として、Access Manager にログインする必要があります。

1. SafeWord 認証を追加する組織に移動します。
2. 「表示」メニューから「サービス」を選択します。

   コアモジュールが追加済みの場合は、ナビゲーション区画に表示されます。追加済みでない場合は、SafeWord 認証モジュールとともに追加できます。
3. ナビゲーション区画で「追加」をクリックします。

   利用可能なモジュールのリストがデータ区画に表示されます。
4. 「SafeWord 認証」のチェックボックスを選択し、「追加」をクリックします。

   SafeWord 認証モジュールがナビゲーション区画に表示され、追加されたことが管理者に示されます。
5. 「SafeWord」認証のプロパティの矢印をクリックします。

   「現在このサービスにはテンプレートが存在しません。新規に作成しますか ?」というメッセージがデータ区画に表示されます。
6. 「作成」をクリックします。

   SafeWord 認証属性がデータ区画に表示されます。必要に応じて属性を修正します。これらの属性の説明については、第 28 章「SafeWord 認証属性」を参照するか、またはコンソール右上の「ヘルプ」リンクをクリックしてください。

7. 「保存」をクリックします。

   SafeWord 認証モジュールが有効になります。

SafeWord 認証を使用してログインする

SafeWord 認証を使用してログインするには、282 ページの「組織認証モジュール」のコア認証モジュール属性を修正し、SafeWord 認証を有効にして選択する必要があります。これにより、ユーザーが `http://` ホスト名 `:` ポート `/` 配備 `_URI/UI/Login?module=SafeWord` を使用してログインするときに、SafeWord 認証のログインウィンドウが表示されます。使用している認証タイプ (ロール、ユーザー、組織など) によっては、認証モジュールをデフォルトとして設定する場合に、URL でモジュール名を指定する必要がありません。

Sun ONE Application Server で SafeWord を設定する

SafeWord クライアントがそのサーバーに対してソケット接続を作成するとき、デフォルトでは、Application Server の `server.policy` ファイルで SocketPermission の `connect` アクセス権だけが与えられています。SafeWord 認証を正常に機能させるには、次のアクションを許可する必要があります。

- accept (受け入れ)
- connect (接続)
- listen (待機)
- resolve (解決)

ソケット接続のアクセス権を与えるには、Application Server の `server.policy` ファイルにエントリを追加します。SocketPermission は、ホストの指定と、そのホストへの接続方法を指定する一連のアクションとで構成されます。ホストは次のように指定されます。

```
host = (hostname | IPaddress)[:portrange] portrange = portnumber |
-portnumberportnumber-[portnumber]
```

ホストは、DNS 名または IP アドレスの数値で表されるか、ローカルマシンの場合は localhost と表されます。DNS 名でホストを指定する場合は、ワイルドカード "*" を 1 つだけ使用できます。ワイルドカードを使用する場合は、`*.example.com` のように、左端に置く必要があります。

ポート (portrange) は省略可能です。N- という形式のポート指定は、N またはそれ以上の番号を持つすべてのポートを表します。ここで、N はポート番号です。-N という形式のポート指定は、N またはそれ以下の番号を持つすべてのポートを表します。

listen アクションは、ローカルホストで使用される場合のみ有効です。resolve (ホスト /IP 解決のネームサービスルックアップ) アクションは、ほかのアクションで暗黙的に使用されます。

たとえば、SocketPermission を作成するときに次のアクセス権をコードに与えると、そのコードは machine1.example.com のポート 1645 に接続することと、そのポート上で接続を受け入れることができます。

```
permission java.net.SocketPermission machine1.example.com:5030,
"connect,accept";
```

同様に、次のアクセス権を与えられたコードは、ローカルホストのポート 1024 ～ 65535 に接続することと、これらのポートで接続受け入れおよび待機を行うことができます。

```
permission java.net.SocketPermission "machine1.example.com:5030",
"connect,accept";

permission java.net.SocketPermission "localhost:1024-",
"accept,connect,listen";
```

注 リモートホストに対する接続受け入れや接続作成のアクセス権をコードに与えると、悪意のあるコードによって、本来アクセス権を持たない第三者に機密データが転送されたり共有されたりしやすくなるので、問題が発生することがあります。適切なアクセス権だけを与えるために、ポート番号を範囲で指定するのではなく、正確なポート番号を指定してください。

SAML 認証

SAML (Security Assertion Markup Language) 認証モジュールでは、ターゲットサーバーで SAML アサーションの受信と確認が行われます。このモジュールが、Access Manager 2004Q2 から Access Manager 2005Q1 などのアップグレード後も含めて、ターゲットマシンで設定されている場合にかぎって、SAML SSO は動作します。

SAML 認証を追加し、有効にする

組織管理者または最上位レベル管理者として Access Manager にログインし、LDAP 認証モジュールを前もって登録しておきます。

1. SAML 認証を追加する組織に移動します。
2. 「表示」メニューから「サービス」を選択します。

   コアモジュールが追加済みの場合は、ナビゲーション区画に表示されます。追加済みでない場合は、SAML 認証モジュールとともに追加できます。
3. ナビゲーション区画で「追加」をクリックします。

   利用可能なモジュールのリストがデータ区画に表示されます。
4. 「SAML 認証」のチェックボックスを選択し、「追加」をクリックします。

   SAML 認証モジュールがナビゲーション区画に表示され、追加されたことが管理者に示されます。
5. 「SAML」認証のプロパティの矢印をクリックします。

   「現在このサービスにはテンプレートが存在しません。新規に作成しますか ?」というメッセージがデータ区画に表示されます。
6. 「作成」をクリックします。

   SAML 認証属性がデータ区画に表示されます。必要に応じて属性を修正します。これらの属性の説明については、第 21 章「HTTP 基本認証属性」を参照するか、またはコンソール右上の「ヘルプ」リンクをクリックしてください。
7. 「保存」をクリックします。

   SAML 認証モジュールが有効になります。

SAML 認証を使用してログインする

SAML 認証を使用してログインするには、282 ページの「組織認証モジュール」のコア認証モジュール属性を修正し、HTTP 基本認証を有効にして選択する必要があります。これにより、ユーザーが `http://` ホスト名 `:` ポート `/` サーバー `_` 配備 `_URI/UI/Login?module=SAML` を使用してログインするときに、認証のログインウィンドウが表示されます。使用している認証タイプ (サービス、ロール、ユーザー、組織など) によっては、認証モジュールをデフォルトとして設定する場合に、URL でモジュール名を指定する必要がありません。

SecurID 認証

RSA の ACE/Server 認証サーバーに対して送られる SecureID 認証要求を処理するように、Access Manager を設定できます。Access Manager では、SecurID 認証のクライアント部分を担当します。ACE/Server は、Access Manager のインストールされているシステムにも、別のシステムにも置くことができます。ローカルで管理されたユーザー ID を認証する (`admintool` (1M) 参照) には、ルートでアクセスする必要があります。

SecurID 認証では、認証ヘルパー `amsecuridd` を使用します。認証ヘルパーは Access Manager のメインのプロセスとは独立したプロセスです。起動すると、このヘルパーは設定情報を得るためにポートで待機します。Access Manager が `nobody` として、または root 以外のユーザー ID で実行するようにインストールされている場合でも、*`AccessManager-base`*`/SUNWam/share/bin/amsecuridd` プロセスは root ユーザーとして実行する必要があります。`amsecuridd` ヘルパーについての詳細は、237 ページの「amsecuridd ヘルパー」を参照してください。

注　このリリースの Access Manager の場合、Linux プラットフォームと Solaris x86 プラットフォームでは SecurID 認証モジュールを使用できません。この 2 つのプラットフォームでは、SecurID 認証モジュールの登録、設定、有効化を行わないでください。SecurID 認証モジュールは、SPARC のみで使用できます。

SecurID 認証を追加し、有効にする

組織管理者または最上位レベル管理者として、Access Manager にログインする必要があります。

1. SecurID 認証を追加する組織に移動します。
2. 「表示」メニューから「サービス」を選択します。

   コアモジュールが追加済みの場合は、ナビゲーション区画に表示されます。追加済みでない場合は、SecurID 認証モジュールとともに追加できます。
3. ナビゲーション区画で「追加」をクリックします。

   利用可能なモジュールのリストがデータ区画に表示されます。
4. 「SecurID 認証」のチェックボックスを選択し、「追加」をクリックします。

   SecurID 認証モジュールがナビゲーション区画に表示され、追加されたことが管理者に示されます。
5. 「SecurID」認証のプロパティの矢印をクリックします。

   「現在このサービスにはテンプレートが存在しません。新規に作成しますか ?」というメッセージがデータ区画に表示されます。
6. 「作成」をクリックします。

   SecurID 認証属性がデータ区画に表示されます。必要に応じて属性を修正します。これらの属性の説明については、第 30 章「SecurID 認証属性」を参照するか、またはコンソール右上の「ヘルプ」リンクをクリックしてください。
7. 「保存」をクリックします。

   SecurID 認証モジュールが有効になります。

SecurID 認証を使用してログインする

SecurID 認証を使用してログインするには、282 ページの「組織認証モジュール」のコア認証モジュール属性を修正し、SecurID 認証を有効にして選択する必要があります。これにより、ユーザーが `http://` ホスト名 `:` ポート / 配備 `_URI/UI/Login?module=SecurID` を使用してログインするときに、SecurID 認証のログインウィンドウが表示されます。使用している認証タイプ (ロール、ユーザー、組織など) によっては、認証モジュールをデフォルトとして設定する場合に、URL でモジュール名を指定する必要がありません。

UNIX 認証

Access Manager がインストールされている Solaris または Linux システムで既知の UNIX ユーザー ID およびパスワードに対する認証要求を処理するように、Access Manager を設定できます。UNIX 認証では組織属性は 1 つだけしかなく、またグローバル属性は少ししかありませんが、システムの観点から検討すべき点があります。ローカルで管理されたユーザー ID を認証する (admintool (1M) 参照) には、ルートでアクセスする必要があります。

UNIX 認証では、認証ヘルパー amunixd を使用します。認証ヘルパーは Access Manager のメインのプロセスとは独立したプロセスです。起動すると、このヘルパーは設定情報を得るためにポートで待機します。UNIX ヘルパーは Access Manager ごとに 1 つだけあり、そのすべての組織で共用されます。

Access Manager が nobody として、または root 以外のユーザー ID で実行するようにインストールされている場合でも、*AccessManager-base*/SUNWam/share/bin/amunixd プロセスは root ユーザーとして実行する必要があります。UNIX 認証モジュールは、UNIX 認証要求を待機するために localhost:58946 へのソケットを開くことで、amunixd デーモンを呼び出します。デフォルトのポートで amunixd ヘルパーを実行するには、次のコマンドを入力します。

```
./amunixd
```

デフォルト以外のポートで amunixd ヘルパーを実行するには、次のコマンドを入力します。

```
./amunixd [-c ポート番号 ] [IP アドレス ]
```

IP アドレスとポート番号は、AMConfig.properties 内の UnixHelper.ipadrs 属性 (IPV4 形式) と UnixHelper.port 属性で指定されています。amunixd を amserver コマンド行ユーティリティーから実行することもできます。amserver は AMConfig.properties からポート番号と IP アドレスを取り出し、このプロセスを自動的に実行します。

/etc/nsswitch.conf ファイル内の passwd エントリでは、/etc/passwd および /etc/shadow ファイル、または NIS を認証で探すかどうかを指定します。

UNIX 認証を追加し、有効にする

以下の手順では、最上位レベル管理者として、Access Manager にログインする必要があります。

1. サービス設定モジュールを選択します。
2. 「サービス名」リストで、「UNIX」認証のプロパティの矢印をクリックします。

   グローバル属性および組織属性が表示されます。1 つの UNIX ヘルパーで Access Manager サーバーの組織すべてにサービスを提供するので、UNIX 属性のほとんどはグローバルです。これらの属性の説明については、第 31 章「UNIX 認証属性」を参照するか、またはコンソール右上の「ヘルプ」リンクをクリックしてください。
3. 「保存」をクリックして、属性の新しい値を保存します。

   組織管理者として Access Manager にログインすると、組織に対する UNIX 認証を有効にできます。
4. UNIX 認証を追加する組織に移動します。
5. 「表示」メニューから「サービス」を選択します。

   コアモジュールが追加済みの場合は、ナビゲーション区画に表示されます。追加済みでない場合は、UNIX 認証モジュールとともに追加できます。
6. ナビゲーション区画で「追加」をクリックします。

   利用可能なモジュールのリストがデータ区画に表示されます。
7. 「UNIX 認証」のチェックボックスを選択し、「追加」をクリックします。

   UNIX 認証モジュールがナビゲーション区画に表示され、追加されたことが管理者に示されます。
8. 「UNIX」認証のプロパティの矢印をクリックします。

   「現在このサービスにはテンプレートが存在しません。新規に作成しますか ?」というメッセージがデータ区画に表示されます。
9. 「作成」をクリックします。

   UNIX 認証の組織属性がデータ区画に表示されます。必要に応じて認証レベル属性を修正します。この属性の説明については、第 31 章「UNIX 認証属性」を参照するか、またはコンソール右上の「ヘルプ」リンクをクリックしてください。
10. 「保存」をクリックします。UNIX 認証モジュールが有効になります。

UNIX 認証を使用してログインする

UNIX 認証を使用してログインするには、282 ページの「組織認証モジュール」のコア認証モジュールを修正し、UNIX 認証を有効にして選択するように修正する必要があります。そうすると、ユーザーが `http://` ホスト名 `:` ポート / 配備 `_URI/UI/Login?module=Unix` を使用してログインするときに、UNIX 認証のログインウィンドウが表示されます。使用している認証タイプ (サービス、ロール、ユーザー、組織など) によっては、認証モジュールをデフォルトとして設定する場合に、URL でモジュール名を指定する必要がありません。

Microsoft Windows デスクトップ SSO 認証

Microsoft Windows デスクトップ SSO 認証モジュールは、Microsoft Windows 2000™ に使用する、Kerberos ベースのプラグインモジュールです。このサービスを使用すると、Kerberos Distribution Center (KDC) で認証されたユーザーは、ログインの条件を再度提示しなくても Identity Server に認証されます。

ユーザーは、SPNEGO (Simple and Protected GSS-API Negotiation Mechanism) プロトコルで Access Manager に Kerberos トークンを送信します。この認証モジュールで Access Manager への Kerberos ベースのシングルサインオンを実行するには、ユーザーが、クライアントサイドにおいて、SPNEGO プロトコルをサポートして自分自身を認証する必要があります。一般的に、このプロトコルをサポートするすべてのユーザーは、このモジュールを使用して Access Manager に認証できます。クライアントサイドでトークンを使用できるかどうかにより、SPENGO トークンか Kerberos トークン (どちらの場合でもプロトコルは同一) がこのモジュールによって提供されます。Microsoft Windows 2000 以上で動作している Microsoft Internet Explorer 5.01 以上では、このプロトコルがサポートされています。Solaris 9 および 10 の Mozilla 1.4 では SPNEGO がサポートされますが、Solaris では SPNEGO がサポートされていないので、返されるトークンは KERBEROS トークンのみです。

注 Kerberos V5 認証モジュールの新機能を使用するには、JDK 1.4 以上を使用する必要があります。この SPNEGO モジュールで Kerberos ベースの SSO を実行するには、Java GSS API を使用する必要があります。

Internet Explorer の既知の制限事項

WindowsDesktopSSO 認証に Microsoft Internet Explorer 6.x を使用しており、WindowsDesktopSSO モジュールで設定されている KDC レルムと一致する、ユーザーの Kerberos/SPNEGO トークンにこのブラウザでアクセスできない場合、WindowsDesktopSSO モジュールへの認証がエラーになったあとで、このブラウザは

その他のモジュールに対して不正に動作します。この問題の直接的な原因は、Internet Explorer が WindowsDesktopSSO モジュールでエラーになると、ブラウザを再起動するまで、コールバックを要求されても、別のモジュールのコールバックを Access Manager に渡すことができなくなることです。このため、WindowsDesktopSSO のあとのすべてのモジュールは、ユーザー資格が NULL であるためにエラーとなります。

関連情報については、次の資料を参照してください。

http://support.microsoft.com/default.aspx?scid=kb;en-us;308074

http://www.wedgetail.com/jcsi/sso/doc/guide/troubleshooting.html#ieNTLM

Microsoft Windows デスクトップ SSO 認証を追加し、有効にする

Microsoft Windows デスクトップ SSO 認証サービスを有効にするには、次の 3 つの手順を実行します。

1. Microsoft Windows 2000 のドメインコントローラにユーザーを作成します。
2. Internet Explorer をセットアップします。
3. Microsoft Windows デスクトップ SSO 認証モジュールを追加し、設定します。

Microsoft Windows 2000 のドメインコントローラにユーザーを作成する

1. ドメインコントローラで、Access Manager 認証モジュール用のユーザーアカウントを作成します。
   a. 「スタート」メニューから、「プログラム」>「管理ツール」に進みます。
   b. 「Active Directory ユーザーとコンピュータ」を選択します。
   c. Access Manager ホスト名がユーザー ID (ログイン名) の新しいユーザーを作成します。Access Manager ホスト名には、ドメイン名を含めないでください。
2. ユーザーアカウントをサービスプロバイダの名前と関連付け、Keytab ファイルを Access Manager がインストールされたシステムにエクスポートします。そのためには、次のコマンドを実行します。

   ```
   ktpass -princ host/hostname.domainname@DCDOMAIN -pass password
   -mapuser userName-out hostname.host.keytab
   ```

   ```
   ktpass -princ HTTP/hostname.domainname@DCDOMAIN -pass password
   -mapuser userName-out hostname.host.keytab
   ```

   ktpass コマンドには、次のパラメータを使用できます。

hostname: Access Manager が稼働する、ドメイン名なしのホスト名です。

domainname: Access Manager のドメイン名です。

DCDOMAIN: ドメインコントローラのドメイン名です。この名前は、Access Manager のドメイン名とは異なる場合があります。

password: ユーザーアカウントのパスワードです。`ktpass` はパスワードを検証しないので、パスワードが正しいことを確認します。

userName: ユーザーアカウント ID です。これはホスト名と同じにする必要があります。

注 両方の Keytab ファイルがセキュリティ保護されているようにします。

3. サーバーを再起動します。

Internet Explorer をセットアップする

この手順は、Microsoft Internet Explorer™ 6 以上に当てはまります。これよりも前のバージョンを使用している場合は、Access Manager がブラウザのインターネットゾーンにあり、ネイティブ Microsoft Windows 認証が有効であることを確認します。

1. 「ツール」メニューで、「インターネットオプション」>「詳細設定」>「セキュリティ」に進みます。
2. 「統合 Microsoft Windows 認証を使用する」オプションを選択します。
3. 「セキュリティ」>「イントラネット」に進みます。
   a. 「レベルのカスタマイズ」を選択します。「ユーザー認証」の「ログオン」で「イントラネットゾーンでのみ自動的にログオンする」オプションを選択します。
   b. 「サイト」に進み、すべてのオプションを選択します。
   c. 「詳細設定」をクリックして、Access Manager をローカルゾーンに追加します (まだ追加されていない場合)。

Internet Explorer の既知の制限事項

WindowsDesktopSSO 認証に Microsoft Internet Explorer 6.x を使用しており、WindowsDesktopSSO モジュールで設定されている KDC レルムと一致する、ユーザーの Kerberos/SPNEGO トークンにこのブラウザでアクセスできない場合、WindowsDesktopSSO モジュールへの認証がエラーになったあとで、このブラウザはその他のモジュールに対して不正に動作します。この問題の直接的な原因は、Internet

Explorer が WindowsDesktopSSO モジュールでエラーになると、ブラウザを再起動するまで、コールバックを要求されても、別のモジュールのコールバックを Access Manager に渡すことができなくなることです。このため、WindowsDesktopSSO のあとのすべてのモジュールは、ユーザー資格が NULL であるためにエラーとなります。

関連情報については、次の資料を参照してください。

http://support.microsoft.com/default.aspx?scid=kb;en-us;308074

http://www.wedgetail.com/jcsi/sso/doc/guide/troubleshooting.html#ieNTLM

Microsoft Windows デスクトップ SSO 認証を追加し、設定する

組織管理者または最上位レベル管理者として、Access Manager にログインする必要があります。

1. Microsoft Windows デスクトップ SSO 認証を追加する組織に移動します。
2. 「表示」メニューから「サービス」を選択します。

   コアモジュールが追加済みの場合は、ナビゲーション区画に表示されます。追加済みでない場合は、Microsoft Windows デスクトップ SSO 認証モジュールとともに追加できます。
3. ナビゲーション区画で「追加」をクリックします。

   利用可能なモジュールのリストがデータ区画に表示されます。
4. 「Window デスクトップ SSO 認証」のチェックボックスを選択し、「追加」をクリックします。

   Microsoft Windows デスクトップ SSO 認証モジュールがナビゲーション区画に表示され、追加されたことが管理者に示されます。
5. 「Window デスクトップ SSO」認証のプロパティの矢印をクリックします。

   「現在このサービスにはテンプレートが存在しません。新規に作成しますか ?」というメッセージがデータ区画に表示されます。
6. 「作成」をクリックします。

   Microsoft Windows デスクトップ SSO 認証属性がデータ区画に表示されます。必要に応じて属性を修正します。これらの属性の説明については、第 32 章「Microsoft Windows デスクトップ SSO 認証属性」を参照するか、またはコンソール右上の「ヘルプ」リンクをクリックしてください。
7. 「保存」をクリックします。Microsoft Windows デスクトップ SSO 認証モジュールが有効になります。

Microsoft Windows デスクトップ SSO 認証を使用してログインする

Microsoft Windows デスクトップ SSO 認証を使用してログインするには、282 ページの「組織認証モジュール」のコア認証モジュール属性を修正し、Microsoft Windows デスクトップ SSO 認証を有効にして選択する必要があります。これで、Microsoft Windows 2000 のドメインコントローラの一部であるホストに `http://` ホスト名 : ポート / 配備 `_URI/UI/Login?module=WindowsDesktopSSO` を使ってすでにドメインユーザーとしてログインしているユーザーが、そのホストからログインしても正しく認証されます。使用している認証タイプ (サービス、ロール、ユーザー、組織など) によっては、認証モジュールをデフォルトとして設定する場合に、URL でモジュール名を指定する必要がありません。

第 8 章

パスワードリセットサービス

Sun Java™ System Access Manager 6 2005Q1 では、Access Manager によって保護されている特定のサービスやアプリケーションにアクセスするためのパスワードをユーザー自身がリセットできるように、パスワードリセットサービスが用意されています。パスワードリセットサービス属性は、最上位レベル管理者によって定義され、ユーザー検証のクレデンシャルを「秘密の質問」形式で制御し、新規または既存のパスワード通知のメカニズムを制御します。また、ユーザー検証が失敗した場合のロックアウト間隔も設定できます。

この章は、次の節で構成されています。

パスワードリセットサービスの登録

ユーザーの属している組織に対しては、パスワードリセットサービスを登録する必要はありません。ユーザーの属している組織にパスワードリセットサービスが存在しない場合は、このサービスについてサービス設定モジュールで定義されている値が継承されます。

別の組織のユーザーに対してパスワードリセットを登録する

1. アイデンティティ管理モジュールで、「組織」を選択し、サービスを登録する組織を選択します。

2. ナビゲーションフレームで「登録」をクリックします。

   利用可能なサービスのリストがデータフレームに表示されます。

3. 「パスワードリセット」のチェックボックスを選択し、「登録」をクリックします。

   パスワードリセットサービスがナビゲーションフレームに表示され、登録されたことが管理者に示されます。

パスワードリセットサービスの設定

パスワードリセットサービスの登録が完了したら、管理者権限を持っているユーザーがこのサービスを設定する必要があります。

サービスを設定する

1. パスワードリセットサービスが登録されている組織を選択します。
2. 「パスワードリセット」の矢印をクリックします。

   「このサービスに利用可能なテンプレートはありません。」というメッセージがデータフレームに表示されます。「作成」をクリックします。

3. パスワードリセット属性がデータフレームに表示され、ここでパスワードリセットサービスの要件を定義できます。パスワードリセットサービスが有効になっていることを確認します (デフォルトでは有効になっている)。少なくとも次の属性を定義する必要があります。

   - ユーザー検証
   - 秘密の質問
   - バインド DN
   - バインドパスワード

   「バインド DN」属性には、パスワードをリセットする権限を持っているユーザー (ヘルプデスク管理者など) を指定する必要があります。Directory Server の制限により、バインド DN が `cn=Directory Manager` の場合はパスワードリセットが機能しません。

   これら以外の属性は省略可能です。パスワードリセット属性の説明については、359 ページの「パスワードリセットサービス属性」を参照するか、またはコンソール右上の「ヘルプ」リンクをクリックしてください。

注 Access Manager では、ランダムなパスワードを生成するパスワードリセット Web アプリケーションが自動的にインストールされます。ただし、パスワードの生成や通知を行う独自のプラグインクラスを記述することもできます。このようなプラグインクラスのサンプルについては、次の場所にある Readme.html ファイルを参照してください。

PasswordGenerator:

```
AccessManager-base/SUNWam/samples/console/PasswordGenerator
```

NotifyPassword:

```
AccessManager-base/SUNWam/samples/console/NotifyPassword
```

4. ユーザー自身が独自の質問を定義できるようにするには、「個人的な質問を有効」属性を選択します。属性を定義し終えたら、「保存」をクリックします。

パスワードリセットのロックアウト

パスワードリセットサービスにはロックアウト機能があり、ユーザーが秘密の質問に回答できる回数を制限します。ロックアウト機能を設定するには、パスワードリセットサービス属性を使用します。これらの属性については、359 ページの「パスワードリセットサービス属性」を参照してください。パスワードリセットでは、メモリロックアウトと物理ロックアウトの 2 種類がサポートされています。

メモリロックアウト

これは一時的なロックアウトであり、「パスワードリセット失敗のロックアウト持続時間」属性の値が 0 より大きく、かつ、「パスワードリセット失敗のロックアウトを有効」属性が有効になっている場合にのみ機能します。これでロックアウトされたユーザーは、パスワードリセット Web アプリケーションを通してパスワードをリセットすることができなくなります。「パスワードリセット失敗のロックアウト持続時間」で指定された時間が経過するまで、あるいはサーバーが再起動されるまで、このロックアウトは持続します。

物理ロックアウト

これは、より永続的なロックアウトです。「パスワードリセット失敗のロックアウトカウント」属性の値が 0 に設定され、かつ、「パスワードリセット失敗のロックアウトを有効」属性が有効になっている場合に、ユーザーが秘密の質問に対して回答を誤ると、ユーザーのアカウント状態は無効に変更されます。

エンドユーザーから見たパスワードリセット

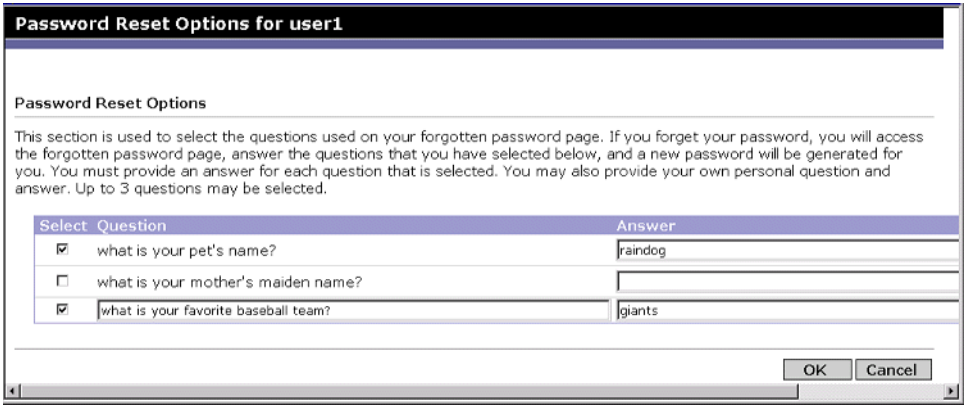
以降の節では、ユーザーの観点からパスワードリセットサービスについて説明します。

パスワードリセットのカスタマイズ

管理者がパスワードリセットサービスを有効にし、属性を定義したら、ユーザーは Access Manager コンソールにログインして秘密の質問をカスタマイズできます。次に例を示します。

1. ユーザーはユーザー名とパスワードを入力して認証を受け、Access Manager コンソールにログインします。
2. 「ユーザープロファイル」ページで、「パスワードリセットのオプション」を選択します。「質問と回答」画面が表示されます。
3. このサービスに対して管理者が定義した、利用可能な質問がユーザーに提示されます。次に例を示します。
   - ペットの名前
   - 好きなテレビ番組
   - 母親の旧姓
   - よく行くレストランの名前
4. 秘密の質問を選択します。管理者が組織に対して定義した最大数 (最大数はパスワードリセットサービスで定義される) まで選択できます。選択した質問に対して回答を指定します。これらの質問と回答は、ユーザーのパスワードをリセットするための基準になります (次の節を参照)。管理者が「個人的な質問を有効」属性を選択した場合は、ユーザーが独自の秘密の質問と回答を指定できるように、テキストフィールドが表示されます。

図 8-1　個人的な質問を有効にした場合の「質問と回答」の画面



5. 「保存」をクリックします。

パスワードを忘れた場合のリセット

ユーザーがパスワードを忘れた場合、Access Manager はパスワードリセット Web アプリケーションを使って新しいパスワードをランダムに生成し、それをユーザーに通知します。パスワードを忘れた場合の典型的なシナリオを次に示します。

1. ユーザーは管理者から与えられた URL を使って、パスワードリセット Web アプリケーションにログインします。次に例を示します。

   `http://` ホスト名 : ポート `/ampassword` (デフォルトの組織の場合)

   または

   `http://` ホスト名 : ポート / 配備 `_uri/UI/PWResetUserValidation?org=`*orgname*。ここで、*orgname* は組織の名前です。

注　パスワードリセットサービスがサブ組織で有効になっていても、親組織で無効になっている場合は、次の構文を使ってサービスにアクセスする必要があります。

`http://` ホスト名 : ポート / 配備 `_uri/UI/PWResetUserValidation?org=`*orgname*

2. ユーザー ID を入力します。

3. パスワードリセットサービスで定義されている質問のうち、ユーザーがカスタマイズで選択したものが提示されます。あらかじめ「ユーザープロファイル」ページにログインして質問をカスタマイズしておかないと、パスワードは生成されません。

   質問に対して正しく回答すると、新しいパスワードが生成され、電子メールで通知されます。回答が正しいかどうかにかかわらず、パスワードリセットが試みられたという通知が送信されます。新しいパスワードやパスワードリセット試行通知を受け取るには、「ユーザープロファイル」ページで電子メールアドレスを入力しておく必要があります。

パスワードポリシー

次のような条件を適用した安全なパスワードポリシーによって、推測されやすいパスワードに関連するリスクを最小限に抑えることができます。

- ユーザーはスケジュールに従ってパスワードを変更しなければならない。
- ユーザーは、自明ではないパスワードを指定しなければならない。
- パスワードを一定回数誤ると、アカウントがロックされることがある。

Directory Server では、いくつかの方法により、ツリー内の任意のノードでパスワードポリシーを設定できます。詳細は、次の Directory Server のマニュアルを参照してください。

`http://docs.sun.com/source/816-6700-10/aci.html#14773`

`http://docs.sun.com/source/816-6698-10/useracct.html#14386`

第 III 部

コマンド行リファレンスガイド

この「コマンド行リファレンスガイド」は、『Sun Java™ System Access Manager 6 2005Q1 管理ガイド』の第 3 部です。次の章で構成されています。

この節で説明するすべてのコマンド行ツールは、デフォルトでは次の場所にあります。

```
AccessManager-base/SUNWam/bin (Solaris)

AccessManager-base/identity/bin (Linux)
```

第 9 章

amadmin コマンド行ツール

この章では、amadmin コマンド行ツールについて説明します。この章は、次の節で構成されています。

amadmin コマンド行実行可能ファイル

コマンド行実行可能ファイル amadmin の第一の目的は、XML サービスファイルを Directory Server にロードして、DIT で管理タスクのバッチ処理を実行することです。amadmin は、AccessManager-base/SUNWam/bin にあり、次の目的に使用します。

- XML サービスファイルのロード - 管理者は sms.dtd で定義された XML サービスファイル形式を使用するサービスを Access Manager にロードします。すべてのサービスは amadmin を使用してロードする必要がありますが、Access Manager コンソールでインポートすることはできません。

注 XML サービスファイルは、Access Manager で参照される XML データの静的なかたまりとして Directory Server に格納されます。この情報は、LDAP だけを使用できる Directory Server では使用されません。

- DIT に対するアイデンティティオブジェクトのバッチ更新の実行 - 管理者は amadmin.dtd に定義されたバッチ処理用 XML ファイル形式を使用して、Directory Server DIT に対するバッチ更新を実行できます。たとえば管理者が組織を 10、ユーザーを 1000、およびグループを 100 作成する場合、この要求を 1 つ以上のバッチ処理用 XML ファイルに置いて、amadmin でロードすることで、1 回で作成できます。詳細については、『Access Manager Developer's Guide』の「Service Management」の章を参照してください。

注 amadmin は、Access Manager コンソールの機能の一部だけをサポートしており、コンソールの代わりに使用することは想定していません。比較的小規模な管理作業にはコンソールを使用し、比較的大規模な管理作業には amadmin を使用することをお勧めします。

amadmin の構文

amadmin を使用するために従わなくてはならない構造的な規則があります。amadmin ツールの一般的な構文は次のとおりです。

- amadmin -u | --runasdn DN 名 -w | --password パスワード [-l | --locale ロケール名] [[-v | --verbose] | [-d |--debug]] -t | --data XML ファイル 1 [XML ファイル 2 ...]
- amadmin -u | --runasdn DN 名 -w | --password パスワード [-l | --locale ロケール名] [[-v | --verbose] | [-d | --debug]] -s | --schema XML ファイル 1 [XML ファイル 2 ...]
- amadmin -u | --runasdn DN 名 -w | --password パスワード [-l | --locale ロケール名] [[-v | --verbose] | [-d | --debug]] -r | --deleteService サービス名 1 [サービス名 2 ...]
- amadmin -u | --runasdn DN 名 -w | --password パスワードまたは -f | --password file パスワードファイル [-c | --continue] [-l | --locale ロケール名] [[-v | --verbose] | [-d | --debug]] -m | --session サーバー名パターン
- amadmin -h | --help
- amadmin -n | --version
- amadmin -u | --runasdn DN 名 -w | --password パスワードまたは - f |--passwordfile パスワードファイル [-l | --locale ロケール名] [[-v | --verbose] | [-d] |--debug]] -a |--addAttributes サービス名 スキーマタイプ XML ファイル [XML ファイル 2] ...

注 2 連続するハイフンは、構文に示すとおりに入力する必要があります。

amadmin のオプション

次に、amadmin コマンド行パラメータオプションの定義を説明します。

--runasdn (-u)

--runasdn は、LDAP サーバーに対してユーザーを認証します。引数は、amadmin を実行できるように承認されたユーザーの識別名 (DN) の値です。たとえば次のようになります。

--runasdn uid=amAdmin,ou=People,o=iplanet.com,o=isp

DN は、ドメイン要素間にスペースを挿入し、DN 全体を二重引用符で囲むこともできます。たとえば次のようになります。--runasdn "uid=amAdmin, ou=People, o=iplanet.com, o=isp"

--password (-w)

--password は必須のオプションであり、--runasdn オプションで指定した DN のパスワードの値になります。

--locale (-l)

--locale は、ロケール名の値になります。メッセージ言語をカスタマイズするために使用できます。このオプションは、メッセージ言語のカスタマイズに使用できます。指定しない場合は、デフォルトのロケールである en_US が使用されます。

--continue (-c)

--continue は、エラーがある場合でも XML ファイルを処理し続けます。たとえば同時にロードされる XML ファイルが 3 つあり、最初の XML ファイルがエラーになった場合、amadmin では残りのファイルをロードし続けます。continue オプションは、個別の要求のみに適用されます。

--session (-m)

--session (-m) は、セッションを管理したり、現在のセッションを表示したりします。--runasdn を指定するときは、AMConfig.properties のスーパーユーザーの DN、または最上位の管理ユーザーの ID と同じでなければなりません。

次の例では、特定のサービスホスト名に対するすべてのセッションを表示します。

```
amadmin -u uid=amadmin,ou=people,dc=iplanet,dc=com -v-w 12345678 -m
http://sun.com:58080
```

次の例では、特定のユーザーのセッションを表示します。

```
amadmin -u uid=amadmin,ou=people,dc=iplanet,dc=com -v -w 12345678 -m
http://sun.com:58080 ユーザー名
```

セッションを中断するには、対応するインデックス番号を入力します。複数のセッションを中断するには、複数のインデックス番号をスペース区切りで入力します。

次のオプションを使用する場合

amadmin -m | --session サーバー名パターン

パターンには、ワイルドカード (*) も使用できます。このパターンにワイルドカード (*) を使用する場合は、メタ文字 (¥) を使ってシェルからエスケープする必要があります。

--debug (-d)

--debug は、*identity_server_root*/var/opt/SUNWam/debug ディレクトリに作成される amAdmin ファイルにメッセージを書き込みます。このメッセージは技術的には詳細なものですが、i18n 互換ではありません。amadmin の操作ログを生成するには、データベースのログ書き込み時に、データベースドライバのクラスパスを手作業で追加する必要があります。たとえば、mysql にログを書き込むときに、amadmin に次の行を追加します。

```
CLASSPATH=$CLASSPATH:/opt/IS61/SUNWam/lib/mysql-connector-java-3.0.6
-stable-bin.jar

export CLASSPATH
```

--verbose (-v)

--verbose は、amadmin コマンドの処理状況の全体を画面に出力します。ファイルには詳細な情報を出力しません。コマンド行のメッセージ出力は、i18n 互換です。

--data (-t)

--data は、インポートされるバッチ処理用 XML ファイルの名前の値です。1 つ以上の XML ファイルを指定できます。この XML ファイルではさまざまなディレクトリオブジェクトを作成、削除、および読み取ることができるほか、サービスを登録および登録解除できます。このオプションに渡される XML ファイルの種類の詳細については、『Access Manager Developer's Guide』の「Service Management」の章を参照してください。

--schema (-s)

--schema は、Access Manager サービスの属性を Directory Server にロードします。サービス属性が定義されている XML サービスファイルを引数に取ります。この XML サービスファイルは、sms.dtd を基にしています。1 つ以上の XML ファイルを指定できます。

注 DIT に対するバッチ更新を設定するか、サービススキーマおよび設定データをロードするかによって、--data または --schema オプションを指定する必要があります。

--deleteservice (-r)

--deleteservice は、サービスとそのスキーマだけを削除します。

--serviceName

`--serviceName` は、XML サービスファイルの `Service name=...` タグに指定されているサービス名の値です。この部分を 219 ページのコード例 9-1 に示します。

コード例 9-1 `sampleMailService.xml` の一部

```
...
<ServicesConfiguration>
    <Service name="sampleMailService" version="1.0">
        <Schema
 serviceHierarchy="/other.configuration/sampleMailService"
            i18nFileName="sampleMailService"
            i18nKey="iplanet-am-sample-mail-service-description">
...
```

--help (-h)

`--help` は、`amadmin` コマンドの構文を表示する引数です。

--version (-n)

`--version` は、ユーティリティ名、製品名、製品バージョン、および法律上の通知を表示する引数です。

amadmin を連携管理に使用する

この節では、連携管理で使用する amadmin のパラメータを示します。連携管理の詳細は『Access Manager Federation Management Guide』を参照してください。

Liberty のメタに準拠した XML を Directory Server にロードする

`amadmin -u|--runasdn < ユーザーの DN >`

`-w|--password < パスワード > または -f|--passwordfile < パスワードファイル >`

`-e|--entityname < エンティティ名 >`

`-g|--import <XML ファイル >`

--runasdn (-u)

ユーザーの DN

--password (-w)

ユーザーのパスワードです。

--passwordfile (-f)

ユーザーのパスワードが書かれているファイルの名前です。

--entityname (-e)

エンティティ名。たとえば、http://www.example.com などです。エンティティは、1 つの組織に属していなければなりません。

--import (-g)

メタ情報を保持する XML ファイルです。このファイルは Liberty のメタ仕様と XSD に従わなければなりません。

エンティティをデジタル署名なしで XML ファイルにエクスポートする

amadmin -u|--runasdn < ユーザーの DN >

-w|--password < パスワード > または -f|--passwordfile < パスワードファイル >

-e|--entityname < エンティティ名 >

-o|--export < ファイル名 >

--runasdn (-u)

ユーザーの DN

--password (-w)

ユーザーのパスワードです。

--passwordfile (-f)

ユーザーのパスワードが書かれているファイルの名前です。

--entityname (-e)

Directory Server にあるエンティティ名です。

--export (-o)

エンティティの XML が書かれているファイルの名前です。XML は Liberty のメタ XSD に準拠していなければなりません。

エンティティをデジタル署名つきで XML ファイルにエクスポートする

amadmin -u|--runasdn < ユーザーの DN >

-w|--password < パスワード > または -f|--passwordfile < パスワードファイル >

```
    -e|--entityname < エンティティ名 >
    -q|--exportwithsing < ファイル名 >
```

--runasdn (-u)

ユーザーの DN

--password (-w)

ユーザーのパスワードです。

--passwordfile (-f)

ユーザーのパスワードが書かれているファイルの名前です。

--entityname (-e)

Directory Server にあるエンティティ名です。

--exportwithsig (-o)

エンティティの XML が書かれているファイルの名前です。このファイルはデジタル署名されています。XML は Liberty のメタ XSD に準拠していなければなりません。

リソースバンドルに amadmin を使用する

以下の節では、amadmin 構文を使ってリソースバンドルの追加、検索、および削除を行う方法を説明します。

リソースバンドルを追加する

```
amadmin -u|--runasdn < ユーザーの DN> -w|--password < ユーザーのパスワード >
        -b|--addresourcebundle < リソースバンドル名 >
        -i|--resourcebundlefilename < リソースバンドルファイル名 >
        [-R|--resourcelocale] < ロケール >
```

リソース文字列を得る

```
amadmin -u|--runasdn < ユーザーの DN> -w|--password < ユーザーのパスワード >
        -z|--getresourcestrings < リソースバンドル名 >
        [-R|--resourcelocale] < ロケール >
```

リソースバンドルを削除する

```
amadmin -u|--runasdn <ユーザーの DN> -w|--password <ユーザーのパスワード>
        -j|--deleteresourcebundle <リソースバンドル名>
        [-R|--resourcelocale] <ロケール>
```

第 10 章

amserver コマンド行ツール

この章では、amserver コマンド行ツールについて説明します。この章は、次の節で構成されています。

amserver コマンド行実行可能ファイル

amserver コマンド行実行可能ファイルでは、それぞれ UNIX 認証モジュールと SecurID 認証モジュールに関連する amunixd ヘルパーと amsercuridd ヘルパーの起動と停止を行います。

amserver の構文

amserver ツールの一般的な構文は次のとおりです。

```
./amserver { start | stop }
```

start

start は、ヘルパーを開始するコマンドです。

stop

stop は、ヘルパーを停止するコマンドです。

第 11 章

am2bak コマンド行ツール

この章では、am2bak コマンド行ツールについて説明します。この章は、次の節で構成されています。

am2bak コマンド行実行可能ファイル

Access Manager の am2bak ユーティリティは AccessManager-base/SUNWam/bin にあります。am2bak ユーティリティは、Access Manager のコンポーネントのすべてまたは一部をバックアップします。ログのバックアップ中は、Directory Server を実行している必要があります。

am2bak の構文

Solaris オペレーティングシステムで am2bak ツールを使用するための一般的な構文は次のとおりです。

```
./am2bak [ -v | --verbose ] [ -k | --backup バックアップ名 ] [ -l |
--location 場所 ] [[-c | --config] | [-b | --debug] | [-g | --log] |
[-t | --cert] | [-d | --ds] | [-a | --all]]*
```

```
./am2bak -h | --help
```

```
./am2bak -n | --version
```

Microsoft Windows 2000 オペレーティングシステムで am2bak ツールを使用するための一般的な構文は次のとおりです。

```
am2bak [ -v | --verbose ] [ -k | --backup バックアップ名 ] [ -l |
--location 場所 ] [[-c | --config] | [-b | --debug] | [-g | --log] |
[-t | --cert] | [-d | --ds] | [-a | --all]]*
```

```
am2bak -h | --help
am2bak -n | --version
```

注	2 連続するハイフンは、構文に示すとおりに入力する必要があります。

am2bak のオプション

--verbose (-v)

--verbose は、バックアップユーティリティを冗長モードで実行するときに使用します。

--backup バックアップ名 (-k)

--backup バックアップ名は、バックアップファイルの名前を定義します。デフォルトは ambak です。

--location (-l)

--location は、バックアップに使用するディレクトリの場所を指定します。デフォルトの場所は AccessManager-base/backup です。

--config (-c)

--config は、設定ファイルのみバックアップすることを指定します。

--debug (-b)

--debug は、デバッグファイルのみバックアップすることを指定します。

--log (-g)

--log は、ログファイルのみバックアップすることを指定します。

--cert (-t)

--cert は、証明書データベースファイルのみバックアップすることを指定します。

--ds (-d)

--ds は、Directory Server のみバックアップすることを指定します。

--all (-a)

--all は、Access Manager 全体を完全バックアップすることを指定します。

--help (-h)

`--help` は、`am2bak` コマンドの構文を表示する引数です。

--version (-n)

`--version` は、ユーティリティ名、製品名、製品バージョン、および法律上の通知を表示する引数です。

バックアップ手順

1. ルートユーザーでログインします。

   このスクリプトを実行するには、ルートユーザーのアクセス権が必要です。

2. 必要に応じて、正しいパスを使用していることを確認するためのスクリプトを実行します。

   このスクリプトでは、次の Solaris™ Operating Environment ファイルをバックアップします。

   - 設定ファイルおよびカスタマイズファイル
     - *AcessManager-base*`/SUNWam/config/`
     - *AcessManager-base*`/SUNWam/locale/`
     - *AcessManager-base*`/SUNWam/servers/httpacl`
     - *AcessManager-base*`/SUNWam/lib/*.properties` (Java プロパティファイル)
     - *AcessManager-base*`/SUNWam/bin/amserver.` インスタンス名
     - *AcessManager-base*`/SUNWam/servers/https-` すべてのインスタンス
     - *AcessManager-base*`/SUNWam/servers/web-apps-`すべてのインスタンス
     - *AcessManager-base*`/SUNWam/web-apps/services/WEB-INF/config`
     - *AcessManager-base*`/SUNWam/web-apps/services/config`
     - *AcessManager-base*`/SUNWam/web-apps/applications/WEB-INF/classes`
     - *AcessManager-base*`/SUNWam/web-apps/applications/console`
     - `/etc/rc3.d/K55amserver.` すべてのインスタンス
     - `/etc/rc3.d/S55amserver.` すべてのインスタンス
     - *DirectoryServer_base*`/slapd-` ホスト `/config/schema/`
     - *DirectoryServer_base*`/slapd-` ホスト `/config/slapd-collations.conf`
     - *DirectoryServer_base*`/slapd-` ホスト `/config/dse.ldif`

- ログファイルおよびデバッグファイル
  - var/opt/SUNWam/logs（Access Manager ログファイル）
  - var/opt/SUNWam/install（Access Manager インストールログファイル）
  - var/opt/SUNWam/debug（Access Manager デバッグファイル）
- 証明書
  - *AcessManager-base*/SUNWam/servers/alias
  - *DirectoryServer_base*/alias

このスクリプトでは、次の Microsoft Windows 2000 オペレーティングシステムファイルもバックアップします。

- 設定ファイルおよびカスタマイズファイル
  - *AcessManager-base*/web-apps/services/WEB-INF/config/*
  - *AcessManager-base*/locale/*
  - *AcessManager-base*/web-apps/applications/WEB-INF/classes/*.properties（Java プロパティファイル）
  - *AcessManager-base*/servers/https- ホスト /config/jvm12.conf
  - *AcessManager-base*/servers/https- ホスト /config/magnus.conf
  - *AcessManager-base*/servers/https- ホスト /config/obj.conf
  - *DirectoryServer_base*/slapd-host/config/schema/*.ldif
  - *DirectoryServer_base*/slapd-host/config/slapd-collations.conf
  - *DirectoryServer_base*/slapd-host/config/dse.ldif
- ログファイルおよびデバッグファイル
  - var/opt/logs（Access Manager ログファイル）
  - var/opt/debug（Access Manager デバッグファイル）
- 証明書
  - *AcessManager-base*/servers/alias
  - *AcessManager-base*/alias

第 12 章

bak2am コマンド行ツール

この章では、bak2am コマンド行ツールについて説明します。この章は、次の節で構成されています。

bak2am コマンド行実行可能ファイル

Access Manager の bak2am ユーティリティは AccessManager-base/SUNWam/bin にあります。bak2am ユーティリティでは、am2back ユーティリティでバックアップした Access Manager コンポーネントを復元します。

bak2am の構文

Solaris オペレーティングシステムでの bak2am ツールの一般的な構文は次のとおりです。

```
./bak2am [ -v | --verbose ] -z | --gzip tar.gz ファイル
./bak2am [ -v | --verbose ] -t | --tar tar ファイル
./bak2am -h | --help
./bak2am -n | --version
```

Microsoft Windows 2000 オペレーティングシステムで bak2am ツールを使用するための一般的な構文は次のとおりです。

```
bak2am [ -v | --verbose ] -d | --directory ディレクトリ名
bak2am -h | --help
bak2am -n | --version
```

注 2 連続するハイフンは、構文に示すとおりに入力する必要があります。

bak2am のオプション

--gzip バックアップ名

--gzip は、tar.gz 形式のバックアップファイルのフルパスとファイル名を指定します。デフォルトのパスは、AccessManager-base/backup です。Solaris 専用のオプションです。

--tar バックアップ名

--tar は、tar 形式のバックアップファイルのフルパスとファイル名を指定します。デフォルトのパスは、AccessManager-base/backup です。Solaris 専用のオプションです。

--verbose

--verbose は、バックアップユーティリティを冗長モードで実行するときに使用します。

--directory

--directory は、バックアップのあるディレクトリを指定します。デフォルトのパスは、AccessManager-base/backup です。Microsoft Windows 2000 専用のオプションです。

--help

--help は、bak2am コマンドの構文を表示する引数です。

--version

--version は、ユーティリティ名、製品名、製品バージョン、および法律上の通知を表示する引数です。

1. ルートユーザーでログインします。

   このスクリプトを実行するには、ルートユーザーのアクセス権が必要です。

2. 入力 tar ファイルを解凍します。

   入力 tar ファイルは、バックアップスクリプトの実行時に作成されています。

第 13 章

ampassword コマンド行ツール

この章では、amPassword コマンド行ツールについて説明します。この章は、次の節で構成されています。

ampassword コマンド行実行可能ファイル

Access Manager の ampassword ユーティリティは etc/opt/SUNWam/bin にあります。ampassword ユーティリティは、管理者またはユーザーの Access Manager パスワードを変更します。

ampassword の構文

ampassword ツールの一般的な構文は次のとおりです。

```
ampassword -a | --admin [ -o | --old 旧パスワード -n | --new 新パスワード ]
ampassword -p | --proxy [ -o | --old 旧パスワード -n | --new 新パスワード ]
ampassword -e | --encrypt [ password ]
```

注 2 連続するハイフンは、構文に示すとおりに入力する必要があります。

ampassword のオプション

--admin (-a)

--admin は、管理者のパスワードを変更します。

--proxy (-p)

--proxy は、プロキシパスワードを変更します。プロキシユーザー (serverconfig.xml でユーザータイプ proxy) に対応しています。

--version

--version は、ユーティリティ名、製品名、製品バージョン、および法律上の通知を表示する引数です。

--encrypt (-e)

--encrypt は、パスワードを暗号化します。コマンド行に出力されます。たとえば、新しい dsamuser パスワードを暗号化するには、次のコマンドを使用します。

```
ampassword -e newPassword
```

次に、新しい dsamuser パスワードを serverconfig.xml に入力し、Web コンテナ (Web Server または Application Server) を再起動します。

SSL での ampassword の実行

SSL (Secure-Socket Layer) モードで実行中の Access Manager で ampassword を実行するには、次の手順を実行します。

1. serverconfig.xml ファイルを修正します。このファイルは、次のディレクトリにあります。

   AccessManager-base/SUNWam/config/

2. サーバー属性 port を Access Manager を実行している SSL ポートに変更します。
3. type 属性を SSL に変更します。

   次に例を示します。

```
<iPlanetDataAccessLayer>

<ServerGroup name="default" minConnPool="1" maxConnPool="10">

    <Server name="Server1" host="sun.com" port="636" type="SSL" />

    <User name="User1" type="proxy">
```

```
        <DirDN>

                cn=puser,ou=DSAME Users,dc=iplanet,dc=com

        </DirDN>

        <DirPassword>

                 AQIC5wM2LY4Sfcy+AQBQxghVwhBE92i78cqf

         </DirPassword>

</User> ...
```

ampassword では、Directory Server 内のパスワードだけが変更されます。Access Manager のすべての認証テンプレートおよび `ServerConfig.xml` にあるパスワードは、手動で変更する必要があります。

第 14 章

VerifyArchive コマンド行ツール

この章では、VerifyArchive コマンド行ツールについて説明します。この章は、次の節で構成されています。

VerifyArchive コマンド行実行可能ファイル

VerifyArchive は、ログアーカイブを検証するために使用します。ログアーカイブとは、タイムスタンプ付きのログと、対応するキーストアのセットのことです。キーストアには、ログファイルの改ざんを検出するための MAC およびデジタル署名を生成するために使用する鍵が含まれます。アーカイブの検証では、アーカイブ内の、改ざんされたり削除されたりした可能性のあるファイルを検出します。

VerifyArchive では、指定された logName に対して、すべてのアーカイブセットと、各アーカイブセットに属するすべてのファイルを抽出します。VerifyArchive を実行すると、各ログレコードで改ざんを探します。改ざんが検出されると、改ざんのあったファイルとそのレコードの番号を知らせるメッセージが出力されます。

VerifyArchive では、アーカイブセットから削除されたファイルも確認します。削除されたファイルが検出されると、検証に失敗したことを知らせるメッセージが出力されます。改ざんまたは削除されたファイルが検出されなかった場合は、アーカイブの検証が正常に終了したことを知らせるメッセージが返されます。

注 管理者権限を持っていないユーザーが amverifyarchive を実行すると、エラーが発生する場合があります。

VerifyArchive の構文

すべてのパラメータは必須です。構文は次のとおりです。

`VerifyArchive -l logName -p` *`path`* `-u` *`uname`* `-w` *`password`*

VerifyArchive のオプション

logName

`logName` は、検証されるログの名前 (`amConsole`、`amAuthentication` など) を指定します。`VerifyArchive` では、指定された logName に対してアクセスログとエラーログの両方を検証します。たとえば `amConsole` を指定すると、`amConsole.access` および `amConsole.error` ファイルが検証されます。その代わりに、`logName` に `amConsole.access` または `amConsole.error` と指定することで、検証をこれらのログに制限できます。

path

`path` は、ログファイルが格納されているディレクトリのフルパスです。

uname

`uname` は、Access Manager 管理者のユーザー ID です。

password

`password` は、Access Manager 管理者のパスワードです。

第 15 章

amsecuridd ヘルパー

この章では、amsecuridd ヘルパーについて説明します。この章は、次の節で構成されています。

amsecuridd ヘルパーコマンド行実行可能ファイル

Access Manager の SecurID 認証モジュールは、Security Dynamic ACE/Client C API と amsecuridd ヘルパーを使って実装されます。このヘルパーは、Access Manager の SecurID 認証モジュールと SecurID Server の間の通信を行います。SecurID 認証モジュールは、localhost:57943 へのソケットを開いて amsecuridd デーモンを呼び出し、SecurID 認証要求を待機します。

注 57943 はデフォルトのポート番号です。このポート番号がすでに使用されている場合は、SecurID 認証モジュールの SecurID ヘルパー認証ポート属性で別のポート番号を指定できます。このポート番号は、すべての組織で一意でなければなりません。

amsecuridd へのインタフェースは、stdin からのクリアテキストなので、ローカルホスト接続だけが許可されています。amsecuridd は、バックエンドで SecurID リモート API (バージョン 5.x) を使ってデータを暗号化します。

amsecuridd ヘルパーは、認定情報を受け取るために、デフォルトではポート番号 58943 で待機します。このポートがすでに使用されている場合は、AMConfig.properties ファイルの securidHelper.ports 属性でポートを変更できます。このファイルはデフォルトで、*AccessManager-base*/SUNWam/config/ にあります。securidHelp.ports 属性には、amsecuridd ヘルパーの各インスタンスのポートが、スペース区切りのリストとして格納されています。AMConfig.properties に加えた変更を保存したら、Identity Sever を再起動してください。

注 異なる sdconf.rec ファイルを持つ別々の ACE/Server と通信する組織ごとに、個別の amsecuridd インスタンスを実行する必要があります。

amsecuridd の構文

構文は次のとおりです。

```
amsecuridd [-v] [-c ポート番号 ]
```

amsecuridd のオプション

verbose (-v)

冗長モードをオンにし、/var/opt/SUNWam/debug/securidd_client.debug にログを記録します。

configure portnumber (-c portnm)

待機ポート番号を設定します。デフォルトは 58943 です。

amsecuridd ヘルパーの実行

amsecuridd は、デフォルトで *AccessManager-base*/SUNWam/share/bin にあります。デフォルトのポートでヘルパーを実行するには、オプションを指定せずに次のコマンドを入力します。

```
./amsecuridd
```

デフォルト以外のポートでヘルパーを実行するには、次のコマンドを入力します。

```
./amsecuridd [-v] [-c ポート番号 ]
```

amsecuridd を amserver コマンド行ユーティリティーから実行することもできますが、常にデフォルトポートでの実行になります。

必要なライブラリ

このヘルパーを実行するには、次のライブラリが必要です。これらのほとんどは、オペレーティングシステムの /usr/lib/ にあります。

- libnsl.so.1
- libthread.so.1
- libc.so.1
- libdl.so.1
- libmp.so.2
- librt.so.1
- libaio.so.1
- libmd5.so.1

注 libaceclnt.so が見つかるように、LD_LIBRARY_PATH を *AccessManager-base*/Sunwam/lib/ に設定します。

第 IV 部

属性リファレンスガイド

この「属性リファレンスガイド」は、『Sun Java System Acceess Manager 管理ガイド』の第 4 部です。Access Manager のデフォルトサービス内で設定済みの属性について説明します。次の章で構成されています。

第 16 章

管理サービス属性

管理サービスにはグローバル属性と組織属性があります。グローバル属性に適用される値は Sun Java System Acceess Manager 設定全体に適用され、設定済みのすべての組織に継承されます。グローバル属性の目的は Access Manager アプリケーションをカスタマイズすることであるため、ロールまたは組織に直接適用することはできません。組織属性に適用される値は設定済みの各組織のデフォルト値で、サービスを組織に登録するときに変更できます。組織属性は組織のエントリに継承されません。管理属性は次のように分けられます。

グローバル属性

管理サービスのグローバル属性は次のとおりです。

- 251 ページの「ユーザープロファイルサービスクラス」
- 251 ページの「DC ノードの属性リスト」
- 252 ページの「削除したオブジェクトの検索フィルタ」
- 252 ページの「デフォルトピープルコンテナ」
- 252 ページの「デフォルトグループコンテナ」
- 252 ページの「デフォルトエージェントコンテナ」

連携管理を有効

このフィールドを選択すると、連携管理が有効になります。デフォルトは有効です。この機能を無効にするには、フィールドの選択を解除します。「連携管理サービス」タブはコンソールに表示されなくなります。

ユーザー管理を有効

このフィールドが true (チェックボックスを選択) の場合、ユーザー管理が有効になります。デフォルトでは、有効になっています。

ピープルコンテナを表示

この属性は、Access Manager コンソールにピープルコンテナを表示するかどうかを指定します。このオプションを選択すると、組織、コンテナ、およびグループコンテナの「表示」メニューに「ピープルコンテナ」メニュー項目が表示されます。「ピープルコンテナ」はフラット DIT の最上位レベルにのみ表示されます。

ピープルコンテナは、ユーザープロファイルを含む組織単位です。DIT で 1 つのピープルコンテナを使用し、ロールの柔軟性を利用してアクセスおよびサービスを管理することをお勧めします。Access Manager コンソールのデフォルトの動作では、ピープルコンテナは非表示です。ただし、DIT に複数のピープルコンテナがある場合は、「ピープルコンテナを表示」を選択して、ピープルコンテナを Access Manager コンソールの管理オブジェクトとして表示します。

表示メニューにコンテナを表示

この属性は、Access Manager コンソールの「表示」メニューにコンテナを表示するかどうかを指定します。デフォルト値は false です。管理者は必要に応じてどちらかを選択できます。

- false (チェックボックスを選択しない) - コンテナは組織およびほかのコンテナの最上位レベルの「表示」メニューの項目に含まれません。
- true (チェックボックスを選択する) - コンテナは組織およびほかのコンテナの最上位レベルの「表示」メニューの項目に含まれます。

グループコンテナを表示

この属性は、Access Manager コンソールにグループコンテナを表示するかどうかを指定します。このオプションを選択すると、組織、コンテナ、およびグループコンテナの「表示」メニューに「グループコンテナ」メニュー項目が表示されます。グループコンテナはグループの組織単位です。

管理されているグループタイプ

このオプションは、コンソールで作成した登録グループがスタティックかダイナミックかを指定します。コンソールは、スタティックでありかつダイナミックである加入グループではなく、スタティックまたはダイナミックのどちらかである登録グループを作成および表示します。フィルタを適用したグループは、この属性に指定された値には関係なく常にサポートされます。デフォルト値はダイナミックです。

- スタティックグループは、groupOfNames または groupOfUniqueNames オブジェクトクラスを使って、各グループメンバーを明示的に一覧表示します。グループエントリには、グループの各メンバーの uniqueMember 属性が含まれます。スタティックグループのメンバーは手動で追加しますが、ユーザーエントリ自体は変更されません。スタティックグループはメンバーの少ないグループに適しています。
- ダイナミックグループは、各グループメンバーのエントリの memberOf 属性を使用します。ダイナミックグループのメンバーは、memberOf 属性を含むすべてのエントリを検索して返す LDAP フィルタを使って生成されます。ダイナミックグループは、メンバーが非常に多いグループに適しています。

- フィルタを適用したグループは、LDAP フィルタを使用して、フィルタの要件を満たすメンバーを検索して返します。たとえば、フィルタは、特定の uid (uid=g*) または電子メールアドレス (mail=*@sun.com) を持つメンバーを生成できます。これらの例では、LDAP フィルタはそれぞれ、uid が g で始まる、または電子メールアドレスが sun.com で終わるすべてのユーザーを返します。フィルタを適用したグループは、「フィルタのメンバーシップ」を選択して、ユーザー管理ビュー内でのみ作成できます。

管理者は次の中から 1 つ選択できます。

- ダイナミック - 登録によるメンバーシップのオプションを使って作成されたグループはダイナミックになります。
- スタティック - 登録によるメンバーシップのオプションを使って作成されたグループはスタティックになります。

デフォルトロールアクセス権

この属性は、新しいロールの作成時に管理者権限の認可に使うデフォルト ACI (アクセス制御命令) または権限のリストを定義します。必要な権限のレベルに応じて、これらの ACI の 1 つを選択します。Access Manager にはデフォルトロール権限が 4 つあります。

アクセス権なし (No permission)

ロールにアクセス権が設定されません。

組織管理者 (Organization Admin)

組織管理者は設定済み組織のすべてのエントリに対する読み取りアクセス権と書き込みアクセス権を持っています。

組織のヘルプデスク管理者 (Organization Help Desk Admin)

組織のヘルプデスク管理者は、設定済み組織のすべてのエントリに対する読み取りアクセス権、および userPassword 属性に対する書き込みアクセス権を持っています。

組織ポリシー管理者 (Organization Policy Admin)

組織のポリシー管理者は、組織のすべてのポリシーに対する読み取りアクセス権と書き込みアクセス権を持っています。組織のポリシー管理者は、ピア組織に対する参照ポリシーを作成できません。

注

ロールは、`aci_name | aci_desc | dn:aci ## dn:aci ## dn:aci` という形式で定義します。

- *aci_name* は ACI の名前です。
- *aci_desc* はこれらの ACI が許可するアクセスの説明です。使いやすくするため、対象読者は ACI やその他のディレクトリの概念に関する知識がないものと仮定します。

aci_name および *aci_desc* は、`amAdminUserMsgs.properties` ファイルに含まれる国際化 (i18n) キーです。コンソールに表示される値は、`.properties` ファイルから取得され、これらのキーを使用して値を検索します。

- *dn:aci* は DN と ACI のペアを表し、## で区切ります。Access Manager は関連付けられた DN エントリの各 ACI を設定します。この形式では、ACI で指定する必要がある値の代わりに ROLENAME、ORGANIZATION、GROUPNAME および PCNAME タグを指定できます。これらのタグを使用することによって、デフォルトとして使用するのに十分に柔軟なロールを定義できます。デフォルトのロールの 1 つに基づいてロールを作成すると、ACI のタグはその新しいロールの DN から取得した値になります。

ドメインコンポーネントツリーを有効

ドメインコンポーネントツリー (DC ツリー) は固有の DIT 構造で、多くの Sun Java System コンポーネントがこれを使用して、DNS 名と組織のエントリ間のマッピングを行います。

このオプションを有効にすると、組織が作成されたときに組織の DNS 名が入力されていれば、組織の DC ツリーエントリが作成されます。DNS 名フィールドは、組織作成ページに表示されます。このオプションは最上位レベルの組織にだけ適用され、サブ組織には表示されません。

組織ツリーで、Access Manager SDK を使用して `inetdomainstatus` 属性の状態を変更すると、対応する DC ツリーエントリの状態が更新されます。Access Manager SDK を使用せずに状態を更新した場合、その内容は同期しません。たとえば、新しい組織 `sun` を sun.com という DNS 名属性で作成すると、DC ツリーに次のエントリが作成されます。

`dc=sun,dc=com,o=internet,`*root suffix*

AMConfig.properties で com.iplanet.am.domaincomponent を設定することによって、この DC ツリーに独自のルートサフィックスを設定することもできます。デフォルトでは、これは Access Manager root に設定されています。異なるサフィックスが望ましい場合は、LDAP コマンドを使ってこのサフィックスを作成する必要があります。組織を作成する管理者の ACI は、新しい DC ツリーのルートに無制限のアクセス権を持つように、修正する必要があります。

管理者グループを有効

このオプションは、DomainAdministrators および DomainHelpDeskAdministrators グループを作成するかどうかを指定します。選択すると (true)、これらのグループが作成され、それぞれ組織管理者ロールおよび組織ヘルプデスク管理者ロールと関連付けられます。このグループが作成されると、これらの関連するロールの 1 つにユーザーを追加したり削除したりしたときに、対応するグループへのユーザーの追加や、グループからのユーザーの削除が自動的に行われます。ただし、逆の処理は行われません。これらのグループの 1 つでユーザーの追加や削除をしても、関連するロールではユーザーの追加や削除は行われません。

DomainAdministrators および DomainHelpDeskAdministrators グループは、このオプションを有効にしたあとに作成された組織でのみ作成されます。

注 このオプションはサブ組織には適用されません。ただし、root org は例外です。root org には、ServiceAdministrators および ServiceHelpDesk 管理者グループが作成され、それぞれ最上位レベル管理者および最上位レベルヘルプデスク管理者のロールと関連付けられます。同じ動作が適用されます。

ユーザー削除を有効

このオプションは、ディレクトリからユーザーのエントリを削除するか、それとも削除マークを付けるだけかを指定します。ユーザーのエントリが削除され、このオプションが選択されている場合 (true)、ユーザーのエントリはまだディレクトリに存在していますが、削除マークは付けられています。削除マークを付けられたユーザーエントリは、ディレクトリサーバーの検索時に返されることはありません。このオプションが選択されていない場合は、ユーザーのエントリはディレクトリから削除されます。

ダイナミック管理ロール ACI

この属性は、Access Manager を使ってグループまたは組織を構成するときにダイナミックに作成される管理者ロールのアクセス制御命令を定義します。これらのロールは、作成したエントリの特定のグループに管理権限を与えるのに使用します。デフォルトの ACI はこの属性リストの下でのみ変更できます。

警告 組織レベルの管理者は、グループ管理者よりも広範なアクセス権を持っています。ただし、デフォルトでは、ユーザーをグループ管理者ロールに追加すると、そのユーザーはグループのすべてのユーザーのパスワードを変更できます。これには、そのグループのメンバーである組織管理者も含まれます。

コンテナヘルプデスク管理者 (Container Help Desk Admin)

コンテナのヘルプデスク管理者ロールは、組織単位のすべてのエントリに対する読み取りアクセス権、およびそのコンテナ単位だけにあるユーザーエントリの `userPassword` 属性に対する書き込みアクセス権を持っています。

組織のヘルプデスク管理者 (Organization Help Desk Admin)

組織のヘルプデスク管理者は、組織のすべてのエントリに対する読み取りアクセス権、および `userPassword` 属性に対する書き込みアクセス権を持っています。

注 サブ組織を作成するときは、管理者ロールは親組織ではなくサブ組織に作成してください。

コンテナ管理者 (Container Admin)

コンテナ管理者ロールは、LDAP 組織単位のすべてのエントリに対する読み取りアクセス権と書き込みアクセス権を持っています。Access Manager では、LDAP 組織単位をコンテナと呼ぶことがあります。

組織ポリシー管理者 (Organization Policy Admin)

組織のポリシー管理者は、組織のすべてのポリシーに対する読み取りアクセス権と書き込みアクセス権を持っており、組織内のすべてのポリシーについて作成、割り当て、修正、および削除ができます。

ピープルコンテナ管理者 (People Container Admin)

デフォルトで、新規に作成した組織のユーザーエントリはその組織のピープルコンテナのメンバーです。ピープルコンテナ管理者は、組織のピープルコンテナのすべてのユーザーエントリに対する読み取りアクセス権と書き込みアクセス権を持っています。なお、このロールは、ロールおよびグループ DN を含む属性に対する読み取りアクセス権と書き込みアクセス権を持っていないため、ロールまたはグループの属性を変更したり、ロールまたはグループからユーザーを消去したりすることができません。

注 ほかのコンテナは、Access Manager とともに設定して、ユーザーエントリ、グループエントリ、またはほかのコンテナを保持することができます。組織を構成したあとで、作成されたコンテナに管理者ロールを適用するには、デフォルトのコンテナ管理者ロールまたはコンテナヘルプデスク管理者を使用します。

グループ管理者 (Group Admin)

グループ管理者は、特定グループのすべてのメンバーに対する読み取りアクセス権および書き込みアクセス権を持っており、新しいユーザーの作成、管理しているグループへのユーザーの割り当て、および作成したユーザーの削除を行うことができます。

グループを作成すると、そのグループを管理するのに必要な権限を持つグループ管理者ロールが自動的に作成されます。このロールはグループのメンバーに自動的には割り当てられません。グループの作成者、またはグループ管理者ロールへのアクセス権を持つ人が割り当てる必要があります。

最上位レベル管理者 (Top-level Admin)

最上位レベル管理者は、最上位レベル組織のすべてのエントリに対する読み取りアクセス権と書き込みアクセス権を持っています。言い換えれば、最上位レベル管理者ロールには、Access Manager アプリケーション内のすべての設定主体に対する権限があります。

組織管理者 (Organization Admin)

組織管理者は、組織のすべてのエントリに対する読み取りアクセス権と書き込みアクセス権を持っています。組織を作成すると、その組織を管理するのに必要な権限を持つ組織管理者ロールが自動的に作成されます。

ユーザープロファイルサービスクラス

この属性は、ユーザープロファイルページでカスタム表示を持つサービスを一覧表示します。サービスによっては、コンソールによって生成されるデフォルト表示では不十分な場合があります。この属性は、どんなサービスでもカスタム表示を作成し、表示するサービス情報の内容や、表示の方法をすべてコントロールすることができます。構文は次のとおりです。

```
サービス名 | 相対 URL
```

注 この属性で一覧表示されるサービスは、ユーザー作成ページには表示されません。カスタムサービス表示のデータ設定は、ユーザープロファイルページで行う必要があります。

DC ノードの属性リスト

オブジェクトが作成されるときに、DC ツリーエントリ内に設定される属性のセットを定義します。デフォルトのパラメータは次のとおりです。

- maildomainwelcomemessage
- preferredmailhost
- mailclientattachmentquota
- mailroutingsmarthost
- mailroutingsmarthost
- mailroutingsmarthost
- mailaccessproxyreplay
- preferredlanguage
- domainuidseparator
- maildomainmsgquota
- maildomainallowedserviceaccess
- preferredmailmessagestore
- maildomaindiskquota
- maildomaindiskquota
- objectclass=maildomain
- mailroutinghosts

削除したオブジェクトの検索フィルタ

このフィールドは、「ユーザー削除を有効」モードが有効であるときに削除される、オブジェクトの検索フィルタを定義します。

デフォルトピープルコンテナ

この属性は、ユーザーが作成されるデフォルトピープルコンテナを指定します。

デフォルトグループコンテナ

この属性は、グループが作成されるデフォルトグループコンテナを指定します。

デフォルトエージェントコンテナ

この属性は、エージェントが作成されるデフォルトエージェントコンテナを指定します。

組織属性

管理サービスの組織属性は次のとおりです。

グループのデフォルトピープルコンテナ

このフィールドは、デフォルトのピープルコンテナを指定します。ユーザーは作成時にこのコンテナに配置されます。デフォルト値はありません。有効な値は、ピープルコンテナの DN です。ピープルコンテナの代替順位については、「グループのピープルコンテナリスト」属性の下にある注を参照してください。

グループのピープルコンテナリスト

このフィールドは、ピープルコンテナのリストを指定します。グループ管理者は、新しいユーザーを作成するときに、このリストから選択できます。ディレクトリツリー内に複数のピープルコンテナがある場合に、このリストを使用できます。このリスト、または「グループのデフォルトピープルコンテナ」フィールドにピープルコンテナが指定されていない場合、ユーザーはデフォルトの Access Manager ピープルコンテナ `ou=people` に作成されます。このフィールドのデフォルト値はありません。この属性の構文は次のとおりです。

dn of group | *dn of people container*

注	ユーザーの作成時に、エントリを入れるコンテナのこの属性がチェックされます。この属性が空の場合、コンテナの「グループのデフォルトピープルコンテナ」属性がチェックされます。後者の属性が空の場合、エントリは ou=people の下に作成されます。

ユーザープロファイル表示クラス

この属性は、Access Manager コンソールがユーザープロファイルページを表示するときに使用する Java のクラスを指定します。

エンドユーザープロファイル表示クラス

この属性は、Access Manager コンソールがエンドユーザープロファイルページを表示するときに使用する Java のクラスを指定します。

ユーザープロファイルページにロールを表示

このオプションは、ユーザーに割り当てられているロールのリストを、ユーザーのユーザープロファイルページの一部として表示するかどうかを指定します。この値が false (選択されていない) の場合、管理者に対してのみユーザープロファイルページにユーザーのロールが表示されます。デフォルト値は false です。

ユーザープロファイルページにグループを表示

このオプションは、ユーザーに割り当てられているグループのリストを、ユーザーのユーザープロファイルページの一部として表示するかどうかを指定します。この値が false (選択されていない) の場合、管理者に対してのみユーザープロファイルページにユーザーのグループが表示されます。デフォルト値は false です。

ユーザーのグループへの自己登録を有効

このオプションは、登録可能なグループにユーザーが自分自身を追加できるかどうかを指定します。この値が false の場合、ユーザーのグループメンバーシップを変更できるのは管理者のみです。デフォルト値は `false` です。

注	このオプションは、「ユーザープロファイルページにグループを表示」オプションが選択されているときにだけ適用されます。

ユーザープロファイル表示オプション

このメニューは、どのサービス属性がユーザープロファイルページに表示されるかを指定します。管理者は次の中から選択できます。

- `UserOnly` - そのユーザーに割り当てられたサービスの、表示可能なユーザースキーマ属性を表示します。

  属性にキーワード「表示」が含まれている場合、ユーザーはユーザーサービス属性の値を見ることができます。詳細は、『Access Manager Developer's Guide』を参照してください。

- `Combined` - そのユーザーに割り当てられたサービスの、表示可能なユーザーおよびダイナミックスキーマ属性を表示します。

ユーザー作成のデフォルトロール

このリストは、新規に作成されたユーザーに自動的に割り当てるロールを定義します。デフォルト値はありません。管理者は 1 つまたは複数のロールの DN を入力できます。

注	このフィールドに指定できるのは、完全識別名のアドレスのみです。ロール名は指定できません。ロールは LDAP (Directory Server) ロールではなく、Access Manager ロールでなければなりません。

管理コンソールタブ

このフィールドは、コンソールの上部に表示されるモジュールの Java クラスを一覧表示します。構文は、`i18N キー | Java クラス名`です。i18N キーは、「表示」メニューのエントリのローカル名に使用します。

検索で返される結果の最大数

このフィールドは検索で返される結果の最大数を定義します。デフォルト値は 100 です。

警告 この属性に大きな値を設定するときは注意してください。サイズの制限については、次の場所にある『Sun Java System Directory Server Installation and Tuning Guide』を参照してください。

http://docs.sun.com/db/doc/816-6697-10

LDAPModify で行った、この属性の修正内容は、Access Manager コンソールで行った修正内容より優先されます。LDAPModify を使用してこの属性を変更する方法については、『Access Manager Developer's Guide』を参照してください。

検索タイムアウト

このフィールドは検索を開始してからタイムアウトするまでの時間 (秒数) を定義します。これは長くかかる可能性のある検索を停止するために使用します。最大検索時間に達すると、エラーが返されます。デフォルト値は 5 秒です。

JSP ディレクトリ名

このフィールドは、コンソールを構築するのに使用する .jsp ファイルを含むディレクトリの名前を指定し、組織に異なった外観を与えます (カスタマイズ)。.jsp ファイルは、このフィールドで指定されたディレクトリにコピーする必要があります。

オンラインヘルプドキュメント

このフィールドは、Access Manager ヘルプのメインページ上に作成されるオンラインヘルプリンクを一覧表示します。これにより、ほかのアプリケーションは、そのオンラインヘルプリンクを Access Manager ページに追加できます。この属性の形式は次のとおりです。

linki18nkey | クリックしたときにロードする html ページ | i18n プロパティファイル | リモートサーバー

注 リモートサーバーはオプションの引数で、オンラインヘルプドキュメントがあるリモートサーバーを指定できます。

次に例を示します。

```
IdentityServer Help | /AMAdminHelp.html | amAdminModuleMsgs
```

必要なサービス

このフィールドは、ユーザーが作成されたときにダイナミックにユーザーのエントリに追加されるサービスを一覧表示します。管理者は、作成時にどのサービスを追加するかを選択できます。

この属性は、コンソールではなく、Access Manager SDK によって使用されます。ダイナミックに作成されるユーザーと、amadmin コマンド行ユーティリティーで作成されるユーザーには、この属性に含まれているサービスが割り当てられます。

ユーザー検索キー

この属性は、ナビゲーションページで単純検索を実行するときに検索対象となる属性の名前を定義します。この属性のデフォルト値は cn です。たとえば、この属性がデフォルト値を使用している場合は、次のようになります。

ナビゲーションフレームの「名前」フィールドに j* を入力すると、「j」または「J」で始まる名前が表示されます。

ユーザー検索により返される属性

このフィールドは、単純検索から返されるユーザーを表示するときに使用する属性名を定義します。この属性のデフォルトは uid cn です。ユーザー ID とユーザーのフルネームが表示されます。

先頭に表示される属性名は、返されるユーザーのセットをソートするためのキーとしても使用されます。パフォーマンスが低下しないようにするには、ユーザーのエントリに値が設定されている属性を使用します。

ユーザー作成通知リスト

このフィールドは、新しいユーザーが作成されたときに通知を送る電子メールアドレスのリストを定義します。次の構文で示されているように、複数の電子メールアドレスを指定できます。

電子メール | ロケール | 文字セット

電子メール | ロケール | 文字セット

電子メール | ロケール | 文字セット

通知リストには、|locale オプションを使って異なるロケールを指定することもできます。たとえばフランスにいる管理者に通知を送信するには、次のようにします。

```
someuser@example.com|fr|fr
```

ロケールのリストについては、285 ページの表 20-1 を参照してください。

注 amProfile.properties のプロパティ 497 を修正することで、送信元の電子メール ID を変更できます。このファイルは、デフォルトで AccessManager-base/SUNWam/locale にあります。

ユーザー削除通知リスト

このフィールドは、ユーザーが削除されたときに通知を送る電子メールアドレスのリストを定義します。次の構文で示されているように、複数の電子メールアドレスを指定できます。

電子メール | ロケール | 文字セット

電子メール | ロケール | 文字セット

電子メール | ロケール | 文字セット

通知リストには、|locale オプションを使って異なるロケールを指定することもできます。たとえばフランスにいる管理者に通知を送信するには、次のようにします。

```
someuser@example.com|fr|fr
```

ロケールのリストについては、285 ページの表 20-1 を参照してください。

注 amProfile.properties のプロパティ 497 を修正することで、送信元の電子メール ID を変更できます。このファイルは、デフォルトで AccessManager-base/SUNWam/locale にあります。デフォルトの送信元の ID は DSAME です。

ユーザー修正通知リスト

このフィールドは、属性および属性に関連する電子メールアドレスのリストを定義します。リストに定義された属性でユーザーの修正が発生すると、その属性に関連する電子メールアドレスに通知が送信されます。各属性は、それぞれ異なるセットの関連アドレスを持つことができます。次の構文で示されているように、複数の電子メールアドレスを指定できます。

属性名 電子メール | ロケール | 文字セット 電子メール | ロケール | 文字セット .....

属性名 電子メール | ロケール | 文字セット 電子メール | ロケール | 文字セット .....

アドレスのいずれか 1 つに `self` キーワードを使用することもできます。このキーワードを使用すると、プロファイルが修正されたユーザーにメールが送信されます。

次に例を示します。

```
manager someuser@sun.com|self|admin@sun.com
```

この場合、`manager` 属性で指定されたアドレス、`someuser@sun.com`、`admin@sun.com` およびユーザーを修正した人 (`self`) にメールが送信されます。

通知リストには、`|locale` オプションを使って異なるロケールを指定することもできます。たとえばフランスにいる管理者に通知を送信するには、次のようにします。

```
manager someuser@sun.com|self|admin@sun.com|fr
```

ロケールのリストについては、285 ページの表 20-1 を参照してください。

注 属性名は Directory Server スキーマに表示されるのと同じものですが、コンソールの表示名とは違います。

ページごとの最大表示エントリ数

この属性を使用して、ページあたりに表示できる最大行数を定義することができます。デフォルトは 25 です。たとえばユーザー検索結果が 100 行の場合、1 ページあたり 25 行のページが 4 ページ表示されます。

イベントリスナークラス

この属性には、作成、修正、および削除の各イベントを Access Manager コンソールから受け取るリスナーのリストが格納されています。

プレおよびポストプロセスクラス

このフィールドは、ユーザー、組織、ロール、およびグループに対するプレおよびポストプロセス操作中にコールバックを受け取るように、`com.iplanet.am.sdk.AMCallBack` クラスを拡張する実装クラスのリストを、プラグインを通じて定義します。操作は次のとおりです。

- 作成
- 削除
- 修正
- ユーザーをロールまたはグループに追加
- ユーザーをロールまたはグループから削除

プラグインの完全なクラス名を入力する必要があります。次に例を示します。

```
com.iplanet.am.sdk.AMCallbacSample
```

そして、プラグインクラスの場所へのフルパスを含むように、Web コンテナのクラスパスを変更する必要があります。これは Access Manager のインストール単位で行います。

外部属性のフェッチを有効

このオプションは、プラグインで外部属性を受け取れるように、コールバックを有効にします。外部属性とは、外部アプリケーション固有の属性のことです。外部属性は Access Manager SDK ではキャッシュされません。そのためこの属性を使用すると、組織単位のレベルで属性を受け取ることができるようになります。このオプションは、デフォルトでは無効になっています。

無効なユーザー ID 文字

この属性により、ユーザーの名前で使用できない文字のリストを定義します。

それぞれの文字は、| 文字で区切る必要があります。次に例を示します。

```
*|(|)|&|!
```

ユーザー ID とパスワードの検証プラグインクラス

このクラスは、ユーザー ID とパスワードの検証プラグインメカニズムを提供します。

このクラスのメソッドは、ユーザーのユーザー ID およびパスワードを検証するプラグインモジュールの実装によりオーバライドする必要があります。実装されたプラグインモジュールは、Access Manager コンソール、`amadmin` コマンド行インタフェース、あるいは SDK を使用して、ユーザー ID またはパスワードが追加または変更されるたびに呼び出されます。

このクラスを継承するプラグインは、組織ごとに設定できます。組織に対してプラグインが設定されていない場合は、グローバルレベルで設定されたプラグインが使用されます。

プラグインの検証に失敗すると、プラグインモジュールは、ユーザーが提供したユーザー ID またはパスワードのエラーをアプリケーションに通知するために例外をスローします。

第 17 章

Active Directory の認証属性

Active Directory の認証属性は組織属性です。サービス設定の下で組織属性に適用される値は、Active Directory 認証テンプレートのデフォルト値になります。組織にサービスを登録したあと、サービステンプレートを作成する必要があります。デフォルト値は組織の管理者が登録後に変更できます。組織属性は組織のエントリに継承されません。Active Directory の認証属性は次のとおりです。

プライマリ Active Directory サーバー

このフィールドは、Access Manager のインストール時に指定するプライマリ Active Directory サーバーのホスト名およびポート番号を指定します。これは、Active Directory 認証で最初に通信するサーバーです。形式は `hostname:port` です。ポート番号がないときは、`389` と想定します。

複数のドメインに Access Manager が配備されている場合は、Access Manager および Directory Server の個々のインスタンス間の通信リンクを、次の形式で指定できます。複数のエントリを指定する場合は、エントリにローカルサーバー名をプレフィックスとして付ける必要があります。

```
local_servername|server:port local_servername2|server2:port2 ...
```

たとえば、異なる場所に配備された 2 つの Access Manager インスタンス (`L1-machine1-IS` および `L2- machine2-IS`) が、それぞれ別の Directory Server インスタンス (`L1-machine1-DS` および `L2-machine2-DS`) と通信する場合は、次のように指定できます。

```
L1-machine1-IS.example.com|L1-machine1-DS.example.com:389
L2-machine2-IS.example.com|L2-machine2-DS.example.com:389
```

セカンダリ Active Directory サーバー

このフィールドは、Access Manager プラットフォームが利用できるセカンダリ Active Directory サーバーのホスト名およびポート番号を指定します。プライマリ Active Directory サーバーが認証要求に応答しない場合は、このサーバーと通信します。プライマリサーバーが起動すると、Access Manager はプライマリサーバーに戻ります。この形式も `hostname:port` です。複数のエントリの場合は、ローカルサーバー名をプレフィックスとして付ける必要があります。

警告 Access Manager を使用する企業から遠隔地にある Directory Server からユーザーを認証する場合は、プライマリとセカンダリ両方の Active Directory サーバーポートに値があることが重要です。1 つの Directory Server の場所の値を両方のフィールドに使用できます。

ユーザー検索の開始 DN

このフィールドは、ユーザーの検索を開始するノードの DN を指定します。性能上の理由から、この DN はできるだけ特定のものにしてください。デフォルト値は、ディレクトリツリーのルートです。すべての有効な DN が認識されます。「検索範囲」属性で「オブジェクト」を選択する場合、DN にはプロファイルが存在するレベルの 1 レベル上を指定する必要があります。

複数のエントリの場合は、ローカルサーバー名をプレフィックスとして付ける必要があります。形式は次のとおりです。

```
servername|search dn
```

複数のエントリを指定する場合は、次のようになります。

```
servername1|search dn servername2|search dn servername3|search dn...
```

同一の検索で複数のユーザーが見つかった場合、認証は失敗します。

root ユーザーバインド DN

このフィールドは、「プライマリ Active Directory サーバー」フィールドで指定した Directory Server に管理者としてバインドするのに使用するユーザーの DN を指定します。ユーザーログイン ID に基づく、一致するユーザー DN を検索するためには、認証サービスをこの DN にバインドする必要があります。デフォルト値は `amLDAPuser` です。すべての有効な DN が認識されます。

パスワードが間違っているとロックアウトされるので、ログアウトする前にパスワードが正しいことを確認してください。ロックアウトされた場合は、`AMConfig.Properties` ファイル内の `com.iplanet.authentication.super.user` プロパティで指定されているスーパーユーザー DN を使ってログインできます。デフォルトでは、これはログインに通常使用する amAdmin アカウントですが、完全な DN を使用する必要があります。次に例を示します。

```
uid_amAdmin,ou=People,AccessManager-base
```

root ユーザーバインドパスワード

このフィールドは、「root ユーザーバインド DN」フィールドで指定される管理者プロファイルのパスワードを指定します。デフォルト値はありません。管理者の有効な Active Directory パスワードだけが認識されます。

root ユーザーバインドパスワード (確認)

パスワードを確認します。

ユーザープロファイルの取得に使用する属性

ユーザー認証が成功したあと、ユーザーのプロファイルを取得します。この属性の値を使用して検索を実行します。このフィールドは、使用する Active Directory 属性を指定します。デフォルトでは、Access Manager はユーザーエントリが `uid` 属性によって識別されるものと想定します。Directory Server で `givenname` などの異なる属性を使用している場合は、このフィールドに属性名を指定します。

注	ユーザー検索フィルタは、検索フィルタ属性とユーザープロファイルの取得に使用する Active Directory 属性の組み合わせです。

認証するユーザーの検索に使用する属性

このフィールドには、認証対象ユーザーの検索フィルタを設定するために使用する属性を一覧表示します。これによりユーザーは、ユーザーのエントリにある複数の属性によって認証を受けることができます。たとえば、このフィールドが `uid`、`employeenumber`、および `mail` に設定されている場合、ユーザーはこれらの名前のどれを使用しても認証を受けることができます。

ユーザー検索フィルタ

このフィールドは、「ユーザー検索の開始 DN」フィールドの下でユーザーの検索に使用する属性を指定します。これはユーザーエントリネーミング属性とともに機能します。デフォルト値はありません。有効なユーザーエントリ属性はすべて認識されます。

検索範囲

このメニューは、一致するユーザープロファイルの検索対象となる、Directory Server 内の階層の数を示します。検索は、265 ページの「ユーザー検索の開始 DN」属性で指定されるノードから開始します。デフォルト値はサブツリーです。次のリスト項目から 1 つ選択できます。

- オブジェクト - 指定したノードだけを検索

- 1 レベル - 指定したノードのレベルとその 1 つ下のレベルで検索
- サブツリー - 指定したノードとその下のノードのエントリすべてを検索

警告 サブ組織のユーザーは、サブ組織の状態が非アクティブであってもログインする可能性があります。これを防ぐには、ユーザーの所属する特定の組織を指定するように検索範囲とベース DN が設定されていることを確認してください。

Active Directory サーバーへの SSL アクセスを有効

このオプションは、「プライマリ Active Directory サーバー」および「セカンダリ Active Directory サーバー」フィールドで指定される Directory Server への SSL アクセスを有効にします。デフォルトでは、これは無効になっているので、Directory Server へのアクセスに SSL プロトコルは使用されません。ただし、この属性が有効になっている場合は、非 SSL サーバーにバインドできます。

SSL が有効な状態で LDAP サーバーが動作している場合は (LDAPS)、必ず適切な信頼された SSL 証明書によって Access Manager を設定し、Access Manager が LDAPS プロトコルで Directory サーバーに接続できるようにする必要があります。

認証するユーザー DN を返す

Access Manager ディレクトリが Active Directory 用に設定されたディレクトリと同じ場合、このオプションを有効にすることができます。オプションを有効にすると、このオプションによって Active Directory 認証モジュールが `userId` ではなく DN を返すことができるため、検索が不要になります。通常、認証モジュールは `userId` のみを返すため、認証サービスはローカルの Access Manager Active Directory でユーザーを検索します。外部の Active Directory ディレクトリが使用された場合、通常このオプションは有効になりません。

Active Directory サーバーのチェック間隔

この属性は Active Directory サーバーのフェイルバックに使用します。Active Directory プライマリサーバーが実行中であることを確認するまでに、スレッドが「スリープ」する時間を分単位で定義します。

ユーザー作成属性リスト

この属性は、外部 Active Directory サーバーとして Active Directory サーバーが設定されているときに、Active Directory 認証モジュールで使用されます。ローカルと外部の Directory Server との間の属性のマッピングが含まれます。この属性は次の形式です。

```
attr1|externalattr1
attr2|externalattr2
```

この属性に値が指定されると、外部属性の値が外部 Directory Server から読み込まれ、内部 Directory Server 属性に対して設定されます。コア認証モジュールのユーザープロファイル属性が「ダイナミックに作成」であり、かつユーザーがローカル Directory Server インスタンスに存在しない場合だけ、外部属性の値が内部属性に設定されます。新しく作成されるユーザーには、ユーザー作成属性リストで指定した内部属性の値と、その値にマッピングされた外部属性の値が含まれます。

認証レベル

認証レベルは認証方法ごとに個別に設定します。この値は、認証の信頼度を示します。ユーザーが認証を受けると、この値がセッションの SSO トークンに格納されます。ユーザーがアクセスしたいアプリケーションに SSO トークンが提供されると、そのアプリケーションは格納されている値を使用して、ユーザーにアクセスを許可するのに十分なレベルかどうかを判別します。SSO トークンに格納されている認証レベルが必要な最小値に満たない場合、アプリケーションはユーザーにより高い認証レベルのサービスで認証を再度受けるよう要求することがあります。デフォルト値は `0` です。

注 認証レベルの指定がない場合、SSO トークンはコア認証属性のデフォルト認証レベルで指定した値を格納します。詳細は、290 ページの「デフォルト認証レベル」を参照してください。

第 18 章

匿名認証属性

匿名認証属性は組織属性です。サービス設定の下で組織属性に適用される値は、匿名認証テンプレートのデフォルト値になります。組織にサービスを登録したあと、サービステンプレートを作成する必要があります。デフォルト値は組織の管理者が登録後に変更できます。組織属性は組織のサブツリーのエントリに継承されません。匿名認証属性は次のとおりです。

有効な匿名ユーザーリスト

このフィールドには、クレデンシャルを指定しないでログインする権限のあるユーザー ID のリストが含まれています。ユーザーのログイン名がこのリストのユーザー ID と一致すれば、アクセスが許可され、指定したユーザー ID にセッションが割り当てられます。

このリストが空の場合、ユーザーは次のデフォルトモジュールログイン URL にアクセスすると、デフォルトの匿名ユーザー名として認証を受けます。

```
protocol://server_host.server_domain:server_port/server_deploy_uri/UI/Login?module=Anonymous&org=org_name
```

このリストが空でない場合、ユーザーはデフォルトモジュールログイン URL (上記と同じ) にアクセスすると、有効な匿名ユーザー名を入力するよう求められます。

このリストが空でない場合、ユーザーは次の URL にアクセスすると、ログインページを表示せずにログインできます。

```
protocol://server_host.server_domain:server_port/server_deploy_uri/UI/Login?module=Anonymous&org=org_name&IDToken1=<有効な匿名ユーザー名>
```

デフォルトの匿名ユーザー名

このフィールドは、有効な匿名ユーザーリストが空の場合で、次のデフォルトモジュールログイン URL がアクセスされたときに、セッションを割り当てるユーザー ID を定義します。

```
protocol://server_host.server_domain:server_port/server_deploy_uri/UI/Login?module=Anonymous&org=org_name
```

デフォルト値は anonymous です。匿名ユーザーは組織にも作成する必要があります。

注　有効な匿名ユーザーリストが空でない場合、デフォルトの匿名ユーザー名に定義されたユーザーを使用すると、ログインページにアクセスせずにログインできます。そのためには、次の URL にアクセスします。

```
protocol://server_host.server_domain:server_port/server_deploy_uri/UI/Login?module=Anonymous&org=org_name&IDToken1=< デフォルトの匿名ユーザー名 >
```

大文字と小文字を区別するユーザー ID を有効

このオプションを有効にすると、ユーザー ID の大文字小文字を区別することができます。デフォルトでは、無効になっています。

認証レベル

認証レベルは認証方法ごとに個別に設定します。この値は、認証の信頼度を示します。ユーザーが認証を受けると、この値がセッションの SSO トークンに格納されます。ユーザーがアクセスしたいアプリケーションに SSO トークンが提供されると、そのアプリケーションは格納されている値を使用して、ユーザーにアクセスを許可するのに十分なレベルかどうかを判別します。SSO トークンに格納されている認証レベルが必要な最小値に満たない場合、アプリケーションはユーザーにより高い認証レベルのサービスで認証を再度受けるよう要求することがあります。デフォルト値は 0 です。

注　認証レベルの指定がない場合、SSO トークンはコア認証属性のデフォルト認証レベルで指定した値を格納します。詳細は、290 ページの「デフォルト認証レベル」を参照してください。2005Q1 のリリースでは、この機能は正常に動作しません。ただし、以前のリリースの場合は正常に動作します。

第 19 章

証明書認証属性

証明書認証属性は組織属性です。サービス設定の下で組織属性に適用される値は、証明書認証テンプレートのデフォルト値になります。組織にサービスを登録したあと、サービステンプレートを作成する必要があります。デフォルト値は組織の管理者が登録後に変更できます。組織属性は組織のサブツリーのエントリに継承されません。証明書認証属性は次のとおりです。

LDAP で証明書を照合

このオプションは、ログイン時に提出されたユーザー証明書が LDAP サーバーに格納されているかをチェックするかどうかを指定します。一致する証明書がない場合、ユーザーはアクセスを拒否されます。一致する証明書があり、かつほかの検証が必要ない場合、ユーザーはアクセスを許可されます。デフォルトでは、証明書認証サービスはユーザー証明書をチェックしません。

注	Directory Server に格納されている証明書は必ずしも有効とは限りません。272 ページの「CRL に対して証明書を照合」を参照してください。ただし、ログイン時に提出されたユーザー証明書が有効かどうかを Web コンテナでチェックすることはできます。

LDAP での証明書の検索に使用するサブジェクト DN 属性

このフィールドは、LDAP で証明書を検索するために使用する証明書の `SubjectDN` 値の属性を指定します。ユーザーエントリを一意に特定する属性でなければなりません。検索には実際の値を使用します。デフォルト値は `CN` です。

CRL に対して証明書を照合

このオプションは、ユーザー証明書と LDAP サーバーの証明書失効リスト (CRL) を比較するかどうかを指定します。CRL は、発行者の `SubjectDN` に含まれている属性名のいずれかによって特定されます。証明書が CRL に載っている場合はユーザーのアクセスが拒否され、載っていない場合は許可されます。デフォルトでは、この属性は無効になっています。

注	証明書の所有者の状態が変わってその証明書を使う権利がなくなった場合、または証明書の所有者の秘密鍵が漏洩した場合は、証明書を取り消す必要があります。

LDAP での CRL の検索に使用する発行者 DN 属性

このフィールドは、LDAP で CRL を検索するのに使用する、受信した証明書の発行者 SubjectDN 値の属性を指定します。このフィールドは、「CRL に対して証明書を照合」属性が有効になっている場合にのみ使用されます。検索には実際の値を使用します。デフォルト値は CN です。

CRL 更新用の HTTP パラメータ

このフィールドは、CRL 更新のためにサーブレットから CRL を取得するための HTTP パラメータを指定します。HTTP パラメータについては、CA の管理者に問い合わせてください。

OCSP 検証を有効

このパラメータは、対応する OCSP レスポンダと連絡することによって実行される OCSP 検証を有効にします。OCSP レスポンダは、実行時に次のように決定されます。

- com.sun.identity.authentication.ocspCheck が true で、OCSP レスポンダが com.sun.identity.authentication.ocsp.repsonder.url 属性で設定されているときは、この属性の値が OCSP レスポンダとして使用されます。
- com.sun.identity.authentication.ocspCheck が true に設定されており、この属性の値が AMConfig.properties ファイルで設定されていないときは、クライアント証明書に示されている OCSP レスポンダが OCSP レスポンダとして使用されます。

com.sun.identity.authentication.ocspCheck が false に設定されているか、com.sum.identity.authentication.ocspCheck が true に設定されており、OCSP レスポンダが見つからない場合は、OCSP 検証が実行されません。

注 OCSP 検証を有効にする前に、Access Manager マシンと OCSP レスポンダマシンの時刻ができるかぎり一致するようにしてください。また、Access Manager マシンの時刻が OCSP レスポンダの時刻より遅れないようにすることが必要です。次に例を示します。

OCSP レスポンダマシン - 12:00:00 pm

Access Manager マシン - 12:00:30 pm

証明書が格納されている LDAP サーバー

このフィールドは証明書を格納する LDAP サーバーの名前とポート番号を指定します。デフォルト値は、Access Manager のインストール時に指定したホスト名とポートです。証明書が格納されている LDAP サーバーのホスト名とポートを使用できます。形式は *hostname:port* です。

LDAP 検索開始 DN

このフィールドは、ユーザーの証明書に対する検索を開始するノードの DN を指定します。デフォルト値はありません。このフィールドは有効な DN をすべて認識します。複数のエントリの場合は、ローカルサーバー名をプレフィックスとして付ける必要があります。形式は次のとおりです。

```
servername|search dn
```

複数のエントリを指定する場合は、次のようになります。

```
servername1|search dn servername2|search dn servername3|search dn...
```

同一の検索で複数のユーザーが見つかった場合、認証は失敗します。

LDAP サーバーの主体ユーザー

このフィールドは、証明書が格納されている LDAP サーバーの主体ユーザー の DN を受け入れます。このフィールドにデフォルト値はありません。有効な DN をすべて認識します。主体ユーザーは読み取り権限を持ち、Directory Server に格納されている証明書情報を検索する必要があります。

LDAP サーバーの主体パスワード

このフィールドは、「LDAP サーバーの主体ユーザー」フィールドで指定されるユーザーに関連付けられた LDAP パスワードを保持します。このフィールドにデフォルト値はありません。指定した主体ユーザーの有効な LDAP パスワードを認識します。

注	この値は読み取り可能テキストとしてディレクトリに格納されます。

プロファイル ID のための LDAP 属性

このフィールドは、証明書と一致する Directory Server エントリの属性を指定します。この証明書の値は、正しいユーザープロファイルの識別に使用します。このフィールドにデフォルト値はありません。ユーザー ID として使用できるユーザーエントリ (cn、sn など) の有効な属性をすべて認識します。

LDAP アクセスに SSL を使用

このオプションは LDAP サーバーへのアクセスに SSL を使用するかどうかを指定します。デフォルトでは、証明書認証サービスは LDAP アクセスに SSL を使用しません。

ユーザープロファイルへのアクセスに使用する証明書フィールド

このメニューでは、一致するユーザープロファイルの検索に使用する証明書のフィールドを指定します。たとえば、`email address` を選択すると、証明書認証サービスはユーザー証明書の属性 `emailAddr` に一致するユーザープロファイルを検索します。その後、ログインするユーザーは一致したプロファイルを使用します。デフォルトのフィールドは `subject CN` です。リストは次のとおりです。

- email address
- subject CN
- subject DN
- subject UID
- other

ユーザープロファイルへのアクセスに使用するその他の証明書フィールド

ユーザープロファイルへのアクセスに使用する証明書フィールド属性の値が `other` に設定されている場合、このフィールドは、受信した証明書の `subjectDN` 値から選択する属性を指定します。認証サービスは、その属性の値に一致するユーザープロファイルを検索します。

信頼できるリモートホスト

この属性では、証明書を Access Manager に送信できると信頼されている、信頼できるホストのリストを定義します。Access Manager では、証明書がこれらのホストの 1 つから送信されたかどうかを確認する必要があります。この設定は Sun Java System Portal Server でのみ使用されます。

この属性には次の値を使用できます。

- none: この属性は無効になります。デフォルトでこの値が設定されます。
- any: 任意のクライアント IP アドレスから Portal Server Gateway スタイルの証明書の認証を受け入れます。
- IP アドレス : 受け入れる Portal Server Gateway スタイルの証明書の認証を要求する発行元の IP アドレス (Gateway の IP アドレス) を一覧表示します。属性は、組織ごとに設定できます。

SSL ポート番号

この属性は、SSL (Secure Socket Layer) 用のポート番号を指定します。現在、この属性は Gateway サーブレットでのみ使用されます。SSL ポート番号の追加や変更を行う前に、『Access Manager Developer's Guide』の第 7 章の「Policy-Based Resource Management」を参照してください。

認証レベル

認証レベルは認証方法ごとに個別に設定します。この値は、認証の信頼度を示します。ユーザーが認証を受けると、この値がセッションの SSO トークンに格納されます。ユーザーがアクセスしたいアプリケーションに SSO トークンが提供されると、そのアプリケーションは格納されている値を使用して、ユーザーにアクセスを許可するのに十分なレベルかどうかを判別します。SSO トークンに格納されている認証レベルが必要な最小値に満たない場合、アプリケーションはユーザーにより高い認証レベルのサービスで認証を再度受けるよう要求することがあります。デフォルト値は `0` です。

注　認証レベルの指定がない場合、SSO トークンはコア認証属性のデフォルト認証レベルで指定した値を格納します。詳細は、290 ページの「デフォルト認証レベル」を参照してください。2005Q1 のリリースでは、この機能は正常に動作しません。ただし、以前のリリースの場合は正常に動作します。

第 20 章

コア認証属性

コア認証サービスは、デフォルトの認証サービスとカスタム認証モジュール属性すべてのための基本サービスです。コア認証は、どの形式であれ認証を使用する組織ごとのサービスとして設定する必要があります。コア認証属性はグローバルおよび組織属性から構成されます。グローバル属性に適用される値は Sun Java System Acceess Manager 設定全体に適用され、設定済みのすべての組織に継承されます。グローバル属性の目的は Access Manager アプリケーションのカスタマイズであるため、ロールまたは組織に直接適用することはできません。サービス設定の下で組織属性に適用される値が、コア認証テンプレートのデフォルト値になります。組織にサービスを追加したあと、サービステンプレートを作成する必要があります。デフォルト値は組織の管理者が追加後に変更できます。組織属性は組織のエントリに継承されません。コア認証属性は次のように分けられます。

グローバル属性

コア認証サービスのグローバル属性には、次のものがあります。

プラグイン可能な認証モジュールクラス

このフィールドは、Access Manager プラットフォーム内で設定されるどの組織でも利用できる認証モジュールの Java クラスを指定します。デフォルトでは、これには LDAP、SafeWord、SecurID、アプリケーション、匿名、HTTP 基本、メンバーシップ、UNIX、証明書、NT、RADIUS、Microsoft Windows デスクトップ SSO などがあります。AMLoginModule SPI または JAAS LoginModule SPI を実装して、カスタム認証モジュールを作成できます。詳細は、『Access Manager Developer's Guide』を参照してください。新しいサービスを定義するには、新しい各認証サービスの完全クラス名 (パッケージ名を含む) を取得するテキスト文字列をこのフィールドに入力する必要があります。

クライアント用にサポートされている認証モジュール

この属性は、特定のクライアント用にサポートされている認証モジュールのリストを指定します。形式は次のとおりです。

```
clientType | module1,module2,module3
```

この属性は、クライアントディテクションが有効になっているときに機能します。

LDAP 接続のプールサイズ

この属性は、特定の LDAP サーバーおよびポートで使用される最大および最小の接続プールを指定します。この属性は、LDAP およびメンバーシップ認証サービス専用です。形式は次のとおりです。

```
host:port:min:max
```

注	この接続プールは、`serverconfig.xml` で構成される SDK 接続プールとは異なります。

LDAP 接続のデフォルトプールサイズ

この属性は、すべての LDAP 認証モジュール設定で使用されるデフォルトの最小および最大接続プールを指定します。ホストおよびポートのエントリが「LDAP 接続のプールサイズ」属性に存在する場合、最小および最大設定は LDAP 接続のデフォルトプールサイズから使用されません。

組織属性

コア認証サービスの組織属性は次のとおりです。

組織認証モジュール

このリストには、登録されていて組織で使用できる認証モジュールが指定されています。管理者は組織ごとに固有の認証タイプを選ぶことができます。複数の認証モジュールには柔軟性がありますが、ユーザーはログイン設定が選択した認証モジュールに適合することを確認する必要があります。デフォルトの認証は LDAP です。Access Manager に含まれている認証サービスは次のとおりです。

注 管理者は作成済み組織にコアおよび認証モジュールのテンプレートを作成、通知して、その組織が正しく機能するようにする必要があります。

ユーザープロファイル

このオプションを使用すると、ユーザープロファイルのオプションを指定することができます。

- 必須 - 認証が成功した場合に認証サービスで SSOToken を発行するには、Access Manager とともにインストールされたローカル Directory Server 内にユーザーのプロファイルが存在している必要があることを指定します。
- 「ダイナミック」- 認証が成功した場合に、ユーザープロファイルがまだ存在していないときに、認証サービスによってユーザープロファイルを作成することを指定します。作成後、SSOToken が発行されます。ユーザープロファイルは、Access Manager とともにインストールされたローカル Directory Server 内に作成されます。
- 「ユーザーエイリアスを使用してダイナミックに」- 認証が成功した場合に、認証サービスが「ユーザーエイリアスリスト」属性を使用することによって、ユーザープロファイルを作成することを指定します。
- 「無視」- 認証が成功した場合に SSOToken を発行する認証サービスに対して、ユーザープロファイルが不要であることを指定します。

管理者認証設定

編集リンクをクリックすることで、管理者専用の認証サービスを定義することができます。この属性は、管理者とエンドユーザーの認証モジュールを別々のものにする必要がある場合に使用できます。Access Manager コンソールにアクセスするときは、この属性で設定されているモジュールが使用されます。次に例を示します。

```
http://servername.port/console_deploy_uri
```

ダイナミックユーザープロファイル作成のデフォルトロール

このフィールドは、282 ページの「ユーザープロファイル」機能でダイナミック作成が選択された場合にプロファイルが作成された、新しいユーザーに割り当てるロールを指定します。デフォルト値はありません。管理者は、新しいユーザーに割り当てられるロールの DN を指定する必要があります。

注 指定するロールは、認証を構成する組織の下にあることが必要です。このロールは、Access Manager ロールか LDAP ロールにすることができますが、フィルタロールにすることはできません。

特定のサービスをユーザーに自動的に割り当てる場合は、ユーザープロファイルで「必要なサービス」属性を設定する必要があります。

持続 Cookie モードを有効

このオプションは、ユーザーがブラウザを再起動したときに認証セッションに戻れるかどうかを指定します。ユーザーセッションは、「持続 Cookie モードを有効」を有効にして保持できます。持続 Cookie モードを有効」を有効にすると、ユーザーセッションは持続 Cookie の期限が切れるか、ユーザーが意図的にログアウトするまで期限切れにはなりません。期限は「Cookie の最大持続時間」で指定します。デフォルトでは、「持続 Cookie モード」が無効になっており、認証サービスはメモリの Cookie だけを使用します。

注 持続 Cookie は、クライアントがログイン URL の `iPSPCookie=yes` パラメータを使用して明示的に要求する必要があります。

Cookie の最大持続時間

このフィールドは、持続 Cookie の期限が切れるまでの間隔を指定します。チェックボックスを選択して「持続 Cookie モードを有効」を有効にする必要があります。この間隔は、ユーザーのセッションの認証が成功したときに始まります。デフォルト値は 2147483 (秒) です。このフィールドには、0 から 2147483 までの任意の整数値を指定できます。

すべてのユーザーのピープルコンテナ

ユーザー認証が成功したあと、ユーザーのプロファイルを取得します。このフィールドの値は、プロファイルの検索先を指定します。一般に、この値はデフォルトのピープルコンテナの DN です。組織に追加されるすべてのユーザーエントリは、自動的にその組織のデフォルトピープルコンテナに追加されます。デフォルト値は ou=People です。一般に、組織名とルートサフィックスで構成されます。このフィールドは組織単位の有効な DN を取得します。

注

認証では、次の方法でユーザープロファイルを検索します。

- デフォルトのピープルコンテナの下を検索する
- 次に、デフォルトの組織の下を検索する
- さらに、エイリアス検索属性名の属性を使用して、デフォルトの組織内でユーザーを検索する

最後に SSO を検索しますが、その場合認証に使用するユーザー名がプロファイルのネーミング属性でないことがあります。たとえば、uid=jamie というプロファイルを持つユーザーが、jn10191 という SafeWord ID を使用して認証を受ける場合などです。

エイリアス検索属性名

ユーザー認証が成功したあと、ユーザーのプロファイルを取得します。このフィールドは、285 ページの「ユーザーネーミング属性」で指定する最初の LDAP 属性に対する検索で一致するユーザープロファイルを見つけられなかった場合に、次に検索する LDAP 属性を指定します。この属性は主に、認証モジュールから返されたユーザーアイデンティティがユーザーネーミング属性で指定したものと異なる場合に使用します。たとえば、RADIUS サーバーが abc1234 を返しても、ユーザー名は abc という可能性があります。この属性のデフォルト値はありません。このフィールドは有効な LDAP 属性をすべて取得します (たとえば、cn)。

ユーザーネーミング属性

ユーザー認証が成功したあと、ユーザーのプロファイルを取得します。この属性の値は、検索に使用する LDAP 属性を指定します。デフォルトでは、Access Manager はユーザーエントリが `uid` 属性によって識別されるものと想定します。Directory Server で `givenname` などの異なる属性を使用している場合は、このフィールドに属性名を指定します。

デフォルト認証ロケール

このフィールドは、認証サービスが使用するデフォルトの言語サブタイプを指定します。デフォルト値は `en_US` です。有効な言語サブタイプのリストを表 20-1 に示します。

別のロケールを使用するには、最初にそのロケールのすべての認証テンプレートを作成する必要があります。次に、それらのテンプレートの新しいディレクトリを作成する必要があります。詳細は、132 ページの「ログイン URL パラメータ」を参照してください。

表 20-1 サポートされている言語ロケール

言語タグ	言語
af	アフリカーンス語
be	ベラルーシ語
bg	ブルガリア語
ca	カタロニア語
cs	チェコ語
da	デンマーク語
de	ドイツ語
el	ギリシャ語
en	英語
es	スペイン語
eu	バスク語
fi	フィンランド語
fo	フェロー語

表 20-1　サポートされている言語ロケール (続き)

言語タグ	言語
fr	フランス語
ga	アイルランド語
gl	ガリシア語
hr	クロアチア語
hu	ハンガリー語
id	インドネシア語
is	アイスランド語
it	イタリア語
ja	日本語
ko	韓国語
nl	オランダ語
no	ノルウェー語
pl	ポーランド語
pt	ポルトガル語
ro	ルーマニア語
ru	ロシア語
sk	スロバキア語
sl	スロベニア語
sq	アルバニア語
sr	セルビア語
sv	スウェーデン語
tr	トルコ語
uk	ウクライナ語
zh	中国語

組織認証設定

この属性は、組織の認証モジュールを設定します。デフォルトの認証モジュールは LDAP です。編集リンクをクリックすると、1 つまたは複数の認証モジュールを選択できます。複数のモジュールが選択された場合、ユーザーはそれらのモジュールの連鎖に沿ってすべての認証に成功する必要があります。

ユーザーが `/server_deploy_uri/UL/Login` 形式を使って認証モジュールにアクセスするとき、この属性に設定されたモジュールが認証に使用されます。詳細は、『Access Manager Developer's Guide』を参照してください。

ログイン失敗時のロックアウトモードを有効

この機能は、最初の認証に失敗した場合に再試行を許可するかどうかを指定します。この属性を選択すると、ロックアウトが有効になります。その場合、ユーザーには 1 回だけ認証を受ける機会が与えられます。デフォルトでは、ロックアウト機能は無効になっています。この属性は、ロックアウト関連および通知関連の属性とともに機能します。

ログイン失敗時のロックアウト回数

この属性は、「ログイン失敗時のロックアウト間隔」で定義された時間内に、ユーザーが認証を試みることができる回数を定義します。この回数を超えると、ユーザーはロックアウトされます。

ログイン失敗時のロックアウト間隔

この属性は、ログインが失敗した場合の、次の再試行までの時間を分単位で定義します。ログイン失敗の後、ロックアウト間隔以内にもう一度ログインが失敗すると、ロックアウト回数が増分されます。それ以外の場合は、ロックアウト回数がリセットされます。

ロックアウト通知の送信先電子メールアドレス

この属性は、ユーザーのロックアウトが発生した場合に通知を受け取る電子メールアドレスを指定します。電子メール通知を複数のアドレスに送信する場合は、電子メールアドレスをスペースで区切ります。英語以外のロケールでの形式は次のとおりです。

```
email_address|locale|charset
```

ユーザーに警告するまでの失敗回数

この属性は、認証に失敗した場合、Access Manager がそのユーザーにロックアウトされるという警告を送信するまでに許可される認証失敗の回数を指定します。

ログイン失敗時のロックアウト持続時間

この属性を選択すると、メモリロックが有効になります。デフォルトでは、このロックアウトメカニズムにより、ロックアウト属性名で定義されているユーザープロファイルが無効になります (ログイン失敗後)。ログイン失敗時のロックアウト持続時間が 0 より大きい値に設定されている場合は、その時間だけメモリとユーザーアカウントがロックされます。

ロックアウト属性名

この属性は、ロックアウトする LDAP 属性を指定します。この属性名のロックアウトを有効にするには、ロックアウト属性値の値も変更する必要があります。デフォルトでは、Access Manager コンソールで「ロックアウト属性名」は空になっています。デフォルトの実装値は、`inetuserstatus` (LDAP 属性) と `inactive` です。「ログイン失敗時のロックアウト持続時間」が 0 に設定されている場合で、ユーザーがロックアウトされたときに適用されます。

ロックアウト属性値

この属性は、「ロックアウト属性名」で指定されている属性について、ロックアウトを有効にするかどうかを指定します。デフォルトでは、`inetuserstatus` に対して、この値は非アクティブに設定されています。

デフォルト成功ログイン URL

このフィールドには、認証成功後にユーザーをリダイレクトする URL を指定する、複数の値のリストを入力します。この属性の形式は、クライアントタイプ |URL ですが、URL の値のみを指定できます。その際、URL のタイプはデフォルトで HTML となることが前提とされています。

注 デフォルト値は /amconsole です。このリリースでは、*protocol*、*host*、*port* の値が必要なくなりました。

リモートコンソールの場合は、この属性を手動で修正し、実際のリモートコンソールホストのコンソールページに向ける必要があります。

デフォルト失敗ログイン URL

このフィールドには、認証が失敗した場合にユーザーをリダイレクトする URL を指定する、複数の値のリストを入力します。この属性の形式は、クライアントタイプ |URL ですが、URL の値のみを指定できます。その際、URL のタイプはデフォルトで HTML となることが前提とされています。

認証ポストプロセスクラス

このフィールドは、ログインの成功または失敗後に実行する、認証後プロセスをカスタマイズするための Java クラス名を指定します。次に例を示します。

```
com.abc.authentication.PostProcessClass
```

この Java クラスでは、次の Java インタフェースを実装する必要があります。

```
com.sun.identity.authentication.spi.AMPostAuthProcessInterface
```

また、このクラスが置かれている場所へのパスを、Web Server の Java Classpath 属性に追加する必要があります。

ユーザー ID 生成モードを有効

この属性は、メンバーシップ認証モジュールによって使用されます。この属性フィールドが有効になっている場合、すでにユーザー ID が存在していれば、自己登録プロセス中にメンバーシップモジュールによって特定ユーザーのユーザー ID が生成可能です。このユーザー ID は、「プラグイン可能なユーザー名ジェネレータクラス」で指定された Java クラスから生成されます。

プラグイン可能なユーザー名ジェネレータクラス

このフィールドは、「ユーザー ID 生成モードを有効」が有効になっている場合にユーザー ID を生成するために使用する Java クラス名を指定します。

デフォルト認証レベル

認証レベルの値は、認証の信頼度を示します。ユーザーが認証を受けると、この値がセッションの SSO トークンに格納されます。ユーザーがアクセスしたいアプリケーションに SSO トークンが提供されると、そのアプリケーションは格納されている値を使用してユーザーにアクセスを許可するのに十分なレベルかどうかを判別できます。SSO トークンに格納されている認証レベルが必要な最小値に満たない場合、アプリケーションはユーザーにより高い認証レベルのサービスで認証を再度受けるよう要求することがあります。

認証レベルは、組織の特定の認証テンプレート内で設定する必要があります。ここで説明するデフォルト認証レベルの値は、特定の組織の、認証テンプレートの認証レベルフィールドに認証レベルが指定されていない場合だけ適用されます。デフォルト認証レベルのデフォルト値は 0 です。この属性の値は Access Manager が使用するものではなく、どの外部アプリケーションでもその値の使用を選択すれば使用できます。2005Q1 のリリースでは、この機能は正常に動作しません。ただし、以前のリリースの場合は正常に動作します。

第 21 章

HTTP 基本認証属性

HTTP 基本認証属性は組織属性です。サービス設定の下で組織属性に適用される値は、HTTP 基本認証テンプレートのデフォルト値になります。組織にサービスを登録したあと、サービステンプレートを作成する必要があります。デフォルト値は組織の管理者が登録後に変更できます。組織属性は組織のエントリに継承されません。

HTTP 基本認証属性は次のとおりです。

認証レベル

認証レベルは認証方法ごとに個別に設定します。この値は、認証の信頼度を示します。ユーザーが認証を受けると、この値がセッションの SSO トークンに格納されます。ユーザーがアクセスしたいアプリケーションに SSO トークンが提供されると、そのアプリケーションは格納されている値を使用して、ユーザーにアクセスを許可するのに十分なレベルかどうかを判別します。SSO トークンに格納されている認証レベルが必要な最小値に満たない場合、アプリケーションはユーザーにより高い認証レベルのサービスで認証を再度受けるよう要求することがあります。デフォルト値は 0 です。

注 認証レベルの指定がない場合、SSO トークンはコア認証属性のデフォルト認証レベルで指定した値を格納します。詳細は、290 ページの「デフォルト認証レベル」を参照してください。2005Q1 のリリースでは、この機能は正常に動作しません。ただし、以前のリリースの場合は正常に動作します。

第 22 章

JDBC 認証属性

JDBC (Java Database Connectivity) 認証属性は組織属性です。サービス設定の下で組織属性に適用される値は、JDBC 認証テンプレートのデフォルト値になります。組織にサービスを登録したあと、サービステンプレートを作成する必要があります。デフォルト値は組織の管理者が登録後に変更できます。組織属性は組織のエントリに継承されません。JDBC 認証属性は次のとおりです。

接続タイプ

このフィールドは、JNDI (Java Naming and Directory Interface) 接続プールまたは JDBC ドライバを使用した、SQL データベースへの接続タイプを指定します。オプションは次のとおりです。

- JNDI を介して接続プールを取得する
- 非持続 JDBC 接続

JNDI 接続プールは、背後の Web コンテナの設定を利用します。

接続プールの JNDI 名

「接続タイプ」で JNDI を選択した場合は、このフィールドで接続プール名を指定します。JDBC 認証では Web コンテナで提供される JNDI 接続プールを使用するので、JNDI 接続プールのセットアップがほかの Web コンテナ間との一貫性を失う可能性があります。

次の例は、Web Server および MySQL 4.0 の接続プールのセットアップ方法を示しています。

1. Web Server コンソールで、次の属性を使用して JDBC 接続プールを作成します。

   poolName: `samplePool`

   DataSource Classname: `com.mysql.jdbc.jdbc2.optional.MysqlDatacSource`

   serverName: MySQL サーバーのサーバー名

   port: MySQL サーバーが稼働するポート番号

   user: データベースのユーザー名

   password: ユーザーのパスワード

   databaseName: データベース名

注 次のステップで説明する DataSource クラスと JDBC ドライバクラスが記述されている jar ファイルを、アプリケーションクラスのパスに追加してください。

---

2. JDBC リソースを設定します。Web Server コンソールで、次の属性を使用して JDBC リソースを作成します。

   JNDI name: `jdbc/samplePool`

   Pool name: `samplePool`

Data Resource Enabled: on

3. アプリケーションの sun-web.xml ファイルに次の行を追加します。

```
<resource-ref>

   <res-ref-name>jdbc/mySQL</res-ref-name>

   <jndi-name>jdbc/samplePool</jndi-name>

</resource-ref>
```

4. アプリケーションの web.xml ファイルに次の行を追加します。

```
<resource-ref>

  <description>mySQL Database</description>

  <res-ref-name>jdbc/mySQL</res-ref-name>

  <res-type>javax.sql.DataSource</res-type>

  <res-auth>Container</res-auth>

</resource-ref>
```

この設定作業が完了すると、この属性の値は次のようになります。

```
java:comp/env/jdbc/mySQL
```

JDBC ドライバ

「接続タイプ」で JNDI を選択した場合は、このフィールドで SQL データベースによって提供される JDBC ドライバを指定します。次に例を示します。

```
com.mysql.jdbc.Driver
```

JDBC URL

「接続タイプ」で JDBC を選択した場合は、このフィールドでデータベース URL を指定します。たとえば、mySQL の URL は次のようになります。

```
jdbc.mysql:// ホスト名 : ポート / データベース名
```

データベースにする接続ユーザー

このフィールドは、JDBC 接続でデータベースへの接続元となるユーザー名を指定します。

データベースへ接続するためのパスワード

このフィールドは、「データベースにする接続ユーザー」で指定したユーザーのパスワードを定義します。

データベースへ接続するためのパスワード (確認)

パスワードを確認します。

データベース内のパスワードカラム

このフィールドは、SQL データベースのパスワード列名を指定します。

準備されているステートメント

このフィールドは、ログインしようとするユーザーのパスワードを取得する SQL 文を指定します。次に例を示します。

```
select Password from Employees where USERNAME = ?
```

パスワード構文を変換するためのクラス

この属性は、パスワードの比較のためにデータベースから取得したパスワードをユーザー入力の形式に変換するクラス名を指定します。このクラスでは `JDBCPasswordSyntaxTransform` インタフェースを実装する必要があります。

認証レベル

認証レベルは認証方法ごとに個別に設定します。この値は、認証の信頼度を示します。ユーザーが認証を受けると、この値がセッションの SSO トークンに格納されます。ユーザーがアクセスしたいアプリケーションに SSO トークンが提供されると、そのアプリケーションは格納されている値を使用して、ユーザーにアクセスを許可するのに十分なレベルかどうかを判別します。SSO トークンに格納されている認証レベルが必要な最小値に満たない場合、アプリケーションはユーザーにより高い認証レベルのサービスで認証を再度受けるよう要求することがあります。デフォルト値は `0` です。

第 23 章

LDAP 認証属性

LDAP 認証属性は組織属性です。サービス設定の下で組織属性に適用される値は、LDAP 認証テンプレートのデフォルト値になります。組織にサービスを登録したあと、サービステンプレートを作成する必要があります。デフォルト値は組織の管理者が登録後に変更できます。組織属性は組織のエントリに継承されません。LDAP 認証属性は次のとおりです。

プライマリ LDAP サーバー

このフィールドは、Access Manager のインストール時に指定するプライマリ LDAP サーバーのホスト名およびポート番号を指定します。これは、LDAP 認証で最初に通信するサーバーです。形式は `hostname:port` です。ポート番号がないときは、`389` と想定します。

複数のドメインに Access Manager が配備されている場合は、Access Manager および Directory Server の個々のインスタンス間の通信リンクを、次の形式で指定できます。複数のエントリを指定する場合は、エントリにローカルサーバー名をプレフィックスとして付ける必要があります。

```
local_servername|server:port local_servername2|server2:port2 ...
```

たとえば、異なる場所に配備された 2 つの Access Manager インスタンス (`L1-machine1-IS` および `L2- machine2-IS`) が、それぞれ別の Directory Server インスタンス (`L1-machine1-DS` および `L2-machine2-DS`) と通信する場合は、次のように指定できます。

```
L1-machine1-IS.example.com|L1-machine1-DS.example.com:389
L2-machine2-IS.example.com|L2-machine2-DS.example.com:389
```

セカンダリ LDAP サーバー

このフィールドは、Access Manager プラットフォームが利用できるセカンダリ LDAP サーバーのホスト名およびポート番号を指定します。プライマリ LDAP サーバーが認証要求に応答しない場合は、このサーバーと通信します。プライマリサーバーが起動すると、Access Manager はプライマリサーバーに戻ります。この形式も `hostname:port` です。複数のエントリの場合は、ローカルサーバー名をプレフィックスとして付ける必要があります。

警告 Access Manager を使用する企業からは遠隔地にある Directory Server からユーザーを認証する場合は、プライマリとセカンダリ両方の LDAP サーバーポートに値があることが重要です。1 つの Directory Server の場所の値を両方のフィールドに使用できます。

ユーザー検索の開始 DN

このフィールドは、ユーザーの検索を開始するノードの DN を指定します。性能上の理由から、この DN はできるだけ特定のものにしてください。デフォルト値は、ディレクトリツリーのルートです。すべての有効な DN が認識されます。「検索範囲」属性で「オブジェクト」を選択する場合は、DN にはプロファイルが存在するレベルの 1 レベル上を指定する必要があります。

複数のエントリの場合は、ローカルサーバー名をプレフィックスとして付ける必要があります。形式は次のとおりです。

```
servername|search dn
```

複数のエントリを指定する場合は、次のようになります。

```
servername1|search dn servername2|search dn servername3|search dn...
```

同一の検索で複数のユーザーが見つかった場合、認証は失敗します。

root ユーザーバインド DN

このフィールドは、「プライマリ LDAP サーバー」フィールドで指定した Directory Server に管理者としてバインドするのに使用するユーザーの DN を指定します。ユーザーログイン ID に基づく、一致するユーザー DN を検索するためには、認証サービスをこの DN にバインドする必要があります。デフォルト値は `amldapuser` です。すべての有効な DN が認識されます。

パスワードが間違っているとロックアウトされるので、ログアウトする前にパスワードが正しいことを確認してください。ロックアウトされた場合は、`AMConfig.Properties` ファイル内の `com.iplanet.authentication.super.user` プロパティで指定されているスーパーユーザー DN を使ってログインできます。デフォルトでは、これはログインに通常使用する amAdmin アカウントですが、完全な DN を使用する必要があります。次に例を示します。

```
uid_amAdmin,ou=People,AccessManager-base
```

root ユーザーバインドパスワード

このフィールドは、「root ユーザーバインド DN」フィールドで指定される管理者プロファイルのパスワードを指定します。デフォルト値はありません。管理者の有効な LDAP パスワードだけが認識されます。

root ユーザーバインドパスワード (確認)

パスワードを確認します。

ユーザープロファイルの取得に使用する LDAP 属性

ユーザー認証が成功したあと、ユーザーのプロファイルを取得します。この属性の値を使用して検索を実行します。このフィールドは、使用する LDAP 属性を指定します。デフォルトでは、Access Manager はユーザーエントリが `uid` 属性によって識別されるものと想定します。Directory Server で `givenname` などの異なる属性を使用している場合は、このフィールドに属性名を指定します。

注	ユーザー検索フィルタは、検索フィルタ属性とユーザープロファイルの取得に使用する LDAP 属性の組み合わせです。

認証するユーザーの検索に使用する LDAP 属性

このフィールドには、認証対象ユーザーの検索フィルタを設定するために使用する属性を一覧表示します。これによりユーザーは、ユーザーのエントリにある複数の属性によって認証を受けることができます。たとえば、このフィールドが `uid`、`employeenumber`、および `mail` に設定されている場合、ユーザーはこれらの名前のどれを使用しても認証を受けることができます。

ユーザー検索フィルタ

このフィールドは、「ユーザー検索の開始 DN」フィールドの下でユーザーの検索に使用する属性を指定します。これはユーザーエントリネーミング属性とともに機能します。デフォルト値はありません。有効なユーザーエントリ属性はすべて認識されます。

検索範囲

このメニューは、一致するユーザープロファイルの検索対象となる、Directory Server 内の階層の数を示します。検索は、301 ページの「ユーザー検索の開始 DN」属性で指定されるノードから開始します。デフォルト値はサブツリーです。次のリスト項目から 1 つ選択できます。

- オブジェクト - 指定したノードだけを検索
- 1 レベル - 指定したノードのレベルとその 1 つ下のレベルで検索
- サブツリー - 指定したノードとその下のノードのエントリすべてを検索

警告 サブ組織のユーザーは、サブ組織の状態が非アクティブであってもログインする可能性があります。これを防ぐには、ユーザーの所属する特定の組織を指定するように検索範囲とベース DN が設定されていることを確認してください。

LDAP サーバーへの SSL アクセスを有効

このオプションは、プライマリおよびセカンダリ LDAP サーバーとポートのフィールドで指定される Directory Server への SSL アクセスを有効にします。デフォルトでは、これは無効になっているので、Directory Server へのアクセスに SSL プロトコルは使用されません。ただし、この属性が有効になっている場合は、非 SSL サーバーにバインドできます。

SSL が有効な状態で LDAP サーバーが動作している場合は (LDAPS)、必ず適切な信頼された SSL 証明書によって Access Manager を設定し、Access Manager が LDAPS プロトコルで Directory サーバーに接続できるようにする必要があります。

認証するユーザー DN を返す

Access Manager ディレクトリが LDAP 用に設定されたディレクトリと同じ場合、このオプションを有効にすることができます。オプションを有効にすると、このオプションによって LDAP 認証モジュールが `userId` ではなく DN を返すことができるため、検索が不要になります。通常、認証モジュールは `userId` のみを返すため、認証サービスはローカルの Access Manager LDAP でユーザーを検索します。外部の LDAP ディレクトリが使用された場合、通常このオプションは有効になりません。

LDAP サーバーのチェック間隔

この属性は LDAP サーバーのフェイルバックに使用します。LDAP プライマリサーバーが実行中であることを確認する前にスレッドが「スリープ」するまでの分単位の時間を定義します。

ユーザー作成属性リスト

この属性は、外部 LDAP サーバーとして LDAP サーバーが設定されているときに、LDAP 認証モジュールで使用されます。ローカルと外部の Directory Server との間の属性のマッピングが含まれます。この属性は次の形式です。

```
attr1|externalattr1
attr2|externalattr2
```

この属性に値が指定されると、外部属性の値が外部 Directory Server から読み込まれ、内部 Directory Server 属性に対して設定されます。コア認証モジュールのユーザープロファイル属性が「ダイナミックに作成」であり、かつユーザーがローカル Directory Server インスタンスに存在しない場合だけ、外部属性の値が内部属性に設定されます。新しく作成されるユーザーには、ユーザー作成属性リストで指定した内部属性の値と、その値にマッピングされた外部属性の値が含まれます。

認証レベル

認証レベルは認証方法ごとに個別に設定します。この値は、認証の信頼度を示します。ユーザーが認証を受けると、この値がセッションの SSO トークンに格納されます。ユーザーがアクセスしたいアプリケーションに SSO トークンが提供されると、そのアプリケーションは格納されている値を使用して、ユーザーにアクセスを許可するのに十分なレベルかどうかを判別します。SSO トークンに格納されている認証レベルが必要な最小値に満たない場合、アプリケーションはユーザーにより高い認証レベルのサービスで認証を再度受けるよう要求することがあります。デフォルト値は 0 です。

注 認証レベルの指定がない場合、SSO トークンはコア認証属性のデフォルト認証レベルで指定した値を格納します。詳細は、290 ページの「デフォルト認証レベル」を参照してください。2005Q1 のリリースでは、この機能は正常に動作しません。ただし、以前のリリースの場合は正常に動作します。

第 24 章

メンバーシップ認証属性

メンバーシップ認証属性は組織属性です。サービス設定の下で組織属性に適用される値は、メンバーシップ認証属性テンプレートのデフォルト値になります。組織にサービスを登録したあと、サービステンプレートを作成する必要があります。デフォルト値は組織の管理者が登録後に変更できます。組織属性は組織のサブツリーのエントリに継承されません。メンバーシップ認証属性は次のとおりです。

パスワードの最少文字数

このフィールドは、自己登録時に設定するパスワードに必要な最少文字数を指定します。デフォルト値は 8 です。

この値を変更すると、次のファイルの登録およびエラーテキストでも値が変更されます。

AcessManager-base/locale/amAuthMembership.properties (PasswdMinChars エントリ)

デフォルトユーザーロール

このフィールドは、自己登録で作成されたプロファイルを持つ新しいユーザーに割り当てるロールを指定します。デフォルト値はありません。管理者は、新しいユーザーに割り当てられるロールの DN を指定する必要があります。

注 指定するロールは、認証を構成する組織の下にあることが必要です。自己登録時には、そのユーザーに割り当て可能なロールだけが追加されます。ほかの DN はすべて無視されます。ロールは Access Manager ロールまたは LDAP ロールにすることができますが、フィルタロールは受け付けられません。

登録後のユーザー状態

このメニューは、自己登録したユーザーにサービスをすぐに利用できるようにするかどうかを指定します。デフォルト値は「アクティブ」なので、新しいユーザーはサービスを利用できます。管理者が「非アクティブ」を選択すると、新しいユーザーはサービスを利用できません。

プライマリ LDAP サーバー

このフィールドは、Access Manager のインストール時に指定するプライマリ LDAP サーバーのホスト名およびポート番号を指定します。これは、LDAP 認証で最初に通信するサーバーです。形式は `hostname:port` です。ポート番号がないときは、`389` と想定します。

複数のドメインに Access Manager が配備されている場合は、Access Manager および Directory Server の個々のインスタンス間の通信リンクを、次の形式で指定できます。複数のエントリを指定する場合は、エントリにローカルサーバー名をプレフィックスとして付ける必要があります。

```
local_servername|server:port local_servername2|server:port ...
```

たとえば、異なる場所に配備された 2 つの Access Manager (`L1-machine1-IS` および `L2- machine2-IS`) が、それぞれ別の Access Manager インスタンス (`L1-machine1-DS` および `L2-machine2-DS`) と通信する場合は、次のように指定できます。

```
L1-machine1-IS.example.com|L1-machine1-DS.example.com:389
L2-machine2-IS.example.com|L2-machine2-DS.example.com:389
```

セカンダリ LDAP サーバー

このフィールドは、Access Manager プラットフォームが利用できるセカンダリ LDAP サーバーのホスト名およびポート番号を指定します。プライマリ LDAP サーバーが認証要求に応答しない場合は、このサーバーと通信します。プライマリサーバーが起動すると、Access Manager はプライマリサーバーに戻ります。この形式も `hostname:port` です。複数のエントリの場合は、ローカルサーバー名をプレフィックスとして付ける必要があります。

警告 Access Manager を使用する企業からは遠隔地にある Directory Server からユーザーを認証する場合は、プライマリとセカンダリ両方の LDAP サーバーポートに値があることが重要です。1 つの Directory Server の場所の値を両方のフィールドに使用できます。

ユーザー検索の開始 DN

このフィールドは、ユーザーの検索を開始するノードの DN を指定します。性能上の理由から、この DN はできるだけ特定のものにしてください。デフォルト値は、ディレクトリツリーのルートです。すべての有効な DN が認識されます。「検索範囲」属性で「オブジェクト」を選択する場合は、DN にはプロファイルが存在するレベルの 1 レベル上を指定する必要があります。

複数のエントリを使用する場合は、エントリにローカルサーバー名をプレフィックスとして付ける必要があります。形式は次のとおりです。

```
servername|search dn
```

複数のエントリを指定する場合は、次のようになります。

```
servername1|search dn servername2|search dn servername3|search dn...
```

同一の検索で複数のユーザーが見つかった場合、認証は失敗します。

root ユーザーバインド DN

このフィールドは、「プライマリ LDAP サーバー」フィールドで指定した Directory Server に管理者としてバインドするのに使用するユーザーの DN を指定します。ユーザーログイン ID に基づく、一致するユーザー DN を検索するためには、認証サービスをこの DN にバインドする必要があります。デフォルトは `amldapuser` です。すべての有効な DN が認識されます。

root ユーザーバインドパスワード

このフィールドは、「root ユーザーバインド DN」フィールドで指定される管理者プロファイルのパスワードを指定します。デフォルト値はありません。管理者の有効な LDAP パスワードだけが認識されます。

root ユーザーバインドパスワード (確認)

パスワードを確認します。

ユーザープロファイルの取得に使用する LDAP 属性

このフィールドは、ユーザーエントリのネーミング規則に使用する属性を指定します。デフォルトでは、Access Manager はユーザーエントリが `uid` 属性によって識別されるものと想定します。Directory Server で `givenname` などの異なる属性を使用している場合は、このフィールドに属性名を指定します。

認証するユーザーの検索に使用する LDAP 属性

このフィールドには、認証対象ユーザーの検索フィルタを設定するために使用する属性を一覧表示します。これによりユーザーは、ユーザーのエントリにある複数の属性によって認証を受けることができます。たとえば、このフィールドが `uid`、`employeenumber`、および `mail` に設定されている場合、ユーザーはこれらの名前のどれを使用しても認証を受けることができます。

ユーザー検索フィルタ

このフィールドは、「ユーザー検索の開始 DN」フィールドの下でユーザーの検索に使用する属性を指定します。これはユーザーネーミング属性とともに機能します。デフォルト値はありません。有効なユーザーエントリ属性はすべて認識されます。

検索範囲

このメニューは、一致するユーザープロファイルの検索対象となる、Directory Server 内の階層の数を示します。検索は、307 ページの「ユーザー検索の開始 DN」属性で指定されるノードから開始します。デフォルト値はサブツリーです。次のリスト項目から 1 つ選択できます。

- オブジェクト - 指定したノードだけを検索
- 1 レベル - 指定したノードのレベルとその 1 つ下のレベルで検索
- サブツリー - 指定したノードとその下のノードのエントリすべてを検索

LDAP サーバーへの SSL アクセスを有効

このオプションは、プライマリおよびセカンダリ LDAP サーバーとポートのフィールドで指定される Directory Server への SSL アクセスを有効にします。デフォルトでは、このチェックボックスは選択されていないので、Directory Server へのアクセスに SSL プロトコルは使用されません。

SSL が有効な状態で LDAP サーバーが動作している場合は (LDAPS)、必ず適切な信頼された SSL 証明書によって Access Manager を設定し、Access Manager が LDAPS プロトコルで Directory サーバーに接続できるようにする必要があります。

認証するユーザー DN を返す

Access Manager ディレクトリが LDAP 用に設定されたディレクトリと同じ場合、このオプションを有効にすることができます。オプションを有効にすると、このオプションによって LDAP 認証モジュールが `userId` ではなく DN を返すことができるため、検索が不要になります。通常、認証モジュールは `userId` のみを返すため、認証サービスはローカルの Access Manager LDAP でユーザーを検索します。外部の LDAP ディレクトリが使用された場合、通常このオプションは有効になりません。

認証レベル

認証レベルは認証方法ごとに個別に設定します。この値は、認証の信頼度を示します。ユーザーが認証を受けると、この値がセッションの SSO トークンに格納されます。ユーザーがアクセスしたいアプリケーションに SSO トークンが提供されると、そのアプリケーションは格納されている値を使用して、ユーザーにアクセスを許可するのに十分なレベルかどうかを判別します。SSO トークンに格納されている認証レベルが必要な最小値に満たない場合、アプリケーションはユーザーにより高い認証レベルのサービスで認証を再度受けるよう要求することがあります。デフォルト値は `0` です。

注 認証レベルの指定がない場合、SSO トークンはコア認証属性のデフォルト認証レベルで指定した値を格納します。詳細は、290 ページの「デフォルト認証レベル」を参照してください。2005Q1 のリリースでは、この機能は正常に動作しません。ただし、以前のリリースの場合は正常に動作します。

第 25 章

MSISDN 認証属性

MSISDN 認証属性は組織属性です。サービス設定の下で組織属性に適用される値は、MSISDN 認証テンプレートのデフォルト値になります。組織にサービスを登録したあと、サービステンプレートを作成する必要があります。デフォルト値は組織の管理者が登録後に変更できます。組織属性は組織のエントリに継承されません。MSISDN 認証属性は次のとおりです。

信頼できるゲートウェイの IP アドレス

この属性は、MSISDN モジュールにアクセス可能な信頼できるクライアントの IP アドレスのリストを指定します。すべてのクライアントの IP アドレスを設定して MSISDN モジュールへのアクセスを許可することができます。設定するには、入力フィールドにたとえば「123.456.123.111」のように入力して、「追加」をクリックします。デフォルトでは、リストは空になっています。この属性を空にすると、すべてのクライアントにアクセスが許可されます。「none」を指定すると、どのクライアントも許可されません。

MSISDN 番号引数

このフィールドは、要求ヘッダーまたは Cookie ヘッダーのどのパラメータから MSISDN 番号を検索するかを識別する、パラメータ名のリストを指定します。たとえば、x-Cookie-Param、AM_NUMBER、および COOKIE-ID を定義する場合、MSISDN 認証サービスは、それらのパラメータから MSISDN 番号を検索します。

LDAP サーバーおよびポート

このフィールドは、MSISDN 番号を使用してユーザーの検索が実行される Directory Server のホスト名とポート番号を指定します。形式は hostname:port です。ポート番号がないときは、389 と想定します。

複数のドメインに Access Manager が配備されている場合は、Access Manager および Directory Server の個々のインスタンス間の通信リンクを、次の形式で指定できます。複数のエントリを指定する場合は、エントリにローカルサーバー名をプレフィックスとして付ける必要があります。

```
local_servername|server:port local_servername2|server2:port2 ...
```

たとえば、異なる場所に配備された 2 つの Access Manager インスタンス (L1-machine1-IS および L2- machine2-IS) が、それぞれ別の Directory Server インスタンス (L1-machine1-DS および L2-machine2-DS) と通信する場合は、次のように指定できます。

```
L1-machine1-IS.example.com|L1-machine1-DS.example.com:389
L2-machine2-IS.example.com|L2-machine2-DS.example.com:389
```

LDAP 検索の開始 DN

このフィールドは、ユーザーの MSISDN 番号に対する検索を開始するノードの DN を指定します。デフォルト値はありません。このフィールドは有効な DN をすべて認識します。複数のエントリの場合は、ローカルサーバー名をプレフィックスとして付ける必要があります。形式は次のとおりです。

```
servername|search dn
```

複数のエントリを指定する場合は、次のようになります。

```
servername1|search dn servername2|search dn servername3|search dn...
```

同一の検索で複数のユーザーが見つかった場合、認証は失敗します。

LDAP の検索に使用する属性

特定のユーザーの検索に使用するための、MSISDN 番号を含んだ、ユーザーのプロファイルにある属性の名前を指定します。デフォルト値は sunIdentityMSISDNNumber です。ユーザーのプロファイルの別の属性に同じ MSISDN 番号があることが確実な場合以外は、この値を変更しないでください。

LDAP サーバーの主体ユーザー

この属性は、Directory Server での MSISDN 検索を許可する LDAP バインド DN を指定します。デフォルトのバインド DN は、`cn=amldapuser,ou=DSAME Users,dc=sun,dc=com` です。

LDAP サーバーの主体パスワード

この属性は、「LDAP サーバーの主体ユーザー」で定義されるバインド DN の LDAP バインドパスワードを指定します。

LDAP サーバーの主体パスワード (確認)

パスワードを確認します。

LDAP アクセス用に SSL をオン

このオプションは、LDAP サーバーとポート属性で指定される Directory Server への SSL アクセスを有効にします。デフォルトでは、これは無効になっているので、Directory Server へのアクセスに SSL プロトコルは使用されません。ただし、この属性が有効になっている場合は、非 SSL サーバーにバインドできます。

MSISDN ヘッダー検索属性

この属性は、要求から MSISDN 番号を検索する場合に使用するヘッダーを指定します。サポートされる値は次のとおりです。

- `SearchCookieHeader` - Cookie 内の検索を実行します。
- `SearchRequestHeader` - 要求ヘッダー内の検索を実行します。
- `SearchRequestParameter` - 要求パラメータ内の検索を実行します。

デフォルトでは、すべてのオプションが選択されています。

認証レベル

認証レベルは認証方法ごとに個別に設定します。この値は、認証の信頼度を示します。ユーザーが認証を受けると、この値がセッションの SSO トークンに格納されます。ユーザーがアクセスしたいアプリケーションに SSO トークンが提供されると、そのアプリケーションは格納されている値を使用して、ユーザーにアクセスを許可するのに十分なレベルかどうかを判別します。SSO トークンに格納されている認証レベルが必要な最小値に満たない場合、アプリケーションはユーザーにより高い認証レベルのサービスで認証を再度受けるよう要求することがあります。デフォルト値は 0 です。

注 認証レベルの指定がない場合、SSO トークンはコア認証属性のデフォルト認証レベルで指定した値を格納します。詳細は、290 ページの「デフォルト認証レベル」を参照してください。

第 26 章

Microsoft Windows NT 認証属性

Microsoft Windows NT 認証属性は組織属性です。サービス設定の下で組織属性に適用される値は、Microsoft Windows NT 認証テンプレートのデフォルト値になります。組織にサービスを登録したあと、サービステンプレートを作成する必要があります。デフォルト値は組織の管理者が登録後に変更できます。組織属性は組織のサブツリーのエントリに継承されません。

Microsoft Windows NT 認証モジュールをアクティブにするには、Samba Client 2.2.2 をダウンロードして次のディレクトリにインストールする必要があります。

```
AcessManager-base/SUNWam/bin
```

Samba Client は、Microsoft Windows マシンと UNIX マシンを共存させるためのファイルサーバー兼プリントサーバーで、専用の Microsoft Windows NT/2000 Server を必要としません。詳細とダウンロードについては、http://wwws.sun.com/software/download/products/3e3af224.html を参照してください。

Red Hat Linux とともに出荷される Samba クライアントは、次のディレクトリに置かれています。

```
/usr/bin
```

Linux 用 Microsoft Windows NT 認証サービスを使って認証を行うためには、クライアントのバイナリを Access Manager の次のディレクトリにコピーします。

```
AccessManager-base/identity/bin
```

Microsoft Windows NT 認証属性は次のとおりです。

Microsoft Windows NT 認証ドメイン

この属性は、ユーザーが属するドメイン名を定義します。

Microsoft Windows NT 認証ホスト

この属性は、Microsoft Windows NT 認証のホスト名を定義します。ホスト名は、完全修飾ドメイン名 (FQDN) ではなく、NetBIOS 名にする必要があります。デフォルトでは、FQDN の先頭部は NetBIOS 名です。

DHCP (ダイナミックホスト構成プロトコル) を使用している場合、Microsoft Windows 2000 マシンの HOSTS ファイルに適切なエントリを設定します。

名前解決は、NetBIOS 名に基づいて行われます。サブネット上で NetBIOS 名の名前解決をするサーバーがない場合、マッピングはハードコードされている必要があります。

たとえば、ホスト名は `example1.company1.com` ではなく `example1` とする必要があります。

Microsoft Windows NT Samba 設定ファイル名

この属性は、Samba 設定ファイル名を定義し、`smbclient` コマンドの `-s` オプションをサポートします。値は、Samba 設定ファイルが格納されている場所へのフルディレクトリパスにする必要があります。次に例を示します。

```
/etc/opt/SUNWam/config/smb.conf
```

認証レベル

認証レベルは認証方法ごとに個別に設定します。この値は、認証の信頼度を示します。ユーザーが認証を受けると、この値がセッションの SSO トークンに格納されます。ユーザーがアクセスしたいアプリケーションに SSO トークンが提供されると、そのアプリケーションは格納されている値を使用して、ユーザーにアクセスを許可するのに十分なレベルかどうかを判別します。SSO トークンに格納されている認証レベルが必要な最小値に満たない場合、アプリケーションはユーザーにより高い認証レベルのサービスで認証を再度受けるよう要求することがあります。デフォルト値は `0` です。

注	認証レベルの指定がない場合、SSO トークンはコア認証属性のデフォルト認証レベルで指定した値を格納します。詳細は、290 ページの「デフォルト認証レベル」を参照してください。2005Q1 のリリースでは、この機能は正常に動作しません。ただし、以前のリリースの場合は正常に動作します。

第 27 章

RADIUS 認証属性

RADIUS 認証属性は組織属性です。サービス設定の下で組織属性に適用される値は、RADIUS 認証テンプレートのデフォルト値になります。組織にサービスを登録したあと、サービステンプレートを作成する必要があります。デフォルト値は組織の管理者が登録後に変更できます。組織属性は組織のエントリに継承されません。RADIUS 認証属性は次のとおりです。

RADIUS サーバー 1

このフィールドは、プライマリ RADIUS サーバーの IP アドレスまたは完全修飾ホスト名を表示します。デフォルト IP アドレスは `127.0.0.1` です。このフィールドは有効な IP アドレスまたはホスト名をすべて認識します。複数のエントリの場合は、次の構文に示されるように、ローカルサーバー名をプレフィックスとして付ける必要があります。

```
local_servername|ip_address local_servername2|ip_adress ...
```

RADIUS サーバー 2

このフィールドは、セカンダリ RADIUS サーバーの IP アドレスまたは完全修飾ドメイン名 (FQDN) を表示します。これはフェイルオーバーサーバーで、プライマリサーバーが通信できない場合に通信するサーバーです。デフォルト IP アドレスは `127.0.0.1` です。複数のエントリの場合は、次の構文に示されるように、ローカルサーバー名をプレフィックスとして付ける必要があります。

local_servername | ip_address local_servername2 | ip_adress ...

RADIUS 共有シークレット

このフィールドは RADIUS 認証の共有シークレットを保持します。共有シークレットは、注意深く選んだパスワードと同じレベルにする必要があります。このフィールドのデフォルト値はありません。

RADIUS 共有シークレット (確認)

RADIUS 認証の共有シークレットを確認します。

RADIUS サーバーのポート

このフィールドは、RADIUS サーバーが待機するポートを指定します。デフォルト値は `1645` です。

タイムアウト

このフィールドは、RADIUS サーバーがタイムアウトするまで応答を待つ時間間隔を秒単位で指定します。デフォルト値は 3 秒です。どんな秒数のタイムアウトを指定しても認識されます。

認証レベル

認証レベルは認証方法ごとに個別に設定します。この値は、認証の信頼度を示します。ユーザーが認証を受けると、この値がセッションの SSO トークンに格納されます。ユーザーがアクセスしたいアプリケーションに SSO トークンが提供されると、そのアプリケーションは格納されている値を使用して、ユーザーにアクセスを許可するのに十分なレベルかどうかを判別します。SSO トークンに格納されている認証レベルが必要な最小値に満たない場合、アプリケーションはユーザーにより高い認証レベルのサービスで認証を再度受けるよう要求することがあります。デフォルト値は 0 です。2005Q1 のリリースでは、この機能は正常に動作しません。ただし、以前のリリースの場合は正常に動作します。

第 28 章

SafeWord 認証属性

SafeWord 認証属性は組織属性です。サービス設定の下で組織属性に適用される値は、SafeWord 認証テンプレートのデフォルト値になります。組織にサービスを登録したあと、サービステンプレートを作成する必要があります。デフォルト値は組織の管理者が登録後に変更できます。組織属性は組織のサブツリーのエントリに継承されません。

このサービスでは、Secure Computing の SafeWord または SafeWord PremierAccess 認証 サーバーを使用して、ユーザーを認証できます。SafeWord 認証属性は次のとおりです。

SafeWord サーバー

このフィールドは、SafeWord または SafeWord PremierAccess サーバー名とポートを指定します。SafeWord サーバーのデフォルトとしてポート 7482 が設定されます。SafeWord PremierAccess サーバーのデフォルトのポート番号は 5030 です。

SafeWord サーバー検証ファイルのディレクトリ

このフィールドは、SafeWord クライアントライブラリがその検証ファイルを置くディレクトリを指定します。デフォルトは次のとおりです。

```
/var/opt/SUNWam/auth/safeword/serverVerification
```

このフィールドに異なるディレクトリを指定する場合は、SafeWord 認証を試みる前にそのディレクトリが存在する必要があります。

SafeWord ログを有効

この属性を選択すると、SafeWord のログが有効になります。デフォルトでは、SafeWord ログが有効になっています。

SafeWord ログレベル

このフィールドは SafeWord ログレベルを指定します。プルダウンメニューでレベルを選択します。選択可能なレベルは、「デバッグ」、「エラー」、「情報」、および「なし」です。

SafeWord ログファイル

この属性は、SafeWord クライアントログのディレクトリパスとログファイル名を指定します。デフォルトのパスは次のとおりです。

```
/var/opt/SUNWam/auth/safeword/safe.log
```

異なるパスまたはファイル名を指定する場合は、SafeWord 認証を試みる前にそれらが存在している必要があります。

SafeWord 認証に複数の組織が構成され別々の SafeWord サーバーが使用されている場合、別々のパスを指定する必要があります。そうしないと、SafeWord 認証が行われる最初の組織だけが認証されます。同様に、組織が SafeWord サーバーを変更した場合、新しく構成された SafeWord サーバーの認証が行われる前に、指定されたディレクトリの `swec.dat` を削除する必要があります。

SafeWord 認証接続タイムアウト

この属性は、SafeWord クライアント (Access Manager) と SafeWord サーバーの間のタイムアウト期間を秒単位で定義します。デフォルト値は 120 秒です。

SafeWord クライアントタイプ

この属性は、モバイルクライアント、VPN、固定パスワード、チャレンジ / レスポンスなどの異なるクライアントと通信するために SafeWord サーバーが使用する、クライアントタイプを定義します。

SafeWord EASSP バージョン

この属性は、EASSP (Extended Authentication and Single Sign-on Protocol) バージョンを指定します。このフィールドでは、標準 (101) またはプレミアアクセス (201) のプロトコルバージョンが許容されています。

SafeWord オーセンティケータ最小強度

この属性は、クライアントまたは SafeWord サーバー認証で使用するオーセンティケータの最小強度を定義します。それぞれのクライアントタイプには異なるオーセンティケータ値があり、値が高いほどオーセンティケータの強度も高くなります。最大値は 20 です。最小値は 0 です。

認証レベル

認証レベルは認証方法ごとに個別に設定します。この値は、認証の信頼度を示します。ユーザーが認証を受けると、この値がセッションの SSO トークンに格納されます。ユーザーがアクセスしたいアプリケーションに SSO トークンが提供されると、そのアプリケーションは格納されている値を使用して、ユーザーにアクセスを許可するのに十分なレベルかどうかを判別します。SSO トークンに格納されている認証レベルが必要な最小値に満たない場合、アプリケーションはユーザーにより高い認証レベルのサービスで認証を再度受けるよう要求することがあります。デフォルト値は 0 です。

注	認証レベルの指定がない場合、SSO トークンはコア認証属性のデフォルト認証レベルで指定した値を格納します。詳細は、290 ページの「デフォルト認証レベル」を参照してください。2005Q1 のリリースでは、この機能は正常に動作しません。ただし、以前のリリースの場合は正常に動作します。

第 29 章

SAML 認証属性

SAML 認証属性は組織属性です。サービス設定の下で組織属性に適用される値は、SAML 認証テンプレートのデフォルト値になります。組織にサービスを登録したあと、サービステンプレートを作成する必要があります。デフォルト値は組織の管理者が登録後に変更できます。組織属性は組織のエントリに継承されません。

SAML 認証属性は次のとおりです。

認証レベル

認証レベルは認証方法ごとに個別に設定します。この値は、認証の信頼度を示します。ユーザーが認証を受けると、この値がセッションの SSO トークンに格納されます。ユーザーがアクセスしたいアプリケーションに SSO トークンが提供されると、そのアプリケーションは格納されている値を使用して、ユーザーにアクセスを許可するのに十分なレベルかどうかを判別します。SSO トークンに格納されている認証レベルが必要な最小値に満たない場合、アプリケーションはユーザーにより高い認証レベルのサービスで認証を再度受けるよう要求することがあります。デフォルト値は 0 です。

注 認証レベルの指定がない場合、SSO トークンはコア認証属性のデフォルト認証レベルで指定した値を格納します。詳細は、290 ページの「デフォルト認証レベル」を参照してください。2005Q1 のリリースでは、この機能は正常に動作しません。ただし、以前のリリースの場合は正常に動作します。

第 30 章

SecurID 認証属性

SecurID 認証属性は組織属性です。サービス設定の下で組織属性に適用される値は、SecurID 認証テンプレートのデフォルト値になります。組織にサービスを登録したあと、サービステンプレートを作成する必要があります。デフォルト値は組織の管理者が登録後に変更できます。組織属性は組織のサブツリーのエントリに継承されません。

このサービスでは、RSA の ACE/Server 認証サーバーを使用して、ユーザーを認証できます。SecurID 認証属性は次のとおりです。

注 このリリースの Access Manager の場合、Linux プラットフォームと Solaris x86 プラットフォームでは SecurID 認証モジュールを使用できません。この 2 つのプラットフォームでは、SecurID 認証モジュールの登録、設定、有効化を行わないでください。SecurID 認証モジュールは、SPARC のみで使用できます。

SecurID ACE/ サーバー設定パス

このフィールドは、SecurID ACE/Server `sdconf.rec` ファイルの存在するディレクトリを指定します。デフォルトは次のとおりです。

`/opt/ace/data`

このフィールドに異なるディレクトリを指定する場合は、SecurID 認証を試みる前にそのディレクトリが存在する必要があります。

SecurID ヘルパー設定ポート

この属性は、起動時に SecurID ヘルパー認証ポート属性に含まれる設定情報について、SecurID ヘルパーがどのポート上で待機するかを指定します。デフォルトは 58943 です。

この属性を変更した場合、`AMConfig.properties` ファイルの `securidHelper.ports` エントリも変更して、Access Manager を再起動することが必要です。`AMConfig.properties` ファイル内のエントリは、SecurID ヘルパーのインスタンスのポートをスペースで区切ったリストです。別の ACE/Server (別の `sdconf.rec` ファイルを持つ) と通信する組織ごとに、別々の SecurID ヘルパーを用意する必要があります。

SecurID ヘルパー認証ポート

この属性は、SecurID ヘルパーインスタンスが認証要求を待機するように、組織の SecurID 認証モジュールで設定するためのポートを指定します。ポート番号は、SecurID または UNIX 認証を使用するすべての組織で一意でなければなりません。デフォルトのポート番号は、57943 です。

認証レベル

認証レベルは認証方法ごとに個別に設定します。この値は、認証の信頼度を示します。ユーザーが認証を受けると、この値がセッションの SSO トークンに格納されます。ユーザーがアクセスしたいアプリケーションに SSO トークンが提供されると、そのアプリケーションは格納されている値を使用して、ユーザーにアクセスを許可するのに十分なレベルかどうかを判別します。SSO トークンに格納されている認証レベルが必要な最小値に満たない場合、アプリケーションはユーザーにより高い認証レベルのサービスで認証を再度受けるよう要求することがあります。デフォルト値は `0` です。

注 認証レベルの指定がない場合、SSO トークンはコア認証属性のデフォルト認証レベルで指定した値を格納します。詳細は、290 ページの「デフォルト認証レベル」を参照してください。2005Q1 のリリースでは、この機能は正常に動作しません。ただし、以前のリリースの場合は正常に動作します。

第 31 章

UNIX 認証属性

UNIX 認証サービスにはグローバル属性と組織属性があります。グローバル属性に適用される値は Sun Java System Acceess Manager 設定全体に適用され、設定済みのすべての組織に継承されます。グローバル属性の目的は Access Manager アプリケーションのカスタマイズであるため、ロールまたは組織に直接適用することはできません。組織属性に適用される値は設定済みの各組織のデフォルト値で、サービスを組織に登録するときに変更できます。組織属性は組織のエントリに継承されません。
UNIX 認証属性は次のように分類できます。

注 UNIX 認証属性を変更する場合は、Access Manager と `amunixd` ヘルパーの両方を再起動する必要があります。

グローバル属性

UNIX 認証サービスのグローバル属性には、次のものがあります。

UNIX ヘルパー設定ポート

この属性は、起動時に、「UNIX ヘルパー認証ポート」、「UNIX ヘルパーのタイムアウト」、および「UNIX ヘルパースレッド」属性に含まれる設定情報について、UNIX ヘルパーがどのポート上で待機するかを指定します。デフォルトは 58946 です。

この属性を変更した場合、`AMConfig.properties` ファイルの `unixHelper.port` エントリも変更して、Access Manager を再起動することが必要です。

UNIX ヘルパー認証ポート

この属性は、構成後に UNIX ヘルパーがどのポート上で認証要求を待機するかを指定します。デフォルトのポート番号は、57946 です。

UNIX ヘルパーのタイムアウト

この属性は、認証の制限時間を分単位で指定します。指定された時間を超えると、認証は自動的に失敗します。デフォルトでは 3 分に設定されています。

UNIX ヘルパースレッド

この属性は、同時に可能な UNIX 認証セッションの最大数を指定します。所定の時間にこの最大数に達すると、いずれかのセッションが解放されるまで、認証を試みても認証は行われません。デフォルトは 5 に設定されています。

組織属性

UNIX 認証サービスの組織属性には、次のものがあります。

認証レベル

認証レベルは認証方法ごとに個別に設定します。認証レベルは認証方法ごとに個別に設定します。この値は、認証の信頼度を示します。ユーザーが認証を受けると、この値がセッションの SSO トークンに格納されます。ユーザーがアクセスしたいアプリケーションに SSO トークンが提供されると、そのアプリケーションは格納されている

値を使用して、ユーザーにアクセスを許可するのに十分なレベルかどうかを判別します。SSO トークンに格納されている認証レベルが必要な最小値に満たない場合、アプリケーションはユーザーにより高い認証レベルのサービスで認証を再度受けるよう要求することがあります。デフォルト値は 0 です。

注 認証レベルの指定がない場合、SSO トークンはコア認証属性のデフォルト認証レベルで指定した値を格納します。詳細は、290 ページの「デフォルト認証レベル」を参照してください。2005Q1 のリリースでは、この機能は正常に動作しません。ただし、以前のリリースの場合は正常に動作します。

第 32 章

Microsoft Windows デスクトップ SSO 認証属性

Microsoft Windows デスクトップ SSO 認証属性は、組織属性です。サービス設定の下で組織属性に適用される値は、Microsoft Windows デスクトップ SSO 認証テンプレートのデフォルト値になります。組織にサービスを登録したあと、サービステンプレートを作成する必要があります。デフォルト値は組織の管理者が登録後に変更できます。組織属性は組織のサブツリーのエントリに継承されません。

この認証モジュールには、ドメインコントローラとして稼働中の Microsoft Windows 2000 サーバーが提供する、Kerberos 認証サービスが必要です。

Microsoft Windows デスクトップ SSO 認証属性は次のとおりです。

サービス主体

この属性は、認証に使用する Kerberos 主体を指定します。次の形式を使用します。

```
HTTP/hostname.domainname@dc_domain_name
```

hostname と *domainame* は、Access Manager インスタンスのホスト名とドメイン名を表します。*dc_domain_name* は、Microsoft Windows 2000 Kerberos サーバー (ドメインコントローラ) が存在する Kerberos ドメインです。このドメインは、Access Manager のドメイン名とは異なる可能性があります。

Keytab ファイル名

この属性は、認証に使用する Kerberos Keytab ファイルを指定します。次の形式を使用します。この形式は必須ではありません。

hostname.HTTP.keytab

hostname は、Access Manager インスタンスのホスト名です。

Kerberos レルム

この属性は、Kerberos Distribution Center (ドメインコントローラ) のドメイン名を指定します。設定によっては、ドメインコントローラのドメイン名は Access Manager のドメイン名とは異なることがあります。

Kerberos サーバー名

この属性は、Kerberos Distribution Center (ドメインコントローラ) のホスト名を指定します。ドメインコントローラの完全修飾ドメイン名 (FQDN) を入力する必要があります。

ドメイン名を含む主体を返す

有効にすると、この属性は Access Manager が認証時にドメインコントローラのドメイン名を含む Kerberos 主体を自動的に返すことを可能にします。

認証レベル

認証レベルは認証方法ごとに個別に設定します。この値は、認証の信頼度を示します。ユーザーが認証を受けると、この値がセッションの SSO トークンに格納されます。ユーザーがアクセスしたいアプリケーションに SSO トークンが提供されると、そのアプリケーションは格納されている値を使用して、ユーザーにアクセスを許可するのに十分なレベルかどうかを判別します。SSO トークンに格納されている認証レベルが必要な最小値に満たない場合、アプリケーションはユーザーにより高い認証レベルのサービスで認証を再度受けるよう要求することがあります。デフォルト値は 0 です。

注	認証レベルの指定がない場合、SSO トークンはコア認証属性のデフォルト認証レベルで指定した値を格納します。詳細は、290 ページの「デフォルト認証レベル」を参照してください。2005Q1 のリリースでは、この機能は正常に動作しません。ただし、以前のリリースの場合は正常に動作します。

第 33 章

認証設定サービス属性

認証設定サービス属性はダイナミックな組織属性です。この属性は、組織、サービス、またはロールに対して定義できます。組織属性はコア認証モジュールで定義されます。

ロールがユーザーに適用されると、またはユーザーが組織に割り当てられると、これらの属性はデフォルトでユーザーに継承されます。認証設定属性は次のとおりです。

認証設定

「編集」リンクをクリックすると、認証設定インタフェースが表示されます。これにより、ロールベースまたは組織ベースの認証用の認証モジュールを設定することができます。

次の表に、認証モジュールの設定オプションのリストを示します。

モジュール名	Access Manager に使用できるデフォルトの認証モジュールのリストから選択できます。

フラグ	プルダウンメニューで認証モジュールの要件を次のいずれかに指定できます。必須 - 認証には認証モジュールが必要です。 • 必須 - 認証には認証モジュールが必要です。認証に成功または失敗すると、認証モジュールリストの次のモジュールへと認証が進行します。 • 必要 - 認証には認証モジュールが必要です。認証に成功すると、認証モジュールリストの次のモジュールへと認証が進行します。認証に失敗すると、制御がアプリケーションに返されます。認証モジュールリストの次のモジュールには認証が進行しません。 • 十分 - 認証に認証モジュールは不要です。認証に成功するとすぐに、制御がアプリケーションに返されます。この場合、認証モジュールリストの次のモジュールには認証が進行しません。認証に失敗すると、リストの次のモジュールへと認証が進行します。 • オプション - 認証に認証モジュールは不要です。認証に成功または失敗しても、認証モジュールリストの次のモジュールへと認証が進行します。 以上のフラグによって、認証モジュールの適用条件が確立されます。適用条件には上下関係があり、「必須」がもっとも高く、「オプション」がもっとも低くなります。 たとえば、管理者が LDAP モジュールに「必須」フラグを設定している場合、ユーザーが特定のリソースにアクセスするためには、ユーザーの資格情報が LDAP の認証条件にパスすることが必要です。 複数の認証モジュールを追加して各モジュールのフラグを「必須」に設定した場合、ユーザーがアクセスするためにはすべての認証条件にパスする必要があります。 フラグの定義の詳細については、次のサイトの JAAS (Java Authentication and Authorization Service) を参照してください。 `http://java.sun.com/security/jaas/doc/module.html`
オプション	モジュールの追加オプションをキー = 値のペアとして指定できます。複数のオプションを指定するときは、スペースで区切ります。

ログイン成功 URL

この属性は、認証が成功した場合にユーザーをリダイレクトする URL を指定します。

ログイン失敗 URL

この属性は、認証が失敗した場合にユーザーをリダイレクトする URL を指定します。

認証ポストプロセスクラス

この属性は、ログインの成功または失敗後に実行する、認証後プロセスをカスタマイズするための Java クラス名を定義します。

競合の解決レベル

この属性は、ロールにだけ適用されます。競合解決レベルは、ロールの認証設定属性の優先順位を設定します。ロールには同一のユーザーが含まれる場合もあります。たとえば、User 1 が Role 1 および Role 2 に割り当てられている場合を想定します。ユーザーが認証を試みたときに、認証の成功または失敗時のリダイレクトや認証後プロセスに対して Role 1 の優先順位がもっとも高くなるように設定することができます。

第 34 章

クライアントディテクションサービス属性

クライアントディテクションサービス属性はグローバル属性です。グローバル化設定のサービス属性に適用される値は、Access Manager 設定全体に適用され、設定済みのすべての組織に継承されます。グローバル属性の目的は Access Manager アプリケーションのカスタマイズなので、セッションサービス属性をロールまたは組織に直接適用することはできません。クライアントディテクション属性は次のとおりです。

クライアントタイプ

クライアントタイプを検出するため、Access Manager はクライアントの特性を認識する必要があります。これらの特性は、サポートされているすべてのタイプのプロパティをクライアントデータの形式で識別します。この属性を使って、クライアントマネージャインタフェースを通してクライアントデータを変更できます。クライアントマネージャにアクセスするには、「編集」リンクをクリックします。

初期状態では、Access Manager には次のクライアントタイプがあります。

- HDML
- HTML
- JHTML
- VoiceX
- WML
- XHTML

- cHTML
- iHTML
- クライアントタイプについては、次の場所にある『Sun Java System Portal Server, Mobile Access 2005Q1 管理ガイド』を参照してください。

  http://docs.sun.com/app/docs/coll/PortalServer_05q1

クライアントマネージャ

クライアントマネージャは、基本クライアント、スタイル、および関連プロパティを一覧表示するインタフェースです。また、デバイスの追加や設定を行うことができます。

基本クライアントタイプ

基本クライアントタイプは、クライアントマネージャの上部に一覧表示されます。基本クライアントタイプにはデフォルトのプロパティがあり、そのクライアントタイプに属するすべてのデバイスは、これらのプロパティを継承できます。

スタイルプロファイル

クライアントマネージャでは、基本クライアントタイプも含め、利用可能なクライアントがすべて「スタイル」プルダウンメニューにまとめられます。選択したスタイル (親プロファイル) によって、設定済みの子デバイスに共通するプロパティが定義されます。デバイスには、親プロファイルのプロパティがダイナミックに継承されます。

「現在のスタイルのプロパティ」リンクをクリックすると、読み取り専用のクライアントエディタウィンドウが開き、スタイルのプロパティが表示されます。

デバイスプロファイル

スタイルを選択すると、そのスタイルに対して設定されているデバイスプロファイルがクライアントマネージャに表示されます。デバイスはユーザーエージェント (デバイス名) に基づいてソートされます。デバイスをフィルタリングするには、「フィルタ」フィールドにユーザーエージェント文字列を入力します (ワイルドカードも使用可)。

各デバイスのクライアントプロパティを修正するには、そのデバイス名の横にある「編集」リンクをクリックします。クライアントエディタウィンドウにプロパティが表示されます。プロパティを編集するには、プルダウンリストから以下の分類を選択します。

ハードウェアプラットフォーム : ディスプレイサイズ、サポートされている文字セットなど、デバイスのハードウェアのプロパティが含まれています。

ソフトウェアプラットフォーム : デバイスのアプリケーション環境、オペレーティングシステム、およびインストール済みソフトウェアのプロパティが含まれています。

ネットワーク特性 : サポートされているベアラなど、ネットワーク環境を記述するプロパティが含まれています。

ブラウザユーザーエージェント : デバイス上で実行中のブラウザユーザーエージェントに関連する属性が含まれています。

WAP 特性 : デバイスでサポートされている WAP (Wireless Application Protocol) 環境のプロパティが含まれています。

Push 特性 : デバイスでサポートされている WAP 環境のプロパティが含まれています。

追加プロパティ : デバイスのプロパティを追加できます。

具体的なプロパティの定義については、次の場所にある Open Mobile Alliance Ltd. (OMA) の『Wireless Application Protocol, Version 20-Oct-2001』を参照してください。

http://www1.wapforum.org/tech/terms.asp?doc=WAP-248-UAProf-20011020-a.pdf

---

注 マニュアルにアクセスするには、最初に WAP Forum™ で登録する必要があります。詳細については、http://www.wapforum.org/faqs/index.htm を参照してください。

プロパティの修正が完了したら、「保存」をクリックします。デバイスには、カスタマイズされたことを示す「**」という文字が表示されます。「デフォルト」リンクを使用すると、カスタマイズしたプロパティを削除し、デバイスをデフォルト設定に戻すことができます。

スタイルに新しいデバイスを追加するには、「新規デバイス」ボタンをクリックします。次のフィールドを持つ「新規デバイスを作成」ウィンドウが表示されます。

スタイル : デバイスの基本スタイルを表示します。たとえば、HTML などです。

デバイスユーザーエージェント : デバイスの名前を指定します。

「次へ」をクリックして、次のフィールドを表示します。

クライアントタイプ名 : HTML などのクライアントタイプを表示します。クライアントタイプ名は、すべてのデバイスで一意でなければなりません。

このデバイスの直接の親 : デバイスの親 (基本) クライアントタイプを指定します。たとえば、HTML などです。

HTTP ユーザーエージェント文字列 : HTTP 要求ヘッダー内のユーザーエージェントを定義します。たとえば、Mozilla/4.0 などです。

「OK」をクリックし、デバイスのプロパティをカスタマイズします。具体的なプロパティの定義については、次の場所にある Open Mobile Alliance Ltd. (OMA) の『Wireless Application Protocol, Version 20-Oct-2001』を参照してください。

http://www1.wapforum.org/tech/

デバイスとそのプロパティを複製するには、「複製」リンクをクリックします。デバイス名は一意でなければなりません。Access Manager では、デフォルトで `copy_of_` デバイス名というデバイス名に変更されます。

デバイスを削除するには、そのデバイスに表示されている「削除」リンクをクリックします。

デフォルトクライアントタイプ

この属性は、クライアントタイプ属性のクライアントタイプのリストの中からデフォルトクライアントタイプを定義します。デフォルトは `genericHTML` です。

クライアントディテクションクラス

この属性は、クライアントディテクション要求のすべてが送信されるクライアントディテクションクラスを定義します。この属性によって返される文字列は、クライアントタイプ属性に指定されているクライアントタイプのいずれかと一致します。デフォルトのクライアントディテクションクラスは、`com.sun.mobile.cdm.FEDIClientDetector` です。Access Manager には、`com.iplanet.services.cdm.ClientDetectionDefaultImpl` も含まれています。

クライアントディテクションを有効

この属性で、クライアントディテクションを有効にすることができます。クライアントディテクションが有効になっている (選択されている) 場合、すべての要求はクライアントディテクションクラス属性で指定されているクラスを使って送信されます。

デフォルトでは、クライアントディテクション機能は有効になっています。この属性が選択されていない場合、Access Manager では、そのクライアントは genericHTML であり、HTML ブラウザからアクセスされると見なされます。

第 35 章

グローバル化設定のサービス属性

グローバル化設定のサービス属性はグローバル属性です。グローバル化設定のサービス属性に適用される値は、Access Manager 設定全体に適用され、設定済みのすべての組織に継承されます。グローバル属性の目的は Access Manager アプリケーションのカスタマイズなので、セッションサービス属性をロールまたは組織に直接適用することはできません。グローバル化設定の属性は次のとおりです。

各ロケールでサポートされる文字セット

この属性では、各ロケールでサポートされる文字セットを一覧表示します。このリストでは、ロケールと文字セットとのマッピングを示します。形式は次のとおりです。

`locale=` ロケール名 `|charset=charset1;charset2;charset3;...;charsetn`

属性の下にあるボタンを使用すると、文字セットを追加、編集、複製、および削除できます。

文字セットのエイリアス

この属性では、応答を送信するために使用するコードセット名を一覧表示します。コードセット名は IANA 名にマップしています。Java コードセット名に一致する必要はありません。現在は、Java 文字セットと IANA 文字セットとの間のマップにはハッシュテーブルが使われます。エイリアスの形式は次のとおりです。

`mimeName=` 文字セット `|javaName=` 文字セット

次に例を示します。

`mimeName=Shift_JIS|javaName=SJIS`

これはどちらも同じ文字セットを示しています。

属性の下にあるボタンを使用すると、文字セットのエイリアスを追加、編集、複製、および削除できます。

自動生成される共通名の形式

この表示オプションではさまざまなロケールおよび文字セットに名前の形式を適応するように、名前を自動生成する方法を定義します。デフォルトの構文は次のとおりです。定義内のコンマやスペースは名前の形式で表示されることに注意してください。

```
en_us = {givenname} {initials} {sn}
```

たとえば中国語の文字セットで、uid (11111) のユーザー (User One) に対して新しい名前の形式で表示するには、次の表現を使用します。

```
zh = {sn}{givenname}({uid})
```

これにより、次のように表示されます。

```
OneUser 11111
```

第 36 章

ログサービス属性

ログサービス属性はグローバル属性です。これらの属性に適用される値は Sun Java System Acceess Manager 設定全体に適用され、設定済みのすべての組織に継承されます。グローバル属性の目的は Access Manager アプリケーションのカスタマイズであるため、ロールまたは組織に直接適用することはできません。ログ属性は次のとおりです。

最大ログサイズ

この属性は、Access Manager ログファイルの最大サイズ (バイト単位) の値を指定します。デフォルト値は 1000000 です。

履歴ファイルの数

この属性は、履歴解析のために保持するバックアップログファイルの数に等しい値を持ちます。入力できる整数値は、ローカルシステムのパーティションサイズと利用可能なディスク容量で決まります。デフォルト値は 3 です。

注	値に 0 を入力すると、値が 1 と同様に解釈されます。つまり、0 を指定すると、バックアップログファイルが作成されます。

ログファイルの場所

ファイルベースのログ機能には、ログファイルを格納する場所が必要です。このフィールドは、その場所の完全なディレクトリパスを指定します。デフォルトの場所は次に示すとおりです。

```
/var/opt/SUNWam/logs
```

デフォルト以外のディレクトリを使う場合、そのディレクトリには Access Manager を実行しているユーザーに対する書き込み権限が必要です。

Oracle や MySQL などの DB (データベース) ログ用にログの場所を設定するとき、ログの場所の記述部分では大文字と小文字が区別されます。

たとえば、Oracle データベースにログを書き込む場合、ログの場所は次のようになります。

```
jdbc:oracle:thin:@machine.domain:port:DBName
```

`jdbc:oracle:thin` は小文字で記述する必要があります。

注	DB へのログの書き込みを設定するには、Web コンテナの JVM クラスパスに JDBC ドライバファイルを追加します。JDBC ドライバファイルは手動で amadmin スクリプトのクラスパスに追加する必要があります。手動で追加しないと、amadmin のログ機能によって JDBC ドライバがロードされません。 ログ属性の値を変更した場合は、変更を有効にするために Access Manager を再起動する必要があります。

ログタイプ

この属性により、フラットファイルログには File、データベースログには DB のいずれかを指定できます。

データベースユーザー名

この属性は、ログタイプ属性が DB に設定されている場合に、データベースに接続するユーザーの名前を指定します。

データベースユーザーパスワード

この属性は、ログタイプ属性が DB に設定されている場合に、データベースユーザーのパスワードを指定します。

データベースユーザーパスワード (確認)

データベースパスワードを確認します。

データベースドライバ名

この属性は、ログ実装クラスに使用するドライバを指定します。

設定可能なログフィールド

このパラメータは、記録されるフィールドのリストを表します。デフォルトでは、次のフィールドが記録されます。

- Domain
- Hostname
- IPAddress
- LoggedBy
- Loglevel
- LoginID
- ModuleName

ログ検証頻度

この属性は、サーバーがログを検証して改ざんを検出する頻度 (秒単位) を設定します。デフォルトの時間は 3600 秒です。このパラメータは、セキュリティ保護されたログにだけ適用されます。

ログ署名時間

このパラメータは、ログに署名する頻度 (秒単位) を設定します。デフォルトの時間は 900 秒です。このパラメータは、セキュリティ保護されたログにだけ適用されます。

セキュリティ保護されたログを有効

この属性は、セキュリティ保護されたログを有効にするかどうかを指定します。デフォルトでは、セキュリティ保護されたログはオフです。セキュリティ保護されたログでは、不正な変更またはセキュリティログの改ざんを検出できます。

レコードの最大数

この属性は、読み取り照会と一致するレコードの数に関係なく、Java LogReader インタフェースが返すレコードの最大数を設定します。デフォルトでは 500 に設定されています。この属性は、LogQuery パラメータを使用してログ API を呼び出すことで無効にできます。

アーカイブごとのファイル数

この属性は、セキュリティ保護されたログにのみ適用可能です。この属性は、セキュリティ保護された後続のログに対して、ログファイルとキーストアをいつアーカイブする必要があるか、およびセキュリティ保護されたキーストアをいつ再生成する必要があるかを指定します。デフォルトでは、各ログで 5 ファイル処理します。

バッファサイズ

この属性は、ログサービスに送られて記録される前にメモリ内のバッファに保存される、ログレコードの最大数を指定します。デフォルトは、1 レコードです。

DB 失敗メモリバッファサイズ

この属性では、データベース (DB) ログが失敗した場合にメモリに保存するログレコードの最大数を定義します。DB ログを指定した場合にかぎって、この属性を適用できます。Access Manager ログサービスが DB との接続を失うと、指定したレコード数までバッファに保存されます。この属性のデフォルトは、バッファサイズ属性で定義した値の 2 倍です。

バッファ時間

この属性は、ログレコードがログサービスに送られて記録される前にメモリ内のバッファに保存される時間を指定します。デフォルト値は 3600 秒です。

時間バッファリングを有効

「オン」にすると、Access Manager では、ログレコードをメモリにバッファする時間の上限を設定します。この時間は、「バッファ時間」属性に設定します。

第 37 章

ネーミングサービス属性

ネーミングサービス属性はグローバル属性です。これらの属性に適用される値は Sun Java System Acceess Manager 設定全体に適用され、設定済みのすべての組織に継承されます。グローバル属性の目的は Access Manager アプリケーションのカスタマイズであるため、ロールまたは組織に直接適用することはできません。

ネーミングサービスを使用すると、プラットフォームで複数の Access Manager が動作していても、クライアントは正しいサービス URL を見つけることができます。ネーミング URL が見つかると、ネーミングサービスはユーザーのセッションを復号化して、プロトコル、ホスト、およびポートをセッションのパラメータでダイナミックに置き換えます。これにより、サービスに対して返された URL が、ユーザーセッションの作成されたホストの URL であることが保証されます。ネーミング属性は次のとおりです。

プロファイルサービス URL

このフィールドは、次の構文を使用して値を取得します。

`%protocol://%host:%port/` サーバー配備 `URI/profileservice`

この構文により、特定のセッションパラメータに基づくプロファイル URL をダイナミックに置き換えることができます。

セッションサービス URL

このフィールドは、次の構文を使用して値を取得します。

`%protocol://%host:%port/` サーバー配備 `URI/sessionservice`

この構文により、特定のセッションパラメータに基づくセッション URL をダイナミックに置き換えることができます。

ログサービス URL

このフィールドは、次の構文を使用して値を取得します。

`%protocol://%host:%port/` サーバー配備 `URI/loggingservice`

この構文により、特定のセッションパラメータに基づくログ URL をダイナミックに置き換えることができます。

ポリシーサービス URL

このフィールドは、次の構文を使用して値を取得します。

`%protocol://%host:%port/` サーバー配備 `URI/policyservice`

この構文により、特定のセッションパラメータに基づくポリシー URL をダイナミックに置き換えることができます。

認証サービス URL

このフィールドは、次の構文を使用して値を取得します。

`%protocol://%host:%port/` サーバー配備 `URI/authservice`

この構文により、特定のセッションパラメータに基づく認証 URL をダイナミックに置き換えることができます。

SAML Web プロファイル / アーティファクト サービス URL

このフィールドは、次の構文を使用して値を取得します。

`%protocol://%host:%port/` サーバー配備 `URI/SAMLAwareServlet`

この構文により、特定のセッションパラメータに基づく SAML Web プロファイル / アーティファクト URL をダイナミックに置き換えることができます。

SAML SOAP サービス URL

このフィールドは、次の構文を使用して値を取得します。

`%protocol://%host:%port/` サーバー配備 `URI/SAMLSOAPReceiver`

この構文により、特定のセッションパラメータに基づく SAML SOAP URL をダイナミックに置き換えることができます。

SAML Web プロファイル /POST サービス URL

このフィールドは、次の構文を使用して値を取得します。

`%protocol://%host:%port/` サーバー配備 `URI/SAMLPOSTProfileServlet`

この構文により、特定のセッションパラメータに基づく SAML Web プロファイル /POST URL をダイナミックに置き換えることができます。

SAML アサーションマネージャサービス URL

このフィールドは、次の構文を使用して値を取得します。

`%protocol://%host:%port/` サーバー配備
`URI/AssertionManagerServlet/AssertionManagerIF`

この構文により、特定のセッションパラメータに基づく SAML アサーションマネージャサービス URL をダイナミックに置き換えることができます。

連携アサーションマネージャサービス URL

このフィールドは、次の構文を使用して値を取得します。

```
%protocol://%host:%port/amserver/FSAssertionManagerServlet/FSAssert
ionManagerIF
```

この構文により、特定のセッションパラメータに基づく連携アサーションマネージャサービス URL をダイナミックに置き換えることができます。

アイデンティティ SDK サービス URL

このフィールドは、次の構文を使用して値を取得します。

```
%protocol://%host:%port/amserver/UserManagementServlet/
```

この構文により、特定のセッションパラメータに基づくアイデンティティ SDK サービス URL をダイナミックに置き換えることができます。

セキュリティトークンマネージャ URL

このフィールドは、次の構文を使用して値を取得します。

```
%protocol://%host:%port/amserver/SecurityTokenManagerServlet/Securi
tyTokenManagerIF/
```

この構文により、特定のセッションパラメータに基づくセキュリティトークンマネージャ URL をダイナミックに置き換えることができます。

JAXRPC エンドポイント URL

このフィールドは、次の構文を使用して値を取得します。

```
%protocol://%host:%port/amserver/jaxrpc/
```

この構文により、特定のセッションパラメータに基づく JAXRPC エンドポイント URL をダイナミックに置き換えることができます。

第 38 章

パスワードリセットサービス属性

パスワードリセットサービス属性は組織属性です。サービス設定の下で組織属性に適用される値は、指定された組織でのパスワードリセットサービスのデフォルト値になります。組織属性は組織のサブツリーのエントリに継承されません。

パスワードリセット属性は次のとおりです。

- 363 ページの「パスワードリセットのロックアウト属性名」
- 363 ページの「パスワードリセットのロックアウト属性値」

ユーザー検証

この属性では、パスワードをリセットするユーザーを検索するための値を指定します。

秘密の質問

このフィールドでは、ユーザーが自分のパスワードをリセットするために使用できる質問のリストを追加できます。質問を追加するには、「秘密の質問」フィールドに質問を入力し、「追加」をクリックします。選択した質問は、ユーザーの「ユーザープロファイル」ページに表示されます。

すると、パスワードをリセットするために、ユーザーは質問を選択できるようになります。「個人的な質問を有効」属性を選択している場合は、ユーザーが独自の質問を作成できます。

検索フィルタ

ユーザーエントリの検索に使用する検索フィルタを指定します。

ベース DN

この属性は、ユーザーの検索をどの DN から開始するかを指定します。DN が指定されていない場合は、組織 DN から検索が始まります。プロキシ認証の競合の原因となるので、ベース DN として `cn=directorymanager` は使用しないでください。

バインド DN

この属性は、バインドパスワードとともに、ユーザーのパスワードをリセットするために使用します。

バインドパスワード

この属性は、バインド DN とともに、ユーザーのパスワードをリセットするために使用します。

パスワードリセットのオプション

この属性は、パスワードをリセットするためのクラス名を指定します。デフォルトのクラス名は次のとおりです。

```
com.sun.identity.password.RandomPasswordGenerator
```

パスワードリセットクラスは、プラグインを使用してカスタマイズできます。このクラスは、`PasswordGenerator` インタフェースで実装する必要があります。詳細は、『Access Manager Developer's Guide』を参照してください。

パスワードの変更通知のオプション

この属性は、パスワードをリセットするときのユーザーへの通知方法を指定します。デフォルトのクラス名は次のとおりです。

```
com.sun.identity.password.EmailPassword
```

パスワード通知クラスは、プラグインを使用してカスタマイズできます。このクラスは、`NotifyPassword` インタフェースで実装する必要があります。詳細は、『Access Manager Developer's Guide』を参照してください。

パスワードリセットを有効

この属性を選択すると、パスワードリセット機能が有効になります。

個人的な質問を有効

この属性を選択すると、ユーザーはパスワードをリセットするための独自の質問を作成できます。

質問の最大数

この値は、パスワードリセットページで確認される質問の最大数を指定します。

次のログイン時にパスワード変更を強制

このオプションを有効にすると、ユーザーは次のログイン時にパスワードの変更を強制されます。最上位レベル管理者以外の管理者にパスワードリセット強制オプションを設定させる場合は、その属性へのアクセスを許可するためにデフォルト権限 ACI を変更する必要があります。

パスワードリセット失敗のロックアウトを有効

この属性は、パスワードリセットアプリケーションを使用してユーザーが最初にパスワードリセットに失敗したときに、ユーザーにパスワードリセットを禁止するかどうかを指定します。デフォルトでは、無効になっています。

パスワードリセット失敗のロックアウトカウント

この属性は、パスワードリセット失敗のロックアウト間隔で定義された時間内に、ユーザーがパスワードのリセットを試みることができる回数を定義します。

この回数を超えると、ユーザーはロックアウトされます。たとえば、パスワードリセット失敗のロックアウトカウントが 5 に設定されていて、ログイン失敗のロックアウト間隔が 5 分に設定されている場合、ユーザーは、ロックアウトになる前の 5 分以内に 5 回パスワードリセットのチャンスがあります。

パスワードリセット失敗のロックアウト間隔

この属性は、パスワードのリセットを試みる回数 (パスワードリセット失敗のロックアウトカウントで定義される) を完了するまでの時間を分単位で定義します。これを超えるとロックアウトされます。

ロックアウト通知の送信先電子メールアドレス

この属性は、ユーザーがパスワードリセットサービスからロックアウトされた場合に通知を受け取る電子メールアドレスを指定します。スペース区切りのリストで複数の電子メールアドレスを指定できます。

ユーザーに警告を出すまでの失敗回数

この属性は、Access Manager がそのユーザーにロックアウトされるという警告を送信するまでに許可されるパスワードリセット失敗の回数を指定します。

パスワードリセット失敗のロックアウト持続時間

この属性は、ロックアウトが発生した際に、ユーザーがパスワードリセットを試みることが許可されない期間を、分単位で定義します。

パスワードリセットのロックアウト属性名

この属性には、パスワードリセットのロックアウト属性値で設定する `inetuserstatus` 値が含まれています。ユーザーがパスワードリセットでロックアウトされて、パスワードリセット失敗のロックアウト持続時間 (分) 変数が 0 に設定されている場合、`inetuserstatus` が無効に設定され、ユーザーはパスワードのリセットを試みることができません。

パスワードリセットのロックアウト属性値

この属性は、ユーザー状態の `inetuserstatus` 値 (パスワードリセットロックアウト属性名に含まれる) を有効と無効のいずれかに指定します。ユーザーがパスワードリセットでロックアウトされて、パスワードリセット失敗のロックアウト持続時間 (分) 変数が 0 に設定されている場合、`inetuserstatus` が無効に設定され、ユーザーはパスワードのリセットを試みることができません。

第 39 章

プラットフォームサービス属性

プラットフォームサービス属性はグローバル属性です。これらの属性に適用される値は Sun Java System Acceess Manager 設定全体に適用され、設定済みのすべての組織に継承されます。グローバル属性の目的は Access Manager アプリケーションのカスタマイズであるため、ロールまたは組織に直接適用することはできません。プラットフォーム属性は次のとおりです。

サーバーリスト

ネーミングサービスは初期化時にこの属性を読み取ります。このリストには、1 つの Access Manager 構成内の Access Manager セッションサーバーが含まれます。たとえば、2 つの Access Manager をインストールして 1 つのサーバーとして動作させる場合は、両方ともこのリストに入れる必要があります。サービス URL の要求で指定したホストがこのリストにない場合、ネーミングサービスは要求を拒否します。このリストの最初の値は、インストール時に指定したサーバーのホスト名およびポートを指定します。リストの最後には、サーバーを一意に特定する 2 バイトの値が入ります。ロードバランスに参加するサーバーには、それぞれに固有の識別子が必要です。これはまた、サーバー URL をサーバー ID にマッピングして Cookie の長さを短くするためにも使用されます。次に例を示します。

```
protocol:// サーバードメイン : ポート |01
```

さらに「protocol:// サーバードメイン ポート |01| インスタンス名」の形式を使用して、サーバーを追加できます。

この属性では、ネーミングサービスプロトコルのみを使用しなければなりません。

プラットフォームロケール

プラットフォームロケールの値は、Access Manager とともにインストールしたデフォルトの言語サブタイプです。認証サービス、ログサービス、および管理サービスは、この値の言語で管理されます。デフォルトは en_US です。サポートされている言語サブタイプの全リストについては、285 ページの表 20-1 を参照してください。

Cookie ドメイン

これはドメインのリストで、認証中にユーザーのブラウザに Cookie を設定するときに Cookie ヘッダーで返されます。空の場合、Cookie ドメインは設定されません。言い換えると、Access Manager セッション Cookie は Access Manager 自体にだけ転送され、ドメインのほかのサーバーには転送されません。ドメインのほかのサーバーで SSO が必要な場合は、Cookie ドメインでこの属性を設定する必要があります。1 つの Access Manager 上で異なるドメインに 2 つのインタフェースがある場合、両方の Cookie ドメインをこの属性に設定する必要があります。ロードバランサを使用する場合、Cookie ドメインは、ロードバランサの背後にあるサーバーではなく、ロードバランサのドメインのものであることが必要です。このフィールドのデフォルト値はインストールされている Access Manager のドメインです。

注	入力した Cookie ドメインが正しいことを確認します。Cookie ドメインが間違っていると、Access Manager にログインできません。

ログインサービス URL

このフィールドはログインページの URL を指定します。この属性のデフォルト値は / サービス配備 URI/UI/Login です。

ログアウトサービス URL

このフィールドはログアウトページの URL を指定します。この属性のデフォルト値は / サービス配備 URI/UI/Logout です。

使用可能なロケール

この属性は、プラットフォーム用に設定したすべての使用可能なロケールを格納します。たとえば、ユーザーにロケールを選択させるアプリケーションを考えます。このアプリケーションは、プラットフォームのプロファイルからこの属性を取得して、ロケールのリストをユーザーに提示します。ユーザーがロケールを選択すると、アプリケーションがそれをユーザーエントリ `preferredLocale` に設定します。

クライアント文字セット

この属性は、プラットフォームレベルのさまざまなクライアント用の文字セットを指定します。これには、クライアントタイプのリスト、および対応する文字セットも含まれます。形式は次のとおりです。

```
clientType| 文字セット
clientType2| 文字セット
```

次に例を示します。

```
genericHTML|UTF-8
```

第 40 章

ポリシー設定サービス属性

ポリシー設定サービス属性には、グローバル属性と組織属性とがあります。グローバル属性に適用される値は Sun Java System Acceess Manager 設定全体に適用され、設定済みのすべての組織に継承されます。グローバル属性の目的は Access Manager アプリケーションのカスタマイズであるため、ロールまたは組織に直接適用することはできません。サービス管理の下で組織属性に適用される値が、ポリシー設定のデフォルト値になります。組織にサービスを登録したあと、サービステンプレートを作成する必要があります。デフォルト値は組織の管理者が登録後に変更できます。組織属性は組織のエントリに継承されません。ポリシー設定属性は次のように分けられます。

グローバル属性

ポリシー設定サービスのグローバル属性には、次のものがあります。

リソースコンパレータ

この属性はリソースコンパレータ情報を指定します。リソースコンパレータは、「ポリシー」ルール定義で指定されたリソースの比較に使用されます。リソースの比較は、ポリシーの作成と評価の両方に使用します。この属性には次の値があります。

`serviceType`　　コンパレータを使用するサービスを指定します。

class	リソース比較アルゴリズムを実装する Java クラスを定義します。
wildcard	リソース名に使用可能なワイルドカードを指定します。
delimiter	リソース名に使用する区切り記号を指定します。
caseSensitivity	リソースを比較する際に、大文字と小文字を区別するかどうかを指定します。False の場合は区別せず、True の場合は区別します。

拒否決定時の評価継続

この属性は、拒否ポリシー決定が存在する場合でも、ポリシーフレームワークが後続のポリシーの評価を続けるかどうかを指定します。選択しない場合 (デフォルト)、ポリシーの評価は、拒否決定を認識すると、後続のポリシーの評価をやめます。

組織属性

ポリシー設定サービスの組織属性は次のとおりです。

- 371 ページの「LDAP サーバーおよびポート」
- 372 ページの「LDAP ベース DN」
- 372 ページの「LDAP ユーザーベース DN」
- 372 ページの「Access Manager ロールベース DN」
- 372 ページの「LDAP バインド DN」
- 372 ページの「LDAP バインドパスワード」
- 373 ページの「LDAP バインドパスワード (確認)」
- 373 ページの「LDAP 組織検索フィルタ」
- 373 ページの「LDAP 組織検索範囲」
- 373 ページの「LDAP グループ検索フィルタ」
- 373 ページの「LDAP グループ検索範囲」
- 374 ページの「LDAP ユーザー検索フィルタ」
- 374 ページの「LDAP ユーザー検索範囲」
- 374 ページの「LDAP ロール検索フィルタ」
- 374 ページの「LDAP ロール検索範囲」

LDAP サーバーおよびポート

このフィールドは、Access Manager インストール時に指定されるプライマリ LDAP サーバーのホスト名とポート番号を指定します。このホスト名とポート番号は、LDAP ユーザー、LDAP ロール、LDAP グループなどのポリシーサブジェクトを検索する際に使用します。形式は *hostname:port* です。次に例を示します。

```
machine1.example.com:389
```

複数の LDAP サーバーホストに対するフェイルオーバー設定の場合は、この値にスペース区切りのリストで複数のホストを指定できます。形式は hostname1:port1 hostname2:port2... です。

次に例を示します。

```
machine1.example1.com:389 machine2.example1.com:389
```

複数のエントリの場合は、ローカルサーバー名をプレフィックスとして付ける必要があります。これにより、特定の Access Manager が特定の Directory Server と通信するように設定できます。

形式は `servername|hostname:port` です。

次に例を示します。

```
machine1.example1.com|machine1.example1.com:389
machine1.example2.com|machine1.example2.com:389
```

フェイルオーバー設定の場合は次のようになります。

```
IS_Server1.example1.com|machine1.example1.com:389 machine2.example.com1:389
IS_Server2.example2.com|machine1.example2.com:389 machine2.example2.com:389
```

LDAP ベース DN

このフィールドは、検索を開始する LDAP サーバー内のベース DN を指定します。デフォルトでは、Access Manager インストールの最上位組織です。

LDAP ユーザーベース DN

この属性は、検索を開始する LDAP サーバー内の LDAP ユーザーサブジェクトで使用するベース DN を指定します。デフォルトでは、Access Manager インストールベースの最上位組織です。

Access Manager ロールベース DN

この属性は、検索を開始する LDAP サーバー内の Access Manager ロールサブジェクトで使用するベース DN を指定します。デフォルトでは、Access Manager インストールベースの最上位組織です。

LDAP バインド DN

このフィールドは、LDAP サーバー内のバインド DN を指定します。

LDAP バインドパスワード

この属性は、LDAP サーバーへのバインドに使用するパスワードを定義します。デフォルトで、インストール中に入力した `amldapuser` パスワードが、バインドユーザーとして使用されます。

LDAP バインドパスワード (確認)

LDAP バインドパスワードを確認します。

LDAP 組織検索フィルタ

組織エントリの検索に使用する検索フィルタを指定します。デフォルトは (objectclass=sunMangagedOrganization) です。

LDAP 組織検索範囲

この属性は、組織エントリの検索範囲を定義します。範囲は次のいずれかにする必要があります。

- SCOPE_BASE
- SCOPE_ONE
- SCOPE_SUB (デフォルト)

LDAP グループ検索フィルタ

グループエントリの検索に使用する検索フィルタを指定します。デフォルトは、(objectclass=groupOfUniqueNames) です。

LDAP グループ検索範囲

この属性は、グループエントリの検索範囲を定義します。範囲は次のいずれかにする必要があります。

- SCOPE_BASE
- SCOPE_ONE
- SCOPE_SUB (デフォルト)

LDAP ユーザー検索フィルタ

ユーザーエントリの検索に使用する検索フィルタを指定します。デフォルトは、(objectclass=inetorgperson) です。

LDAP ユーザー検索範囲

この属性は、ユーザーエントリの検索範囲を定義します。範囲は次のいずれかにする必要があります。

- SCOPE_BASE
- SCOPE_ONE
- SCOPE_SUB (デフォルト)

LDAP ロール検索フィルタ

ロールのエントリ検索に使用する検索フィルタを指定します。デフォルトは、(&(objectclass=ldapsubentry)(objectclass=nsroledefinitions)) です。

LDAP ロール検索範囲

この属性は、ロールのエントリ検索範囲を定義します。範囲は次のいずれかにする必要があります。

- SCOPE_BASE
- SCOPE_ONE
- SCOPE_SUB (デフォルト)

Access Manager ロール検索範囲

この属性は、Access Manager ロールサブジェクトのエントリ検索範囲を定義します。範囲は次のいずれかにする必要があります。

- SCOPE_BASE
- SCOPE_ONE
- SCOPE_SUB (デフォルト)

LDAP 組織検索属性

このフィールドは、組織に対して検索を行うための属性タイプを定義します。デフォルトは o です。

LDAP グループ検索属性

このフィールドは、グループに対して検索を行うための属性タイプを定義します。デフォルトは cn です。

LDAP ユーザー検索属性

このフィールドは、ユーザーに対して検索を行うための属性タイプを定義します。デフォルトは uid です。

LDAP ロール検索属性

このフィールドは、ロールに対して検索を行うための属性タイプを定義します。デフォルトは cn です。

検索で返される結果の最大数

このフィールドは検索で返される結果の最大数を定義します。デフォルト値は 100 です。指定された最大数を検索結果が上回った場合、その時点までに検索されたエントリが返されます。

検索タイムアウト

この属性は、検索タイムアウトが発生するまでの時間を指定します。指定した時間を過ぎた場合は、その時点までに検索されたエントリが返されます。

LDAP SSL を有効

この属性は、LDAP サーバーが SSL を実行するかどうかを指定します。選択した場合、SSL は有効になり、選択しない場合 (デフォルト)、SSL は無効になります。

SSL が有効な状態で LDAP サーバーが動作している場合は (LDAPS)、必ず適切な信頼された SSL 証明書によって Access Manager を設定し、Access Manager が LDAPS プロトコルで Directory サーバーに接続できるようにする必要があります。

LDAP 接続プールの最小サイズ

この属性は、LDAP サーバー属性に指定されたとおり、Directory Server への接続に使用する接続プールの最小サイズを指定します。デフォルトは 1 です。

LDAP 接続プールの最大サイズ

この属性は、LDAP サーバー属性に指定されたとおり、Directory Server への接続に使用する接続プールの最大サイズを指定します。デフォルトは 10 です。

選択したポリシーサブジェクト

この属性を使用すると、組織内のポリシー定義に使用できるサブジェクトタイプのセットを選択できます。

選択したポリシー条件

この属性を使用すると、組織内のポリシー定義に使用できる条件タイプのセットを選択できます。

選択したポリシー参照

この属性を使用すると、組織内のポリシー定義に使用できる参照タイプのセットを選択できます。

サブジェクト結果の有効時間

この属性は、キャッシュされたサブジェクト結果を使用して、シングルサインオントークンに基づく同じポリシー要求を評価できる時間 (分単位) を指定します。

ポリシーが SSO トークンに対して最初に評価される場合に、そのポリシー内のサブジェクトインスタンスが評価され、ポリシーを特定のユーザーに適用できるかどうかが判別されます。サブジェクト結果は SSO トークン ID に合わされ、ポリシーにキャッシュされます。「サブジェクト結果の有効時間」属性で指定された時間内に、同一の SSO トークン ID に対する同一のポリシーで別の評価が発生した場合、ポリシーフレームワークはキャッシュされたサブジェクト結果を検索します。サブジェクトインスタンスは評価しません。これにより、ポリシー評価の時間が大幅に短縮されます。

ユーザーエイリアスを有効

リモート Directory Server でリソースのサブジェクトのメンバーがローカルユーザーをエイリアス化するとき、そのリソースを保護するためのポリシーを作成する場合は、この属性を有効にする必要があります。

リモート Directory Server で `uid=rmuser` を作成し、Access Manager で `rmuser` をエイリアスとしてローカルユーザーに追加する (`uid=luser` など) ような場合、この属性は有効でなければなりません。`rmuser` としてログインすると、ローカルユーザー (`luser`) でセッションが作成され、ポリシーの適用が成功します。

第 41 章

SAML サービス属性

SAML (Security Assertion Markup Language) サービス属性はグローバル属性です。これらの属性に適用される値は Sun Java System Acceess Manager 設定全体に適用され、設定済みのすべての組織に継承されます。グローバル属性の目的は Access Manager アプリケーションのカスタマイズであるため、ロールまたは組織に直接適用することはできません。

SAML サービスのアーキテクチャの詳細は、『Access Manager Developer's Guide』を参照してください。

SAML 属性は次のとおりです。

サイト ID とサイト発行者名

この属性にはエントリのリストが含まれ、各エントリにはインスタンス ID、サイト ID、およびサイト発行者名が含まれます。インストール時にはデフォルト値が割り当てられます。形式は次のとおりです。

`instanceid=serverprotocol://servername:portnumber|siteid=`*site_id*
`|issuerName=`*site_issuer_name*

ソースサイトと目的サイトの両方で SSL 用にこの属性を設定したら、`instanceid` のプロトコルが `HTTPS//` であることを確認してください。

署名 SAML 要求

この属性は、配信前にすべての SAML 要求にデジタル署名 (`XML DSIG`) するかどうかを指定します。このオプションをクリックすると、この機能が有効になります。

署名 SAML 応答

この属性は、配信前にすべての SAML 応答にデジタル署名 (XML DSIG) するかどうかを指定します。このオプションをクリックすると、この機能が有効になります。

このオプションを有効にするかどうかにかかわらず、SAML Web Post プロファイルが使用するすべての SAML 応答にデジタル署名が行われます。

署名アサーション

この属性は、配信前にすべての SAML アサーションにデジタル署名 (`XML DSIG`) するかどうかを指定します。このオプションをクリックすると、この機能が有効になります。

SAML アーティファクト名

この属性は、SAML サービス設定で定義されている SAML アーティファクトに変数名を割り当てます。SAML アーティファクトはサイズ制限付きのデータであり、アサーションとソースサイトを特定します。URL の照会文字列の一部として送られ、目的サイトへのリダイレクトによって転送されます。デフォルト値は SAMLart です。たとえば、デフォルトの SAMLart サービス設定を使用する場合、リダイレクトの照会文字列は次のようになります。

```
http:/ ホスト : ポート / 配備
_URI/SamlAwareServlet?TARGET=http://URL/&SAMLart=artifact123
```

ターゲット指定子

この属性は、リダイレクトに使用される目的サイトの URL に変数名を割り当てます。デフォルト値は Target です。

アーティファクトのタイムアウト

この属性は、アーティファクト用に作成したアサーションのタイムアウトを指定します。デフォルトは 400 です。

notBefore 時間のアサーションスキュー係数

この属性を使用して、アサーションの notBefore 時間を計算します。たとえば、IssueInstant が 2002-09024T21:39:49Z で、「notBefore 時間のアサーションスキュー係数」の値が 300 秒 (180 秒がデフォルト値) に設定されている場合、アサーションの条件要素の notBefore 属性は 2002-09-24T21:34:49Z になります。

アサーションのタイムアウト

この属性は、アサーションのタイムアウトが発生するまでの秒数を指定します。デフォルトは 420 です。

注	アサーションの有効な総時間は、「notBefore 時間のアサーションスキュー係数」属性、および「アサーションのタイムアウト」属性の両方の値に設定された値で決まります。

信頼パートナーサイト

この属性は、あるサイトが別のパートナーサイトと信頼関係を確立して通信できるように、パートナーの情報を保存します。

この属性にはエントリのリストが含まれ、各エントリにはキーとその値が「|」記号で区切られたペアの形で含まれます。各エントリにはソース ID が必要です。次に例を示します。

`SourceID=siteid|SOAPURL=https://servername:portnumber/amserver/SAMLSOAPReceiver|AuthType=SSL|hostlist=ipaddress` （または、サーバーの DNS 名または証明書エイリアス）

パラメータは次のとおりです。

表 41-1 信頼パートナーサイトのパラメータ

`SourceID`	「サイト ID とサイト発行者名」と同様に定義される 20 バイトのシーケンス。

target	このパラメータは、特定のドメインとして定義されます。ポート番号を含む場合と含まない場合とがあります。特定のドメインにホスティングされている Web ページにアクセスしたい場合、target はその後の処理のためパラメータ SAMLUrl または POSTUrl で定義される URL へのリダイレクトを指定します。 「信頼パートナーサイト」属性に指定された同一のドメインを持つエントリで、ポート番号を含むものと含まないものの 2 つのエントリがある場合、ポート番号を含むエントリが優先されます。 たとえば次のように、信頼されたパートナーサイトの定義が 2 つある場合を考えます。 target=sun.com|SAMLUrl=http://machine1.sun.com:8080/amserver/SAMLAwareServlet および target=sun.com:8080|SAMLUrl=httyp://machine2.sun.com:80/amserver/SAMLAwareServlet 両方とも次のページを検索しているものとします。 http://somemachine.sun.com:8080/index.html 上記の場合、一致するドメインとポートの両方が 2 番目の定義の target パラメータ内にあるので、2 番目の定義が SAML サービスプロバイダとして選択されます。
SAMLUrl	SAML サービスを提供する URL を定義します。URL に定義されたサーブレットは、「OASIS-SAML Bindings and Profiles」仕様に定義された「Web-browser SSO with Artifact」プロファイルを実装します。
POSTUrl	SAML サービスを提供する URL を定義します。URL に定義されたサーブレットは、「OASIS-SAML Binding and Profiles」仕様に定義された「Web-browser SSO with POST」プロファイルを実装します。
issuer	Access Manager 内で生成されたアサーションの作成者を定義します。構文は hostname:port です。
SOAPUrl	SOAP 受信者サービス URL を指定します。

AuthType	SAML で使用する認証タイプを定義します。次のいずれかになります。 • NOAUTH • BASICAUTH • SSL • SSLWITHBASICAUTH このパラメータは省略可能です。指定しない場合、デフォルトは NOAUTH です。 BASICAUTH または SSLWITHBASICAUTH が指定されている場合、User パラメータは必須で、SOAPUrl には HTTPS を指定する必要があります。
User	パートナーの SOAP 受信者の保護に使用するパートナーの uid を定義します。
version	SAML 要求を送信するために使用する SAML のバージョンを定義します。SAML バージョンには、1.0 または 1.1 のどちらかを指定します。このパラメータを定義しない場合は、AMConfig.properties からの次のデフォルト値が使用されます。 com.example.identity.saml.asertion.version=1.1 com.example.identity.saml.protocol.version=1.1
hostlist	この属性は、特定のパートナーサイトに対して要求を送信可能なすべてのホストの IP アドレスと certAlias の両方または一方を一覧表示します。これによって、要求の送信者が確実に SAML アーティファクトの本来の受信者であると保証されます。 要求者のホストまたはクライアントの証明書が、受信者のサイトでこのリストに含まれている場合は、サービスが続行されます。ホストまたはクライアントの証明書が、このホストリストに含まれているどのホストや証明書にも一致しない場合、SAML サービスは要求を拒否します。
AccountMapper	アサーションのサブジェクトと目的サイトでの位置付けとを関連付ける方法を定義する、プラグイン可能なクラスを指定します。デフォルトでは、次のようになります。 com.sun.identity.saml.plugins.DefaultAccountMapper

PartnerAccountMapper	クラス PartnerAccountMapper は、Sun Java System Access Manager でパートナーアカウントをユーザーアカウントにマップするために実装されているインタフェースです。
attributeMapper	attributeMapper がある場所へのパスを持つクラスを指定します。アプリケーションは、attributeMapper を展開して、SSOToken ID または AuthenticationStatement を含むアサーションを照会から取得できます。その後、マッパーを使用してサブジェクトの属性を取得します。attributeMapper が指定されていない場合は、DefaultAttributeMapper が使用されます。
actionMapper	actionMapper がある場所へのパスを持つクラスを指定します。アプリケーションは、actionMapper を展開して、SSOToken ID または AuthenticationStatement を含むアサーションを照会から取得できます。その後、マッパーを使用して、照会に定義されているアクションの認証決定を取得します。actionMapper が指定されていない場合は、DefaultActionMapper が使用されます。
siteAttributeMapper	siteAttributeMapper がある場所へのパスを持つクラスを指定します。アプリケーションは、siteAttributeMapper を展開して、SSO 中にアサーションに組み込む属性を取得できます。siteAttributeMapper が見つからない場合、属性は SSO 中にアサーションに組み込まれません。
PartnerSiteAttributeMapper	ブラウザアーティファクトと POST プロファイルの SSO シナリオ中にパートナーに返される認証アサーションの一部として、AttributeStatements 要素として返されることを要求される属性オブジェクトのリストを返すには、パートナーサイトでこのインタフェースを実装する必要があります。
certAlias=*aliasName*	パートナーがアサーションに署名しているのに、署名されたアサーションの KeyInfo 部分にパートナーの証明書が見つからない場合に、アサーションで署名を検証するために使用する certAlias 名を指定します。

信頼されたパートナーサイトに対する設定例を、以下の表に示します。必ずしもすべてのパラメータを指定する必要はありません。省略可能なパラメータはカギかっこ ([ ]) で示しています。

	送信者	受信者
アーティファクト	sourceid	sourceid
	target	SOAPUrl
	SAMLUrl	[accountMapper]
	hostlist	[AuthType]
	[siteAttributeMapper]	[User]
		[certAlias]
POST プロファイル	sourceid	sourceid
	target	issuer
	POSTUrl	[accountMapper]
	[siteAttributeMapper]	[certAlias]
SOAP 要求		sourceid
		hostlist
		[attributeMapper]
		[actionMapper]
		[certAlias]
		[issuer]

ターゲット URL への POST

アーティファクトプロファイルまたは POST プロファイルの SSO 経由でサイトによって受信されたターゲット URL がこの属性に含まれている場合、SSO から受信したアサーションは http: FORM POST によってターゲット URL に送信されます。POST にテストの URL またはその他の URL は使用しないでください。

第 42 章

セッションサービス属性

セッションサービス属性はグローバルおよびダイナミック属性です。グローバル属性に適用される値は Access Manager の設定全体に適用され、設定済みのすべての組織に継承されます。グローバル属性の目的は Access Manager アプリケーションのカスタマイズなので、セッションサービス属性をロールまたは組織に直接適用することはできません。

ダイナミック属性に適用される値は、ロールまたは組織に適用されます。ロールがユーザーに適用されると、またはユーザーが組織に割り当てられると、これらの属性はデフォルトでユーザーに継承されます。デフォルトセッションの値は、サービス設定で Access Manager のすべての登録組織に対して設定されます。これらの値は、セッションサービスを特定の組織に登録し、テンプレートを作成して、デフォルト値以外の値を入力することによって、個々の組織に対して異なる設定にすることができます。

セカンダリ設定インスタンス

インスタンス名

このフィールドは、セカンダリインスタンスの名前を定義します。

セッションストアユーザー

このフィールドは、セッションデータを取得して保存するために使用されるデータベースユーザーを定義します。

セッションストアパスワード

このフィールドは、「セッションストア」で定義したデータベースユーザーのパスワードを定義します。

セッションストアパスワード (確認)

パスワードを確認します。

セッションクラスタサーバーリスト

この属性は、同じセッションのフェイルオーバークラスタに参加する Access Manager サーバーインスタンスの一意の ID (2 バイト値、プラットフォームサービスのサーバーリストのエントリに対応) を一覧表示します。

最大待ち時間

このフィールドは、JDBC 接続オブジェクトを取得するまでスレッドが待機する合計時間を定義します。値はミリ秒単位で指定します。

JDBC ドライバ実装クラス

このフィールドは、JDBC 接続プールのセットアップに使用するリポジトリ依存ファクトリクラスの名前を指定します。Access Manager には、出荷時に HADB および Oracle の両方の実装クラスが組み込まれています。

JDBC URL

このフィールドは JDBC の URL を指定します。

最小プールサイズ

この属性は、接続プールで作成する JDBC 接続の最小数を定義します。

最大プールサイズ

この属性は、接続プールで作成する JDBC 接続の最大数を定義します。

グローバル属性

グローバル属性は次のとおりです。

- 389 ページの「検索結果の最大数」
- 389 ページの「検索のタイムアウト (秒)」

検索結果の最大数

この属性はセッション検索で返される結果の最大数を指定します。デフォルト値は 120 です。

検索のタイムアウト (秒)

この属性はセッション検索を終了するまでの最大時間を定義します。デフォルト値は 5 秒です。

ダイナミック属性

ダイナミック属性は次のとおりです。

- 390 ページの「最大セッション時間 (分)」
- 390 ページの「最大アイドル時間 (分)」
- 390 ページの「最大キャッシュ時間 (分)」

最大セッション時間 (分)

この属性は、セッションが期限切れになるまでの最大時間を分単位で表す値を指定します。期限が切れると、ユーザーはアクセスするために再度認証を受ける必要があります。有効な値は 1 以上です。デフォルト値は 120 です。セキュリティと利便性の要求のバランスをとるためには、最大セッション時間の間隔にはより大きい値を設定し、最大アイドル時間の間隔には比較的小さい値を設定してください。最大セッション時間では、セッションの有効期限を指定します。設定した値を超えて延長されることはありません。

最大アイドル時間 (分)

この属性は、セッションが期限切れになるまでのアクティビティのない最大時間に等しい値 (分単位) を指定します。期限が切れると、ユーザーはアクセスするために再度認証を受ける必要があります。有効な値は 1 以上です。デフォルト値は 30 です。セキュリティと利便性の要求のバランスをとるためには、最大セッション時間の間隔にはより大きい値を設定し、最大アイドル時間の間隔には比較的小さい値を設定してください。

最大キャッシュ時間 (分)

この属性は、クライアントが Access Manager と通信してキャッシュされたセッション情報を更新するまでの最大間隔に等しい値 (分単位) を指定します。有効な値は 0 以上です。デフォルト値は 3 です。最大キャッシュ時間は常に最大アイドル時間より小さくなるように設定してください。

第 43 章

SOAP バインドサービス属性

SOAP バインドサービス属性はグローバル属性です。これらの属性に適用される値は Sun Java System Acceess Manager 設定全体に適用され、設定済みのすべての組織に継承されます。グローバル属性の目的は Access Manager アプリケーションのカスタマイズであるため、ロールまたは組織に直接適用することはできません。

SOAP バインドサービス属性は次のとおりです。

要求ハンドラリスト

この属性は、Access Manager に配備された Web サービスプロバイダ (WSP) についての情報を格納します。「|」記号で区切られた、キーと値のペアが含まれるエントリを一覧表示します。次に例を示します。

```
key=disco|class=com.example.identity.liberty.ws.disco.DiscoveryServ
ice|soapActions=sa1 sa2 sa2
```

新しい要求ハンドラを追加するには、「追加」ボタンをクリックします。キーパラメータとクラスパラメータが必要です。パラメータは次のとおりです。

key: WSP の SOAP エンドポイントの URI パスの 2 番目の部分を定義します。最初の部分は、SOAP サービスが Liberty として定義しています。たとえば、disco をキーとして定義すると、ディスカバリサービスの SOAP エンドポイントは次のようになります。

プロトコル :// ホスト名 : ポート / 配備 _URI/Liberty/disco

class: このパラメータは、WSP の実装クラスの名前を指定します。Liberty SOAP レイヤーは、要求されたメッセージを処理してから応答を返すために各 WSP が実装する、ハンドラインタフェースを提供します。

soapActions: サポートする SOAPActions を指定するパラメータです。このパラメータはオプションです。このパラメータを指定しない場合は、すべての SOAPActions をサポートします。Web サービスコンシューマ (WSC) がサポートされていない SOAPAction で要求を送信すると、その要求は SOAP レイヤーに拒否され、対応する WSP に渡されません。

Web サービス認証

この属性は、`WebServiceAuthenicator` インタフェースの実装クラスを定義します。要求に基づいて、Web サービスコンシューマ (WSC) に対する資格情報を認証および生成します。

サポートされている認証メカニズム

この属性は、SOAP エンドポイントがサポートされている認証メカニズムを指定します。デフォルトでは、すべてのメカニズムが選択されています。選択されていない認証メカニズムを使用して WSC が要求を送信すると、その要求は SOAP レイヤーに拒否され、対応する WSP に渡されません。

第 44 章

ユーザー属性

ユーザー属性を格納する場所は、「サービス設定」ウィンドウと「ユーザー管理」ウィンドウの 2 箇所です。「サービス設定」ウィンドウには、登録されている組織のデフォルト属性が含まれます。「ユーザー管理」ウィンドウには、ユーザーエントリ属性が含まれます。

ユーザーサービス属性

ユーザーサービス属性はダイナミック属性です。ダイナミック属性に適用される値は、Access Manager で設定されるロールまたは組織に割り当てられます。ロールがユーザーに割り当てられるか、ユーザーが組織に割り当てられる場合は、ダイナミック属性がそのユーザーの特性になります。ユーザー属性は次のように分けられます。

- ユーザー設定言語
- ユーザー設定タイムゾーン
- 継承するロケール
- 管理者 DN 開始表示
- デフォルトユーザー状態

デフォルトユーザーの値は、Access Manager のすべての登録組織に対して設定されます。これらの値は、ユーザーサービスを特定の組織に登録し、テンプレートを作成して、デフォルト値以外の値を入力することによって、個々の組織に対して異なる設定にすることができます。

ユーザー設定言語

このフィールドは、Access Manager コンソールに表示されるテキスト言語に関するユーザーの選択項目を指定します。デフォルト値は en です。この値によってユーザーセッションへの地域対応化キーのセットがマップされて、画面のテキストがユーザーに合った言語で表示されます。

ユーザー設定タイムゾーン

このフィールドは、ユーザーが Access Manager コンソールにアクセスするタイムゾーンを指定します。デフォルト値はありません。

継承するロケール

このフィールドは、ユーザーのロケールを指定します。デフォルト値は en_US です。285 ページの表 20-1 のすべての値を使用できます。

管理者 DN 開始表示

このユーザーが Access Manager 管理者の場合、このフィールドはユーザーがログインするときに Access Manager コンソールに表示される開始点となるノードを指定します。デフォルト値はありません。ユーザーが少なくとも読み取りアクセス権を持っている有効な DN を使用することができます。

デフォルトユーザー状態

このオプションは、新しく作成したユーザーのデフォルト状態を示します。ユーザーエントリ状態の方がこの状態よりも優先されます。アクティブなユーザーだけが Access Manager を使用して認証を受けることができます。デフォルト値は「アクティブ」です。プルダウンメニューから次のどちらかを選択することができます。

- アクティブ - ユーザーは Access Manager を使用して認証を受けることができます。
- 非アクティブ - ユーザーは Access Manager を使用して認証を受けることはできませんが、ユーザープロファイルはそのままディレクトリに格納されます。

個々のユーザー状態は、ユーザーサービスを登録し、値を選択してロールに適用し、そのロールをユーザープロファイルに追加することによって設定します。

ユーザープロファイル属性

ユーザープロファイル属性はユーザープロファイルのデフォルト属性です。この値は、管理者またはユーザーがログイン時にユーザープロファイル表示で設定します。管理者は、自分のユーザー属性をユーザープロファイルに追加したり、新しいサービスを作成したりできます。詳細は、『Access Manager Developer's Guide』を参照してください。

注 Access Manager ではユーザーエントリ内の属性の一意性は必ずしも必要ではありません。たとえば `userA` と `userB` はどちらも同じ組織で作成されているとします。どちらの場合も、電子メールアドレス属性を `jimb@madisonparc.com` に設定できます。管理者は、Sun Java System Directory Server の属性の一意性プラグインを設定することによって、一意な属性値になるようにすることができます。詳細は、この章の末尾にある「ユーザー ID の一意性」または『Sun Java System Directory Server 管理ガイド』を参照してください。

名 (ファーストネーム)

このフィールドはユーザーの名 (ファーストネーム) を取得します。名 (ファーストネーム) の値と姓 (ラストネーム) の値によって、Access Manager コンソールの右上隅にあるログイン名を示すフィールドのユーザーが識別されます。

姓 (ラストネーム)

このフィールドはユーザーの姓 (ラストネーム) を取得します。名 (ファーストネーム) の値と姓 (ラストネーム) の値によって、Access Manager コンソールの右上隅にあるログイン名を示すフィールドのユーザーが識別されます。

フルネーム

このフィールドはユーザーのフルネームを取得します。

パスワード

このフィールドは、ユーザー ID フィールドで指定した名前のパスワードを取得します。

パスワード (確認)

パスワードを確認します。

電子メールアドレス

このフィールドはユーザーの電子メールアドレスを取得します。

社員番号

このフィールドはユーザーの社員番号を取得します。

電話番号

このフィールドはユーザーの電話番号を取得します。

ホームアドレス

このフィールドはユーザーのホームアドレスを取得します。

ユーザー状態

このオプションは、Access Manager による認証をユーザーに許可するかどうかを指定します。アクティブなユーザーだけが Access Manager を使用して認証を受けることができます。デフォルト値は「アクティブ」です。プルダウンメニューから次のどちらかを選択することができます。

- アクティブ - ユーザーは Access Manager を使用して認証を受けることができます。
- 非アクティブ - ユーザーは Access Manager を使用して認証を受けることはできませんが、ユーザープロファイルはそのままディレクトリに格納されます。

注 ユーザー状態を「非アクティブ」に変えても、Access Manager による認証に影響するだけです。Directory Server は、nsAccountLock 属性を使用してユーザーアカウント状態を判別します。認証を無効にしたユーザーアカウントでも、Access Manager を必要としないタスクは実行できます。Access ManagerAccess Manager 認証だけではなく、ディレクトリのユーザーアカウントも無効にするには、nsAccountLock の値を「true」に設定します。サイトの委託管理者がユーザーを定期的に無効にしている場合は、nsAccountLock 属性を Access Manager ユーザープロファイルのページに追加することを考慮してください。詳細は、『Access Manager Developer's Guide』を参照してください。

アカウント有効期限

この属性が存在する場合、指定されたアカウント有効期限が現在の日付以前であれば、認証サービスはログインを無効にします。この属性の形式は次のとおりです。

```
(mm/dd/yyyy hh:mm)
```

ユーザー認証設定

この属性は、ユーザーの認証方法を設定します。デフォルトの認証方法は LDAP です。1 つまたは複数の認証方法を、「編集」リンクをクリックすることによって選択できます。複数の認証方法を選択した場合、選択した方法すべてに対してユーザーは認証に成功する必要があります。

ユーザーエイリアスリスト

このフィールドは、ユーザーに適用される可能性のあるエイリアスを定義します。この属性に設定されたエイリアスを使用するために、iplanet-am-user-alias-list 属性を LDAP サービスのユーザーエントリ検索属性フィールドに追加して、LDAP サービスを修正する必要があります。

設定ロケール

このフィールドは、ユーザーのロケールを指定します。デフォルト値は en_US です。285 ページの表 20-1 のすべての値を使用できます。

プルダウンメニューで次の属性のどれかを選択できます。

- 無視
- カスタマイズ
- 継承

成功 URL

このフィールドには、認証成功後にユーザーをリダイレクトする URL を指定する、複数の値のリストを入力します。この属性の形式は、クライアントタイプ | URL ですが、URL の値のみを指定できます。その際、URL のタイプはデフォルトで HTML となることが前提とされています。

失敗 URL

このフィールドには、認証が失敗した場合にユーザーをリダイレクトする URL を指定する、複数の値のリストを入力します。この属性の形式は、クライアントタイプ | URL ですが、URL の値のみを指定できます。その際、URL のタイプはデフォルトで HTML となることが前提とされています。

ユーザー ID の一意性

Access Manager アプリケーション内で uid の一意性を実現するには、Directory Server で利用可能なプラグインを次のように設定する必要があります。

```
dn: cn=uid uniqueness,cn=plugins,cn=config

objectClass: top

objectClass: nsSlapdPlugin

objectClass: extensibleObject

cn: uid uniqueness

nsslapd-pluginPath: /ids908/lib/uid-plugin.so

nsslapd-pluginInitfunc: NSUniqueAttr_Init
```

```
nsslapd-pluginType: preoperation
nsslapd-pluginEnabled: on
nsslapd-pluginarg0: attribute=uid
nsslapd-pluginarg1: markerObjectClass=nsManagedDomain
nsslapd-plugin-depends-on-type: database
nsslapd-pluginId: NSUniqueAttr
nsslapd-pluginVersion: 6.1
nsslapd-pluginVendor: Sun | SunONE
nsslapd-pluginDescription: Enforce unique attribute values
```

nsManagedDomain オブジェクトクラスは、uid の一意性を必要とする組織にマークを付けるために使用することをお勧めします。プラグインは、デフォルトでは有効ではありません。

組織ごとに uid の一意性を設定するには、プラグインエントリに各組織の DN を追加するか、またはマーカーオブジェクトクラスオプションを使用して nsManagedDomain を最上位レベルの組織エントリのそれぞれに追加します。

```
nsslapd-pluginEnabled: on
nsslapd-pluginarg0: attribute=uid
nsslapd-pluginarg1: markerObjectClass=nsManagedDomain
```

エラーコード

この付録では、Sun Java System Acceess Manager によって生成されるエラーメッセージのリストを示します。このリストにすべてが網羅されているわけではありませんが、この章の情報は一般的な問題に対処するための開始点として役立ちます。この付録の各表には、エラーコード、エラーの説明や考えられる原因、および、発生した問題を修正する方法が示されています。

この付録では、次の機能分野に関連するエラーコードのリストを示します。

エラー診断に支援が必要な場合は、次の Web サイトから Sun テクニカルサポートに連絡してください。

http://www.sun.com/service/sunone/software/index.html

Access Manager コンソールのエラー

次の表は、Access Manager コンソールによって生成され表示されるエラーコードのリストです。

表 A-1 Access Manager コンソールのエラー

エラーメッセージ	説明 / 考えられる原因	対処方法
次のものを削除中にエラーが発生しました。	現在のユーザーが削除を行う前に、そのオブジェクトはほかのユーザーによって削除された可能性があります。	削除しようとしているオブジェクトを再表示し、操作をやり直します。

表 A-1　Access Manager コンソールのエラー (続き)

エラーメッセージ	説明 / 考えられる原因	対処方法
入力した URL が無効です。	Access Manager コンソールウィンドウの URL が正しく入力されなかった場合に発生します。	
検索条件と一致するエントリがありません。	検索ウィンドウまたはフィルタフィールドに入力されたパラメータが、ディレクトリ内のどのオブジェクトにも一致しませんでした。	パラメータを変更して検索をやり直します。
表示する属性がありません。	選択されたオブジェクトのスキーマには、編集可能な属性が定義されていません。	
このサービスのために表示する情報がありません。	サービス設定モジュールから表示するサービスに、グローバル属性または組織ベースの属性がありません。	
検索サイズの上限を超えました。検索を絞り込んでください。	指定されたパラメータによる検索で、許容数を超えるエントリが返されました。	管理サービスの「検索で返される結果の最大数」属性を、より大きな値に修正します。検索パラメータをより厳しい条件に修正することもできます。
検索時間が指定された時間を過ぎました。検索を絞り込んでください。	指定されたパラメータによる検索に、許容値より長い検索時間がかかりました。	管理サービスの「検索のタイムアウト」属性を、より大きな値に修正します。より多数の値を取得するために、検索パラメータをより緩やかな条件に修正することもできます。
ユーザーの開始位置が無効です。管理者に連絡してください。	ユーザーエントリの開始位置 DN が無効です。	ユーザープロファイルページで、開始 DN の値を有効な DN に変更します。
アイデンティティオブジェクトを作成できませんでした。ユーザーに適切なアクセス権がありません。	必要なアクセス権を持っていないユーザーが操作を実行しました。ユーザーが実行できる操作は、持っているアクセス権によって決定します。	

認証エラーコード

次の表は、認証サービスによって生成されるエラーコードのリストです。これらのエラーは、認証モジュールでユーザーや管理者に表示されます。

表 A-2 認証エラーコード

エラーメッセージ	説明 / 考えられる原因	対処方法
authentication.already.login.	ユーザーはすでにログインし、有効なセッションを持っているが、成功の場合のリダイレクト URL が定義されていません。	ログアウトするか、Access Manager コンソールを使ってログイン成功リダイレクト URL をセットアップします。管理コンソール URL として、この値に 'goto' 照会パラメータを使用します。
logout.failure.	ユーザーが Access Manager からログアウトできません。	サーバーを再起動します。
uncaught_exception	不正なハンドラが原因で、認証の例外がスローされました。	ログイン URL に無効な文字や特殊文字が含まれていないかどうかをチェックします。
redirect.error	Access Manager で成功 URL または失敗 URL にリダイレクトできません。	Web コンテナのエラーログをチェックして、エラーがないかどうかを確認します。
gotoLoginAfterFail	このリンクは、ほとんどのエラー発生時に生成されます。ユーザーはこのリンクをクリックして、元のログイン URL ページに戻ります。	
invalid.password	入力したパスワードが無効です。	パスワードは 8 文字以上である必要があります。パスワードに適切な文字数が含まれていることと、パスワードの有効期限が切れていないことを確認します。
auth.failed	認証に失敗しました。これは汎用のエラーメッセージであり、デフォルトのログイン失敗テンプレートで表示されます。もっとも一般的な原因は、クレデンシャルが無効または不正であることです。	有効なユーザー名とパスワード (呼び出される認証モジュールに対して必要なクレデンシャル) を正しく入力します。

表 A-2　認証エラーコード (続き)

エラーメッセージ	説明 / 考えられる原因	対処方法
nouser.profile	その組織には、入力されたユーザー名に一致するユーザープロファイルが見つかりませんでした。このエラーは、メンバーシップ (自己登録) 認証モジュールにログインするときに表示されます。	ログイン情報を入力し直します。はじめてログインする場合は、ログイン画面で「新規ユーザー」を選択します。
notenough.characters	入力されたパスワードの文字数が不足しています。このエラーは、メンバーシップ (自己登録) 認証モジュールにログインするときに表示されます。	デフォルトでは、ログインパスワードは 8 文字以上である必要があります。この値は、メンバーシップ認証モジュールを通して設定できます。
useralready.exists	その組織には、この名前を持つユーザーがすでに存在しています。このエラーは、メンバーシップ (自己登録) 認証モジュールにログインするときに表示されます。	ユーザー ID は組織内で一意にする必要があります。
uidpasswd.same	「ユーザー名」と「パスワード」のフィールドは同じ値にすることはできません。このエラーは、メンバーシップ (自己登録) 認証モジュールにログインするときに表示されます。	ユーザー名とパスワードは必ず異なる値にします。
nouser.name	ユーザー名が入力されませんでした。このエラーは、メンバーシップ (自己登録) 認証モジュールにログインするときに表示されます。	必ずユーザー名を入力します。
no.password	パスワードが入力されていません。このエラーは、メンバーシップ / 自己登録認証モジュールにログインするときに表示されます。	必ずパスワードを入力します。
missing.confirm.passwd	パスワードの確認フィールドが入力されていません。このエラーは、メンバーシップ (自己登録) 認証モジュールにログインするときに表示されます。	必ず「パスワードの確認」フィールドにパスワードを入力します。
password.mismatch	パスワードと確認のパスワードが一致しません。このエラーは、メンバーシップ (自己登録) 認証モジュールにログインするときに表示されます。	パスワードと確認のパスワードは必ず同じ値にします。

表 A-2　認証エラーコード (続き)

エラーメッセージ	説明 / 考えられる原因	対処方法
An error occurred while storing the user profile.	ユーザープロファイルの格納時にエラーが発生しました。このエラーは、メンバーシップ (自己登録) 認証モジュールにログインするときに表示されます。	`Membership.xml` ファイル内の「自己登録」の属性と要素が有効で正しいことを確認します。
orginactive	この組織はアクティブではありません。	Access Manager コンソールを使って、組織の状態を非アクティブからアクティブに変更します。
internal.auth.error	内部認証エラー。これは一般的な認証エラーであり、さまざまな環境や設定の問題によって発生します。	
usernot.active	ユーザーの状態はアクティブでなくなっています。	管理コンソールを使って、ユーザーの状態を非アクティブからアクティブに変更します。 メモリロックによってユーザーがロックアウトされている場合は、サーバーを再起動します。
user.not.inrole	ユーザーは、指定されたロールには属していません。このエラーは、ロールベースの認証で表示されます。	ロールベースの認証に指定されているロールに、ログインするユーザーが属していることを確認します。
session.timeout	ユーザーのセッションがタイムアウトしました。	ログインし直します。
authmodule.denied	指定された認証モジュールは拒否されています。	要求された認証モジュールが要求された組織で登録されていること、そのモジュールのテンプレートが作成され保存されていること、および、コア認証モジュールの「組織認証モジュール」リストでそのモジュールが選択されていることを確認します。
noconfig.found	設定が見つかりません。	認証設定サービスをチェックして、必要な認証方法があるかどうかを確認します。
cookie.notpersistent	持続 Cookie ユーザー名が、持続 Cookie ドメインに存在しません。	

表 A-2　認証エラーコード (続き)

エラーメッセージ	説明 / 考えられる原因	対処方法
nosuch.domain	その組織が見つかりません。	有効な組織を正しく入力します。
userhasnoprofile.org	ユーザーには、指定された組織におけるプロファイルがありません。	ローカル Directory Server 内で、指定された組織にそのユーザーが存在し、有効になっていることを確認します。
reqfield.missing	必須フィールドのどれかが未記入のままになっています。必ずすべての必須フィールドに入力します。	必ずすべての必須フィールドに入力します。
session.max.limit	大セッション数の限度に達しました。	ログアウトし、ログインし直します。

ポリシーエラーコード

次の表は、ポリシーフレームワークによって生成され、Access Manager コンソールに表示されるエラーコードのリストです。

表 A-3　ポリシーエラーコード

エラーメッセージ	説明 / 考えられる原因	対処方法
illegal_character_/_in_name	ポリシー名に不正な文字「/」が含まれています。	ポリシー名に「/」という文字が含まれていないことを確認します。
policy_already_exists_in_org	同じ名前を持つルールがすでに存在しています。	別の名前を使ってポリシーを作成します。
rule_name_already_present	同じ名前を持つルールがすでに存在しています。	別のルール名を使ってポリシーを作成します。
rule_already_present	同じルール値を持つルールがすでに存在しています。	別のルール値を使用します。
no_referral_can_not_create_policy	組織 {0} への参照が存在しません	組織への参照が存在しません。サブ組織にポリシーを作成するには、その親組織に参照ポリシーを作成して、このサブ組織に対して参照可能なリソースを示す必要があります。

表 A-3　ポリシーエラーコード (続き)

エラーメッセージ	説明 / 考えられる原因	対処方法
ldap_search_exceed_size_limit	LDAP 検索のサイズの上限を超えました。検索で見つかった結果が最大数を超えたのでエラーが発生しました。	検索制御パラメータの組織の検索パターンまたはポリシーの設定を変更します。検索サイズの上限は、ポリシー設定サービスにあります。
ldap_search_exceed_time_limit	LDAP 検索の時間の上限を超えました。検索で見つかった結果が最大数を超えたのでエラーが発生しました。	検索制御パラメータの組織の検索パターンまたはポリシーの設定を変更します。最大セッション時間は、ポリシー設定サービスにあります。
ldap_invalid_password	無効な LDAP バインドパスワード。	ポリシー設定で定義されている LDAP バインドユーザーのパスワードが間違っています。これが原因で、ポリシー操作を実行するための認証済み LDAP 接続を取得できません。
app_sso_token_invalid	アプリケーション SSO トークンが無効です。	サーバーがアプリケーション SSO トークンを検証できませんでした。SSO トークンの有効期限が切れている可能性があります。
user_sso_token_invalid	ユーザー SSO トークンが無効です。	サーバーがユーザー SSO トークンを検証できませんでした。SSO トークンの有効期限が切れている可能性があります。
property_is_not_an_Integer	プロパティ値が整数ではありません。	このプラグインのプロパティ値は整数にする必要があります。
property_value_not_defined	プロパティ値を定義する必要があります。	そのプロパティに値を指定します。
start_ip_can_not_be_greater_than_end_ip	開始 IP が終了 IP より大きくなっています。	IP アドレス条件に、終了 IP アドレスより大きい開始 IP アドレスを設定しようとしました。開始 IP を終了 IP より大きくすることはできません。
start_date_can_not_be_larger_than_end_date	開始日が終了日より大きくなっています。	ポリシーの時間条件に、終了日より大きい開始日を設定しようとしました。開始日を終了日より大きくすることはできません。

表 A-3　ポリシーエラーコード (続き)

エラーメッセージ	説明 / 考えられる原因	対処方法
policy_not_found_in_organization	組織内にそのポリシーが見つかりません。組織内に存在していないポリシーを見つけようとしてエラーが発生しました。	指定された組織にそのポリシーが存在していることを確認します。
insufficient_access_rights	ユーザーに適切なアクセス権がありません。ユーザーは、ポリシー操作を実行するために必要なアクセス権を持っていません。	適切なアクセス権を持っているユーザーでポリシー操作を実行します。
invalid_ldap_server_host	無効な LDAP サーバーホスト。	ポリシー設定サービスに入力された無効な LDAP サーバーホストを変更します。

amadmin エラーコード

次の表は、`amadmin` コマンド行ツールによって生成され Access Manager のデバッグファイルに書き込まれるエラーコードのリストです。

表 A-4　amadmin エラーコード

エラーメッセージ	コード	説明 / 考えられる原因	対処方法
nocomptype	1	引数が足りません。	必須の引数 (`--runasdn`、`--password`、`--passwordfile`、`--schema`、`--data`、および `--addAttributes`) とそれぞれの値がコマンド行で指定されていることを確認します。
file	2	入力 XML ファイルが見つかりませんでした。	構文をチェックし、入力 XML ファイルが有効であることを確認します。
nodnforadmin	3	`--runasdn` の値としてユーザー DN が指定されていません。	`--runasdn` の値としてユーザー DN を指定します。
noservicename	4	`--deletservice` の値としてサービス名が指定されていません。	`--deleteservice` の値としてサービス名を指定します。
nopwdforadmin	5	`--password` の値としてパスワードが指定されていません。	`--password` の値としてパスワードを指定します。

表 A-4　amadmin エラーコード (続き)

エラーメッセージ	コード	説明 / 考えられる原因	対処方法
nolocalename	6	ロケール名が指定されませんでした。ロケールには en_US が指定されます。	ロケールのリストについては、「デフォルト認証ロケール」を参照してください。
nofile	7	入力 XML ファイルが指定されていません。	処理する入力 XML ファイルの名前を少なくとも 1 つ指定します。
invopt	8	1 つ以上の引数が間違っています。	すべての引数が有効であることを確認します。有効な引数を一覧表示するには、`amadmin --help` と入力します。
oprfailed	9	操作に失敗しました。	`amadmin` の失敗時には、このエラーを示す詳細なエラーコードが生成されます。これらのエラーコードを参照して問題を評価します。
execfailed	10	要求を処理できません。	`amadmin` の失敗時には、このエラーを示す詳細なエラーコードが生成されます。これらのエラーコードを参照して問題を評価します。
policycreatexception	12	ポリシーを作成できません。	`amadmin` では、このエラーを示す例外メッセージが生成されます。これらのメッセージを参照して問題を評価します。
policydelexception	13	ポリシーを削除できません。	`amadmin` では、このエラーを示す例外メッセージが生成されます。これらのメッセージを参照して問題を評価します。
smsdelexception	14	サービスを削除できません。	`amadmin` では、このエラーを示す例外メッセージが生成されます。これらのメッセージを参照して問題を評価します。
ldapauthfail	15	ユーザーを認証できません。	ユーザー DN とパスワードが正しいことを確認します。
parserror	16	入力 XML ファイルをパースできません。	XML の形式が正しく、`amAdmin.dtd` に従っていることを確認します。

表 A-4　amadmin エラーコード (続き)

エラーメッセージ	コード	説明 / 考えられる原因	対処方法
parseiniterror	17	アプリケーションエラーまたはパーサ初期化エラーのため、パースできません。	XML の形式が正しく、amAdmin.dtd に従っていることを確認します。
parsebuilterror	18	指定したオプションを持つパーサをビルドできないため、パースできません。	amadmin では、このエラーを示す例外メッセージが生成されます。これらのメッセージを参照して問題を評価します。
ioexception	19	入力 XML ファイルを読み取ることができません。	amadmin では、このエラーを示す例外メッセージが生成されます。これらのメッセージを参照して問題を評価します。
fatalvalidationerror	20	XML ファイルが有効なファイルではないため、パースできません。	構文をチェックし、入力 XML ファイルが有効であることを確認します。
nonfatalvalidationerror	21	XML ファイルが有効なファイルではないため、パースできません。	amadmin では、このエラーを示す例外メッセージが生成されます。これらのメッセージを参照して問題を評価します。
validwarn	22	ファイルに対する XML ファイル検証警告。	amadmin では、このエラーを示す例外メッセージが生成されます。これらのメッセージを参照して問題を評価します。
failedToProcessXML	23	XML ファイルを処理できません。	amadmin では、このエラーを示す例外メッセージが生成されます。これらのメッセージを参照して問題を評価します。
nodataschemawarning	24	--data オプションと --schema オプションのどちらもコマンド行に指定されていません。	すべての引数が有効であることを確認します。有効な引数を一覧表示するには、amadmin --help と入力します。
doctyperror	25	XML ファイルが正しい DTD に従っていません。	XML ファイルの DOCTYPE 要素を確認します。
statusmsg9	26	無効な DN、パスワード、ホスト名、またはポート番号が原因で LDAP 認証に失敗しました。	ユーザー DN とパスワードが正しいことを確認します。
statusmsg13	28	サービスマネージャ例外 (SSO 例外)。	amadmin では、このエラーを示す例外メッセージが生成されます。これらのメッセージを参照して問題を評価します。

表 A-4　amadmin エラーコード (続き)

エラーメッセージ	コード	説明 / 考えられる原因	対処方法
statusmsg14	29	サービスマネージャ例外。	amadmin では、このエラーを示す例外メッセージが生成されます。これらのメッセージを参照して問題を評価します。
statusmsg15	30	スキーマファイルの入力ストリーム例外。	amadmin では、このエラーを示す例外メッセージが生成されます。これらのメッセージを参照して問題を評価します。
statusmsg30	31	ポリシーマネージャ例外 (SSO 例外)。	amadmin では、このエラーを示す例外メッセージが生成されます。これらのメッセージを参照して問題を評価します。
statusmsg31	32	ポリシーマネージャ例外。	amadmin では、このエラーを示す例外メッセージが生成されます。これらのメッセージを参照して問題を評価します。
dbugerror	33	複数のデバッグオプションが指定されています。	デバッグオプションは 1 つだけ指定する必要があります。
loginFalied	34	ログインに失敗しました。	amadmin では、このエラーを示す例外メッセージが生成されます。これらのメッセージを参照して問題を評価します。
levelerr	36	属性値が無効です。	LDAP 検索に設定されているレベルを確認します。SCOPE_SUB または SCOPE_ONE である必要があります。
failToGetObjType	37	オブジェクトタイプの取得時にエラーが発生しました。	XML ファイル内の DN が値であることと、正しいオブジェクトタイプを持っていることを確認します。
invalidOrgDN	38	無効な組織 DN。	XML ファイル内の DN が有効であることと、組織オブジェクトであることを確認します。
invalidRoleDN	39	無効なロール DN。	XML ファイル内の DN が有効であることと、ロールオブジェクトであることを確認します。

表 A-4　amadmin エラーコード (続き)

エラーメッセージ	コード	説明 / 考えられる原因	対処方法
invalidStaticGroupDN	40	無効なスタティックグループ DN。	XML ファイル内の DN が有効であることと、スタティックグループオブジェクトであることを確認します。
invalidPeopleContainer DN	41	無効なピープルコンテナ DN。	XML ファイル内の DN が有効であることと、ピープルコンテナオブジェクトであることを確認します。
invalidOrgUnitDN	42	無効な組織単位 DN。	XML ファイル内の DN が有効であることと、コンテナオブジェクトであることを確認します。
invalidServiceHostName	43	無効なサービスホスト名。	有効なセッションを取得するためのホスト名が正しいことを確認します。
subschemaexception	44	サブスキーマのエラー。	サブスキーマはグローバル属性と組織属性でのみサポートされています。
serviceschemaexception	45	サービスのサービススキーマを見つけることができません。	XML ファイル内のサブスキーマが有効であることを確認します。
roletemplateexception	46	ロールテンプレートは、スキーマタイプがダイナミックの場合にのみ true となります。	XML ファイル内のロールテンプレートが有効であることを確認します。
cannotAddusersToFileredRole	47	フィルタリングされたロールにはユーザーを追加できません。	XML ファイル内のロール DN が、フィルタリングされたロールでないことを確認します。
templateDoesNotExist	48	テンプレートが存在しません。	XML ファイル内のサービステンプレートが有効であることを確認します。
cannotAdduUersToDynamicGroup	49	ダイナミックグループにはユーザーを追加できません。	XML ファイル内のグループ DN がダイナミックグループでないことを確認します。
cannotCreatePolicyUnderContainer	50	コンテナの子組織である組織にはポリシーを作成できません。	ポリシーの作成先となる組織が、コンテナの子でないことを確認します。
defaultGroupContainer NotFound	51	グループコンテナが見つかりませんでした。	親組織または親コンテナにグループコンテナを作成します。

表 A-4　amadmin エラーコード (続き)

エラーメッセージ	コード	説明 / 考えられる原因	対処方法
cannotRemoveUserFromFilteredRole	52	フィルタリングされたロールからはユーザーを削除できません。	XML ファイル内のロール DN が、フィルタリングされたロールでないことを確認します。
cannotRemoveUsersFromDynamicGroup	53	ダイナミックグループからはユーザーを削除できません。	XML ファイル内のグループ DN がダイナミックグループでないことを確認します。
subSchemStringDoesNotExist	54	サブスキーマ文字列が存在しません。	XML ファイル内にサブスキーマ文字列があることを確認します。
defaultPeopleContainerNotFound	59	ユーザーを組織またはコンテナに追加しようとしています。しかし、デフォルトのピープルコンテナが、組織およびコンテナには存在しません。	デフォルトのピープルコンテナがあることを確認します。
nodefaulturlprefix	60	defaultURLPrefix 引数の URL プレフィックスが見つかりません。	適切なデフォルトの URL プレフィックスを指定します。
nometaalias	61	-metaalias 引数の後にメタエイリアスが見つかりません。	適切なメタエイリアスを指定します。
missingEntityName	62	エンティティ名が指定されていません。	エンティティ名を指定します。
missingLibertyMetaInputFile	63	メタデータをインポートするファイル名がありません。	メタデータを含むファイルの名前を指定します。
missingLibertyMetaOutputFile	64	エクスポートされたメタデータを格納するためのファイル名がありません。	メタデータを格納するファイルの名前を指定します。
cannotObtainMetaHandler	65	メタ属性にハンドラを取得できません。指定されたユーザー名とパスワードが誤っている可能性があります。	ユーザー名とパスワードが正しいことを確認します。
missingResourceBundleName	66	ディレクトリサーバーに保存されるリソースバンドルを追加、閲覧、または削除しようとしましたが、リソースバンドル名がありません。	リソースバンドル名を指定します。

表 A-4　amadmin エラーコード (続き)

エラーメッセージ	コード	説明 / 考えられる原因	対処方法
missingResourceFileName	67	リソースバンドルをディレクトリサーバーに追加しようとしましたが、リソース文字列を含むファイルの名前がありません。	有効なファイル名を指定します。
failLoadLibertyMeta	68	ディレクトリサーバーへの Liberty メタの読み込みに失敗しました。	メタデータをチェックしてから再度読み込みます。

用語集

このマニュアルセットの中で使われる用語については、『Sun Java™ Enterprise System 用語集』を参照してください。

http://docs.sun.com/doc/819-1933?l=ja

索引

A

B

C

D

F

G

H

I

J

L

M

N

O

P

R

S

U

V

W

あ

い

え

お

か

き

く

け

こ

さ

し

す

せ

そ

た

つ

て

と

に

ね

は

ひ

ふ

へ

ほ

め

も

ゆ

り

る

れ

ろ

ろ