



Sun Java™ System

Identity Manager 7.0 の 配備に関する技術概要

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

Part No: 820-0156

Copyright © 2006 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

Sun Microsystems, Inc. は、この製品に含まれるテクノロジーに関する知的所有権を保持しています。特に限定されることなく、これらの知的所有権は <http://www.sun.com/patents> に記載されている 1 つ以上の米国特許および米国およびその他の国における 1 つ以上の追加特許または特許出願中のものが含まれている場合があります。

この製品は SUN MICROSYSTEMS, INC. の機密情報と企業秘密を含んでいます。SUN MICROSYSTEMS, INC. の書面による許諾を受けることなく、この製品を使用、開示、複製することは禁じられています。

U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

本製品には、サードパーティーが開発した技術が含まれている場合があります。

本製品の一部は Berkeley BSD システムより派生したもので、カリフォルニア大学よりライセンスを受けています。UNIX は、X/Open Company, Ltd. が独占的にライセンスしている米国ならびにほかの国における登録商標です。

Sun、Sun Microsystems、Sun ロゴ、Java、Solaris、JDK、Java Naming and Directory Interface、JavaMail、JavaHelp、J2SE、iPlanet、Duke ロゴ、Java Coffee Cup ロゴ、Solaris ロゴ、SunTone Certified ロゴおよび Sun ONE のロゴマークは、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標もしくは登録商標です。

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用されている、米国およびその他の国における同社の商標または登録商標です。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャに基づくものです。

Legato および Legato のロゴマークは Legato Systems, Inc. の商標であり、Legato NetWorker は同社の商標または登録商標です。Netscape Communications Corp のロゴマークは Netscape Communications Corporation の商標または登録商標です。

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

この製品は、米国の輸出規制に関する法規の適用および管理下であり、また、米国以外の国の輸出および輸入規制に関する法規の制限を受ける場合があります。核、ミサイル、生物化学兵器もしくは原子力船に関連した使用またはかかる使用者への提供は、直接的にも間接的にも、禁止されています。このソフトウェアを、米国の輸出禁止国へ輸出または再輸出すること、および米国輸出制限対象リスト (輸出が禁止されている個人リスト、特別に指定された国籍者リストを含む) に指定された、法人、または団体に輸出または再輸出することは一切禁止されています。

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されない、明示的であるか黙示的であるかを問わない、なんらの保証も行われぬものとします。

目次

はじめに	ix
対象読者	ix
内容の紹介	x
表記上の規則	xi
表記上の規則	xi
記号	xi
シェルプロンプト	xii
関連ドキュメントとヘルプ	xii
オンライン上の Sun リソースへのアクセス	xiii
Sun テクニカルサポートへのお問い合わせ	xiv
関連するサードパーティー Web サイト	xiv
ご意見、ご要望の送付先	xiv
 第 1 章 属性の操作	1
関連する章	1
属性とは	1
属性のタイプ	2
アイデンティティ属性の操作	11
アイデンティティ属性を使用する理由	11
アーキテクチャーの概要	12
 第 2 章 データ読み込みと同期	17
データ読み込みの種類	18
探索	18
調整	21
Active Sync	22
データ読み込みの種類の概要	23
調整を管理する	24
調整ポリシー	24
リソースのスケジューリング	30

調整設定オブジェクト	31
Active Sync を管理する	32
Active Sync 対応アダプタが機能する仕組み	32
フォームを使用する	36
第 3 章 データ読み込みのシナリオ	45
環境を評価する	45
最初のリソースを選択する	46
最初のデータ読み込みプロセスを選択する	48
ファイルから読み込み	49
リソースから読み込み	50
一括操作の作成	51
調整	52
データ読み込みの準備を行う	53
アダプタを設定する	53
アカウント ID ポリシーとパスワードポリシーを設定する	53
データ読み込み用のアカウントを作成する	55
ユーザーフォームを割り当てる	55
ほかのリソースへアカウントをリンクする	57
カスタム相関キーを定義する	58
カスタム規則を作成する	60
手動でアカウントをリンクする	61
シナリオの例	62
Active Directory、SecurID、および Solaris	63
LDAP、PeopleSoft、および Remedy	68
高速一括追加のシナリオ	73
第 4 章 ユーザーアクションを設定する	75
カスタムタスクを追加する	75
カスタムタスクへのアクセス権を設定する	75
タスクをリポジトリに追加する	78
ユーザーアクションを設定する	81
ユーザーアクションの設定	82
第 5 章 Identity Manager のプライベートトラベリング	87
プライベートトラベリングのタスク	87
アーキテクチャー機能	88
スタイルシート	88
デフォルトテキスト	89
テキスト属性	89
デフォルトスタイル設定	89
カスタマイズするファイル	89

JSP ファイル	90
WPMessages_ja.properties ファイル	90
ヘッダーのカスタマイズ	90
ヘッダー外観を変更する	90
Identity Manager ページをカスタマイズする	91
ホームページをカスタマイズする	91
Identity Manager ユーザーインタフェースのホームページに表示されるデフォルト情報を 変更する	94
ユーザーインタフェースのナビゲーションメニューの外観を変更する	95
フォント特性を変更する	96
ラベリング演習のサンプル	96
Identity Manager ロゴのカスタムロゴへの置き換え	97
マストヘッドの外観を変更する	97
ナビゲーションタブを変更する	99
タブパネルのタブを変更する	100
ソートテーブルのヘッダーを変更する	101
ユーザー / リソーステーブルのコンポーネントを変更する	101
よく使われるページでの Identity Manager の動作を変更する	103
 第 6 章 メッセージカタログをカスタマイズする	107
カスタムメッセージカタログの利点	107
Identity Manager がメッセージカタログエントリを取得する方法	107
メッセージカタログの形式	108
カスタマイズしたメッセージカタログを作成する	108
例	109
 付録 A 設定オブジェクトを編集する	111
設定オブジェクトについて	111
設定オブジェクトを表示および編集する	111
UserUIConfig オブジェクト	112
属性タイプ	113
オブジェクト属性	113
 付録 B 国際化サポートを有効にする	117
アーキテクチャの概要	117
一般的なエントリ	118
複数言語のサポートの有効化	119
手順 1: ローカライズされたファイルをダウンロードしてインストールする	119
手順 2: Waveset.Properties ファイルを編集する	121
複数言語のリンクをログインページに作成する	121

付録 C リソース上の 1 ユーザーに対して複数のアカウントを割り当てる	123
1 つのリソース上の 1 ユーザーに複数のアカウントを割り当てる理由	123
アカウントタイプを設定する	123
アカウントタイプを割り当てる	124
アカウントタイプを削除する	124
複数アカウントユーザーと調整	125
アカウントタイプとユーザー用画面	125
 付録 D Sun Communications Services 配備シナリオ	 127
機能	127
前提条件	128
シナリオをロードおよび設定する	129
手順 1: Directory Server を設定する	129
手順 2: Messaging Server を設定する	130
手順 3: シナリオをロードする	130
手順 4: Sun Communications Services リソースアダプタを設定する	131
手順 5: 電子メールテンプレートを設定する	132
サンプルディレクトリの概要	133
Identity Manager オブジェクトを作成および設定する	134
Sun JSCS リソース属性	134
管理者ロール	136
ロール	136
オブジェクトグループ	137
ログインアプリケーションとログインモジュールグループ	137
一般的な設定	137
ユーザーアカウントを作成する	138
調整を使用してユーザーを作成する	138
ActiveSync を使用してユーザーを作成する	140
Identity Manager を使用してユーザーを作成する	143
ユーザーアカウントの更新	147
調整によって検出される更新	147
ActiveSync によって検出される更新	148
Identity Manager で実行された更新	148
変更をユーザーインターフェースに実装する	149
秘密の質問に対する回答を設定する	149
アカウント属性を変更する	151
マネージャーが部下のデータを編集する	152
リソースへのアクセスをリクエストする	153
変更を管理者インターフェースに実装する	155
承認ページを修正する	155
「Find User Form」を修正する	155

付録 E	HR データベース /Active Directory 配備シナリオ	157
機能		158
内容のサンプル		159
前提条件		160
シナリオをロードおよび設定する		160
Active Directory の設定		160
PeopleSoft の設定		161
Identity Manager の設定		161
配備の確認		163
ユーザーの作成および反映		164
ユーザーの修正		164
監査の設定		165
リンクの確認		165
索引		167

はじめに

本書『Sun Java™ System Identity Manager の配備に関する技術概要』では、Sun Java™ System Identity Manager を環境に合わせてカスタマイズするときに使用する参照情報と手順の概要について説明します。

対象読者

『Sun Java™ System Identity Manager の配備に関する技術概要』は、製品配備のさまざまな段階で Identity Manager を顧客インストール用にカスタマイズするのに必要なワークフロー、画面、規則、システム設定、およびその他の設定ファイルを作成および更新するデブロイヤーおよび管理者に向けて作成されました。

デブロイヤーは、プログラミングに関する予備知識があり、XML、Java、Emacs や IDE (Eclipse または NetBeans など) に精通していることが望まれます。

内容の紹介

『Identity Manager の配備に関する技術概要』は、次の章で構成されています。

- 第 1 章「[属性の操作](#)」 - アイデンティティ属性と、この機能を使って Identity Manager 配備中のデータフローを整理する方法について説明します。
- 第 2 章「[データ読み込みと同期](#)」 - アカウント情報を Identity Manager に読み込むための調整とその他の機構の概要を説明します。調整では、Identity Manager で定義されたユーザーのセットと Identity Manager リソースで定義されたアカウントのセットを比較します。
- 第 3 章「[データ読み込みのシナリオ](#)」 - アカウント情報を Identity Manager に読み込むときに考慮すべきヒントについて説明します。これには、発生する可能性がある問題を示すサンプルシナリオが含まれます。
- 第 4 章「[ユーザーアクションを設定する](#)」 - Identity Manager の管理者インタフェースにカスタムタスクを追加する方法と、インタフェースの 2 つのエリアから実行できるユーザーアクションを設定する方法について説明します。
- 第 5 章「[Identity Manager のプライベートトラベリング](#)」 - 組織のスタイル標準に合わせて IDM の色、ロゴ、およびヘッダーとフッターの内容をカスタマイズする方法について説明します。
- 付録 A「[設定オブジェクトを編集する](#)」 - 設定オブジェクトの概要と UserUIConfig オブジェクトについて説明します。
- 付録 B「[国際化サポートを有効にする](#)」 - Identity Manager で複数の言語を使用する場合や英語以外の言語を表示する場合の設定に関する情報を提供します。
- 付録 C「[リソース上の 1 ユーザーに対して複数のアカウントを割り当てる](#)」 - 1 つのリソースに対してユーザーごとに複数のアカウントを割り当てる場合の基本概念について説明します。
- 付録 D「[Sun Communications Services 配備シナリオ](#)」 - LDAP 配備シナリオについて説明します。このシナリオは、サンプルディレクトリ内のスクリプトから Identity Manager に読み込むことができます。
- 付録 E「[HR データベース /Active Directory 配備シナリオ](#)」 - ユーザーアカウントが人事データベース (たとえば、PeopleSoft) で定義され、IT インフラストラクチャーへのアクセスが Active Directory によって制御されている環境に Identity Manager を配備する方法について説明します。

表記上の規則

この項の表は、本書で使用する次の表記規則について説明しています。

- [表記上の規則](#)
- [記号](#)
- [シェルプロンプト](#)

表記上の規則

次の表は、本書で使用する表記上の規則について説明しています。

表 1 表記上の規則		
字体または記号	意味	例
AaBbCc123 (モノスペース)	API および言語の要素、HTML タグ、Web サイトの URL、コマンド名、ファイル名、ディレクトリパス名、画面出力の表示、サンプルコードを示します。	.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表示します。 % You have mail.
AaBbCc123 (太字のモノスペース)	ユーザーが入力する文字を、画面上のコンピュータ出力とは区別して示します。	% su Password:
AaBbCc123 (イタリック)	実際の名前または値によって置き換えられるコマンドまたはパス名の可変部分。	これらを、 <i>class</i> オプションと呼びます。 このファイルは、 <i>install-dir/bin</i> ディレクトリにあります。

記号

次の表は、本書で使用する記号の表記規則を示しています。

表 2 記号の表記規則			
記号	説明	例	意味
[]	省略可能なコマンドオプションが入ります。	ls [-l]	-l オプションは省略可能です。
-	同時に押すキーを連結します。	Control-A	Ctrl キーと A キーを同時に押します。

表 2 記号の表記規則 (続き)

記号	説明	例	意味
+	連続して押すキーを連結します。	Ctrl+A+N	Ctrl キーを押し、離してから、以後のキーを続けて押します。
>	グラフィカルユーザーインタフェースで選択するメニュー項目を示します。	「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。

シェルプロンプト

次の表は、本書で使用するシェルプロンプトを示しています。

表 3 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	<i>machine-name%</i>
UNIX または Linux の C シェルのスーパーユーザー	<i>machine-name#</i>
UNIX または Linux の Bourne シェルおよび Korn シェル	\$
UNIX または Linux の Bourne シェルおよび Korn シェルのスーパーユーザー	#
Windows のコマンド行	C:¥

関連ドキュメントとヘルプ

Sun Microsystems は、Identity Manager をインストール、使用、および設定する際に役立つ次のような追加のドキュメントと情報を提供しています。

- 『Identity Manager Installation』: Identity Manager と関連ソフトウェアをインストールおよび設定する手順と参照情報が記載されています。
- 『Identity Manager Upgrade』: Identity Manager と関連ソフトウェアをアップグレードおよび設定する手順と参照情報が記載されています。
- 『Identity Manager 管理ガイド』: Identity Manager を使用して企業情報システムへのセキュリティ保護されたユーザーアクセスを実現するために、手順、チュートリアル、実例を説明します。

- 『Identity Manager Deployment Tools』 : Identity Manager のさまざまな配備ツールの使用方法を示す参照情報と手順が記載されています。これらの情報は、Identity Manager サーバーによって提供される規則と規則ライブラリ、共通のタスクとプロセス、辞書サポート、および SOAP ベースの Web サービスインタフェースを対象としています。
- 『Identity Manager Workflows, Forms, and Views』 : Identity Manager のワークフロー、フォーム、および画面の使用方法を示す参照情報と手順が記載されています。この中には、これらのオブジェクトをカスタマイズするのに必要なツールに関する情報が含まれます。
- 『Identity Manager Resources Reference』 : アカウント情報をリソースから Sun Java™ System Identity Manager に読み込んで同期する方法を示す参照情報と手順が記載されています。
- 『Identity Manager Tuning, Troubleshooting, and Error Messages』 : Sun Java™ System Identity Manager のチューニングに関するガイダンス、問題の追跡とトラブルシューティングの手順、およびこの製品を操作したときに発生する可能性があるエラーメッセージと例外についての説明を提供する参照情報と手順が記載されています。
- 『Identity Manager Service Provider Edition Deployment』 : Sun Java™ System Identity Manager Service Provider Edition の計画と実装の方法を示す参照情報と手順が記載されています。
- Identity Manager ヘルプ : Identity Manager の完全な手順、参照情報、用語の説明を記載したオンラインガイダンス、オンライン情報です。ヘルプにアクセスするには、メニューバーの「ヘルプ」リンクをクリックします。主要なフィールドには、ガイダンス (フィールド固有の情報) があります。

オンライン上の Sun リソースへのアクセス

製品のダウンロード、プロフェショナルサービス、パッチとサポート、および開発者向け追加情報については、次の Web サイトにアクセスしてください。

- ダウンロードセンター
<http://www.sun.com/software/download/>
- プロフェショナルサービス
<http://www.sun.com/service/sunps/sunone/index.html>
- Sun Enterprise サービス、Solaris パッチ、およびサポート
<http://sunsolve.sun.com/>
- 開発者向け情報
<http://developers.sun.com/prodtech/index.html>

Sun テクニカルサポートへのお問い合わせ

製品のドキュメントで解決できない、本製品に関する技術的な質問については、次のいずれかの方法でカスタマサポートにお問い合わせください。

- オンラインサポート Web サイト <http://www.sun.com/service/online/us>
- 保守契約に基づいて提供されるサポート電話番号

関連するサードパーティー Web サイト

Sun は、本書に記載されているサードパーティー Web サイトの利用について責任を負いません。Sun は、このようなサイトまたはリソースで得られるあらゆる内容、広告、製品、およびその他素材を保証するものではなく、責任または義務を負いません。Sun は、このようなサイトまたはリソースで得られるあらゆるコンテンツ、製品、またはサービスによって生じる、または生じたと主張される、または使用に関連して生じる、または信頼することによって生じる、いかなる損害または損失についても責任または義務を負いません。

ご意見、ご要望の送付先

Sun ではマニュアルの品質向上のため、お客様のご意見、ご要望をお受けしております。

コメントをお送りになる場合は、<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で、ドキュメントのタイトルと Part No. を入力します。Part No. は、マニュアルのタイトルページまたは最上部に記載されている 7 桁または 9 桁の番号です。

たとえば、本書のタイトルは『Sun Java™ System Identity Manager の配備に関する技術概要』であり、Part No. は 820-0156 です。

属性の操作

この章では、Identity Manager 配備で使用する属性の概念的な概要について説明します。

関連する章

属性は、Identity Manager のさまざまな処理の中で操作され、このマニュアルセット全体で説明されています。各種の属性に関する実質的な説明は、次の章にあります。

- ビュー属性、属性の登録、および deferred 属性に関する説明の詳細は、『Identity Manager Workflows, Forms, and Views』の「Views」の章にあります。
- リソース属性に関する説明は、『Identity Manager Resources Reference』にあります。

属性とは

属性は、Identity Manager オブジェクトや外部リソースの特性を定義および操作するために使われる名前と値のペアです。フォーム、ワークフロー、規則などの Identity Manager コンポーネントは、通常の操作でデータアクセスやデータ変換の主要部分として属性を呼び出します。

属性のタイプ

Identity Manager 配備のオブジェクトにはさまざまな属性を入れることができますが、次の表で紹介するタイプを使用する可能性がもっとも多くなります。個々の属性タイプについては、あとの項で詳しく説明します。

表 1-1 Identity Manager の属性タイプ

属性タイプ	説明
概要の属性	オブジェクトを書き込むときに使用できる属性を定義します。
クエリー可能な属性	検索時にアクセスできる属性を定義します。
インライン属性	データベースを介したクエリーを最適化します。
ユーザー拡張属性	デフォルトのオブジェクト定義に含まれていない、ユーザーによって追加される省略可能な名前と値のペアを定義します。
ビュー属性	画面と呼ばれる動的なデータモデルに組み合わせて構成されるユーザーアカウント情報の名前と値のペアを定義します。
オペレーショナル属性	リポジトリが正常に機能するために必要です。
リソースユーザー属性	リソース上のユーザーアカウントの特性を定義します。
アイデンティティーシステムユーザー属性	リソースユーザー属性に対応する Identity Manager の値を定義します。Identity Manager では、スキーママップの左側にアイデンティティーシステムユーザー属性が表示されます。この用語は、アカウント属性と同じ意味で使われます。
アイデンティティー属性	Identity Manager 配備の内部の属性フローを設定および表示する中心点になります。

概要の属性

すべての持続オブジェクトは、概要の属性のセットを公開します。概要の属性には、リスト操作の結果として各項目に返される値が含まれます。各 PersistentObject サブクラスは、getSummaryAttributes メソッドを上書きすることにより、デフォルトの属性セットを拡張できます。

概要の属性は、直列化されて文字列になったときに概要の属性全体の長さが制限されるため、通常は単一の値です。

ユーザーオブジェクト用のこれらの属性は、`UserUIConfig` オブジェクトを使ってすぐに設定できます。詳細については、「`UserUIConfig` オブジェクト」を参照してください。

クエリー可能な属性

すべての持続オブジェクトは、クエリー可能な属性のセットを公開します。各サブクラスは、`getQueryableAttributes` メソッドを上書きすることにより、デフォルトのセットを拡張できます。

クエリー可能な属性は複数值であり、フィルタや照合に使用する値のセットを格納できます。

ユーザーオブジェクト用のこれらの属性は、`UserUIConfig` オブジェクトを使ってすぐに設定できます。詳細については、「`UserUIConfig` オブジェクト」を参照してください。

インライン属性

オブジェクトの型ごとに、クエリー可能な属性を最大 5 個までインライン属性として指定できます。属性をインラインとして指定すると、その属性に対するクエリーのパフォーマンスを最適化しようデータストアに要求します。

`Identity Manager` は、通常、メインオブジェクトテーブルとは別個の属性テーブル内の行としてクエリー可能な属性の各値を格納します。属性テーブルをオブジェクトテーブルに結合することにより、`AttributeCondition` と一致するオブジェクトを選択できます。

ただし、`Identity Manager` はインライン属性の値をその型のオブジェクトテーブルに直接格納します。属性をインラインとして指定すると、`Identity Manager` はより効率的な SQL を生成できます。メインオブジェクトテーブル上の列の式は、対応する属性テーブルへの JOIN (または対応する属性テーブルに対する EXISTS 述語) よりも高速になります。これにより、属性に対するすべてのクエリーのパフォーマンスが向上します。

インライン属性の特徴は次のとおりです。

- **インライン属性は、単一値である必要があります。**これは、インライン属性の値がオブジェクトテーブルの親行の 1 つの列に保存されるためです。
- **インラインにできるクエリー可能な属性は、1 つの型につき最大 5 個です。**これは、任意の属性値の格納に使用できる列がオブジェクトテーブルに 5 つしかないためです。

- 同じクエリー可能な属性のセットが、1つの型の全インスタンスでインラインとして指定されます。これは、列の値と属性の名前の対応がこの設定のみで指定されるためです。つまり、ある型に対するインライン属性の設定は、どの属性がどの列に格納されているかをリポジトリが識別するための唯一の方法です。

ユーザー拡張属性

ユーザーオブジェクトの拡張属性のデフォルトリストに、属性を追加できます。これらの属性は、ユーザー拡張属性と呼ばれ、リポジトリの `User Extended Attributes` 設定オブジェクトに格納されます。

この設定オブジェクトは、リポジトリのユーザーオブジェクトに格納された追加属性を定義します。システムは、ユーザー画面の `accounts[lighthouse]` 名前空間にある、設定オブジェクトに登録されていない属性値をすべて無視します。

ユーザー拡張属性の詳細については、『*Identity Manager Workflows, Forms, and Views*』の「*Views*」の章にある、ユーザー画面の `accounts[lighthouse]` 属性についての説明を参照してください。

オペレーショナル属性

Identity Manager では、リポジトリが正常に機能するために必要ないくつかの属性が事前に定義されています。ID、type、および name は特に重要です。

ID、type、および name

リポジトリに格納されているすべての *PersistentObject* には、グローバルに一意的な内部識別子 (ID) があります。ID 値は時間と空間を越えて一意であり、生成された ID 値は再使用されません。事前に定義された *Identity Manager* オブジェクトの中には、プログラム定数として定義された既知の識別子を持つものがあります。これらは、擬似 ID と呼ばれます。リポジトリでは、オブジェクトの ID が変更されないことが保証されます。

同じ型のオブジェクトは、通常、同じ *Java* クラスにマップされます。つまり、直列化復元されるときに同じ *Java* クラスのインスタンスとして構築されます。型と *Java* クラスが 1 対 1 で対応しない場合は、少なくとも同じ型のすべてのオブジェクトが同じ機構を使って、対応する *Java* クラスを検索します。たとえば、一部の型のオブジェクトが完全修飾クラス名を含む `class` 属性を公開するなどです。

オブジェクトの名前は、型の中で一意である必要があります。つまり、ある型の 1 つのオブジェクトは特定の名前しか持つことができません。ただし、型の異なる別のオブジェクトが同じ名前を持つことは可能です。これにより、型ごとに従属する名前空間が効果的に定義されます。オブジェクトの名前は変更できますが、オブジェクトの ID は変更できません。

modified

Java の Long 型の *modified* には、オブジェクトが Identity Manager を介して変更されるごとに増分される値が含まれています。

更新の直後から同じ型の次のオブジェクトが変更されるまでの間、この *modified* の値はその型の特別な LASTMODIFIED 項目の counter 属性の値と一致します。

lockinfo

Java の String 型の *lockinfo* には、論理ロックを表す値が含まれています。

lockinfo 文字列には、ロッカーの名前と論理ロックの期限が切れる時間が含まれています。

counter

Java の Long 型の *counter* には、オブジェクトに対する論理ロックを必要とせず非常に効率的に増分または設定できる任意の値が含まれています。

その他の標準的な属性

MemberObjectGroups

すべての持続オブジェクトは、少なくとも 1 つのオブジェクトグループに属しています。このような複数値属性の各値は、ObjectGroup オブジェクトの ID です。

ObjectGroup は、Identity Manager の管理者インタフェースとユーザーインタフェースでは組織として公開されます。ObjectGroup メンバーシップは、セッションレベルでのアクセス権限 (つまり、管理者およびユーザーのリポジトリオブジェクトへのアクセス) を制御しますが、リポジトリ自体はオブジェクトグループのメンバーシップを無視します。

creator、*createDate*、*lastModifier*、および *lastModDate*

これらの値は、各オブジェクトの履歴情報を記録します。これらの属性はリポジトリが管理しますが、リポジトリが使用することはありません。

PropertyList

すべての持続オブジェクトは、任意のプロパティのリストを格納できます。この機能は、一般にあまり使用されません。

subType

すべての持続オブジェクトは *subType* 属性を持つことができます。たとえば、Identity Manager は Attribute.SUBTYPE を使って利用可能な関連規則と確認規則の個々のリストを選択します。

authType

`authType` 属性を用いると、組織 (オブジェクトグループ) を管理しないユーザーに、限られた範囲内での、限定された権限を付与することができます。これらの主体は、**Identity Manager** の標準の権限付与スキームにおいては、これ以外の方法では権限を得ることができません。

ビュー属性

画面とは、リポジトリに格納された 1 つ以上のオブジェクトを組み合わせて構成した名前と値のペア、またはリソースから読み取った名前と値のペアの集合です。ビュー属性では、文字列などの個々の値、リストなどの集合、または別のオブジェクトの参照を値として使用できます。

Identity Manager の管理者インタフェースまたはユーザーインタフェースからユーザーアカウントの作成または変更を実行する場合は、ユーザー画面を間接的に処理します。ワークフロープロセスも、ユーザー画面と連携しています。リクエストをワークフロープロセスに渡すと、属性が画面としてプロセスに送信されます。ワークフロープロセスの中で手動プロセスをリクエストすると、ユーザー画面の属性を表示して詳細に変更できます。

画面の操作については、『**Identity Manager Workflows, Forms, and Views**』の「**Views**」の章で詳細に説明しています。

リソースユーザー属性

リソースユーザー属性は、**Identity Manager** のアカウント属性をスキーママップ (右側) のリソースアカウント属性にマップします。属性のリストはリソースごとに異なります。使用していない属性は、スキーママップページから削除できます。ただし、属性の追加には、アダプタのコードを編集する必要がある場合があります。

リソースユーザー属性は、アダプタがリソースと通信しているときにだけ使用されます。

リソースユーザー属性の操作については、『**Identity Manager Resources Reference**』で詳細に説明しています。

アイデンティティシステムユーザー属性

アイデンティティシステムユーザー属性は、リソースユーザー属性に対応する **Identity Manager** 内部の値を定義します。アイデンティティシステムユーザー属性は、規則、フォーム、およびその他の **Identity Manager** 固有の機能で使用できます。**Identity Manager** では、スキーママップの左側にこれらの属性が表示されます。

アイデンティティシステムユーザー属性の操作については、『**Identity Manager Resources Reference**』で詳細に説明しています。

属性を表示させるかどうかを制御する (シークレット属性)

Identity Manager では、編集フォームにアスタリスク付きで表示されるように属性を設定した場合でも、結果ページの属性の値は平文形式で表示されます。属性値がキャッシュに表示されないようにする場合は、属性をシークレットとして登録できます。シークレット属性は、ブラウザのキャッシュに値が平文形式で表示されない属性ですが、その点を除けばほかの属性と同じように Identity Manager で処理されます。

Identity Manager は「シークレット」属性を次のように処理します。Identity Manager は、結果テーブルをレンダリングするときに、シークレットとして登録されている属性があるかどうかを確認し、シークレット属性の値をアスタリスクのみで表示します。

社会保障番号は、シークレット属性として登録することが望ましい典型的な属性です。

シークレット属性を登録する

シークレット属性を登録するには、次のようにしてシークレット属性を System Configuration オブジェクトに追加します。

```
<Attribute name='secretAttributes'>
  <List>
    <String>email</String>
    <String>myAttribute</String>
  </List>
</Attribute>
```

属性条件

属性条件は、属性の値を判定する式です。属性条件は、特定の条件に一致するオブジェクトのサブセットを選択するためによく使われます。

個々の属性条件は、1つの条件を表し、次の要素で構成されます。

- (クエリー可能な属性の) 属性名
- 演算子 (実行するチェックや比較の種類)
- オペランド (指定された値のセット)

属性条件の演算子

AttributeCondition には、次のような演算子が定義されています。

表 1-2 属性条件の演算子

演算子	説明
EQ, EQUALS	指定された属性に関して、(大文字小文字は無視して)字句的にオペランドと等しい値がオブジェクトに少なくとも1つあります。
NE, NOT_EQUALS	指定された属性に関して、(大文字小文字は無視して)字句的にオペランドと等しい値がオブジェクトにありません。
GT, GREATER_THAN	指定された属性に関して、(大文字小文字は無視して)字句的にオペランドより大きい値がオブジェクトに少なくとも1つあります。
GE	指定された属性に関して、(大文字小文字は無視して)字句的にオペランドと等しいか、それより大きい値がオブジェクトに少なくとも1つあります。
LE	指定された属性に関して、(大文字小文字は無視して)字句的にオペランドと等しいか、それより小さい値がオブジェクトに少なくとも1つあります。
LT, LESS_THAN	指定された属性に関して、(大文字小文字は無視して)字句的にオペランドより小さい値がオブジェクトに少なくとも1つあります。
STARTS_WITH	指定された属性に関して、(大文字小文字は無視して)オペランドの部分文字列の先頭と一致する値がオブジェクトに少なくとも1つあります。
ENDS_WITH	指定された属性に関して、(大文字小文字は無視して)オペランドの部分文字列の末尾と一致する値がオブジェクトに少なくとも1つあります。
CONTAINS	指定された属性に関して、(大文字小文字は無視して)オペランドの部分文字列と一致する値がオブジェクトに少なくとも1つあります。
IS_PRESENT	指定された属性の値がオブジェクトに少なくとも1つあります。この演算子はオペランドを取りません。
NOT_PRESENT	指定された属性の値がオブジェクトにありません。この演算子はオペランドを取りません。

表 1-2 属性条件の演算子 (続き)

演算子	説明
IN、IS_ONE_OF	指定された属性に関して、(大文字小文字は無視して) 字句的に (リスト) オペランド内のいずれかの値と等しい値がオブジェクトに少なくとも 1 つあります。
注	RelationalDataStore は、各属性条件を演算の WHERE 句の一部となる適切な述語に変換することにより、評価を最適化します。しかしながら、複数値の属性の処理に特別なロジックは必要ありません。 RelationalDataStore は、これを処理する適切な SQL DML を自動的に生成します。

属性条件は、属性の値ごとに適用されます。特に演算子 **NE** は、指定された属性について、指定されたオペランドと等しい値がオブジェクトにない場合かつその場合にかぎり **true** になります。演算子 **EQ** は、指定された属性の、指定されたオペランドと一致する値がオブジェクトに少なくとも 1 つある場合に **true** になります。

暗黙的な AND 結合

属性条件のセットは、暗黙的に **AND** 結合されます。つまり、属性条件のセットは、セットに含まれるすべての属性条件が **true** と評価された場合かつその場合にかぎり **true** と評価されます。逆に、セット内のいずれかの属性条件が **false** と評価されると、ただちに属性条件のセットが **false** と評価されます。

Identity Manager の属性条件には、一般的に使用される演算子が公開されています。通常は、**IDM** の属性条件を使って一連の選択条件を表現できます。表現できない条件もごく一部ありますが、多くの場合、クエリー可能な属性を追加するか、クエリー可能な属性の表現を変更することによって、そのような条件にも適切に対応できます。

組織にユーザーメンバー規則を設定する

特定の組織に含まれるユーザーのセットは、次の方法で特定できます。

- **(Identity Manager から見て) 外部のリソースアカウント属性**。この場合、複数の **Identity Manager** ユーザーが異なるリソース上で同じ **acctid** を持っている可能性があるため、一致する **Identity Manager** ユーザーを見つけるためには、リソースアカウント ID とリソース名の両方 (たとえば、**acctid:resname**) が必要です。
- **Identity Manager のユーザーアカウント属性** (たとえば、名前、場所、マネージャー)。

効果的に「OR 結合」の結果が得られるように、複数の属性条件を使用しないでください。代わりに、次のようにオペランドのリストとともに「is one of」演算子を使用します。

```
<list>
  <new class='com.waveset.object.AttributeCondition'>
    <s>firstname</s>
    <s>is one of</s>
    <list>
      <s>Nicola</s>
      <s>Paolo</s>
    </list>
  </new>
</list>
```

シナリオの例

問題

指定された管理者ロールを持つユーザーを除くすべてのユーザーを含める規則が必要である。

解決方法

属性条件は暗黙的に AND 結合されるため、次の 2 つの属性条件を使用できます。

- 少なくとも 1 つの管理者ロールを持つユーザーを選択する (つまり、管理者ではないユーザーを除外する) 条件。この条件に一致するユーザーは、adminRoles 属性の値を少なくとも 1 つ持つことになります。

```
<AttributeCondition>
  <s>adminRoles</s>
  <s>exists</s>
</AttributeCondition>
```

- 特定の管理者ロール (の任意のセット) を持つユーザーを除外する条件。次に示す条件の例では、adminRoles 属性の値が ar1 か ar2 であるユーザーが除外されます。

```

<AttributeCondition>
  <s>adminRoles</s>
  <s>is not</s>
  <list>
    <s>ar1</s>
    <s>ar2</s>
  </list>
</AttributeCondition>

```

この2つの条件を組み合わせることにより、指定されたリストに含まれない管理者ロールを持ったユーザーを返す規則を作成できます。

アイデンティティ属性の操作

アイデンティティ属性は、情報の発生元やリソースの数 (SPML、Active Sync リソース、ユーザーインタフェースなど) に関係なく、Identity Manager 配備の内部の属性フローを設定および表示するための中心点になります。

Identity Manager では、配備環境のどの部分にアイデンティティ属性を組み込むかを選択できます。アイデンティティ属性を配備全体で使用することも、「アプリケーション」(たとえば、Identity Manager 管理者インタフェース、Identity Manager ユーザーインタフェース、Active Sync の処理中など) だけにその実装を限定することもできます。

アイデンティティ属性は、Identity Manager 管理者インタフェースの「設定」>「アイデンティティ属性」ページで定義します。属性の作成、削除、編集の手順については、これらのページとともに表示されるオンラインヘルプを参照してください。

特別な配備のニーズに対応するため、ワークフローやフォームをカスタマイズする代替りの手段としてアイデンティティ属性を使用できます。たとえば、lastname に firstname を追加することで fullname という名前のアイデンティティ属性を作成できます。これにより、この値を取得するためのワークフローを作成する必要がなくなります。

アイデンティティ属性を使用する理由

アイデンティティ属性には、配備に関して次の利点があります。

- 配備内で一貫した方法でデータマッピング規則を適用できます。

- フォームのデータ変換機能をデータ表示機能から分離できます。以前に **Identity Manager** 配備で操作したことがあれば、**Identity Manager** のフォームがデータ変換とデータ表示という 2 つの目的を果たすことに気づくはずですが。アイデンティティ属性を使用すると、フォームベースのデータ変換の量が減少します。
- **Identity Manager** 配備全体のデータの流れを一元的に表示できます。

アーキテクチャーの概要

アイデンティティ属性の実装には、次の **Identity Manager** コンポーネントを必要とします。より重要なコンポーネントについては、あとで説明します。

表 1-3 アイデンティティ属性コンポーネントの概要

コンポーネント	説明
属性のソース	属性のソースは、リソース、規則、変換、ほかのアイデンティティ属性、または定数です。
属性のターゲット	属性のターゲットは、リソース、拡張属性、または ID ユーザーのオペレーショナル属性です。
メタビュー	アイデンティティ属性の作成と操作をサポートするオブジェクトです。このビューの名前空間はユーザー画面内に存在します。このオブジェクトには、ほかのオブジェクトと同様、 Identity Manager IDE を使ってアクセスできます。
ユーザーメタビューフォーム	アイデンティティ属性が設定または変更されたあと、 Identity Manager が自動的に生成するフォーム。
UserUIConfig オブジェクト	Identity Manager の概要の属性、クエリー可能な属性、および拡張ユーザー属性に関する情報を格納するオブジェクト。たとえば、新たに作成されたアイデンティティ属性をリポジトリに格納した場合、 Identity Manager はその属性をリポジトリ内の拡張ユーザー属性設定オブジェクトに追加します。

属性のソースとターゲット

各アイデンティティ属性は、1 つ以上のソースからの入力を使用できます。入力には、次のものがあります。

- リソース属性。たとえば、ほかのリソース上の `firstname` を設定するために、アイデンティティ情報の源泉として信頼性の高い **Active Sync** リソースを割り当てることができます。

- 変換。規則または変換を使って、ほかのアイデンティティ属性 (またはユーザービュー属性) から属性を取得できます。メタビュー変換は、変更の発生元に関係なく、システム全体に一律に適用されます。Identity Manager は、属性が Active Sync イベント、調整プロセス、ユーザー入力情報のいずれからユーザーインタフェース内のフォームに取得されたかに関係なく、これらの変換を適用します。
- 定数値。アイデンティティ属性に定数値を割り当てることにより、利用できない属性やそれ以外の方法で値を取得できない属性 (たとえば、password) のデフォルト値を設定できます。

アイデンティティ属性のターゲットは、リソース属性、拡張属性、またはオペレーショナル属性です。オペレーショナル属性は、ID ユーザーに対する、そのユーザーの動作を制御する属性 (たとえば、規則や割り当てられたリソース) です。出力がリソースである場合、Identity Manager は属性が変更された時点でその属性をリソースに書き込みます。

属性をターゲットリソースにプッシュするときの条件を割り当てることもできます。これらの条件は、true または false の値を返す規則であり、イベントの種類 (たとえば、作成や更新) によって限定することもできます。

メタビュー情報の部分読み込み

Identity Manager が「調整」、「リソースから読み込み」、または「ファイルから読み込み」用のユーザー画面に値を設定すると、Identity Manager ユーザーに割り当てられたリソースからリソースアカウントデータのみを読み込むように指定できます。これらのリソースは、読み込み中のリソースをソースとするターゲットでもあるため、変更される可能性があります。Identity Manager ユーザーの、アカウントに影響を与えないオペレーショナル属性 (たとえば、アカウント ID や組織) は、firstname や lastname などのユーザー拡張属性と同様、常に画面内にあります。

メタビュー

メタビューは、アイデンティティ属性の作成と操作をサポートします。配備に含まれるすべてのリソースとその関係を表す統合画面を提供します。

アイデンティティ属性を定義するたびに、Identity Manager はユーザー画面の metaView 名前空間内に属性を作成します。ユーザー画面には次に示すように、その属性の値が含まれます。

表 1-4 メタビュー属性と関連するユーザービュー属性

メタビュー属性	ユーザービュー属性
firstname	metaView.firstname
lastname	metaView.lastname

表 1-4 メタビュー属性と関連するユーザービュー属性

メタビュー属性	ユーザービュー属性
waveset.roles	metaView.waveset.roles

ユーザーメタビューフォーム

アイデンティティ属性を設定または変更すると、Identity Manager はユーザーメタビューフォームを自動的に生成します。このフォームは、有効なアプリケーションがアイデンティティ属性に最初にアクセスしたときに作成されます。たとえば、有効なアプリケーションとして Identity Manager の管理者インタフェースを指定すると、Identity Manager はユーザーが「ユーザーの編集」ページまたは「ユーザーの作成」ページに最初にアクセスしたときにこのフォームを作成します。

注 このフォームを変更しないでください。

アイデンティティ属性変換の適用方法

アイデンティティ属性変換は、拡張、取得、およびデフォルトの式を使ってユーザーメタビューフォームを標準の画面サイクルプロセスに取り込むことにより、ユーザー画面に適用されます。このサイクルは、ユーザー画面の作成 (新しいユーザーの場合)、チェックアウト (既存のユーザーの場合)、更新、またはチェックインが行われたときに発生します。ユーザーメタビューフォームは、アプリケーションの標準のユーザーフォーム (たとえば、Tabbed User Form) の直前に処理されます。このため、標準のユーザーフォームはアイデンティティ属性によって設定された値を上書きし、metaView 名前空間の値にアクセスすることができます。

Active Sync と SPML では、複数のフォームを操作するため、やや異なる画面処理サイクルが使用されます。したがって、入力フォームではアイデンティティ属性から設定された値を上書きできません。

サンプルシナリオ

次の表は、アイデンティティ属性とリソースの関係の例を示しています。この表で、firstname アイデンティティ属性は PeopleSoft リソースによって設定されています。Identity Manager は、firstname 属性と lastname 属性を、fullname アイデンティティ属性と user ID アイデンティティ属性を生成する規則のソースとして使用します。

表 1-5 サンプルシナリオ:アイデンティティ属性

アイデンティティ属性	属性のソース	属性のターゲット
firstname	PeopleSoft	- Active Directory - AIX
lastname	PeopleSoft	- Active Directory - AIX
fullname	- firstname (規則により生成: first space last) - lastname (規則により生成: first initial last)	Active Directory
user ID	- firstname - lastname	- Active Directory - AIX
description	Active Directory	Identity Manager ユーザー (ローカルに格納)

属性値の格納

アイデンティティ属性の値は、ローカルに格納するか、実行時に動的に取得できます。Identity Manager は、ローカルに格納された属性をユーザー拡張属性として設定します。拡張ユーザー属性リストに属性を追加すると、Identity Manager は、その属性をユーザーオブジェクトとともにリポジトリに格納して管理します。

属性値をこのように格納することで、次の 2 つの利点が得られます。

- メタビュー変換の処理速度が上がる
- Identity Manager が複数リソースにわたって検索を実行できる

通常、Identity Manager 配備ではユーザー画面を読み込むとすぐに属性が取得されます。

規則を使用してアイデンティティ属性を生成する

規則を使ってアイデンティティ属性の値を生成できます。規則をこのように使用する場合は、次の点に留意してください。

- 通常、規則は MetaView 名前空間を含むユーザー画面全体にアクセスできます。Active Sync リソースからイベントが発生した場合は、activeSync 名前空間にもアクセスできます。

- メタビュー変換に使用する規則では、メタビュー属性のみを操作の対象にしてください。ユーザービュー属性は変更しないでください。
- 規則は、アイデンティティ属性のソースとしてのみ使用してください。

データ読み込みと同期

この章では、アカウント情報をリソースから Identity Manager に読み込んで同期するために使用できる方法の概要を説明します。

アイデンティティシステム ユーザーとリソースアカウントの違いを明確に区別することが重要です。次の定義は、このトピックを理解するのに役立ちます。

- **ユーザー** : Identity Manager が管理する仮想アイデンティティです。Identity Manager ユーザーは複数のアカウントを参照する場合があります。
- **アカウント** : リソース (正確には、Identity Manager 内でリソースオブジェクトとして表現される外部システムまたはアプリケーション) で管理される実際のアイデンティティです。たとえば、UNIX システムの `/etc/passwd` にあるエントリ、Windows システムの SAM データベースにあるエントリ、RACF のあらゆる表現アカウントにある `UserProfile` などです。
- **管理者** : Identity Manager システムの設定および保守の責任を負っている人です。

Identity Manager は、既知のリソースアカウントとユーザーの情報をアカウントインデックスに格納します。アカウントインデックスの各エントリには、最小限でもアカウント ID と Identity Manager リソース ID が含まれています。エントリには、ネイティブの GUID やアカウントのステータス (有効または無効) などの追加情報が含まれる場合もあります。エントリには、アカウントの所有者として Identity Manager ユーザーの ID も記録でき、所有者候補のリストも記録できます。

データ読み込みの種類

データ読み込みは、アカウント情報をリソースから Identity Manager にインポートし、それらのアカウントを Identity Manager ユーザーに割り当てるプロセスです。

Identity Manager は、リソースからアカウントデータを読み込む次の機能をサポートします。

- 「探索」：リソースアカウントを最初に Identity Manager に読み込む基本機能を提供します。
- 「調整」：リソースアカウント情報を定期的に Identity Manager に読み込み、設定されたポリシーに従って各アカウントに適切な処置を行います。
- 「Active Sync」：アイデンティティ情報の源泉として信頼性の高い外部リソース（アプリケーションやデータベースなど）に格納されている情報を Identity Manager のユーザーデータと同期できるようにします。Active Sync 対応アダプタは、アイデンティティ情報の源泉として信頼性の高いリソースに対する変更を「待機」（ポーリング）します。

これらの概念について、それぞれ詳しく説明します。データ読み込みの種類を比較した表を「データ読み込みの種類の概要」に示します。

探索

探索プロセスは、リソースを初めて配備するときに使うことを目的に設計されています。アカウント情報を Identity Manager にすばやく読み込むための手段を提供します。そのため、調整や Active Sync に含まれる機能の一部が備えられていません。たとえば、探索プロセスではアカウントインデックスにエントリが追加されません。また、探索の前後にワークフローを実行することもできません。ただし、探索プロセスでは、相関規則が予期したとおりに機能するかどうかをより早く判定できます。

探索プロセスを開始すると、Identity Manager は入力アカウントが既存のユーザーと一致する（つまり相関する）かどうかを判定します。一致した場合、探索プロセスはアカウントをユーザーにマージします。一致しない入力アカウントがある場合は、新しい Identity Manager ユーザーが作成されます。

Identity Manager には、次の探索機能があります。

- 「ファイルから読み込み」：ファイル内に指定されたアカウントを読み取り、Identity Manager に読み込みます。
- 「リソースから読み込み」：リソースからアカウントを抽出し、Identity Manager に直接読み込みます。
- 一括操作の作成：ファイルに一覧されているユーザー作成コマンドを実行します。

個々の探索プロセスの詳細については、以降の項を参照してください。

ファイルから読み込み

「ファイルから読み込み」探索プロセスは、XML ファイルまたは CSV (Comma-Separated Value) ファイルに書き込まれたアカウント情報を読み取ります。

一部のリソース (Active Directory など) には、ネイティブのアカウント情報を CSV (Comma-Separated Value) 形式にエクスポートする機能があります。これらの CSV ファイルを使って、Identity Manager アカウントを作成できます。CSV 形式の詳細については、『Identity Manager 管理ガイド』を参照してください。

ファイルから読み込むときは、どのアカウント関連規則および確認規則を使用するかを指定する必要があります。詳細については、「関連と確認の規則」を参照してください。

リソースから読み込み

「リソースから読み込み」機能は、ターゲットシステムを走査し、すべてのユーザーの情報を返します。その後、Identity Manager がユーザーを作成して更新します。リソースから読み込む前に、アダプタがそのリソース用に設定されている必要があります。

リソースから読み込むときには、どのアカウント関連規則および確認規則を使用するかを指定する必要があります。詳細については、「関連と確認の規則」を参照してください。

一括操作の作成

一括操作では、複数のアカウントを同時に操作できます。一括操作を使って Identity Manager アカウントおよびリソースアカウントの作成、更新、および削除ができますが、ここでは Identity Manager アカウントの作成のみについて説明します。一括操作の詳細については、『Identity Manager 管理ガイド』を参照してください。

一括操作は、CSV (Comma-Separated Value) を使って指定します。これらの値の構造は、「ファイルから読み込み」プロセスで指定される値とは異なります。

この CSV 形式は、2 行以上の入力行で構成されます。各行は、コンマで区切られた値のリストで構成されます。1 行目にはフィールド名が入ります。残りの各行は、Identity Manager ユーザー、ユーザーのリソースアカウント、またはその両方に対して実行される操作に対応します。各行には、同じ数の値が含まれるようにしてください。値を空にすると、対応するフィールドの値は変更されません。

一括操作の CSV 入力には、常に次の 2 つのフィールドが必要です。

- **user:** Identity Manager ユーザーの名前を含みます。
- **command:** Identity Manager ユーザーに対して実行する操作を含みます。Identity Manager ユーザーを作成する場合は、この値を **Create** にする必要があります。

3 番目以降のフィールドは、ユーザービューのフィールドです。使用されるフィールド名は、ビュー内の属性のパス表現です。ユーザービューで利用可能な属性の詳細については、『*Identity Manager Workflows, Forms, and Views*』の「*Understanding the User View*」を参照してください。カスタマイズしたユーザーフォームを使用している場合は、フォーム内のフィールド名に使用可能なパス表現の一部が表示されます。

一括操作でよく使われるパス表現の一部を次に示します。

- `waveset.roles`: **Identity Manager** アカウントに割り当てる 1 つ以上のロール名のリストです。
- `waveset.resources`: **Identity Manager** アカウントに割り当てる 1 つ以上のリソース名のリストです。
- `waveset.applications`: **Identity Manager** アカウントに割り当てる 1 つ以上のリソースグループのリストです。
- `waveset.organization`: **Identity Manager** アカウントが配置される組織名です。
- `accounts[resource_name].attribute_name`: リソースアカウント属性です。属性の名前はリソースのスキーマに一覧されています。

複数の値を指定できるフィールドもあります。たとえば、`waveset.resources` フィールドを使って、1 人のユーザーに複数のリソースを割り当てることができます。フィールド内の複数の値を区切るには、縦棒 (|) 文字 (「パイプ」文字とも呼ぶ) を使用します。複数の値の構文は、次のようにして指定できます。

```
value0 | value1 [ | value2 ... ]
```

一括操作の作成の例を次に示します。

```
command,user,waveset.resources,password.password,password.confirmPassword,accounts[AD].description,accounts[Solaris].comment
Create,John Doe,AD|Solaris,changeit,changeit,John Doe,John Doe
Create,Jane Smith,AD,changeit,changeit,Jane Smith,
```

一括操作の作成は、「ファイルから読み込み」プロセスより汎用性があります。「ファイルから読み込み」では一度に 1 つのリソースから情報を読み込みますが、一括操作では複数のリソースを操作できます。

調整

調整では、アカウントインデックスの内容と各リソースに現在含まれている内容と比較します。調整では、次の機能を実行できます。

- 新規アカウントと削除されたアカウントの検出
- アカウント属性値の変更の検出
- アカウントと Identity Manager ユーザーの関連付け
- Identity Manager ユーザーと関連のないアカウントの検出
- 検出した個々のアカウント状況に応じたワークフローの実行
- リソース上の特定のコンテナからリソース上の別のコンテナにユーザーを移動した時刻の検出

注	調整を行う前に、アダプタがそのリソース用に設定されている必要があります。アダプタの詳細については、『Identity Manager Resources Reference』を参照してください。
---	---

調整には、完全調整と差分調整の2つの種類があります。

完全調整

完全調整は、アダプタがリストする各アカウント ID に対して、存在、所有権、状況を再計算します。これは所有権を再計算するために、リソースを要求する各 Identity Manager ユーザーをチェックします。

Identity Manager ユーザーは、次の方法でリソースを要求できます。

- リソースを間接的に指定するロールを使用する方法
- 直接的なリソース割り当てを使用する方法
- このリソースのアカウントを参照する方法
- リソースグループを使用する方法

調整プロセスは、アカウントインデックスに記録された Identity Manager 所有者がまだ存在し、アカウントを要求していることを、アカウントごとに確認します。所有者のいないすべてのアカウントは、このリソースの調整ポリシーで相関規則が指定されているかぎり、Identity Manager ユーザーと関連付けられます。相関規則で1人以上の所有者候補がいることがわかると、それぞれの候補が確認規則で二重にチェックされます(指定されている場合)。規則の詳細については、「相関と確認の規則」を参照してください。

アカウントに対して状況が設定されると、調整はこのリソースの調整ポリシーで設定された応答を実行します。調整ポリシーがアカウント単位で実行するワークフローを指定している場合、完全調整は、状況操作を実行したあとに、調整するアカウントごとにこれを実行します。ワークフローの詳細については、「調整ワークフロー」を参照してください。

差分調整

差分調整は、差分バックアップと同質のものです。完全調整よりも高速であり、必要な処理のほとんどを実行しますが、完全調整ほどには完璧ではありません。

差分調整では、アカウントインデックスで保守している情報が正確であると信頼しています。既知のアカウント ID のリストが正確であると信頼し、Identity Manager 所有者によるアカウントの所有は正確に記録されていると信頼しているので、差分調整はいくつかの処理段階を省略または短縮することができます。

差分調整は、リソースを要求している Identity Manager ユーザーを検査する手順を省略しています。また差分調整は前回のリソース調整のあとに追加または削除されたアカウントだけを対象にして状況を算出します。リソースのアカウントインデックスにあるアカウント ID のリストとリソースアダプタが返したアカウント ID のリストを比較して、この計算を実行します。新規アカウントは、既存のアカウントとして記録され、削除されたアカウントはもはや存在しないアカウントとして記録されます。これ以降、これらの 2 種類のアカウントだけが処理されます。

差分調整は完全調整よりもはるかに高速であり、使用する処理サイクルも少ないので、差分調整の実行頻度を高くし、完全調整の頻度を低くすると快適な環境にできます。

Active Sync

Active Sync では、リソースに対する変更を「待機」(ポーリング)し、差分変更をリアルタイムで検出します。Active Sync は変更を検出するように設計されているため、アカウント情報をはじめて Identity Manager に読み込むときには使用しないでください。代わりに、調整または探索プロセスを使用します。

一般に、次のような場合は Active Sync リソースに対して調整を実行します。

- リソースの最初の読み込みを実行する場合
- 属性を無視または除外するように Active Sync が設定されているために Identity Manager で更新されていない属性を検出する場合

Active Sync は、次の点で調整と異なります。

- Active Sync では、複数のアカウントにわたる属性の同期が取れているかどうかを確認するためのユーザーフォームを管理者が指定できます。

- すべての Active Sync 処理を完全に制御する処理規則を実装できます。これは、一般にリソース上のアカウントが変更されたときに特別な操作 (リポジトリ内の複数のオブジェクトを編集するなど) を実行する必要がある場合に使用します。

Active Sync では、適切に設定された Active Sync 対応アダプタを使用する必要があります。リソースを設定して Active Sync を実装する方法の詳細については、『Identity Manager 管理ガイド』を参照してください。

データ読み込みの種類の概要

次の表は、探索と調整の機能を比較したものです。

表 2-1 データ読み込みの種類の概要

機能	探索	調整	Active Sync
新規アカウントの検出	あり	あり	あり
削除されたアカウントの検出	なし	あり	あり
アカウント属性値の変更の検出	なし	あり	あり
Identity Manager ユーザーと関連のないアカウントの検出	あり	あり	あり
リソース上の特定のコンテナからリソース上の別のコンテナにユーザーを移動した時刻の検出	なし	あり	あり
アカウントと Identity Manager ユーザーの関連付け	あり	あり	あり
検出した個々のアカウント状況に応じたワークフローの実行	なし	あり	あり
スケジュール可能	なし	あり	あり
差分モード	なし	あり	該当しない
アカウントインデックスへのエントリの追加	なし	あり	あり
複数リソース上の属性の同期	なし	なし	あり

調整を管理する

調整プロセスは、主に管理者インタフェースを介して管理されます。ただし、このインタフェースから実行できない調整の機能もあります。たとえば、新しい相関規則や確認規則の作成、調整ワークフローの作成、または調整設定オブジェクトの編集が必要な場合があります。以降の項では、これらの機能およびその他について説明します。調整ポリシーの定義の概要については、『Identity Manager 管理ガイド』を参照してください。

調整ポリシー

調整ポリシーでは、調整タスクごとに (リソースによる) 一連の応答を設定できます。ポリシー内では、調整を実行するサーバーを選択し、調整の実行回数と実行時間を決定し、調整中に発生する個々の状況に対する応答を設定します。アカウント属性に対してネイティブに (Identity Manager を介さずに) 行われた変更を検出するように調整を設定することもできます。

これらの各ポリシー設定は、異なる範囲で定義できます。

- グローバルに (全リソースタイプに対して)
- 特定のリソースタイプに対して
- 個々のリソースインスタンスに対して

それぞれの範囲の値は、下位の各範囲のデフォルトになります。したがって、調整ポリシーは、継承ツリーを定義します。

- グローバルの値は、全リソースタイプのデフォルトになります。
- 各リソースタイプは、グローバル値を継承するか、特定の値を指定します。
- リソースタイプ値は、このタイプの全リソースインスタンスのデフォルトです。
- 各リソースインスタンスは、親リソースタイプの値を継承できます。また特定の値を指定できます。

継承は、リソースが多い場合に、ポリシーの管理を容易にします (特に多くのリソースが同じ設定である場合)。

たとえば、すべてのリソースを同じ方法で処理したい場合、1 セットのポリシー設定をグローバルレベルで管理するだけで済みます。すべての Windows リソースを 1 つの方法で処理し、すべての Solaris リソースを別の方法で処理したい場合、2 つのリソースタイプにそれぞれ 1 つずつポリシーを設定し、それぞれの範囲内でポリシー設定を管理する必要があります。一部のリソースインスタンスのリソースタイプレベル

で定義するポリシーに例外がある場合、これらの個々のリソースに対応するために、必要なポリシー設定を優先する（つまり、指定する）ことができます。各ポリシー設定は個々に継承されるので、指定する必要があるのは異なる設定だけです。その他のポリシー設定は、以前と変わらずに、上位から値を継承することができます。

関連と確認の規則

Identity Manager は、ユーザーにリンクされていないリソースアカウントと Identity Manager ユーザーを次の 2 段階で照合します。

- **関連** - 所有者の候補を検出します。
- **確認** - それぞれの所有者候補をテストします。

関連規則は、アカウントを所有している可能性がある Identity Manager ユーザーを検索します。確認規則は、ユーザーが実際にアカウントを所有しているかどうかを確認するために、アカウントに対して Identity Manager ユーザーをテストします。所有者候補をすばやく検索したり（名前または属性に基づいて）、負担の大きなチェックの実行を所有者候補だけに制限したりするので、この 2 段階のアプローチによって、Identity Manager は関連を最適化することができます。

調整ポリシーでは、各リソースに対して、関連規則と確認規則を選択できます。また、「確認規則なし」も指定できます。デフォルトの関連規則は、入力したアカウントのアカウント ID と厳密に一致する名前を持つユーザーを検索します。デフォルトでは、「確認規則なし」が使用されています。

注 関連規則と確認規則は、探索や Active Sync でも使用できます。

Identity Manager は、いくつかの関連規則と確認規則を sample/reconRules.xml 内で事前に定義しています。また独自の関連規則や確認規則を記述することもできます。SUBTYPE_ACCOUNT_CORRELATION_RULE または SUBTYPE_ACCOUNT_CONFIRMATION_RULE というサブタイプを持つすべての規則オブジェクトは、適切な調整ポリシー選択リストに自動的に表示されます。

関連規則

関連規則は、リソースアカウントの属性値に基づいて、ユーザー名のリストを生成できます。また関連規則は、ユーザーの選択に使用する属性条件のリストを生成できます（ユーザーオブジェクトのクエリー可能な属性を参照）。

要求されていない各アカウントに対して、関連規則は 1 回ずつ実行されます。

注 関連規則は、比較的「負担の大きくない処理」ですが、使用する場合は状況を選ぶべきです。可能であれば、負担の大きな処理は確認規則に割り当ててください。

Identity Manager は、sample/reconRules.xml 内でいくつかの関連規則を事前に定義しています。

- **アカウント ID と一致するユーザー名** : accountId 属性の値を返します。これはリソースアカウント ID と一致する名前を持つあらゆる Identity Manager ユーザーを所有者候補として選択します。これがデフォルトの関連規則です。
- **一致するアカウント ID を所有するユーザー** : 属性条件のリストを返します。これは同じ accountId 値に一致するリソースアカウントを持っているすべての Identity Manager ユーザーを所有者候補として選択します。
- **ユーザーの電子メールがアカウントの電子メールに一致** : アカウントの email 属性に基づいて Identity Manager ユーザーを選択する属性条件のリストを返します。

関連規則の入力は、アカウント属性のマップです。出力は次のいずれかでなければなりません。

- 文字列 (ユーザー名または ID を含む)
- 文字列要素のリスト (各ユーザー名または ID)
- WSAttribute 要素のリスト
- AttributeCondition 要素のリスト

より複雑な規則で、アカウント属性値を組み合わせたり、処理したりして、名前のリストや属性条件のリストを生成できます。

注 属性条件は、UserUIConfig オブジェクトの QueryableAttrNames として設定されているクエリー可能な属性を参照する必要があります。

たとえば、reconRules.xml には、関連規則の 4 番目のサンプル、「アカウントフルネームと一致するユーザーフルネーム」があります。追加の設定を実行しなければ、このサンプルは正常に動作しないので、この規則は XML コメントで無効にしています。この規則は、fullname に基づいて Identity Manager ユーザーを検索しますが、この属性はデフォルトではクエリー可能になっていません。

拡張属性の関連には、特別な設定が必要です。

- 拡張属性は、UserUIConfig でクエリー可能と指定する必要があります (QueryableAttrNames のリストに追加)。
- UserUIConfig の変更を有効にするには、Identity Manager アプリケーション (またはアプリケーションサーバー) を再起動する必要があります。

確認規則

確認規則は、関連規則が返す一致ユーザーごとに 1 回ずつ実行されます。

一般的な確認規則は、ユーザー画面の内部の値とアカウント属性の値を比較します。相関処理のオプションの第2段階として、確認規則は相関規則で表現できないチェックを実行します（または、相関規則では負担が大きいため評価できないチェック）。一般的に、確認規則が必要なのは、次の場合だけです。

- 相関規則が一致するユーザーを複数返す可能性がある場合
- 比較する必要があるユーザー値をクエリーできない場合

Identity Manager は、sample/reconRules.xml 内で2つの確認規則を事前に定義しています。

- **ユーザーの電子メールがアカウントの電子メールに一致**：ユーザーの電子メールがアカウントの電子メールに一致すると、true が値として返されます。ここでは、所有に関する多くの判断を、相関規則または確認規則で表現できるという事実を説明します。ただし、Identity Manager ユーザーの email 属性は、自動的にクエリー可能になるので、ほとんどの場合、これを相関規則で表現する方法が、より効率的だと思われます。
- **ユーザーの名前および姓がアカウントと一致**：XPRESS 言語を使用して、ユーザーの名前と姓を、アカウントの対応する値と比較します。

確認規則には、次のものを入力します。

- **ユーザー画面**：Identity Manager ユーザーの完全な画面です。
- **アカウント**：リソースアカウント属性のマップです。

ユーザーがアカウントを所有している場合、確認規則は文字列形式でブール値の true を返します。それ以外の場合、値として false を返します。

デフォルトの確認規則は、「確認規則なし」です。これは、相関規則がアカウントごとにはほぼ1人のユーザーを検出するという状況を前提としています。相関規則が複数のユーザーを選択する場合、アカウントの状況は「DISPUTED」になります。

調整ワークフロー

ユーザー定義のワークフローに対応するいくつかの接続ポイントを提供することにより、典型的な調整処理を拡張できます。

プレリソースワークフロー

プレリソースワークフローは、ほかの調整処理の前に起動できます。「Notify Reconcile Start」ワークフローは、プレリソースワークフローの例です。

「Notify Reconcile Start」ワークフローは、あるリソースに対する調整が開始したことを通知する電子メールを管理者に送信します。このワークフローを実行する前に、「Notify Reconcile Start」電子メールテンプレートを設定する必要があります。

プレリソースワークフローには、次のパラメータが渡されます。

- **resourceName** - これから調整するリソースの名前です。
- **resourceId** - これから調整するリソースのオブジェクト ID です。

アカウント単位ワークフロー

アカウント単位ワークフローは、応答 (使用する場合) が完了したあと、調整によって処理されるアカウントごとに起動されます。応答のタイプと応答の結果は、このワークフローに影響を与えません。

「Notify Reconcile Response」ワークフローは、アカウント単位ワークフローの例です。調整プロセスが、探索された状況に対して自動的に応答しようとしたときに管理者に電子メールを送信します。このワークフローを実行する前に、「Notify Reconcile Response」電子メールテンプレートを設定する必要があります。

アカウント単位ワークフローには、次のパラメータが渡されます。

- **accountId** - 調整されたリソースアカウントの名前です。
- **resourceId** - 調整しているリソースのオブジェクト ID です。
- **resourceName** - 調整しているリソースの名前です。
- **userId** - アカウントの所有者として識別される Identity Manager ユーザーのオブジェクト ID です (状況に応じて、要求または相関によって識別)。アカウントに関連付けられたユーザーがいない場合、この値は NULL になります。
- **userName** - アカウントの所有者として識別される Identity Manager ユーザーの名前です (状況に応じて、要求または相関によって識別)。アカウントに関連付けられたユーザーがいない場合、この値は NULL になります。
- **initialSituation** - アカウントに関して最初に探索され、応答を開始した状況です。値は有効なメッセージキーです。
- **responseSuccess** - 応答が正常に完了したかどうかを示すブール値です。実行された応答がない場合、値は NULL になります。
- **finalSituation** - 応答を適用したあとのアカウントの調整状況です。アカウントがすでに存在せず、Identity Manager にこのアカウントへの参照がない場合、この値は NULL になります。

ポストリソースワークフロー

ポストリソースワークフローは、ほかの調整処理がすべて完了したあとに起動できます。「Notify Reconcile Finish」ワークフローは、ポストリソースワークフローの例です。

「Notify Reconcile Finish」ワークフローは、あるリソースに対する調整が完了したことを通知する電子メールを管理者に送信します。このワークフローを実行する前に、「Notify Reconcile Finish」電子メールテンプレートを設定する必要があります。

ポストリソースワークフローには、次のパラメータが渡されます。

- **resourceName** - 調整したリソースの名前です。
- **resourceId** - 調整したリソースのオブジェクト ID です。

ネイティブ変更を監査する

「Audit Native Change To Account Attributes」ワークフローが起動されるのは、調整またはプロビジョナが、**Identity Manager** による変更ではなく、リソースアカウント属性のネイティブ変更を検出したときです。変更の監視が実行されるのは、ユーザーが指定した属性だけです。デフォルトでは、属性の監視は実行されません。

このワークフローには、次のパラメータが渡されます。

- **resource** - アカウントがネイティブに変更されたリソースオブジェクトです。
- **accountID** - ネイティブに変更されたリソースアカウントの名前です。
- **prevAttributes** - **Identity Manager** によって記録される監視対象のリソースアカウント属性を含むマップです。
- **newAttributes** - リソース上で現在設定されている監視対象のリソースアカウント属性を含むマップです。
- **attributeChanges** - どの属性が変更されたかを示す汎用オブジェクトのリストを含むマップです。各オブジェクトには、前の値と新しい値が格納されます。
- **formattedChanges** - 監査レコードに適した簡潔な形式で属性の変更を表す文字列です。

ネイティブ変更を監査するには、次を実行する必要があります。

- 「調整ポリシーの編集」ページで、「属性レベル調整」ドロップダウンメニューから「アカウント属性へのネイティブ変更を検出する」オプションを選択します。属性のリストを表示するには、「リソースタイプポリシーを継承」チェックボックスの選択を解除する必要がある場合があります。監査する属性を選択します。
- 監査イベントのリストに「**Identity Manager** の外部での変更」を追加します。そのためには、「設定」タブを選択し、左側の「監査イベント」を選択します。

リソースのスケジューリング

調整は、それぞれのリソースのために、差分調整と完全調整という 2 つの別々のスケジュールを保守します。

各リソースは、個々の「要求者」タスクによってスケジュールが管理されます。調整ポリシー GUI でリソースに対して調整スケジュールを設定すると、要求者タスク用の TaskSchedule が設定されます。必要があれば、これで外部のタスクスケジューラによって調整を管理できるようになります。またこれで調整タスクの負担が最小限になります。何も処理を実行していない調整デーモンは、ほとんどリソースを消費しません。このデーモンは、メモリー内のキューを定期的にポーリングしているからです (調整の準備が整ったリソースを探して、データベースのクエリーを実行することはありません)。

調整は、リソースアダプタを通じて、各リソースにアクセスします。調整はアダプタを直接的に呼び出して、アカウントのリスト、アカウントの反復、または個人用リソースアカウントのフェッチを実行します。また調整は、プロビジョナを通じて間接的にリソースにアクセスし、プロビジョナを使用してリソースアカウントを作成したり、リソースアカウントから Identity Manager ユーザーを作成したりします。

調整とプロビジョナは、両方ともアカウントインデックスの保守を実行します。また、リソースの調整を行うたびに、アカウントインデックスから不要な情報が除去されます。調整タスクは、Identity Manager ユーザーがアカウントを所有していないかぎり、リソースにもはや存在しないアカウントのエントリを自動的に削除します。したがって、アカウントインデックスにあるリソースの情報を手動でクリアする必要はありません。

各 Identity Manager サーバーは、デーモントaskとして調整を実行します。つまり、Identity Manager スケジューラは、調整タスクをすぐに開始し、これが停止した場合は、タスクを自動的に再開します。

注	リソース調整は自動的に再開されません。新しいリクエストに応答できるように、ReconTask デーモン自体は、自動的に再起動されますが、ホストサーバーが停止したとき (またはアプリケーションサーバーがシャットダウンされたとき) に実行していたリクエストは失われます。デーモンはリソース調整を再開しません。リクエストがないときに、リソースを調整するのは不適切だからです。
---	--

調整設定オブジェクト

ReconcileConfiguration オブジェクトには、「調整ポリシーの編集」ページで編集できない複数の属性が含まれます。

次の表で、調整の属性について説明します。ReconcileConfiguration オブジェクト (#ID#Configuration:ReconcileConfiguration) の属性値を編集するには、デバッグページを使用します。

表 2-2 ReconcileConfiguration オブジェクトの主要な属性

属性	説明
fetchTimeout	調整プロセスがアカウントをフェッチするときにリソースからの応答を待機するミリ秒数です。デフォルト値は、1 分 (60,000 ミリ秒) です。
listTimeout	調整プロセスがアカウントをリストするときにリソースからの応答を待機するミリ秒数です。デフォルト値は、10 分 (600,000 ミリ秒) です。
maxConcurrentResources	サーバーが同時に調整するリソースの最大数です。デフォルト値は、3 です。
maxQueueSize	調整サーバーの作業キューに含まれるエントリの最大数です。デフォルト値は、1000 です。

次の例では、ReconcileConfiguration オブジェクトのデフォルト値を示します。

コード例 2-1 ReconcileConfiguration オブジェクトのデフォルト値

```
<Configuration id='#ID#Configuration:ReconcileConfiguration'
name='Reconcile Configuration'>
  <Extension>
    <Object>
      <Attribute name='listTimeout' value='600000' />
      <Attribute name='fetchTimeout' value='60000' />
      <Attribute name='maxConcurrentResources' value='3' />
      <Attribute name='maxQueueSize' value='1000' />
    </Object>
  </Extension>
  <MemberObjectGroups>
    <ObjectRef type='ObjectGroup' id='#ID#All' name='All' />
  </MemberObjectGroups>
</Configuration>
```

Active Sync を管理する

Active Sync 対応アダプタは、管理者インタフェースで管理できます。このインタフェースには、1 つのアダプタに対する Active Sync のほとんどの設定項目を管理者が完全に設定できるウィザードが含まれます。このウィザードでは、管理者が Identity Manager IDE を使わずにリソース、入力、またはフォームを構築することもできます。Active Sync ウィザードの詳細については、『Identity Manager 管理ガイド』を参照してください。

Active Sync 対応アダプタが機能する仕組み

この項では、次の内容を説明します。

- アダプタ処理の基本的な手順の概要
- 規則の使用
- フォームの使用
- ワークフロープロセスの起動

アダプタ処理の基本的な手順

すべての Active Sync 対応アダプタは、Identity Manager で定義されているリソースの変更を知るために、次の基本的な手順でリスニングまたはポーリングを実行します。変更されたリソースを検出すると、Active Sync 対応アダプタは次の手順を実行します。

1. リソースから変更された情報を抽出します。
2. どの Identity Manager オブジェクトに関係があるか判断します。
3. システムに渡すユーザー属性のマップを、アダプタの参照および追加オプションのマップとともに生成します。これにより、Identity Application Programming Interface (IAPI) オブジェクトが作成されます。
4. IAPI オブジェクトを ActiveSync マネージャーに送信します。
5. ActiveSync マネージャーは、このオブジェクトを処理し、処理の成功を Active Sync 対応アダプタに通知する WaveSetResult オブジェクトをアダプタに返します。このオブジェクトには、ID の更新のために Identity Manager システムが使用するさまざまな手順の結果を多く含めることができます。一般に、ワークフローは Identity Manager 内のエラーにも対応し、多くの場合、担当管理者の承認にあとを任せます。

規則を使用する

Active Sync 対応アダプタは、リソース上でのアカウントの変更を検出すると、受け取った属性を Identity Manager ユーザーにマップします。あるいは、一致するアカウントがない場合や、アカウントを作成するように Active Sync リソースが設定されている場合は、Identity Manager ユーザーアカウントを作成します。

Active Sync ウィザードでは、さまざまな条件が発生したときの処理を制御するための規則を指定できます。次の表は、規則の種類についてそれぞれ説明しています。

表 2-3 規則の種類

パラメータ	説明
処理規則	TaskDefinition の名前、またはフィールド内のすべてのレコードに対して実行される TaskDefinition の名前を返す規則のいずれかです。この処理規則は、activeSync 名前空間内のリソースアカウント属性を、リソース ID およびリソース名とともに取得します。 処理規則は、システムがリソース上の変更を検出したときに実行されるすべての機能を制御します。アカウント処理を完全に制御する必要がある場合に使用します。この結果、処理規則はほかのすべての規則より優先されます。 処理規則が指定されると、このアダプタ上にほかのどんな設定があっても、すべての行に対してその処理が実行されます。 処理規則は、少なくとも次の機能を実行する必要があります。 • 一致するユーザービューに対するクエリー。 • ユーザーが存在する場合は、ビューのチェックアウト。ユーザーが存在しない場合は、ユーザーの作成。 • ビューの更新またはビューへの設定。 • ユーザービューのチェックイン。 ユーザー以外のオブジェクト (LDAP ロールなど) を同期することもできます。

表 2-3 規則の種類 (続き)

パラメータ	説明
相関規則	リソースアカウントを所有する Identity Manager ユーザーのリソース情報が特定されない場合、Identity Manager は相関規則を呼び出し、(アカウントの名前空間内の) リソースアカウント属性に基づいて、ユーザーの照合に使用する、一致する可能性のあるユーザーまたはアカウント ID の候補のリスト、あるいは属性条件を特定します。 規則は、エントリを既存の Identity Manager アカウントに関連付けるために使用できる次のいずれかの情報を返します。 • Identity Manager ユーザー名 • WSAttribute オブジェクト (属性ベースの検索に使用) • AttributeCondition 型または WSAttribute 型の項目のリスト (AND 結合による属性ベースの検索) • String 型の項目のリスト (各項目は Identity Manager アカウントの Identity Manager ID またはユーザー名) 相関規則によって複数の Identity Manager アカウントが識別された場合は、複数の一致を処理するために確認規則またはプロセス解決規則が必要です。 データベーステーブル、フラットファイル、および PeopleSoft コンポーネントの Active Sync アダプタの場合は、デフォルトの相関規則はリソース上の調整ポリシーから継承されます。 調整と Active Sync で同じ相関規則を使用できます。詳細については、「相関と確認の規則」を参照してください。
確認規則	相関規則によって返されるすべてのユーザーを対象にして評価される規則です。ユーザーごとに、Identity Manager の ID と (「account.」名前空間にある) リソースアカウント情報の相関を示す完全なユーザー画面が確認規則に渡されます。確認規則は、ブール値で表すことができる値を返すことが期待されます。たとえば、「true」または「1」または「yes」と、「false」または「0」または NULL です。 データベーステーブル、フラットファイル、および PeopleSoft コンポーネントの Active Sync アダプタの場合は、デフォルトの確認規則はリソース上の調整ポリシーから継承されます。 調整と Active Sync で同じ確認規則を使用できます。詳細については、「相関と確認の規則」を参照してください。

表 2-3 規則の種類 (続き)

パラメータ	説明
削除規則	<code>activeSync</code>、または <code>account</code>、という形式のキーを持つ値すべてのマップを期待できる規則です。プロキシ管理者のセッションに基づく <code>LighthouseContext</code> オブジェクト (<code>display.session</code>) は、この規則のコンテキストで利用できます。この規則は、ブール値で表すことができる値を返すことが期待されます。たとえば、「true」または「1」または「yes」と、「false」または「0」または NULL です。 あるエントリに関してこの規則によって <code>true</code> が返された場合、アダプタの設定方法に応じて、フォームとワークフローを介してアカウント削除リクエストが処理されます。
プロセス解決規則	<code>TaskDefinition</code> の名前、またはフィード内のあるレコードに対して複数の一致がある場合に実行される <code>TaskDefinition</code> の名前を返す規則のいずれかです。プロセス解決規則は、リソースアカウント属性をリソース ID およびリソース名とともに取得します。 この規則は、一致がなく、「一致しないアカウントの作成」が選択されていない場合にも必要です。 このワークフローは、管理者による手動操作を求める処理にすることもできます。
一致しないアカウントの作成	<code>true</code> に設定すると、一致する <code>Identity Manager</code> ユーザーが見つからない場合に、リソース上にアカウントが作成されます。<code>false</code> に設定すると、処理規則が設定され、その規則が識別するワークフローによって新しいアカウントが保証されていることが確認されないかぎり、<code>Identity Manager</code> はアカウントを作成しません。デフォルトは <code>true</code> です。
グローバルで利用	<code>true</code> に設定すると、<code>activeSync</code> 名前空間に加えてグローバル名前空間にも値が入力されます。デフォルト値は、<code>false</code> です。

アダプタがユーザーを検出しなかった場合

`Identity Manager` が既存の `Identity Manager` ユーザーと一致するものを見つけられない場合は、「一致しないアカウントの作成」が `true` に設定されているか、プロセス解決ワークフローで `feedOp of create` が指示されているか、更新操作が作成操作に変更されます。

`feedOp` フィールドを利用できるのは、ユーザーの作成、削除、または更新を実行するロジックを設定したフォームです。このフィールドを使用すると、特定イベントの固有なフィールドを無効化、あるいは有効化することができます (たとえば、`feedOp` フィールドが作成に設定されると、パスワードを生成するなど)。

この例の feedOp フィールドがパスワードを作成するのは、Active Sync 対応アダプタが Identity Manager ユーザーと一致しないユーザーをリソース内で検出し、Identity Manager 内にこのユーザーを作成するときだけです。

コード例 2-2 feedOp フィールドの例

```
<Field name='waveset.password'>
  <Disable>
    <neq>
      <ref>feedOp</ref>
      <s>create</s>
    </neq>
  </Disable>
  <expression>
    <cond>
      <notnull>
        <ref>activeSync.password</ref>
      </notnull>
      <ref>activeSync.password</ref>
      <s>change12345</s>
    </cond>
  </expression>
</Field>
```

フォームを使用する

一般的に、Active Sync 対応アダプタは、処理中にリソースフォームとユーザーフォームの2つのフォームを使用します。

フォーム処理は、次の3つの手順で実行されます。

1. Active Sync フィールドには、属性とリソース情報が入れられます。activeSync 名前空間を使用してリソース上の属性を取得および設定します。
2. リソースフォームが拡張され、取得されます。この拡張処理の間、すべてのユーザービュー属性が利用できます。
3. ユーザーフォームが拡張され、取得されます。

\$WSHOME/sample/forms ディレクトリには、末尾に ActiveSyncForm.xml が付けられたサンプルフォームがあります。これらのフォームには、新規ユーザーと既存ユーザーの対応を分けるロジックのほかに、リソース上で削除を検出したときに Identity Manager ユーザーの無効化または削除を行うロジックも含まれています。

注	リソースフォームにはリソース固有のロジックだけを設定し、ユーザーフォームに共通のロジックを設定します。リソース固有のロジックは、feedop フィールドが NULL 以外の値になったときに有効になります。リソースフォームを「なし」に設定すると、すべての Active Sync 属性 (accountId を除く) がグローバル属性と見なされ、自動的にその設定がほかの属性に反映されます。
---	--

リソースフォームにはリソース固有のロジックだけを設定し、ユーザーフォームに共通のロジックを設定します。リソース固有のロジックは、feedop フィールドが NULL 以外の値になったときに有効になります。リソースフォームを「なし」に設定すると、すべての Active Sync 属性 (accountId を除く) がグローバル属性と見なされ、自動的にその設定がほかの属性に反映されます。

リソースフォーム

リソースフォームは、リソースの作成または編集時に、管理者がプルダウンメニューから選択するフォームです。選択したフォームの参照は、リソースオブジェクトに保存されます。

リソースフォームは、次のように Active Sync 対応アダプタで使用されます。

- スキーママップから受け取った属性を変換します。
- パスワード、ロール、および組織などのフィールドを生成します。
- 新規ユーザーと既存ユーザーの対応を分けるロジック、削除を検出したときに Identity Manager ユーザーの無効化または削除を行うロジックなど、カスタマイズした処理のために、簡単な制御ロジックを提供します。
- ユーザーフォームが入力として使用するフィールドに、ActiveSync から取得した属性をコピーし、オプションでこの値を変換します。作成処理に必要なフィールドは、waveset.accountId と waveset.password です。ほかのフィールドも設定できます (たとえば、accounts[NT].email または waveset.resources)。
- IAPI.cancel に true を設定して、ユーザーの処理をキャンセルします。多くの場合、これは特定のユーザーの更新を無視するために使用します。

次の例では、「Doe」という姓を持つ全ユーザーを無視する簡単なフィールドを説明します。

コード例 2-3 「Doe」という姓を持つ全ユーザーを無視するフィールド

```

<Field name='IAPI.cancel'>
  <Disable>
    <neq>
      <ref>activeSync.lastName</ref>
      <s>Doe</s>
    </neq>
  </Disable>
  <expression>
    <s>true</s>
  </expression>
</Field>

```

リソースフォームには、新規ユーザーと既存ユーザーの対応を分けるロジック、削除を検出したときに Identity Manager ユーザーの無効化または削除を行うロジックがあります。

ユーザーフォーム

ユーザーフォームは、Identity Manager インタフェース上での編集作業に使用します。プロキシ管理者をアダプタに割り当てることで、ユーザーフォームを割り当てます。プロキシ管理者が自分に関連付けられたユーザーフォームを持っている場合、処理の実行時に、このフォームがユーザー画面に適用されます。

プロキシ管理者とユーザーフォーム

ProxyAdministrator 属性を使用して、アダプタに対してプロキシ管理者を設定します。この属性は、あらゆる Identity Manager 管理者に設定できます。すべての Active Sync 対応アダプタの処理は、プロキシ管理者が実行したかのように実行されます。プロキシ管理者が割り当てられていない場合、デフォルトユーザーフォームが指定されます。

属性を処理する代替フォーム

ベストプラクティスとしては、名と姓からフルネームを導出するなど、共通の変更はユーザーフォームに設定することをお勧めします。リソースフォームには、HR ステータスの変更時にユーザーを無効化するなど、リソース固有の変更だけを設定します。ただし、目的の属性が incoming など共通のパスに入れられたあとは、組み込みフォームなどに代替的に共通の変更を設定できます。

```
<Form>
  <Field name='incoming.lastname'>
    <ref>activeSync.lastname</ref>
  </Field>
  <Field name='incoming.firstname'>
    <ref>activeSync.firstname</ref>
  </Field>
</Form>
```

これ以降、共通のフォームでは、共通のロジックのために `incoming.xxx` を参照します。

```
<Form>
  <Field name='fullname'>
    <concat>
      <ref>incoming.firstname</ref>
      <s> </s>
      <ref>incoming.lastname</ref>
    </concat>
  </Field>
</Form>
```

キャンセル操作の実行

ユーザーの処理をキャンセルするには、リソースフォームで `IAPI.cancel` に **true** を設定します。これを使用すると、特定のユーザーへの更新を無視できます。

注 `Active Sync` フォームで `IAPI.cancel` に **true** の値が設定された場合、`IAPISync` イベントまたは `IAPISync` イベントに関連付けられたプロセスは起動されません。

次の例では、「Doe」という姓を持つ全ユーザーを無視する、リソースフォーム内の簡単なフィールドを説明します。

```
<Field name='IAPI.cancel'>
  <Disable>
    <eq><ref>activeSync.lastName</ref><s>Doe</s></eq>
  </Disable>
  <Expansion>
    <s>true</s>
  </Expansion>
</Field>
```

ワークフロープロセスを起動する

管理者は、Active Sync ウィザードを使ってポーリング前またはポーリング後のワークフローを指定できます。これらのワークフローは、「調整ワークフロー」で説明したワークフローと概念上ほぼ同じです。

一部の Active Sync 対応アダプタは、取得したユーザー画面の変更をチェックする代わりに、指定されたワークフローを実行するリソース属性をサポートします。このワークフローは、Active Sync データだけで構成される入力変数で実行されます。個々のプロセスをサポートしないアダプタの場合、または標準的なユーザーフォームを使用してプロセスを起動したいアダプタの場合、オプションを設定してプロセスよりも優先させることができます。

```
<Form>
  <Field name='sourceOptions.Process'>
    <Expansion>
      <s>My workflow process name</s>
    </Expansion>
  </Field>
</Form>
```

フォームで指定したワークフローは、標準的なプロビジョニングワークフローと同じように呼び出されます。カスタムワークフローは、標準的な作成ワークフローや更新ワークフローに基づいて作成することを強くお勧めします。workflow.xml 内のユーザー作成ワークフローやユーザー更新ワークフローを参考にしてください。

例：Active Sync 対応アダプタからアカウントを無効化する

この例では、企業における従業員の現在のステータスでリソース (HR データベース) を更新できます。Active Sync 対応アダプタは、この HR データベースからの入力に基づいて、Identity Manager リポジトリを更新し、企業全体を通じてユーザーアカウントの無効化、削除、作成、あるいはその他の操作を実行できます。

次のコード例では、Status という属性を受け取り、これがアクティブ (「A」) でない場合、従業員の全アカウントを無効化します。次の表では、この属性の 4 つの状態を示します。

表 2-4 属性の状態

状態	説明
A	アクティブ
T	終了
L	中断
S	変更の保留

Status 属性の値に基づいて、アカウントの有効化または無効化を設定できます。

コード例 2-4 受け取った非アクティブな Status 属性に基づくアカウントの無効化

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE Configuration PUBLIC 'waveset.dtd' 'waveset.dtd'>
<Configuration wstype='UserForm' name='PeopleSoft ActiveSync Form'>
  <Extension>
    <Form>
<!-- this is a sample of how to map the accountID to a different
field than the one from the schema map
Commented out because we want to use the default account
ID mapped from the resource Schema Map.
    <Field name='waveset.accountId'>
      <Disable><neq><ref>feedOp</ref><s>create</s></neq></Disable>
      <Expansion>
        <concat><s>ps</s><ref>waveset.accountId</ref></concat>
      </Expansion>
    </Field>
  -->

  <!-- this is the real one, limited to create -->
```

コード例 2-4 受け取った非アクティブな **Status** 属性に基づくアカウントの無効化 (続き)

```

    <Field name='waveset.accountId'>
      <Disable><neq><ref>feedOp</ref><s>create</s></neq></Disable>
      <Expansion>
        <ref>activeSync.EMPLID</ref>
      </Expansion>
    </Field>

<!-- we need to make up a password for accounts that are being
      createdThis picks the last six digits of the SSN.
-->
    <Field name='waveset.password'>
      <Disable><neq><ref>feedOp</ref><s>create</s></neq></Disable>
      <expression>
        <s>change123456</s>
      </expression>
    </Field>

    <Field name='waveset.resources'>
<!--      <Disable><neq><ref>feedOp</ref><s>create</s></neq></Disable> -->
<!-- Don't change the resources list if it already contains peoplesoft -->
      <Disable>
        <member>
          <ref>activeSync.resourceName</ref>
          <ref>waveset.resources</ref>
        </member>
      </Disable>

      <expression>
        <appendAll>
          <ref>waveset.resources</ref>
          <ref>activeSync.resourceName</ref>
        </appendAll>
      </expression>
    </Field>

    <!-- Status is mapped by the schema map to PS_JOB.EMPL_STATUS which
has at least four states - A for active, T terminated,
L laid off, and S which is a pending change.
The audit data tells us what the state was, and the global
data tells us what it is. Based on the change we can disable
or enable the account
Note that this can happen on a create also!
-->
    <Field>
      <Disable><eq><ref>activeSync.Status</ref><s>A</s></eq></Disable>

    <Field name='waveset.disabled'>

```

コード例 2-4 受け取った非アクティブな Status 属性に基づくアカウントの無効化 (続き)

```

        <Expansion>
            <s>true</s>
        </Expansion>
    </Field>
    <FieldLoop for='name' in='waveset.accounts[*].name'>
        <Field name='accounts[$(name)].disable'>
            <expression>
                <s>true</s>
            </expression>
        </Field>
    </FieldLoop>

</Field>

<!-- Status is mapped by the schema map to PS_JOB.EMPL_STATUS which has at least four
states - A for active, T terminated, L laid off, and S which is a pending change.

This is the enable logic. It is disabled if the account status is <> A or is already enabled
-->
    <Field>
        <Disable>
            <neq>
                <ref>activeSync.Status</ref>
                <s>A</s>
            </neq>
        </Disable>

        <Field name='waveset.disabled'>
            <Disable><eq><ref>waveset.disabled</ref><s>>false</s></eq></Disable>
            <Expansion>
                <s>>false</s>
            </Expansion>
        </Field>

        <FieldLoop for='name' in='waveset.accounts[*].name'>
            <Field name='accounts[$(name)].disable'>
                <Expansion>
                    <s>>false</s>
                </Expansion>
            </Field>
        </FieldLoop>
    </Field>

```

コード例 2-4 受け取った非アクティブな **Status** 属性に基づくアカウントの無効化 (続き)

```
</Form>
</Extension>
<MemberObjectGroups>
  <ObjectRef type='ObjectGroup' id='#ID#Top' name='Top' />
</MemberObjectGroups>
</Configuration>
```

データ読み込みのシナリオ

この章では、アカウント情報を **Identity Manager** に読み込むための準備をするときに考慮すべきヒントを紹介します。遭遇する可能性のあるいくつかの問題について解説するためにシナリオ例も示します。

環境を評価する

Identity Manager へのユーザーアカウント情報の読み込みを開始する前に、次の質問の中で自分の環境に該当するものがあるかどうかを判断してください。

- すべてのユーザー ID に対して、アイデンティティ情報の源泉として信頼性の高いリソースが存在しますか。

答えが「はい」の場合、**Identity Manager** へのユーザーアカウントの読み込みは容易です。そのリソースを最初のリソースとして使用し、ほかのリソースからアカウントを読み込み、相関規則を使用してアカウントどうしをリンクします。

- 重複しているが相関キーが存在するリソースから、ユーザーの完全なリストを取得できますか。

答えが「はい」の場合、ユーザーアカウントを読み込むプロセスはほぼ同じになります。ほかのリソースからアカウントを読み込む前に、必ず重複のあるリソースからユーザーアカウントが **Identity Manager** に読み込まれるようにしてください。

- 重複していて相関キーが存在しないリソースからユーザーのリストを取得できますか。

答えが「はい」の場合、それらのリソースからユーザーの一意の集合を最も簡単に検出できる方法を判断します。多くの場合、リソースごとにユーザーを手動で相関させたり削除したりする必要があります。

これらすべての質問に対して答えが「いいえ」の場合、アカウントを読み込むプロセスは難しくなります。可能な範囲でユーザーアカウントを読み込み、ほかの Identity Manager ユーザーの作成は部署の管理者またはエンドユーザーに委任することを検討してください。

最初のリソースを選択する

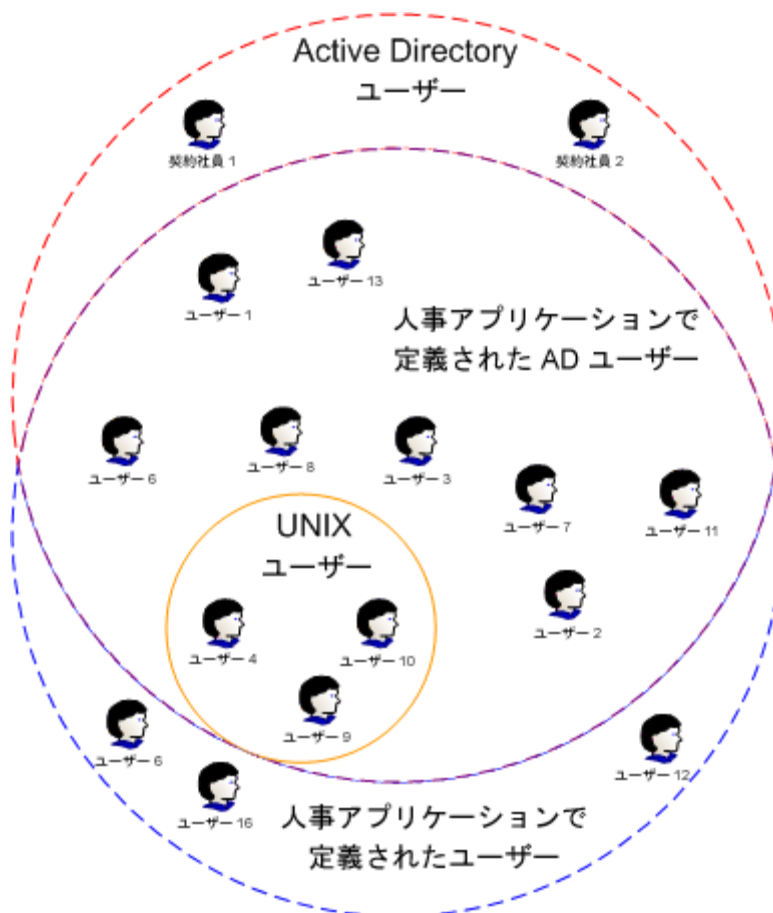
Identity Manager にアカウントを読み込むために使用する最初のリソースは、次の特徴を備えていることが理想的です。

- **広範囲のユーザーの集合を参照していること。**最初の読み込みの目的は、できるかぎり多くのアカウントを Identity Manager に取り込むことです。したがって、次のアプリケーションが選択肢として適しています。
 - PeopleSoft や SAP などの人事アプリケーション (アプリケーションに契約社員やその他の臨時職員のデータが含まれていない場合は、それらの従業員のアカウントを別途読み込むようにしてください)。
 - LDAP や Active Directory などのディレクトリベースアプリケーション。多くの場合、大半のユーザーが中央の組織または組織単位に定義されます。
- **Identity Manager のアカウント ID を構築するための十分な情報を保持していること。**個々の Identity Manager アカウント ID は一意である必要があります。一意性が保証されており、Identity Manager のアカウント ID として使用できる属性がリソースに存在していることが理想的です。そのような属性の例としては、従業員 ID や Active Directory の sAMAccountName 属性などがあります。姓と名を組み合わせてアカウント ID を作成することもできますが、この方法では一意の Identity Manager アカウントが作成されることを保証できません。
- **関連キーとして使用できるユーザー属性を格納していること。**Identity Manager でリソースアカウントをリンクするには、複数のリソースにまたがって同じ値を持つ属性が必要です。二次リソースでの値が常に、最初のリソースでの値と完全に一致するのが理想的です。さらに、二次リソースでの値がそのリソース内で一意であればなお理想的です。最も適した属性としては従業員 ID やフルネームがありますが、それら以外でも、すでにあるもので、複数のリソース間で値が一貫していればどのような属性でも使用できます。
- **アイデンティティ情報の源泉として信頼性が高いとみなせるデータを保持していること。**ユーザーが自分自身のアカウントデータを編集できる場合、システム間でデータの一貫性が失われる可能性があります。

次の図は、3 種類のリソースが存在するある会社のシナリオを例示したものです。この会社のほとんどの従業員は、PeopleSoft や SAP などの人事アプリケーションで定義されています。ただし、この会社では契約社員の情報を人事アプリケーションに入力しないため、このアプリケーションを使用して Identity Manager に契約社員のデータを読み込むことはできません。Active Directory でもほとんどのユーザーを定義して

いますが、すべてを網羅しているわけではありません(コンピュータへのアクセスを必要としない工場従業員などがこれに該当する場合があります)。このように、大部分のユーザーは両方のリソースで定義されていますが、どちらのリソースもすべてのユーザーを網羅しているわけではありません。一部の従業員は UNIX アカウントも持っています。

図 3-1 小規模なデータ読み込みのシナリオ



どのリソースを最初のリソースとして選択すべきでしょうか。UNIX リソースはごく少数のユーザーしか網羅していないため、問題なく検討から外すことができます。Active Directory と人事アプリケーションにはほとんど同数のユーザーが登録されているため、どちらにも明確な利点はありません。

Active Directory と人事アプリケーションのどちらのリソースを最初に読み込むべきかの判断材料となる要素として、次の点が挙げられます。

- Identity Manager 内でアカウントを管理する緊急性。人事アプリケーションでのみ定義され Active Directory では定義されていない従業員に、UNIX やその他のシステムのアカウントなど、追加のリソースアカウントが一切付与されていない場合、人事アプリケーションは Active Directory リソースほどには重要でない可能性があります。
- 相関キー。UNIX アカウントにも存在する複数の属性が一方のリソースに存在する場合、その属性がより適切な選択肢である場合があります。
- Identity Manager ログイン名。一方のリソースが、Identity Manager のログイン名としてより適切な名前を生成する場合、これが決定的な要因となる可能性があります。

最初のデータ読み込みプロセスを選択する

ユーザーデータを Identity Manager に読み込むために最初に使用するリソースを選択したあとは、使用するプロセスを決定する必要があります。次の表は、各種のデータ読み込みプロセスの長所と短所をまとめたものです。それぞれのデータ読み込みプロセスについては、次節以降で詳しく検討します。

表 3-1 データ読み込みプロセスの概要

データ読み込みプロセス	長所	短所
ファイルから読み込み	• 最も高速な読み込みプロセス。 • どの属性を読み込むかの制御が容易。 • 調整よりも設定が容易で高速。	• 顧客側にリソースから CSV ファイルを生成する時間が必要。 • 本稼働環境への投入前に完全調整が必要。 • アカウントの更新には使用できない。
リソースから読み込み	• すべてのリソースに対して有効 • 調整よりも設定が容易	• 読み込むリソースアカウントを自由に選択できない。 • 本稼働環境への投入前に完全調整が必要。
一括操作の作成	1 名の Identity Manager ユーザーに複数のアカウントを同時に追加できます。	• リソースからの読み込みや調整と比べて低速。 • リソースから CSV ファイルを容易に生成できない。 • この機能を十分に活用するには Identity Manager の詳細な知識が必要。 • 本稼働環境への投入前に完全調整が必要。

表 3-1 データ読み込みプロセスの概要 (続き)

データ読み込みプロセス	長所	短所
調整	- 調整ポリシーのすべての側面を実装可能 - 事前の調整を使用することで最終段階での不測の事態を回避できる	- 読み込むリソースアカウントを自由に選択できない。 - 大規模環境 (従業員が 50,000 名を超える) では全アカウントの読み込みに膨大な時間がかかる可能性がある
ActiveSync	アカウント情報を読み込む手段として ActiveSync を選ぶことは可能なかぎり避けてください。ActiveSync は変更を検出するように設計されており、結果として初回の読み込みは低速です。	

ファイルから読み込み

「ファイルから読み込み」プロセスは、Identity Manager アカウントにアカウント ID、姓、名、電子メールアドレスなどの基本的な値を割り当てます。アカウント ID は唯一の必須属性です。

「ファイルから読み込み」プロセスでは、CSV (Comma-Separated Value) ファイルの内容を Identity Manager にインポートします。このファイルの先頭行には、属性名のリストをコンマ区切りで記述します。それ以降の行には、対応する一連の属性値を記述します。すべての属性はコンマで区切る必要があります。

注	「ファイルから読み込み」では XML ファイルからの読み込みも可能ですが、XML ファイルの構文が、「ファイルへ抽出」機能によって生成されるファイルの構文と一致する必要があります。この形式について、ここでは詳しく説明しません。
---	---

CSV ファイル内のデータは多くの場合、リソースからエクスポートされたものです。たとえば、Microsoft 管理コンソール (MMC) の「Active Directory ユーザーとコンピュータ」では、組織単位の内容を CSV ファイルに直接エクスポートすることができます。組織単位に定義されているすべてのユーザーが、表示されている属性とともにエクスポートされます。したがって、「Active Directory ユーザーとコンピュータ」MMC でエクスポートを実行するときは、Identity Manager によって管理される属性だけが表示されていることを確認してください。不必要な属性が含まれていると、読み込みにかかる時間が長くなります。

リソースの中には、ユーザーアカウント情報を CSV 形式に直接エクスポートできないものもあります。そのようなリソースからデータを読み込む場合、別途プログラムを使用して情報を抽出したり、手動でデータを追加したりするなどの手順が必要な場合があります。

たとえば、CSV ファイルの最初の 3 行が次のようになっていますとします。

```
accountId,firstname,lastname,EmployeeID
```

```
Josie.Smith,Josie,Smith,1436
```

```
AJ.Harris,Anthony,Harris,c310
```

CSV ファイルに記述されている属性が、ユーザー画面の属性としてあらかじめ定義されている必要があります。accountId、email、password、confirmpassword などの基本属性はあらかじめ定義されています。その他の属性は、拡張ユーザー属性設定オブジェクトで定義されます。このオブジェクトはデフォルトで、firstname、lastname、fullname を利用可能な属性のリストに追加します。

Identity Manager の定義済みでない従業員 ID などの属性値を保持したい場合、拡張ユーザー属性設定オブジェクトにそれらの属性を追加する必要があります。

「ファイルから読み込み」プロセスの設定ページでは、関連規則と確認規則の指定を求められます。これが最初のデータ読み込みであるため、「アカウント ID と一致するユーザー名」関連規則を選択します。確認規則は不要です。

重要な点として、CSV ファイルに含まれているデータは Identity Manager アカウントを構成する目的のみに使用されます。たとえば、データが Active Directory から直接エクスポートされた場合でも、そのデータはどの Active Directory アカウントまたは Active Directory リソースにもリンクされません (リンクを行うことを目的としたカスタムユーザーフォームを作成した場合を除く)。カスタムユーザーフォームがない場合、リソースのアカウントデータを Identity Manager のユーザーとリンクするには、別のデータ読み込み機構を使用する必要があります。ユーザーフォームについては「[ユーザーフォームを割り当てる](#)」で簡単に説明します。

注	「ファイルから読み込み」プロセスでは、Identity Manager アカウントインデックスにエントリが追加されません。したがって、Identity Manager の配備が完了する前に、完全調整またはユーザーの更新を実行する必要があります。また、「ファイルから読み込み」では、Identity Manager でのユーザー作成時にワークフローは一切実行されません。
---	--

リソースから読み込み

「リソースから読み込み」プロセスと「ファイルから読み込み」プロセスの設定ページはほとんど同じ内容ですが、「リソースから読み込み」は機能的には調整に似ています。「リソースから読み込み」プロセスと調整プロセスでは、データをリソースから抽出したあとで、見つかったアカウントを Identity Manager に追加します。したがって、これらの操作を実行する前にアダプタを設定する必要があります。

「リソースから読み込み」は調整と比べて初回の実行が高速ですが、アカウントインデックスに値が設定されません。差分モードでの 2 回目の実行時の調整プロセスは、「リソースから読み込み」よりも高速です。調整が長期的なソリューションとして望ましいツールである場合は、ユーザーの初回読み込みに調整を使用してください。

ユーザーフォームは、アカウント ID の設定、組織へのユーザーの配置、ユーザー作成に関連するその他のタスクの実行に使用できます。詳細については、「[ユーザーフォームを割り当てる](#)」を参照してください。

最初に読み込むリソースに対して、「リソースから読み込み」を使用して Identity Manager アカウントに属性値を割り当てるとき、関連規則と確認規則はそれほど重要な意味を持ちません。「アカウント ID と一致するユーザー名」関連規則を選択します。確認規則は不要です。

注	「リソースから読み込み」プロセスでは、Identity Manager アカウントインデックスにエントリが追加されません。したがって、Identity Manager の配備が完了する前に、完全調整またはユーザーの更新を実行する必要があります。また、「ファイルから読み込み」では、Identity Manager でのユーザー作成時にワークフローは一切実行されません。
---	--

一括操作の作成

一括操作の作成では、CSV ファイルからデータを読み込みます。一括操作の作成では「ファイルから読み込み」プロセスと異なり、Identity Manager に固有の属性、グローバル属性、リソースアカウント属性などを含む、任意の書き込み可能な属性をユーザービューで定義することができます。この柔軟性の代償として、一括操作の CSV ファイルの作成は多くの場合、多少手順が複雑になります。複数のリソースを一括操作で処理する場合、リソースデータを単一の CSV ファイルにマージする方法を探す必要があります。ただし、一括操作の比較的単純なファイルを定義し、あとで更新操作を使用してデータを Identity Manager ユーザーアカウントに読み込むという方法もあります。

一括操作では、作成、更新、削除の各操作に対してデフォルトのワークフローを実行します。これによりユーザーアカウントの読み込み処理の速度は低下しますが、柔軟性が大きく向上します。

次の例では、一括操作の作成のみを使用します。command 属性と user 属性は必須です。

```
command,user,waveset.resources,password.password,password.confirmPassword,accounts[MyAD].description,accounts[MySolaris].comment  
  
Create,John Doe,MyAD|MySolaris,changeit,changeit,John Doe,John Doe  
Create,Jane Smith,MyAD,changeit,changeit,Jane Smith
```

次の例は、2つの個別のファイルで一括操作の作成と一括更新操作を使用する方法を示しています。

```
command,user,waveset.resources,password.password,password.confirmPassword,accounts[MyAD].description,  
Create,John Doe,MyAD,changeit,changeit,John Doe,John Doe  
Create,Jane Smith,MyAD,changeit,changeit,  
Jane Smith  
  
command,user,waveset.resources,password.password,password.confirmPassword,accounts[MySolaris].comment  
Update,John Doe,MySolaris,changeit,changeit,John Doe  
Update,Jane Smith,MySolaris,changeit,changeit,Jane Smith
```

一括操作を使用してアカウントを作成するとき、**Identity Manager** のアカウントインデックスにエントリは追加されません。したがって、**Identity Manager** の配備が完了する前に完全調整を実行する必要があります。

注 一括操作を使用してアカウントを作成するとき、**Identity Manager** のアカウントインデックスにエントリは追加されません。したがって、**Identity Manager** の配備が完了する前に完全調整を実行する必要があります。

調整

多くの場合、リソースの初回の調整には2回目以降の調整よりも長い時間がかかります。一般的に、リソースの初回の調整では多数のアカウントインデックスエントリが追加されます。

データ読み込みの準備を行う

Identity Manager へのアカウント情報の読み込みプロセスを開始する前に、次の各項の内容を確認してください。

- アダプタを設定する
- アカウント ID ポリシーとパスワードポリシーを設定する
- データ読み込み用のアカウントを作成する
- フォームを割り当てる

アダプタを設定する

リソース上のアカウントを管理するには、アカウント情報のソースごとにアダプタを設定する必要があります。「ファイルから読み込み」プロセスまたは一括操作を使用する場合、アダプタの設定は調整の準備ができた時点で行えば問題ありません。それ以外の場合は、Identity Manager にデータを読み込む前にアダプタを設定する必要があります。

アダプタの設定についての全般的な情報は、『Identity Manager 管理ガイド』を参照してください。個別のアダプタに関する詳しい情報は、『Identity Manager Resources Reference』またはオンラインヘルプを参照してください。

アカウント ID ポリシーとパスワードポリシーを設定する

「リソースから読み込み」、調整、または ActiveSync によってリソースからアカウントデータを読み込むとき、Identity Manager はリソースからパスワードを取得しません (リソースからパスワードが渡された場合、リソース上のセキュリティが侵害される可能性がある)。したがって、Identity Manager アカウントのパスワードは、リソース上のパスワードと同じにはなりません。デフォルトでは、Identity Manager によってランダムなパスワードが生成され、このパスワードを再設定する必要があります。ただし、ユーザーフォームのパスワード画面を使用して、全ユーザーに共通の、または Identity Manager アカウント ID と同一のリテラル文字列などを一時パスワードとして指定することもできます。詳細については、「ユーザーフォームを割り当てる」と、「Identity Manager の画面」の章を参照してください。

一括操作および「ファイルから読み込み」では、パスワード値を CSV ファイルで指定できます。これらのパスワードは、ユーザーが変更しなければならない一時パスワードとして考えるべきです。

ポリシーは Identity Manager アカウントに適用される制限を設定し、次のように分類されます。

- Identity Manager アカウントポリシー -- ユーザー、パスワード、および認証のポリシーオプションを設定するために使用します。Identity Manager のアカウントポリシーは組織またはユーザーに割り当てられます。
- リソースパスワードおよびアカウント ID ポリシー -- 長さ規則、文字タイプ規則、使用可能な単語および属性値を設定または選択するために使用します。

デフォルトポリシーを更新する場合は必ず、Identity Manager へのアカウント情報の読み込みを開始する前に行ってください。

次の表は、Identity Manager に用意されているポリシーとそのデフォルト設定の一覧です。

表 3-2 デフォルトの Identity Manager ポリシー

ポリシー名	デフォルト特性
アカウント ID ポリシー	アカウント ID の長さは 4 文字以上 16 文字以下である必要があります。
デフォルトの Lighthouse アカウントポリシー	アカウント ID ポリシーを「アカウント ID ポリシー」に、パスワードポリシーを「パスワードポリシー」に設定します。パスワードはユーザーではなく Identity Manager によって生成されます。
パスワードポリシー	パスワードの長さは 4 文字以上 16 文字以下である必要があります。パスワードにはユーザーの電子メールアドレス、姓、名、またはフルネームを含めることはできません。
Windows 2000 パスワードポリシー	パスワードの長さは 6 文字以上である必要があります。パスワードには次の文字タイプのうち 3 種類以上が含まれている必要があります。 • 数字 (1) • 大文字のアルファベット (1) • 小文字のアルファベット (1) • 特殊文字 (1) また、パスワードにはアカウント ID を含めることはできません。

アカウントポリシーとパスワードポリシーの詳細については、『Identity Manager 管理ガイド』の「Identity Manager ユーザー」の章を参照してください。

データ読み込み用のアカウントを作成する

次の理由により、データ読み込みを実行するための管理者アカウントを別途作成することをお勧めします。

- 監査を有効にしたときにデータ読み込み操作の追跡が容易になります。
- すべての Identity Manager 管理者に、Identity Manager ユーザーを作成および編集するユーザーフォームが割り当てられます。データ読み込み用のアカウントを作成すると、デフォルトのフォームよりも実行が高速な簡易形式のフォームを指定できます。詳細については、次の項を参照してください。

アカウントの作成の詳細については、『Identity Manager 管理ガイド』を参照してください。

ユーザーフォームを割り当てる

データ読み込みのコンテキストでは、ユーザーフォームはバックグラウンド処理を実行する目的で使用されます。たとえば、フォームとリソースアダプタと一緒に使用して、外部リソースから得た情報を処理し、Identity Manager リポジトリにこれを保存できます。ユーザーフォームは、入力ユーザーデータに基づいて正しい組織にユーザーを配置する目的にも使用できます。

ユーザービューは、Identity Manager ユーザーの利用可能な全情報が含まれるデータ構造です。このビューには、以下の項目が含まれます。

- Identity Manager リポジトリに格納される属性
- リソースアカウントからフェッチされる属性
- リソース、ロール、および組織など、ほかのソースから取得される情報

ビューには多くの属性があり、ビュー属性はビュー内にある名前を付けられた値です（たとえば、`waveset.accountId` は、Identity Manager アカウント名を値とする、ユーザービュー中の属性）。

ほとんどのフォームフィールドの名前は、ビュー属性と関連付けられています。フォームフィールドの名前として、ビュー属性の名前を指定することで、フィールドとビュー属性を関連付けます。ユーザービューの全属性の参照情報など、ユーザービューの詳細については、「Views」の章を参照してください。

ユーザーを読み込むユーザーフォームには、多くの場合次のフィールドが含まれます。

- `firstname` (名)
- `lastname` (姓)
- `fullname` (フルネーム)

- email (電子メール)
- waveset.accountId
- waveset.organization
- EmployeeId (従業員 ID)

waveset.accountId および waveset.organization は、Identity Manager に固有の値です。EmployeeId 属性はカスタマイズした属性です。その用途については、「カスタム関連キーを定義する」で例示しています。

Identity Manager には、システムにインストール済みの多数のフォームが用意されています。また、追加のフォームが \$WSHOME/sample/forms ディレクトリに収められています。このディレクトリ内のフォームの多くは、リソース固有の内容になっています。Identity Manager IDE を使用してこれらのフォームを確認し、本稼働環境で使用するかどうかを判断することができます。

一括操作中のパフォーマンスを向上させるために、管理者に割り当てられるユーザーフォームはできるだけ簡易なものにしてください。データ読み込み用のフォームを作成する場合、データ表示のために記述されたコードは削除できます。フォーム簡易化の別の例は、一括追加操作を使用する場合です。CSV ファイルでは、firstname や lastname などの基本属性を定義できます。これらの属性は、あとで管理者のユーザーフォームから削除できます。フォームの作成と編集の詳細については、「Identity Manager のフォーム」の章を参照してください。

注	Identity Manager に用意されているフォームを直接修正しないでください。代わりに、これらのフォームのコピーを作成して一意の名前を付け、このファイル名を変更したコピーフォームを編集するようにしてください。こうすることにより、カスタマイズしたコピーがアップグレードやサービスパックの更新の際に上書きされるのを防止できます。
---	--

ほかのリソースへアカウントをリンクする

Identity Manager では、関連規則と確認規則を使用してアカウントどうしをリンクします。関連規則は、アカウントを所有している可能性がある Identity Manager ユーザーを検索します。関連規則で定義された条件に一致するユーザーのリストが返されます。確認規則は、ユーザーが実際にアカウントを所有しているかどうかを確認するために、アカウントに対して Identity Manager ユーザーをテストします。true または false の値が返されます。所有者候補をすばやく検索したり (名前または属性に基づいて)、負担の大きなチェックの実行を所有者候補だけに制限したりするので、この 2 段階のアプローチによって、Identity Manager は関連を最適化することができます。

関連規則と確認規則の使用を開始する前に、最初のデータ読み込みの時点から含まれるデータを確認しておく必要があります。Identity Manager の accountId 属性は常に含まれます。「ファイルから読み込み」または一括操作の作成を実行した場合、CSV ファイルの先頭行に記述された属性値も含まれます。「リソースから読み込み」または調整を実行した場合、リソース上の一部のキー属性は含まれますが、ほかの属性は明示的に保存された場合にのみ含まれます。

また、二次リソース上に格納されたアカウントデータについても確認しておく必要があります。二次リソースには、Identity Manager にすでに存在するデータと重複するデータが含まれているのが理想的です。

これは意外に難しい条件である場合があります。多くの場合、異なるリソース間ではユーザーアカウントについての要件も異なります。例として、次の表では Windows のアカウント名と Solaris のアカウント名について要件および制限を比較しています。

表 3-3 Windows アカウント名と Solaris アカウント名の要件の比較

特性	Windows	Solaris
最大長	20 文字	8 文字
使用できる特殊文字	"/ ¥ [] ; = , + * ? を除く すべて < >	ピリオド (.)、下線 (_)、ハイフン (-) のみ
フルネームを指定可能	はい	コメントフィールドが 1 つあり。このコメントフィールドには伝統的にユーザーのフルネームを記述することになっていますが、その他の情報が含まれている可能性があります。
従業員 ID などの追加コメントを指定できるか	はい	

Windows と Solaris のアカウント名の違いから、アカウントのリンクに伴ういくつかの問題が明らかになります。

- 最大長の違いが理由で、これら 2 つのシステムではアカウント名が異なるのが一般的です。Windows では、ユーザーが自分の姓および名と一致するアカウント名を持つことがよくあります。Solaris では、名の先頭の文字と、名の先頭から 7 文字を組み合わせたものをアカウント名として使用する場合があります。したがって、アカウント ID を比較する関連規則は、Solaris アカウントを Windows アカウントとリンクする手段として確実ではありません。
- Windows でユーザーアカウントを作成するとき、管理者は表示名を指定する必要があります。Solaris のユーザーアカウントには表示名は不要ですが、オプションの GECOS フィールドを使用してユーザーの名前を指定できます。このフィールドの内容または形式についてのガイドラインはありません。ユーザーの GECOS フィールドは空にすることも、Windows の表示名とは関係のないテキストを入れることもできます。したがって、フルネームを比較する関連規則は、一般的に考えられているほど多くは利用されません。

アカウントをリンクするための準備をするときは、次の質問の答えを検討してください。

- Mary A. Jones と Mary B. Jones、または John Doe Jr. と John Doe III などの、よく似た名前を持つユーザーがいますか。答えが「はい」の場合、それらのユーザーをどのようにして区別しますか。
- 各リソース上でアカウント名を定義する方法は共通化されていましたか。答えが「はい」の場合、Identity Manager に格納される値を使用してリソース上のアカウント ID を比較する規則を容易に定義できます。
- リソースのアカウントデータを編集する機能がユーザーに許可されていましたか。答えが「はい」の場合、そのデータは、ユーザーが変更できないシステム上に格納される値と一致しない可能性があります。

カスタム関連キーを定義する

規則では、値がシステムに格納されないかぎり、リソース上のアカウントの値を Identity Manager の値と比較することはできません。accounts[Lighthouse] 属性にはこれらの値の多くが格納されますが、それら以外の値は、拡張ユーザー属性設定オブジェクトを使用して追加する必要があります。設定オブジェクトに登録されていない属性はシステムに保存されません。

デフォルトでは、次の属性が拡張ユーザー属性として含まれています。

- firstname (名)
- lastname (姓)
- fullname (フルネーム)

注	拡張ユーザー属性 <code>fullname</code> は、 <code>QueryableAttrNames</code> のリストに追加する必要があります。
---	---

これまでに述べた以外の、従業員 ID などの属性を相関規則の一部として使用したい場合、その属性を拡張ユーザー属性設定オブジェクトに追加する必要があります。追加は次の手順で行います。

1. Identity Manager のデバッグページ (<http://PathToIDM/debug>) にアクセスします。「System Settings」ページが表示されます。
2. 「List Objects」プルダウンメニューから「設定」を選択します。「List Objects of type: Configuration」ページが表示されます。
3. 「User Extended Attributes」の「edit」リンクを選択します。
4. 次の例のように、List 要素に新しい属性を追加します。
`<String>EmployeeId</String>`
5. リソースの「アカウント属性 (スキーママップ)」ページで、属性を Identity Manager 属性として定義する必要があります。
6. 変更を保存します。Identity Manager の「System Settings」デバッグページに戻ります。

カスタム属性は、UserUIConfig 設定オブジェクトの `QueryableAttrNames` 要素にも追加する必要があります。

1. 「List Objects」プルダウンメニューから「設定」を選択します。「List Objects of type: Configuration」ページが表示されます。
2. 「UserUIConfig」の「edit」リンクを選択します。
3. 次の例のように、`<QueryableAttrNames><List>` 要素に新しい属性を追加します。
`<String>EmployeeId</String>`
4. 変更を保存します。Identity Manager の「System Settings」デバッグページに戻ります。
5. アプリケーションサーバーを再起動します。

カスタム規則を作成する

Identity Manager は、いくつかの関連規則と確認規則を `sample/reconRules.xml` 内で事前に定義しています。これらは、独自の規則を作成するためのベースとして使用できます。規則には `SUBTYPE_ACCOUNT_CORRELATION_RULE` または `SUBTYPE_ACCOUNT_CONFIRMATION_RULE` のサブタイプを割り当てる必要があります。

次の規則は、二次リソースで定義された属性である `account.EmployeeId` を、以前に Identity Manager に読み込まれた属性である `EmployeeId` と比較します。二次リソースに `account.EmployeeId` の値が存在する場合、`EmployeeId` と一致するユーザーのリストが関連規則によって返されます。

```
<Rule subtype='SUBTYPE_ACCOUNT_CORRELATION_RULE' name='Correlate
Employee IDs'
  <cond>
    <ref>account.EmployeeId</ref>
    <list>
      <new class='com.waveset.object.AttributeCondition'>
        <s>EmployeeId</s>
        <s>equals</s>
        <ref>account.EmployeeId</ref>
      </new>
    </list>
  </cond>
</Rule>
```

この例では、`EmployeeId` 属性はあらかじめユーザー拡張属性設定オブジェクトおよび `UserUIConfig` 設定オブジェクトに追加されています。リソースの側でデフォルトの Identity Manager 属性名にこの属性が含まれていなかった場合、リソースのスキーママップでこの属性を追加または編集する必要もあります。

関連規則は見つかった一致のリストを返します。従業員 ID のように、結果で 1 つの一致しか返されないことが予想される場合、確認規則は必要ありません。一方で、複数の一致が考えられる場合 (たとえば、姓名が一致する複数のアカウントが関連規則で検出された場合など) は、一致をさらに検証するための確認規則が必要です。

規則を Identity Manager に追加する方法には、Identity Manager IDE を使用する方法、XML ファイルをインポートする方法、デバッグページで既存の規則を編集してその名前を変更する方法があります。

手動でアカウントをリンクする

Identity Manager には、関連規則および確認規則で一致が見つからないときにアカウントを割り当てるために使用できる複数の機構が用意されています。

アカウントインデックスを使用する

アカウントインデックスには、Identity Manager が認識している各リソースアカウントの最新の状態が記録されています。このインデックスは主に調整によって管理されますが、Identity Manager のほかの機能でも必要に応じてアカウントインデックスの更新が行われます。

注 「リソースから読み込み」、「ファイルから読み込み」、および一括操作では、アカウントインデックスの更新は行われません。

アカウントインデックスを確認するには、「リソース」タブをクリックし、左側の「アカウントインデックス」リンクをクリックします。リソースへ移動すると、対象のリソース上のすべてのアカウントの状態が表示されます。

関連のないアカウント（「アカウントインデックス」テーブルで、状況が「UNMATCHED」、所有者が「_UNKNOWN_」として表示される）を右クリックすると、Identity Manager のメニューが表示されます。このメニューから、新しい Identity Manager ユーザーアカウントの作成、リソースに対して有効な調整ポリシーを使用した単一アカウントに対する調整の実行、所有者の指定、リソースアカウントの削除または無効化などの処理を実行できます。「所有者の指定」オプションを選択すると、Identity Manager は、指定された条件に一致する所有者を検索するための画面を表示します。詳細については、『Identity Manager 管理ガイド』を参照してください。

自己検索を有効にする

Identity Manager のユーザーが自分自身のリソースアカウントの検索を行えるように、Identity Manager のユーザーインターフェースを設定することができます。つまり、Identity Manager の ID を持つユーザーはこの機能によって、既存のまだ関連付けられていないリソースアカウントに自分の ID を関連付けることができます。自己検索を有効にすることができるリソースは、パススルー認証をサポートするリソースに限られます。

自己検索を有効にするには、エンドユーザーリソース設定オブジェクトを編集して、ユーザーによるアカウント検索を許可するリソースの名前をそれぞれこのオブジェクトに追加する必要があります。次の手順で実行します。

1. Identity Manager のデバッグページ (<http://PathToIDM/debug>) にアクセスします。「System Settings」ページが表示されます。

2. 「List Objects」プルダウンメニューから「設定」を選択します。「List Objects of type: Configuration」ページが表示されます。
3. 「End User Resources」の「edit」リンクを選択します。
4. <List> 要素のあとに、<String>Resource</String>を追加します。ここで、Resource はリポジトリ内のリソースオブジェクトの名前です。たとえば、ユーザーがリソース NT および Solaris 上のアカウントを自己検索できるようにするには、<List> 要素を次のように編集します。

```
<List>

  <String>NT</String>

  <String>Solaris</String>

</List>
```

5. 変更を保存します。Identity Manager の「System Settings」デバッグページに戻ります。

自己検索を有効にすると、Identity Manager のユーザーインターフェースに新しいメニュー項目が追加されます（「他のアカウントについて Identity Manager に知らせる」）。ユーザーはこの領域で、利用可能なリストからリソースを選択し、リソースアカウント ID とパスワードを入力して、アカウントを自分の Identity Manager ID とリンクすることができます。

シナリオの例

この項では、1 つまたは複数のリソースからアカウントを読み込むプロセスを例示するシナリオを提供します。次のシナリオでは、さまざまな環境で発生する可能性がある問題について検討します。

- Active Directory、SecurID、および Solaris
- LDAP、PeopleSoft、および Remedy
- 高速一括追加

Active Directory、SecurID、および Solaris

ある会社で、Identity Manager を使用して Active Directory、SecurID、および Solaris のアカウントを管理することを検討しています。すべての従業員が Active Directory アカウントを、ほとんどの従業員が SecurID アカウントを持っています。ごく一部の従業員が Solaris アカウントを持っています。各リソース上のアカウントデータを検証したあとで、Identity Manager 管理者は、関連キーとして次の属性を使用できると判断しました。

表 3-4 関連キーの候補

関連キーの候補	Active Directory	SecurID	Solaris
アカウント ID が AD と一致	N/A	使用可	使用不可
従業員 ID	使用可	使用不可	使用不可
フルネーム	使用可	使用可	使用可 (Description 属性)

すべての従業員が Active Directory アカウントを持っているため、最初のデータ読み込みリソースには Active Directory を使用します。SecurID リソース上のアカウント ID は Active Directory 上のアカウント ID と一致するため、SecurID リソースを 2 番目に読み込みます。アカウント ID は常に一意であることから、フルネームよりも関連キーとして適しています。Active Directory アカウントおよび SecurID アカウントの間には、問題なく関連が成立すると考えられます。

Solaris アカウントについては、関連の成立は難しくなります。Solaris アカウントに存在する属性のうち、唯一の関連属性はユーザーのフルネームです。Solaris アカウントには、姓と名を定義するための独立した属性はありません。結果として、関連規則では、Solaris の `useradd -c` コマンドで定義される文字列と、Active Directory の `fullname` の値を比較することになります。この比較は、ニックネームの使用や不要なスペースまたは句読文字が原因で、よく失敗します。

ユーザーの例

このシナリオでは、次のユーザーが存在する環境で、アカウントの読み込み時に発生する可能性があるいくつかの問題の例を示します。

表 3-5 データ読み込みのシナリオ: アカウントの読み込み時に発生する可能性のある問題

従業員名	AD および SecurID ロ グオン名	AD フルネーム	Solaris アカウント名	Solaris の Description
Anthony Harris	AJ Harris	Anthony J Harris	ajharris	A.J. Harris

表 3-5 データ読み込みのシナリオ: アカウントの読み込み時に発生する可能性のある問題 (続き)

従業員名	AD および SecurID ロ グオン名	AD フルネーム	Solaris アカウント名	Solaris の Description
Isabelle Moreno	Isabelle Moreno	Isabelle Moreno	imoreno	Isabelle Moreno
John Thomas (Sr.)	John Thomas	John Thomas	jthomas	John Thomas
John Thomas (Jr.)	John P. Thomas	John P. Thomas	jthomas2	John Thomas
Robert Blinn	Robert Blinn	Bob Blinn	rblinn	Bob Blinn
Theodore Benjamin	Theodore Benjamin	Theodore Benjamin	tbenjami	Ted Benjamin

Active Directory アカウントを読み込む

調整を使用して Active Directory アカウントを Identity Manager に読み込むためのガイドラインとして、次の手順を使用します。

1. 管理者インタフェースの「リソース」ページで、「新規リソース」プルダウンメニューから「Windows 2000 / Active Directory」リソースを選択します。次に、アダプタを設定します。

Identity Manager のユーザー属性 accountId または fullname をスキーママップから削除しないように注意してください。また、アイデンティティテンプレートが正しいことを確認してください。アダプタの設定の詳細については、オンラインヘルプおよび『Identity Manager Resources Reference』を参照してください。

2. (オプション) アカウントポリシーとパスワードポリシーを必要に応じて変更します。詳細については、「アカウント ID ポリシーとパスワードポリシーを設定する」を参照してください。
3. (オプション) 調整に使用するユーザーフォームを作成します。詳細については、「ユーザーフォームを割り当てる」を参照してください。
4. (オプション) データ読み込みを実行するための Identity Manager ユーザーを作成します。前の手順で作成したユーザーフォームをユーザーに割り当てます。
5. リソースの調整ポリシーを設定します。最初のリソースについては、相関規則は重要ではなく、確認規則は Identity Manager ユーザーの作成時には使用されません。これは最初のリソースであるため、値「リソースアカウントに基づく新規 Identity Manager ユーザーの作成」に UNMATCHED 状況を割り当てるのが適切と考えられます。
6. データ読み込みを実行するためのユーザーを作成したら、そのユーザーとしてログインします。調整の場合はこの手順は不要ですが、「ファイルから読み込み」、「リソースから読み込み」、または一括操作の場合は必要です。
7. Active Directory リソースを調整します。

結果

デフォルトの Identity Manager アカウントポリシーとデフォルトの Active Directory アイデンティティテンプレートを使用した場合、Theodore Benjamin の名前が 16 文字を超えているため、Identity Manager はこのユーザーの Active Directory アカウントにリンクする Identity Manager ユーザーを作成しません。これを回避するために、この例では、アカウント ID ポリシーを 25 文字に設定しました。

Identity Manager は、状況が CONFIRMED 状態であるすべてのリソースアカウントに対してユーザーアカウントを作成します。パスワードポリシーとアカウント ID ポリシーをパスしたすべてのユーザーがこれに該当します。ユーザーフォームでほかの設定を指定した場合を除き、Identity Manager のアカウント名は Active Directory のログイン名と同じになります。

SecurID アカウントを読み込む

SecurID を実装するとき、SecurID のユーザーレコードは通常、Microsoft セキュリティアカウントマネージャー (SAM) データベースまたは LDAP サーバーからインポートされます。その結果、SecurID のアカウント ID はインポートソースのアカウント ID と一致します。SecurID アカウントと Active Directory アカウントの間には 1 対 1 の相関が成立するため、ユーザー間の相関確立は比較的容易になります。これらのアカウントを迅速にリンクする目的で、「アカウント ID と一致するユーザー名」の相関規則を使用できます。

SecurID アカウントを読み込むには、「[Active Directory アカウントを読み込む](#)」で説明した手順を、次のように変更して実行します。

- SecurID アダプタを設定しているときは、Identity Manager のユーザー属性である `accountId` を削除しないように注意してください。
- 調整ポリシーを次のように設定します。
 - 相関規則を「アカウント ID と一致するユーザー名」に設定します。
 - Active Directory はアイデンティティ情報の源泉として信頼性の高いリソースと考えられ、SecurID は Active Directory のアカウント情報に依存するため、UNMATCHED 状況時の動作オプションを「リソースアカウントの削除」または「リソースアカウントの無効化」に設定することをお勧めします。UNASSIGNED 状況は「Identity Manager ユーザーへリソースアカウントをリンク」に設定します。

結果

すべての SecurID アカウントについて、Active Directory アカウントとの相関が確立されます。必要に応じて、UNMATCHED または DISPUTED 状況を解決するための手順を追加します。

Solaris アカウントを読み込む

このシナリオでは、fullname 属性が唯一の相関キーです。スペースや句読文字などの違いにより確実に一致が失敗するため、これは弱い相関キーです。また、ユーザーは自分の表示名を Solaris の chfn コマンドで変更できます。フルネームが当初一致していたとしても、ユーザーが chfn コマンドを実行したことが原因で相関を確立できない可能性があります。

デフォルトでは、fullname 属性はクエリー可能ではありません。クエリー可能にするには、UserUIConfig 設定オブジェクトを編集し、fullname 属性を <QueryableAttrNames><List> 要素に追加する必要があります。詳細については、「カスタム相関キーを定義する」を参照してください。

fullname 属性どうしを相関させるためのカスタム規則を作成する必要もあります。この例では、「Correlate Full Names」という名前の規則を作成して相関処理を実行させます。この規則では、Solaris リソースの account.Description 属性の値を、Active Directory から読み込まれたシステム属性である fullname 属性と比較します。

```
<Rule subtype='SUBTYPE_ACCOUNT_CORRELATION_RULE' name='Correlate Full
Names'
  <cond>
    <ref>account.Description</ref>
    <list>
      <new class='com.waveset.object.AttributeCondition'>
        <s>fullname</s>
        <s>equals</s>
        <ref>account.Description</ref>
      </new>
    </list>
  </cond>
</Rule>
```

この規則では、Solaris リソースからの Description 属性を、Identity Manager の fullname 属性と比較します。2つの属性が一致する場合、アカウント間に相関が成立し、状況が CONFIRMED に変化します。

Solaris アカウントを読み込むには、「[Active Directory アカウントを読み込む](#)」で説明した手順を、次のように変更して実行します。

- Solaris アダプタを設定しているときは、Identity Manager のユーザー属性である accountId または Description を削除しないように注意してください。
- 調整ポリシーを次のように設定します。

- 関連規則を「Correlate Full Names」(サンプル規則) に設定します。
- すでに Identity Manager に読み込まれているアカウントと関連のない Solaris アカウントが数多く存在する可能性があります。UNASSIGNED 状況時の動作オプションを「Identity Manager ユーザーへリソースアカウントをリンク」に設定します。UNMATCHED 状況はほとんどの場合、「何もしない」に設定するのが適切です。一致しないユーザーを削除または無効化すると、データの消失や生産性の低下を招く可能性があります。

結果

表 3-6 データ読み込みのシナリオのユーザー例

従業員名	AD フルネーム	Solaris アカウント名	Solaris の Description
Anthony Harris	Anthony J Harris	ajharris	A.J. Harris
Isabelle Moreno	Isabelle Moreno	imoreno	Isabelle Moreno
John Thomas (Sr.)	John Thomas	jthoma	John Thomas
John Thomas (Jr.)	John P. Thomas	jthomas2	John Thomas
Robert Blinn	Bob Blinn	rblinn	Bob Blinn
Theodore Benjamin	Theodore Benjamin	tbenjami	Ted Benjamin

この例では、Isabelle Moreno のアカウントのみに相関が成立することが予測されます。

- Active Directory の fullname 属性は Solaris の Description 属性と完全には一致しないため、Anthony Harris、John Thomas (Jr.)、Robert Blinn、および Theodore Benjamin のアカウントについては相関が成立しません。これらのアカウントの状況は UNMATCHED となります。このシナリオでは、Solaris のアカウント名は名のイニシャルと姓の組み合わせで構成されます。John Thomas のアカウントの例外を除き、これら不一致の Solaris アカウントの割り当ては簡単です。
- Solaris アカウント jthomas および jthomas2 は、DISPUTED の状況となります。これらのアカウントは Description の値がともに John Thomas です。Solaris アカウント jthomas および jthomas2 がどのユーザーに属するのかを決定する別の手段を見つける必要があります。確認規則を使用して 2 つのアカウントを区別できれば理想的です。しかしながら、Solaris アカウントと Active Directory アカウントには、確認規則を作成するための十分な共通属性が含まれていません。

LDAP、PeopleSoft、および Remedy

このシナリオでは、理論的に LDAP リソースまたは PeopleSoft リソースが一次リソースとなりえます。

- すべての従業員および契約社員が PeopleSoft で追跡される場合、このアプリケーションをアイデンティティ情報の源泉として信頼性の高いリソースとして考えることができます。
- すべての従業員が LDAP アカウントを持つ場合、LDAP をアイデンティティ情報の源泉として信頼性の高いリソースとして考えることができます。

Remedy アカウントを持つ従業員の割合はごく限られているため、Remedy は一次リソースの候補とはなりません。

複数のアイデンティティ情報の源泉として信頼性の高いリソースが存在する場合、それらのうちのどのリソースを最初に読み込んでもかまわないケースがほとんどです。ただし、PeopleSoft コンポーネントアダプタは ActiveSync 機能のみを実行します (別の PeopleSoft アダプタも利用できるが、範囲が限定されている)。PeopleSoft コンポーネントアダプタは調整を実行しないため、結果的に、リソースに対して調整ポリシーを設定できません。利用可能な関連規則がないため、PeopleSoft アカウントを最初に読み込む必要があります。Identity Manager アカウント名は、PeopleSoft の EMPLID (従業員 ID) の値と一致します。

PeopleSoft の従業員 ID は、システムで定義されるすべてのユーザーについて一意であるため、関連キーとして理想的です。LDAP の inetOrgPerson オブジェクトには employeeNumber 属性が含まれます。従業員 ID は、Description などのラベルが付いた属性、またはカスタム属性に格納することもできます。このシナリオでは、LDAP の employeeNumber 属性が使用されていると想定します。

Remedy アダプタにはデフォルト属性が用意されていません。環境に合わせてアダプタをカスタマイズする必要があります。Remedy アプリケーションは、システムがリクエストを受け取ったときに電子メールを送信するように設定されることが多いため、電子メール属性が利用可能であると想定されます。LDAP の inetOrgPerson オブジェクトにもメール属性が含まれます。したがって、電子メールアドレスが関連キーとなります。

次の表は、このシナリオにおける各リソースの関連キーの一覧です。

表 3-7 データ読み込みのシナリオ : 各リソースの関連キー

関連キーの候補	PeopleSoft	LDAP	Remedy
従業員 ID	使用可	使用可	使用不可
電子メールアドレス	使用不可	使用可	使用可

ユーザーの例

このシナリオでは、次のユーザーが存在する環境で、アカウントの読み込み時に発生する可能性があるいくつかの問題の例を示します。

表 3-8 配備シナリオ：アカウントの読み込み時に発生する可能性のある問題

従業員名	PeopleSoft EMPLI	LDAP 従業員番号	LDAP 電子メール (@example.com)	Remedy 電子メール (@example.com)
Robert Blinn	945	945	Bob.Blinn	bblinn
William Cady	なし	なし	William.Cady	William.Cady
Eric D'Angelo	1096	1096	Eric.D'Angelo	Eric.D'Angelo
Renée LeBec	891	なし	なし	なし
Josie Smith	1436	1463	Josie.Smith	なし
John Thomas	509	509	John.Thomas	なし
John P. Thomas	なし	なし	John.P.Thomas	John.P.Thomas

PeopleSoft ユーザーを読み込む

ActiveSync を使用する PeopleSoft アカウントを、調整を使用して Identity Manager に読み込むためのガイドラインとして、次の手順を使用します。

1. Identity Manager 管理者インタフェースの「リソース」ページで、「新規リソース」プルダウンメニューから「PeopleSoft コンポーネント」リソースを選択します。このリソースが表示されない場合、「管理するリソースの設定」ボタンをクリックし、カスタムリソースとして `com.waveset.adapter.PeopleSoftComponentActiveSyncAdapter` を追加します。このアダプタは、PeopleSoft が提供する JAR ファイルのインストールを必要とします。詳細については、『Identity Manager Resources Reference』を参照してください。
2. アダプタを設定します。Identity Manager のユーザー属性 `accountId` または `fullname` をスキーママップから削除しないように注意してください。また、アイデンティティテンプレートが正しいことを確認してください。
3. (オプション) アカウントポリシーとパスワードポリシーを必要に応じて変更します。詳細については、「アカウント ID ポリシーとパスワードポリシーを設定する」を参照してください。
4. (オプション) データの読み込みに使用するユーザーフォームを作成します。`$WSHOME/sample/forms/PeopleSoftForm.xml` ファイルをベースとして使用できます。詳細については、「ユーザーフォームを割り当てる」を参照してください。

5. PeopleSoft アダプタ上で ActiveSync を開始します。

結果

ユーザーフォームでアカウントを読み込まないよう指示された場合を除き、Identity Manager はすべてのユーザーを読み込みます。このシナリオで、Renée LeBec は LDAP アカウントと Remedy アカウントを持っていません。この人物はすでに会社を退職していると考えられます。デフォルトの PeopleSoft フォームを使用した場合、Identity Manager によって、退職した従業員の PeopleSoft アカウントは無効にされます。

William Cady および John P. Thomas は PeopleSoft システムで定義されていないため、これらのユーザーのアカウントは作成されません。

LDAP ユーザーを読み込む

このシナリオでは、LDAP の inetOrgPerson オブジェクトの employeeNumber 属性が相関キーです。この属性は LDAP アダプタのスキーママップにデフォルトで入っていないため、手動で追加する必要があります。この例の場合、属性 EmployeeId をスキーママップの「Identity Manager ユーザー属性」の側に、employeeNumber を「リソースユーザー属性」の側に追加します。

注	PeopleSoft アダプタは、Identity Manager の属性名 EmployeeId をデフォルトで使用します。LDAP と PeopleSoft の間で一貫性を保つためにこの値を選択しましたが、これは必須ではありません。
---	--

電子メールアドレスは Remedy リソースの相関キーとなりますが、LDAP アカウントを読み込む前に設定しておく必要があります。inetOrgPerson オブジェクトには、Remedy アカウントを読み込むための相関キーとなる mail 属性が含まれます。mail 属性もスキーママップに追加する必要があります。email 属性をスキーママップの「Identity Manager ユーザー属性」の側に、mail 属性を「リソースユーザー属性」の側に追加します。email は事前に定義された Identity Manager 属性であるため、この属性を使用するほうが、ユーザー拡張属性設定オブジェクトまたは UserUIConfig 設定オブジェクトを編集して mail 属性を追加するよりも簡単です。

Identity Manager では、アカウント ID は User オブジェクトの resourceAccountIds 属性に格納されます。これは複数の値をとる属性であり、各値の形式は accountId@objectId です。LDAP の EmployeeId の値を PeopleSoft の accountId と比較する、次のような規則を作成できます。

コード例 3-1

LDAP の EmployeeId 値と PeopleSoft の accountId の比較

```

<Rule subtype='SUBTYPE_ACCOUNT_CORRELATION_RULE' name='Correlate EmployeeId with
  accountId'>
  <cond>
    <ref>account.EmployeeId</ref>
    <list>
      <new class='com.waveset.object.AttributeCondition'>
        <s>resourceAccountIds</s>
        <s>startsWith</s>
        <concat>
          <ref>account.EmployeeId</ref>
          <s>@</s>
        </concat>
      </new>
    </list>
  </cond>
</Rule>

```

注 このシナリオでは、accountId および email 属性はシステム上で常に利用可能であるため、ユーザー拡張属性設定オブジェクトまたは **UserUIConfig** 設定オブジェクトに属性を追加する必要はありません。

LDAP アカウントを読み込むには、次の手順を実行します。

1. 管理者インタフェースの「リソース」ページで、「新規リソース」プルダウンメニューから「LDAP」リソースを選択します。その後、アダプタを次のように設定します。
 - a. Identity Manager のユーザー属性 EmployeeId および email を追加します。
 - b. Identity Manager のユーザー属性である accountId をスキーママップから削除しないように注意してください。
 - c. アイデンティティテンプレートが正しいことを確認します。
アダプタの設定の詳細については、オンラインヘルプおよび『Identity Manager Resources Reference』を参照してください。
2. リソースの調整ポリシーを次のように設定します。
 - a. 関連規則を「Correlate EmployeeId with accountId」に設定します。

- b. 次の状況の値を設定します。

UNASSIGNED 状況時の動作オプションを「Identity Manager ユーザーへリソースアカウントをリンク」に設定します。

UNMATCHED 状況のオプションを適切な操作に設定します。ほかのリソース上に検出されたユーザーを追加できるかどうかについては、PeopleSoft 管理者への確認が必要な場合があります。「リソースアカウントに基づく新規 Identity Manager ユーザーの作成」オプションを選択する場合、Identity Manager ユーザーにはデフォルトで、LDAP の cn 属性に基づくアカウント名が付与されます。

3. LDAP リソースを調整します。

結果

このシナリオでは、William Cady、Josie Smith、および John P. Thomas のアカウントは UNMATCHED 状態になります。Josie Smith については、従業員 ID の値が PeopleSoft リソース上と LDAP リソース上で一致しません。従業員 ID は PeopleSoft によって生成されるため、LDAP リソース側の値が正しくありません。誤りを訂正し、もう一度調整を行ってください。

William Cady および John P. Thomas は PeopleSoft で定義されていません。LDAP アカウントの読み込み手順の手順 2 の記述に従って、アカウントを PeopleSoft に追加する必要があるかどうかを検討してください。

Remedy ユーザーを読み込む

Remedy アダプタには事前に定義されたアカウント属性はありません。これらの属性をスキーママップに追加する必要があります。Remedy では、追跡する各属性を一意に識別するために整数を使用します。たとえば、Remedy のアカウント ID には 1002000100 のような値が割り当てられる場合があります。これらの Remedy 属性番号を、スキーママップのリソースユーザー属性に追加する必要があります。最低限、次の Identity Manager ユーザー属性を追加する必要があります。

- accountId
- email (関連キー)

USER_EMAIL_MATCHES_ACCOUNT_EMAIL_CORR 相関規則は、Remedy アカウントを Identity Manager アカウントにリンクします。

Remedy アカウントを読み込むには、次の手順を実行します。

1. Identity Manager 管理者インタフェースの「リソース」ページで、「新規リソース」プルダウンメニューから「Remedy」リソースを選択します。その後、アダプタを次のように設定します。

最低限、Identity Manager のユーザー属性 `accountId` および `email` を追加します。ほかの属性を追加することもできます。

アダプタの設定の詳細については、オンラインヘルプおよび『Identity Manager Resources Reference』を参照してください。

2. リソースの調整ポリシーを次のように設定します。
 - a. 関連規則を `USER_EMAIL_MATCHES_ACCOUNT_EMAIL_CORR` に設定します。
 - b. 次の状況の値を設定します。
 - I. `UNASSIGNED` 状況時の動作オプションを「Identity Manager ユーザーへリソースアカウントをリンク」に設定します。
 - II. `UNMATCHED` 状況を適切な操作に設定します。
3. Remedy リソースを調整します。

結果

William Cady、Eric D'Angelo、および John P. Thomas の Remedy アカウントについては、LDAP と Remedy で定義されている電子メールアドレスが一致したため、正常に相関が確立されました。Robert Blinn の Remedy アカウントについては、相関を確立できませんでした。Remedy 上の電子メールアドレスが別名で設定されていたことが原因です。このシナリオのほかのユーザーは Remedy アカウントを持っていません。

高速一括追加のシナリオ

次の手順は、「作成」操作を使用して、数名のユーザーを Identity Manager に迅速に追加する方法を示します。CSV ファイルをシステムに読み込む前に、CSV ファイルで参照されるすべてのリソースが Identity Manager 内で定義されている必要があります。

1. 一括操作の作成を実行するために必要な情報を格納した CSV ファイルを生成します。CSV ファイルの形式の詳細については、「一括操作の作成」を参照してください。
2. CSV ファイルにパスワードが含まれていない場合、デフォルトのパスワードポリシーオプションを設定し、パスワードがシステムによって生成されるようにします。これを行うには、「設定」タブをクリックし、左側の「ポリシー」をクリックします。「Default Lighthouse Account Policy」リンクをクリックします。次に、「パスワード提供者」ドロップダウンメニューから「生成」オプションを選択します。
3. デフォルトの「ユーザー作成」プロビジョニングタスクに基づいて、新しいプロビジョニングタスクを作成します。

XML 画面で、TaskDefinition タグの resultLimit パラメータの値を 0 に編集します。この値は、タスクの結果を保持する秒数を指定します。

次に、タスクから次のセクションを削除します。

```
<Activity id='1' name='Approve'>
```

```
...
```

```
</Activity>
```

```
<Activity id='3' name='Notify'>
```

```
...
```

```
</Activity>
```

注 プロビジョナを呼び出すアクティビティは残す必要があります。残っていないと、ユーザーが正しく作成されません。

タスクの名前を「Fast Create User」のような値に忘れずに変更してください。

4. デフォルトのユーザーフォームに基づく新しいフォームを作成します。

XML 画面で、<Form>... </Form> 構造全体を次のように置き換えます。

```
<Form help='account/modify-help.xml'>
  <Field name="viewOptions.Process">
    <Expansion>
      <s>Fast Create User</s>
    </Expansion>
  </Field>
</Form>
```

「Fast User Form」のようにユーザーフォームの名前を忘れずに変更してください。

5. アカウントを Identity Manager に読み込むために使用する Identity Manager アカウントを作成します。手順 4 で作成したユーザーフォームをこのユーザーに割り当てます。
6. 前の手順で作成したアカウントを使用して、Identity Manager にログインします。
7. 一括操作の作成を実行します。

ユーザーアクションを設定する

この章では、Identity Manager の管理者インタフェースにカスタムタスクを追加する方法と、次の 2 つのインタフェース領域から実行できるユーザーアクションの設定方法について説明します。

- 「ユーザーアカウントの検索結果」 ページ
- 「アカウント」 ページ上の「ユーザー」 アプレット

カスタムタスクを追加する

カスタムタスクを追加するには、次の一般的な手順に従います。

- タスクへのアクセス権限の設定
- タスクのリポジトリへの追加

カスタムタスクへのアクセス権を設定する

通常は、タスクへのアクセスを特定の管理者の集合に制限するように、カスタムタスクへのアクセス権限を設定します。アクセス権限を設定するには、次の手順を実行します。

1. 新しい AuthType をタスクのリポジトリに追加します。
2. タスクの新しい AdminGroup (機能) を作成します。
3. 新しい機能を 1 名以上の管理者に許可します。

手順 1 : AuthType を作成する

新しい AuthType は、既存の TaskDefinition、TaskInstance、および TaskTemplate の各 AuthType を拡張して作成してください。AuthType を追加するには、リポジトリの AuthType 設定オブジェクトを編集して、タスクの新しい AuthType 要素を追加します。

<AuthType> 要素を使用して、新しい AuthType を作成します。この要素には、1 つの必須プロパティ name があります。次の例は、<AuthType> 要素の正しい構文を示します。

AuthType の作成後、リポジトリの AuthType 設定オブジェクトを編集して、新しい <AuthType> 要素を追加する必要があります。

次の例は、複数のユーザーを新しい組織に移動するカスタムタスクの追加方法を示します。

コード例 4-1 新しい組織への複数ユーザーの移動

```
<Configuration name='AuthorizationTypes'>
  <Extension>
    <AuthTypes>
      <AuthType name='Move User'
extends='TaskDefinition,TaskInstance,TaskTemplate' />
    </AuthTypes>
  </Extension>
</Configuration>
```

手順 2 : AdminGroup を作成する

次に、新しく作成した AuthType に Right.VIEW を許可する AdminGroup を作成します。これを行うには、新しい管理者グループを定義した XML ファイルを作成し、このファイルを Identity Manager リポジトリにインポートする必要があります。

コード例 4-2 AdminGroup の作成

```
<?xml version='1.0' encoding='UTF-8'?>
  <!DOCTYPE Waveset PUBLIC 'waveset.dtd' 'waveset.dtd'>
  <Waveset>
    <AdminGroup name='Move User'
      protected='true'
      displayName='UI_ADMINGROUP_MOVE_USER'
      description='UI_ADMINGROUP_MOVE_USER_DESCRIPTION'>
      <Permissions>
        <Permission type='Move User' rights='View' />
      </Permissions>
      <MemberObjectGroups>
        <ObjectRef type='ObjectGroup' id='#ID#All' name='All' />
      </MemberObjectGroups>
    </AdminGroup>
  </Waveset>
```

注 `displayName` 属性と `description` 属性はメッセージカタログキーです。メッセージカタログに見つからない場合、これらの属性は属性に見つかった時点で表示されます。メッセージカタログキーを使用する場合、`WPMessages.properties` またはカスタムメッセージカタログのどちらかにメッセージを追加する必要があります。

手順 3 : 管理者に機能を許可する

最後に、新しく定義したタスクを実行するためのアクセス権限を管理者に許可する必要があります。これは、次の 2 つの方法のいずれかで行うことができます。

- 新しい機能を直接割り当てる
- 新しい機能を管理者ロールに (直接、または機能規則を使用して) 追加し、そのロールを割り当てる

タスクをリポジトリに追加する

タスクへのアクセス権限を設定したあとで、タスクをリポジトリに追加することができます。タスクは、Identity Manager IDE を使用しての定義または XML 形式でのインポートが可能な標準の **TaskDefinition** です。たとえば、複数のユーザーの組織を変更するタスクは次の例のようになります (この例は `samples` ディレクトリにある)。

コード例 4-3 複数ユーザーの組織の変更

```
<?xml version='1.0' encoding='UTF-8'?>
  <!DOCTYPE TaskDefinition PUBLIC 'waveset.dtd' 'waveset.dtd'>
  <!-- MemberObjectGroups="#ID#Top" authType="Move User" name="Change Organizations"
taskType="Workflow" visibility="runschedulable"-->
    <TaskDefinition authType='MoveUser'
      name='Change Organizations' taskType='Workflow'
      executor='com.waveset.workflow.WorkflowExecutor'
      suspendable='true'
      syncControlAllowed='true' execMode='sync'
      execLimit='0' resultLimit='0'
        resultOption='delete' visibility='runschedulable'
        progressInterval='0'>
      <Form name='Change Organization Form'
        title='Change Organization Form'>
        <Display class='EditForm' />
        <Include>
          <ObjectRef type='UserForm' name='User Library' />
          <ObjectRef type='UserForm' name='Organization Library' />
        </Include>
        FieldRef name='namesList' />
        <FieldRef name='orgsList' />
        <FieldRef name='waveset.organization' />
      </Form>
      <Extension>
        <WFProcess name='Change Organizations' title='Change Organizations'>
          <Variable name='waveset.organization' />
          <Variable name='userObjectIds' input='true'>
            <Comments>The names of the accounts to change the organization on.</Comments>
          </Variable>
```

コード例 4-3 複数ユーザーの組織の変更 (続き)

```

<?xml version='1.0' encoding='UTF-8'?>
  Activity id='0' name='start'>
    <ReportTitle>
      <s>start</s>
    </ReportTitle>
    <Transition to='Process Org Moves' />
  </Activity>
  <Activity id='1' name='Process Org Moves'>
    <Action id='0' process='Move User'>
      <Iterate for='currentAccount' in='userObjectIds' />
      <Argument name='userId' value='${currentAccount}' />
      <Argument name='organizationId'
        value='${waveset.organization}' />
    </Action>
    <Transition to='end' />
  </Activity>

  <Activity id='2' name='end' />
</WFProcess>
</Extension>
<MemberObjectGroups>
  <ObjectRef type='ObjectGroup' id='#ID#Top' name='Top' />
</MemberObjectGroups>
</TaskDefinition>

```

例について

前掲の例については、次の点に注意してください。

- タスクの `authType` 属性は「**Move User**」に設定されます。これにより、このタスクにアクセスできるのは、この **AuthType** を実行する機能を割り当てられたユーザーに限定されます。
- フォームには `namesList` および `orgsList` への **FieldRef** (フィールド参照) が含まれます。これらのフィールドは、それぞれユーザーライブラリと組織ライブラリで定義されます。これらのフィールドを含めると、すべての選択対象ユーザーおよび選択対象組織の名前のリストが表示されます。危険を伴う可能性のあるタスクについては、タスクを実行することの潜在的な影響をユーザーが認識できるように、これらのフィールドのどちらかまたは両方を含めるようにしてください。

- タスクには `userObjectIds` という名前の入力変数があります。この変数の内容は、「ユーザーアカウントの検索結果」ページ、または「アカウント」ページのユーザーアプレットで選択されたユーザーの名前または ID のリストです。すべての選択したユーザーに対して目的のアクションを実行するには、この変数を反復処理します。

次の表は、タスクへの入力として使用できる変数の一覧です。

表 4-1 タスク変数	
変数	説明
<code>userObjectIds</code>	選択されたユーザーの ID のリスト。「ユーザーアカウントの検索結果」ページおよび「アカウント」ページから利用できます。「ユーザーアカウントの検索結果」ページから呼び出された場合、このリストの内容は選択されたユーザーの名前になります。
<code>userNames</code>	選択されたユーザーの名前のリスト。「ユーザーアカウントの検索結果」ページおよび「アカウント」ページから利用できます。
<code>orgObjectIds</code>	選択された組織の ID のリスト。「アカウント」ページからのみ利用できます。
<code>orgNames</code>	選択された組織の名前のリスト。「アカウント」ページからのみ利用できます。

このワークフローを有効にするには、次の例に示すように、ユーザーの組織を変更するためのサブプロセスをリポジトリに追加する必要があります。

コード例 4-4 ユーザーの組織の変更

```
<?xml version='1.0' encoding='UTF-8'?>
  <!DOCTYPE Configuration PUBLIC 'waveset.dtd' 'waveset.dtd'>
  <!-- MemberObjectGroups="#ID#Top" configType="WFProcess" name="Move
User"-->
  <Configuration name='Move User' createDate='1083353996807'>
    <Extension>
      <WFProcess name='Move User' title='Move User'>
        <Variable name='userId' input='true'>
          <Comments> 移動するユーザーのアカウント ID。 </Comments>
        </Variable>
        <Variable name='organizationId' input='true'>
          <Comments> ユーザーの移動先組織の ID。 </Comments>
        </Variable>
      </WFProcess>
    </Extension>
  </Configuration>
</-->
```

コード例 4-4 ユーザーの組織の変更 (続き)

```

<Activity id='0' name='Start'>
  <Transition to='Update Organization' />
</Activity>

<Activity id='1' name='Update Organization'>
  <Action id='0' process='Update User View'>
    <Argument name='accountId' value='${userId}' />
    <Argument name='updates'>
      <map>
        <s>waveset.organization</s>
        <ref>organizationId</ref>
      </map>
    </Argument>
  </Action>
  <Transition to='End' />
</Activity>

<Activity id='2' name='End' />
</WFProcess>
</Extension>
<MemberObjectGroups>
  <ObjectRef type='ObjectGroup' id='#ID#Top' name='Top' />
</MemberObjectGroups>
</Configuration>

```

ユーザーアクションを設定する

カスタムアクションを開始するボタンおよびアクションメニュー選択の定義を設定する必要があります。「ユーザーアカウントの検索結果」ページと「アカウント」ページに表示されるボタンおよびアクションメニュー項目の定義は、**User Actions Configuration** 設定オブジェクトに格納されます。

User Actions Configuration オブジェクトを直接編集することはできるかぎり避けてください。その代わりに、ユーザーアクションを設定するために推奨される方法は次のとおりです。

- **User Actions Configuration** 設定オブジェクトを新しい設定オブジェクトにコピーします。
- 新しい設定オブジェクトを指すように **System Configuration** オブジェクトを修正します。

ユーザーアクションの設定

一般に、ユーザーアクションの設定は次の手順で行うようにしてください。

- User Actions Configuration 設定オブジェクトを新しい XML ファイルにコピーします。
- 新しいオブジェクトの名前を「My User Actions Configuration」に変更します。
- My User Actions Configuration オブジェクトに必要な変更を行います。
- 「交換ファイルのインポート」ページから、XML ファイルを Identity Manager にインポートします。
- SystemConfiguration の userActionsConfigMapping 属性の値を「My User Actions Configuration」に変更します。

設定オブジェクトは、次の設定セクションで構成されます。

表 4-2 User Actions Configuration 属性

属性	説明
findUsersButtons	管理者インタフェースの「ユーザーアカウントの検索結果」ページのボタン定義のリスト。
userApplet.userMenu	ユーザーに対する操作メニューのメニュー項目定義のリスト。このメニューは、管理者インタフェースの「アカウント」ページのアプレット内でユーザーを右クリックしたときに表示されます。
userApplet.organizationMenu	組織に対する操作メニューのメニュー項目定義のリスト。このメニューは、「アカウント」ページのアプレット内で組織を右クリックしたときに表示されます。

各セクションでは、インタフェースで表示するユーザーアクションのリストが定義されます。ボタンおよびメニューの設定項目には共通の基本プロパティがあります。どちらの項目にも、インタフェースに特有のいくつかの拡張が含まれます。

各リストへ組織変更タスクを追加する

次の抜粋部分は、各リストに Change Organization (組織変更) タスクを追加するためのカスタマイズされたユーザーアクション設定の例です。

コード例 4-5 各リストへの組織変更タスクの追加

```

<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE Waveset PUBLIC 'waveset.dtd' 'waveset.dtd'>
<Waveset>

<Configuration name='My User Actions Configuration'>
  <Extension>
    <Object>
      <!-- ユーザー検索の結果ページのボタン -->
      <Attribute name='findUsersButtons'>
        <List>
          <Object>
            <Attribute name='textKey'          value='UI_NEW_LABEL' />
            <Attribute name='commandName'      value='New' />
            <Attribute name='requiredPermission'>
              <Object>
                <Attribute name='objectType' value='User' />
                <Attribute name='rights'     value='Create' />
              </Object>
            </Attribute>
            <Attribute name='alwaysDisplay' value='true' />
          </Object>
          ...
          <Object>
            <Attribute name='textKey'          value='UI_CHANGE_ORGANIZATIONS_LABEL' />
            <Attribute name='commandName'      value='Change Organizations' />
          </Object>
        </List>
      </Attribute>
      <Attribute name='userApplet'>
        <Object>
          <!-- ユーザーが選択されたときに表示するメニュー -->
          <Attribute name='userMenu'>
            <List>
              <Object>
                <Attribute name='textKey'
value='UI_ACCT_JAVA_MENU_NEW_ORG' />
                <Attribute name='commandName'
value='New Organization' />
                <Attribute name='requiredPermission'>
                  <Object>
                    <Attribute name='objectType' value='ObjectGroup' />
                    <Attribute name='rights'     value='Create' />
                  </Object>
                </Attribute>
              </Object>
              ...
              <Object>
                <Attribute name='separator' value='separator' />
              </Object>
              <Object>
                <Attribute name='textKey'
value='UI_CHANGE_ORGANIZATIONS_MENU_LABEL' />
                <Attribute name='commandName'

```

コード例 4-5 各リストへの組織変更タスクの追加 (続き)

```
value='Change Organizations' />
  </Object>
</List>
</Attribute>
  <!-- 組織が選択されたときに表示するメニュー -->
  <Attribute name='organizationMenu'>
    <List>
      <Object>
        <Attribute name='textKey'
value='UI_ACCT_JAVA_MENU_NEW_JUNCTION' />
        <Attribute name='commandName'
value='New Directory Junction' />
        <Attribute name='requiredPermission'>
          <Object>
            <Attribute name='objectType' value='ObjectGroup' />
            <Attribute name='rights' value='Create' />
          </Object>
        </Attribute>
        <Attribute name='orgTypes' value='normal,dynamic' />
      </Object>
      ...
      <Object>
        <Attribute name='separator' value='separator' />
      </Object>
      <Object>
        <Attribute name='textKey'
value='UI_CHANGE_ORGANIZATIONS_MENU_LABEL' />
        <Attribute name='commandName'
value='Change Organizations' />
      </Object>
    </List>
  </Attribute>
</Object>
</Attribute>
</Object>
</Extension>
<MemberObjectGroups>
  <ObjectRef type='ObjectGroup' name='All' />
</MemberObjectGroups>
</Configuration>
</Waveset>
```

ユーザーアクション定義は、次のコア属性をサポートします。

表 4-3 サポートされるコア属性

属性	説明
textKey	ボタンまたはメニュー項目のテキストのメッセージカタログキー。

表 4-3 サポートされるコア属性 (続き)

属性	説明
commandName	実行するコマンドの名前。この属性は、(「新規」や「ユーザーの削除」などの) ネイティブにサポートされているコマンド、または実行する TaskDefinition の名前に設定できます。
requiredPermission.objectType	この項目を表示するために権限が必要なオブジェクトの種類。これは、ネイティブにサポートされているコマンドのみに適用されます。タスク定義については、アクセスの制御に AuthType を使用するようにしてください。
requiredPermission.rights	指定された objectType がこの項目を表示するために必要な権限名のコンマ区切りのリスト。これは、ネイティブにサポートされているコマンドのみに適用されます。タスク定義については、アクセスの制御に AuthType を使用するようにしてください。
alwaysDisplay	オプション。このボタンを常に表示するかどうかを指定します。この値を true に設定した場合、ユーザー検索で結果が何も返されない場合でもボタンが表示されます。この属性のデフォルト値は false です。 findUsersButtons セクションのみに適用されます。

userApplet セクションのユーザーアクション定義は、次の表に示す属性もサポートします。

表 4-4 **userApplet** ユーザーアクションでサポートされる属性

属性	説明
orgTypes	組織メニューに項目を表示する組織タイプのコンマ区切りリスト。指定できる値は、通常組織を表す normal 、動的組織を表す dynamic 、仮想組織を表す virtual です。 この属性を指定しない場合、すべての組織タイプに対してメニュー項目が表示されます。
separator	<Object><Attribute name='separator' value='separator' /></Object> の形式の特殊項目。セパレータは管理者インタフェースに水平バーとして表示され、選択できません。

ユーザーアクションを設定する

Identity Manager のプライベートラベリング

この章では、会社のイントラネット環境または企業のスタイルガイドラインに合わせて Identity Manager インタフェースを再定義するために必要な基本コンポーネントについて説明します。プライベートラベリングとは、このような企業のガイドラインに合わせてインタフェースをカスタマイズすることを言います。

プライベートラベリングのタスク

プライベートラベリングのタスクは、3つのカテゴリに大別されます。

- **デフォルトのヘッダーコンテンツの変更。** ユーザーインタフェースと管理者インタフェースの両方で、企業ロゴの取り込み、デフォルトテキストの変更、色の変更を行います。
- **表示フォントとコンポーネントの色の変更。** `styles\customStyle.css` のスタイルシートを使用して、アプリケーション全体にわたる変更を行います。
- **よく使用するページでの Identity Manager の動作の変更。** System Configuration オブジェクトを編集することによって行います。「パスワードをお忘れですか？」ボタンの無効化を含むこれらのタスクは、製品インタフェースの再定義の間にユーザーによってたびたび実行されます。

アーキテクチャー機能

プライベートラベリングの作業には、次の表に示すコンポーネントの編集が含まれます。

表 5-1 カスタマイズ可能なコンポーネント

コンポーネント	インタフェース
<code>\$WSHOME/styles/customStyle.css</code>	管理者 / ユーザー
<code>\$WSHOME/WEB-INF/lib/idmcommon.jar</code>	管理者 / ユーザー
<code>\$WSHOME/user/userFooter_beforeFirstTableRowTag.jsp</code>	ユーザーインタフェース
<code>\$WSHOME/user/userFooter_beforeEndBodyTag.jsp</code>	ユーザーインタフェース
<code>\$WSHOME/user/userFooter_beforeLastEndTableRowTag.jsp</code>	ユーザーインタフェース
<code>\$WSHOME/includes/bodyEnd_beforeFirstTableRowTag.jsp</code>	管理者インタフェース
<code>\$WSHOME/includes/bodyEnd_beforeEndBodyTag.jsp</code>	管理者インタフェース
<code>\$WSHOME/includes/bodyEnd_beforeLastEndTableRowTag.jsp</code>	管理者インタフェース
<code>\$WSHOME/index_quickLinks.jsp</code>	管理者 / ユーザー

スタイルシート

製品インタフェースでのテキストの表示特性に影響する、4つのスタイルシートがあります。

- `lockhart.css` - Web アプリケーション用の Sun コーポレートインタフェーススタイルが含まれます。
- `style.css` - 両方のインタフェースにわたる、ページの表示属性を定義します。このファイルは、ヘッダーに含まれる画像も制御します。
- `customStyle.css` - `style.css` および `lockhart.css` に含まれるデフォルト設定への変更が含まれます。このファイルの設定は、`style.css` および `lockhart.css` の設定より優先されます。これら 2 つのファイルを編集する代わりに、`customStyle.css` でカスタマイズを行うことをお勧めします。
- `Styles-Help.css` - オンラインヘルプおよびポップアップヘルプ (i-Help) で使用されるスタイルクラスを定義します。

デフォルトテキスト

デフォルトテキストは、製品インタフェース全体を通して次の場所に出現します。

- フォームのタイトル、サブタイトル、ボタン、奇数行、偶数行、セクションヘッダー
- 一般テキスト
- 警告メッセージ
- ナビゲーションボタンのテキスト。利用可能および選択状態のナビゲーションボタンの両方を含む
- テーブルヘッダー / 本文テキスト

テキスト属性

次を含む属性を表示します。

- title - font-family、font-size、font-weight、color
- button - text-alignment、background-color
- text - title と同じ属性および text-decoration、white-space

デフォルトスタイル設定

\$WSHOME/styles/style.css および lockhart.css の各ファイルには、デフォルトのスタイル設定が格納されています。これらのファイルは編集しないでください。

カスタマイズするファイル

customStyle.css ファイルにはカスタマイズの定義が格納されます。このファイルは製品のアップグレード時に上書きされません。このファイルで定義される設定は、style.css および lockhart.css の設定より優先されます。すべてのカスタマイズ定義を customStyle.css に含めるようにしてください。

JSP ファイル

ヘッダーのデフォルト設定を格納する JSP ファイルに、`userHeader.jsp`、`bodyEnd.jsp`、および `bodyStart.jsp` があります。これらのファイルは編集しないでください。製品アップグレード時にカスタマイズ内容を保持するには、このファイルに代わりに、「[アーキテクチャー機能](#)」の一覧に示されている JSP のみを編集してください。

WPMessages_ja.properties ファイル

`$WSHOME/WEB-INF/lib/idmcommon.jar` ファイルにはメッセージカタログのエントリが格納されています。JAR ファイルから `WPMessages_ja.properties` ファイルを抽出し、そのファイル上でエントリを編集できます。編集結果は、必要に応じて `WPMessages_ja.properties` にも反映させてください。

ヘッダーのカスタマイズ

カスタマイズのタスクは両方のインタフェースで同じですが、編集する必要のあるファイルは異なります。

注	指定されたファイル以外の <code>.jsp</code> ファイルを編集しないでください。編集する必要がある場合は、 <code>.jsp</code> ファイルのコピー、編集、名前変更を行う前に、まずそのファイルを安全な場所にバックアップしてください。
---	---

ヘッダー外観を変更する

最も一般的なラベリングのタスクでは、次のことを行います。

- ページのヘッダーセクションで参照される画像を、デフォルトの **Sun** ロゴから会社ごとの標準画像に変更します。`customStyle.css` を編集することにより、画像を置換または削除します。
- Identity Manager ロゴを非表示にします。
- Look & Feel に関する社内ガイドライン (特に境界、ヘッダー、背景色) を使用します。

- 「次のユーザーとしてログイン」テキストを変更します。これは .jsp ファイルからの変更はできません。カスタムメッセージカタログを編集することができます。
「[Identity Manager ユーザーインタフェースのホームページに表示されるデフォルト情報を変更する](#)」を参照してください。

Identity Manager ページをカスタマイズする

一般的なカスタマイズには、次のものがあります。

- ホームページのカスタマイズ
- Identity Manager ユーザーインタフェースのホームページのデフォルト文字列の変更

ホームページをカスタマイズする

- クイックリンクのリストの追加
- デフォルトのログインテキストの変更
- ページタイトルとサブタイトルの変更
- ブラウザのタイトルバーのカスタマイズ

クイックリンクのリストを追加する

ホームページへの一般的なカスタマイズでは、環境内でユーザーが頻繁にアクセスするタスクまたはリソースに、カスタマイズしたリンクリストを追加します。これらのクイックリンクには、製品インタフェースからよく使う項目へのショートカットとしての機能があります。

クイックリンクのリストを追加するには、次の手順を実行します。

1. `index_quickLinks.jsp` でリストにリンクを追加します。
2. 前後の `<%--` および `--%>` タグを削除して、`list` セクションのコメントを解除します。
3. ファイルを保存します。アプリケーションサーバーを再起動する必要はありません。

デフォルトの「次のユーザーとしてログイン」テキストを変更する

1. 次の XML ファイルをインポートします。

```
<?xml version='1.0' encoding='UTF-8'?>

<!DOCTYPE Configuration PUBLIC 'waveset.dtd' 'waveset.dtd'>
```

```
<Configuration name='AltMsgCatalog'>
  <Extension>
    <CustomCatalog id='AltMsgCatalog' enabled='true'>
      <MessageSet language='en' country='US'>
        <Msg id='UI_BROWSER_TITLE_PROD_NAME_OVERRIDE'>Override
Name</Msg>
      </MessageSet>
    </CustomCatalog>
  </Extension>
</Configuration>
```

2. **System Configuration** オブジェクトの <Configuration><Extension><Object> 要素内に次の行を追加します。

```
<Attribute name='customMessageCatalog' value='AltMsgCatalog' />
```

3. 次の行を追加します。

```
<msg id='UI_NAV_FOOT_LOG'>mytext{0}</msg>
```

4. 変更を保存し、アプリケーションサーバーを再起動します。

ページタイトルとサブタイトルを変更する

ログインページのデフォルトのタイトル、サブタイトル、「ようこそ」メッセージを変更するには、カスタムメッセージカタログの次のエントリを変更します。

- UI_LOGIN_TITLE
- UI_LOGIN_TITLE_TO_RESOURCE
- UI_LOGIN_WELCOME2

このテキストを変更するには、「[デフォルトの「次のユーザーとしてログイン」テキストを変更する](#)」で詳しく説明している、WPMessages_ja.properties ファイルの抽出および編集の手順に従います。

コード例 5-1

デフォルトのメッセージカタログ設定

```

UI_LOGIN_IN_PROGRESS_TITLE= ログイン（進行中）
UI_LOGIN_CHALLENGE={0} パスワードを入力してください
要求したアクションを完了するには、[PRODUCT_NAME] への
ログイン時に指定したパスワードを入力する必要があります。
UI_LOGIN_TITLE_LONG=[PRODUCT_NAME] ログイン
UI_LOGIN_WELCOME=Sun Java&#8482; System [PRODUCT_NAME] システムへようこそ。
要求される情報を入力し、<b> ログイン</b> をクリックしてください。
UI_LOGIN_WELCOME2=Sun Java&#8482; System [PRODUCT_NAME] システムへようこそ。
ユーザー ID とパスワードを入力し、「<b> ログイン</b>」をクリックしてください。パスワードを思い出せない
ときは、「<b> パスワードをお忘れですか？</b>」をクリックしてください。
UI_LOGIN_TITLE= ログイン
UI_LOGIN_TITLE_TO_RESOURCE=<b>{0}</b> へのログイン

```

ログインページの背景画像を変更する

次のように customStyle.css を編集することにより、背景画像を変更できます。

```

div#loginFormDiv {
    background:
url(.../images/other/login-backimage2.jpg)
no-repeat;
    margin-left: -10px;
    padding: 0px 0px 280px;
    height: 435px;
}

```

ブラウザのタイトルバーをカスタマイズする

ブラウザのタイトルバーの製品名文字列を、任意のローカライズ可能な文字列で置き換えることができるようになりました。

1. 次の XML ファイルをインポートします。

コード例 5-2 インポートする XML

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE Configuration PUBLIC 'waveset.dtd' 'waveset.dtd'>
<Configuration name='AltMsgCatalog'>
  <Extension>
    <CustomCatalog id='AltMsgCatalog' enabled='true'>
      <MessageSet language='en' country='US'>
        <Msg id='UI_BROWSER_TITLE_PROD_NAME_OVERRIDE'>Override Name</Msg>
      </MessageSet>
    </CustomCatalog>
  </Extension>
</Configuration>
```

- 2. Identity Manager IDE を使用して、System Configuration オブジェクトを編集のためにロードします。<Configuration><Extension><Object> 要素内に次の行を追加します。

```
<Attribute name='customMessageCatalog' value='AltMsgCatalog' />
```

- 3. System Configuration オブジェクトの変更を保存し、アプリケーションサーバーを再起動します。

Identity Manager ユーザーインターフェースの ホームページに表示されるデフォルト情報を変更する

Identity Manager ユーザーインターフェースのホームページには、ログイン中のアカウントについての基本情報を表示する「ダッシュボード」領域があります。表示される情報には、次の表に示すデフォルト文字列が含まれます。

すべての属性は `ui.web.user.dashboard` オブジェクトに属します。

表 5-2 `ui.web.user.dashboard` オブジェクトのデフォルト設定

設定オブジェクトのデフォルト設定	説明
<code>displayLoginFailures</code>	アカウントのポリシーで最大ログイン試行回数が設定されている場合に、ログイン試行でのパスワード入力または秘密の質問への回答の失敗回数を表示します。

表 5-2 ui.web.user.dashboard オブジェクトのデフォルト設定

設定オブジェクトのデフォルト設定	説明
displayPasswordExpirationWarning	パスワードポリシーがアカウントに適用される場合に、パスワードの有効期限に関連するメッセージを表示します。
displayApprovals	すべてのユーザーを対象に、「承認」、「アテステーション」、「是正」、および「その他」の各作業項目タイプの表示を有効にします。
displayAttestationReviews	
displayOtherWorkItems	
displayRemediations	注: 設定オブジェクトで特定のタイプに対する設定が true であっても、アカウントユーザーに許可されたアクセス権によっては、インタフェース文字列が表示されない場合があります。
displayRequests	アカウントのロール、グループ、またはリソースの更新要求のうち未処理のものの数を示します。
displayDelegations	ユーザーが承認の委任を定義したことを示す文字列を表示します。「有効」または「無効」のオプションがあります。

デフォルトでは、これらの設定オブジェクトの値は **true** に設定され、これらの文字列は Identity Manager ユーザーインタフェースに表示されます。任意の文字列を非表示にするには、System Configuration オブジェクトでその文字列を **false** に設定します。

System Configuration オブジェクトは Identity Manager IDE を通じて編集できます。Identity Manager IDE の使用方法については、『Identity Manager Deployment Tools』の「Using the Identity Manager IDE」を参照してください。System Configuration オブジェクトについての全般的な情報は、「[設定オブジェクトを編集する](#)」を参照してください。

ユーザーインタフェースのナビゲーションメニューの外観を変更する

ユーザーインタフェースのナビゲーションメニューに関して、System Configuration オブジェクトで次の 2 つの設定を行う必要があります。

- ui.web.user.showMenu を **true** に設定する必要があります。
- ui.web.user.menuLayout を **horizontal** (デフォルト) に設定すると、ナビゲーションメニューの水平タブが描画されます。値を **vertical** に設定すると、垂直ツリーメニュー形式でメニューが描画されます。

フォント特性を変更する

表示属性では一般に、次の基本フォント表示特性を指定します。

表 5-3 フォント関連の表示特性	
表示属性	説明
family	Helvetica、Arial など
size	ポイントサイズで指定 (例 : 14 ポイント)
weight	未指定の場合は通常の書体。指定する場合、bold など
color	通常は black を指定

一部のコンポーネントは、追加の特性によってより詳細な定義が可能です。たとえば、ボタンの背景色を定義することができます。テキストやボタンラベルの配置も特性の一つです。

フォント特性を編集する

編集を行うには、styles.css からコピーして customStyle.css にペーストします。その後、選択した設定を customStyle.css で修正します。

例

次のエントリは、各ページタイトルのデフォルト設定を表します。

```
.title {  
    font-family: Arial, Helvetica, sans-serif;  
    font-size: 16pt;  
    font-weight: bold;  
    color: C;
```

ラベリング演習のサンプル

次の例は、Identity Manager のロゴを非表示にし、ページのマストヘッドでカスタム画像を参照する方法を示します。sample/rebranding ディレクトリに収められたサンプルファイルを参照してください。

- 製品名の変更

- マストヘッドの色の変更
- ナビゲーションタブの色の変更
- よく使われるページでの Identity Manager の動作の変更

Identity Manager ロゴのカスタムロゴへの置き換え

管理者インタフェースまたはユーザーインタフェースのロゴを変更するには、styles/style.css の次の部分を customStyle.css にコピーし、Identity Manager のロゴ画像を独自の画像ファイルに置き換えます。

```
td.MstTdLogo {
    width: 31px;
    height: 55px;
    background: url(../images/other/logo.jpg) no-repeat 5px;
}

td.MstTdTtlProdNam {
    background: url(../images/other/xyz_masthead.jpg) no-repeat 10px
30px;
    padding:10px 10px 0px 10px;
    vertical-align:top;
    white-space:nowrap;
    height: 75px;
    width: 350px;
}
```

注 最適な結果を得るには、ロゴ画像の高さを 50 ～ 60 ピクセルとしてください。

マストヘッドの外観を変更する

Identity Manager の Look & Feel を変更するには、styles/customStyle.css ファイルを編集します。

styles.css の次のセクションを customStyle.css にコピーし、必要に応じて変更します。

コード例 5-3 styles/customStyle.css ファイル

```
MstDiv {background-image:url(..images/other/dot.gif);background-repeat:repeat-x;
background-position:left top;background-color:#000033}
.MstTblEnd {background: url(..images/other/dot.gif);background-color: #666;height:
1px;}
td.MstTblEndImg {
    background-color: #666;
    height: 1px;
    background: url(..images/other/dot.gif);
    font-size:1px;
}
td.MstTdLogo {
    width: 31px;
    height: 55px;
    background: url(..images/other/logo.jpg) no-repeat 5px;
}
td.MstTdSep {
    width: 1px;
    height: 61px;
    background: url(..images/other/dot.gif) no-repeat center;
}
td.MstTdTtlProdNam {
    background: url(..images/other/xyz_masthead.jpg) no-repeat 10px 30px;
    padding:10px 10px 0px 10px;
    vertical-align:top;
    white-space:nowrap;
    height: 75px;
    width: 350px;
}
```

ナビゲーションタブを変更する

Identity Manager ユーザーインタフェースのナビゲーションバーをカスタマイズする

Identity Manager のユーザーインタフェースは、ナビゲーションバーを含む 2 番目の XPRESS フォームを実装しています。これは、描画されるページに 2 つの <FORM> タグがあり、各タグで name 属性の設定が次のように異なることを意味します。

```
<form name="endUserNavigation">
```

および

```
<form name="mainform">
```

Identity Manager ユーザーインタフェースのナビゲーションバーに JavaScript コードを挿入するときは、必ず正しいフォームを参照するようにしてください。そのためには、name 属性を使用して、どちらの <FORM> タグを参照するか (document.mainform または document.endUserNavigation のどちらか) を指定します。

ナビゲーションリンクをカスタマイズする

ページの上部を横切るナビゲーションリンクをカスタマイズするには、次のコードをコピーして編集します。背景色を適切な色に変更します。

コード例 5-4 ナビゲーションリンクのカスタマイズ

```
* LEVEL 1 TABS */
.TabLvl1Div {
    background-image:url(../images/other/dot.gif);
    background-repeat:repeat-x;
    background-position:left bottom;
    background-color:#333366;
    padding:6px 10px 0px;
}
a.TabLvl1Lnk:link, a.TabLvl1Lnk:visited {
    display:block;
    padding:4px 10px 3px;
    font: bold 0.95em sans-serif;
    color:#FFF;
    text-decoration:none;
    text-align:center;
}
table.TabLvl1Tbl td {
    background-image:url(../images/other/dot.gif);
    background-repeat:repeat-x;
    background-position:left top;
    background-color:#666699;
    border:solid 1px #abab5;
}
table.TabLvl1Tbl td.TabLvl1TblSelTd {
```

コード例 5-4 ナビゲーションリンクのカスタマイズ(続き)

```
background-color:#9999CC;
background-image:url(../images/other/dot.gif);
background-repeat:repeat-x;
background-position:left bottom;
border-bottom:none;
```

タブパネルのタブを変更する

コード例 5-5 タブパネルのタブの変更 Identity Manager

```
.TabLvl2Div {
    background-image:url(../images/other/dot.gif);
    background-repeat:repeat-x;
    background-position:left bottom;
    background-color:#9999CC;
    padding:6px 0px 0px 10px
}
a.TabLvl2Lnk:link, a.TabLvl2Lnk:visited{
    display:block;
    padding:3px 6px 2px;
    font: 0.8em sans-serif;
    color:#333;
    text-decoration:none;
    text-align:center;
}
table.TabLvl2Tbl div.TabLvl2SelTxt {
    display:block;
    padding:3px 6px 2px;
    font: 0.8em sans-serif;
    color:#333;
    font-weight:normal;
    text-align:center;
}
table.TabLvl2Tbl td {
    background-image:url(../images/other/dot.gif);
    background-repeat:repeat-x;
    background-position:left top;
    background-color:#CCCCFF;
    border:solid 1px #abab5;
}
table.TabLvl2Tbl td.TabLvl2TblSelTd {
    border-bottom:none;
    background-image:url(../images/other/dot.gif);
    background-repeat:repeat-x;
    background-position:left bottom;
    background-color:#FFF;
    border-left:solid 1px #abab5;
    border-right:solid 1px #abab5;
    border-top:solid 1px #abab5;
```

コード例 5-5

タブパネルのタブの変更 Identity Manager (続き)

```
table.Tab2TblNew td
{background-image:url(..images/other/dot.gif);background-repeat:repeat-x;background-p
osition:left top;background-color:#CCCCFF;border:solid 1px #8f989f}
table.Tab2TblNew td.Tab2TblSelTd
{border-bottom:none;background-image:url(..images/other/dot.gif);background-repeat:re
peat-x;background-position:left bottom;background-color:#FFF;border-left:solid 1px
#8f989f;border-right:solid 1px #8f989f;border-top:solid 1px #8f989f}
```

ソートテーブルのヘッダーを変更する

```
.tablehdr {
    background-image: url(..images/other/dot.gif);
    background-repeat: repeat-x;
    background-position: left bottom;
    background-color: #9999CC;
}
```

ユーザー / リソーステーブルのコンポーネントを変更する

コード例 5-6

ユーザー / リソーステーブルのコンポーネントの変更

```
.tablesorthdr {
    /*background-color: #BEC7CC;*/
    background-image:url(..images/other/dot.gif);
    background-repeat:repeat-x;
    background-position:left bottom;
    background-color:#CCCCFF;
    border:solid 1px #abab5;
}

.treeContentLayout {
    background-color: #9999CC;
}

.treeBaseRow {
    padding-top: 0px;
    padding-left: 10px;
    padding-right: 10px;
```

コード例 5-6 ユーザー / リソーステーブルのコンポーネントの変更 (続き)

```
padding-bottom: 0px;
background-color: #fff;
border-left: solid 1px #8F989F;
border-right: solid 1px #8F989F;
border-bottom: solid 1px #8F989F;
}

.treeButtonCell {
    background-image: url(../images/other/dot.gif);
    background-color: #666699;
    color: #fff;
}

}

.treeMastHeadRow {
    background-color: #333366;
}

.treeMastHeadRow {
    background-color: #333366;
}

.treeMastHeadText {
    background-color: #333366;
    background-image: url(../images/other/dot.gif);
}
```

同じ手順に従うことにより、その他のさまざまなオプション (テキストのスタイルやサイズ、配置、その他のオブジェクトの色や設定など) をカスタマイズできます。

注	サーバーまたはブラウザを再起動せずに変更を確認するには、ページ上で Ctrl キーを押しながら「更新」を実行します。
----------	--

よく使われるページでの Identity Manager の動作を変更する

System Configuration オブジェクトの設定を変更することにより、よく使われるページでの Identity Manager の動作をカスタマイズできます。

System Configuration オブジェクトを使用したカスタマイズ

System Configuration オブジェクトを編集することにより、ユーザーインターフェースまたは管理者インターフェースの、よく変更される多くのプロパティをカスタマイズすることができます。製品インターフェースを制御するのは、属性 `<Attribute name='ui'>` とそのサブオブジェクトです。この属性の下位属性を変更することにより、Identity Manager の動作を変えることができます。

その他の変更: ファイルの *admin* セクション

System Configuration オブジェクトファイルの *admin* セクションには、管理者インターフェースに関連するさまざまな属性が含まれます。

- 管理者ログインページの「パスワードをお忘れですか?」ボタンを無効にするには、`disableForgotPassword` を **true** に設定します。
- `supressHostName` を **true** に設定すると、タスク詳細情報ページでプロセスのホスト名が非表示になります。

コード例 5-7 ファイルの *admin* セクションの変更

```
<Attribute name='admin'>
  <Object>
    <Attribute name='disableForgotPassword'>
      <Boolean>>false</Boolean>
    </Attribute>
    <Attribute name='workflowResults'>
      <Object>
        <Attribute name='supressHostName'>
          <Boolean>>false</Boolean>
        </Attribute>
      </Object>
    </Attribute>
  </Object>
</Attribute>
```

その他の変更: ファイルの user セクション

System Configuration オブジェクトファイルの user セクションには、ユーザーインタフェース関連のオプションがあります。

- disableForgotPassword を true に設定すると、「パスワードをお忘れですか?」ボタンが無効になります。

workflowResults 属性には、管理者でないユーザーのワークフローの表示をカスタマイズするための属性が、次の表に示すように含まれます。

表 5-4 管理者でないユーザーのワークフローをカスタマイズするための属性

属性	説明
anonSuppressReports	匿名ユーザーのワークフローステータスページ (anonProcessStatus.jsp) にワークフロー図を表示するかどうかを制御します。
suppressHostName	エンドユーザーのワークフローステータスページ (processStatus.jsp) にホスト名の情報を含めるかどうかを制御します。
suppressReports	匿名でないすべてのユーザーに対してワークフロー図を表示するかどうかを制御します (processStatus.jsp)。

コード例 5-8 管理者でないユーザーのワークフローのカスタマイズ

```

<Attribute name='user'>
  <Object>
    <Attribute name='disableForgotPassword'>
      <Boolean>false</Boolean>
    </Attribute>
    <Attribute name='workflowResults'>
      <Object>
        <Attribute name='anonSuppressReports'>
          <Boolean>false</Boolean>
        </Attribute>
        <Attribute name='suppressHostName'>
          <Boolean>false</Boolean>
        </Attribute>
        <Attribute name='suppressReports'>
          <Boolean>false</Boolean>
        </Attribute>
      </Object>
    </Attribute>
  </Object>
</Attribute>

```

- パスワードおよび秘密の質問の回答を非表示にするには、obfuscateAnswers を true に設定します。このように設定すると、管理者インタフェースとユーザーインタフェースの両方で、入力したパスワードや回答がアスタリスクでマスクされます。

```

<Attribute name="obfuscateAnswers">
  <Boolean>true</Boolean>
</Attribute>

```

ラベリング演習のサンプル

メッセージカタログをカスタマイズする

メッセージカタログエントリを追加したり、システムから提供されるエントリを修正したりするために、カスタマイズしたメッセージカタログを作成できます。

カスタムメッセージカタログの利点

カスタムメッセージカタログには、次の利点があります。

- クラスタ環境での保守作業の軽減。個別のメッセージカタログを保守すると、複数の `WPMessages.properties` ファイルを編集する必要がなくなります。
- バージョン制御の簡素化。カスタマイズしたメッセージが 1 か所にあると、変更の追跡や改訂版のバックアップが簡単になります。
- 製品メッセージカタログへのアップグレードは、カスタマイズしたエントリに加えられた変更で失敗することはありません。

Identity Manager がメッセージカタログエントリを取得する方法

Identity Manager は、次の順序でメッセージカタログエントリを取得します。

- ユーザー定義のメッセージカタログ (許容されるユーザー定義メッセージカタログは 1 つのみ)
- システム定義の `defaultCustomCatalog` メッセージカタログ
- `config/WPMessages.properties` ファイル
- `idmcommon.jar` ファイル内の `WPMessages.properties` ファイル。

メッセージカタログの形式

WPMessages.properties ファイルでは、エントリは *KeyName=MessageText* の形式で定義されます。カスタマイズしたメッセージカタログでは、各エントリは個別の `Msg` 要素で指定されます。*KeyName* は `id` 属性で指定されます。*MessageText* は `<Msg>` タグと `</Msg>` タグの間にあるテキストです。次の例は、メッセージカタログエントリを示しています。

```
<Msg id='UI_REMEMBER_PASSWORD'>Remember to set your password.</Msg>
```

メッセージテキストに **HTML** タグを含めてテキストのレンダリング方法を制御できませんが、これはお勧めできません。**HTML** タグを使用する必要がある場合は、`<` や `>` などの記号の代わりに `<` や `>` などのコードを使用してください。

メッセージテキストには、文字列が表示されるときに **Identity Manager** によって文字列に挿入されるデータの変数を含めることもできます。次の例は、`AR_CORRELATED_USER` キーのデフォルトメッセージです。

```
Correlated account with user {0}.
```

レンダリングされたものは次のようになります。

```
Correlated account with user jdoe.
```

カスタマイズしたメッセージカタログを作成する

次の手順では、ユーザー定義のメッセージカタログを作成する方法について説明します。

1. デフォルトのメッセージカタログエントリを上書きする場合は、`WPMessages.properties` ファイルで該当するエラーメッセージキーの場所を特定します。これらのキーは、カスタマイズしたメッセージカタログで指定する必要があります。

新しいメッセージを作成する場合は、これらのキーが `WPMessages.properties` ファイルにないことを確認します

2. 次の構造で **XML** ファイルまたはブロックを作成します。

```
<?xml version='1.0' encoding='UTF-8'?>

<!DOCTYPE Configuration PUBLIC 'waveset.dtd' 'waveset.dtd'>

<Configuration name='CatalogName'>

    <Extension>

        <CustomCatalog id='CatalogName' enabled='true'>

            <MessageSet language='en' country='US'>
```

```

        <Msg id='KeyName'>MessageText</Msg>

        <Msg id='KeyName'>MessageText</Msg>

        ...

    </MessageSet>

</CustomCatalog>

</Extension>

</Configuration>

```

各表記の意味は次のとおりです。

CatalogName は、メッセージカタログの名前です。この値は、**System Configuration** オブジェクトのカタログの定義にも使用されます。

KeyName はメッセージキー名です。

MessageText は、グラフィカルユーザーインターフェースに表示される文字列です。このテキストには、**HTML** タグと変数を含めることができます。

en_US 以外のロケールをサポートする場合は、**language** 属性と **country** 属性を変更します。複数のロケールをサポートする場合は、ロケールごとに個別の **MessageSet** 要素を作成します。

参考例については、「例」の項を参照してください。

3. ファイルまたはブロックを **Identity Manager** にインポートします。
4. **System Configuration** オブジェクトを読み込み、次の行を
`<Configuration><Extension><Object>` 要素内に追加します。
`<Attribute name='customMessageCatalog' value='CatalogName' />`
5. 変更を **System Configuration** オブジェクトに保存します。
6. アプリケーションサーバーを再起動します。これで、新しいメッセージカタログエントリがシステムで利用できるようになりました。

例

次の例では、**myCustomCatalog** という名前のカスタマイズしたメッセージカタログを作成します。ここでは、「交換ファイルのインポート」サブタブのラベルとヘルプテキストを置き換えます。

コード例 6-1

カスタマイズした myCustomCatalog メッセージカタログを作成する

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE Configuration PUBLIC 'waveset.dtd' 'waveset.dtd'>
<Configuration name='myCustomCatalog'>
  <Extension>
    <CustomCatalog id='myCustomCatalog' enabled='true'>
      <MessageSet language='en' country='US'>
        <Msg id='UI_SUBNAV_CONFIGURE_IMPORT_EXCHANGE'>XML ファイル
のインポート </Msg>
        <Msg id='UI_SUBNAV_CONFIGURE_IMPORT_EXCHANGE_HELP'>指定し
た XML ファイルをインポートします。 </Msg>
      </MessageSet>
    </CustomCatalog>
  </Extension>
</Configuration>
```

設定オブジェクトを編集する

この章では、設定オブジェクトと呼ばれる Identity Manager コンポーネントについて説明します。設定オブジェクトプロパティの編集は、Identity Manager の動作に持続的な変更を実装する方法の 1 つです。

設定オブジェクトについて

設定オブジェクトは、持続的なカスタマイズを Identity Manager に格納します。設定オブジェクトはキャッシュされるオブジェクトタイプなので、すべての設定オブジェクトはメモリーに格納され、その後設定オブジェクトが変更されるごとにそのキャッシュはフラッシュされます。

設定オブジェクトを表示および編集する

設定および汎用オブジェクトを表示するには、Identity Manager 統合開発環境 (IDE) を使用します。これらの補足的な設定オブジェクトには、設定オブジェクトカテゴリの下にある IDE からアクセスできます。

Identity Manager IDE の使用方法の詳細については、「Identity Manager IDE の概要」を参照してください。

UserUIConfig オブジェクト

UserUIConfig オブジェクトは、アカウントの検索と編集のために Identity Manager のユーザーインタフェースと管理者インタフェースの表示を制御します。また、内部システム関数も制御します。

配備時にこのオブジェクトを設定すると、Identity Manager のパフォーマンスが向上します。

このオブジェクトを使用して次のことができます。

- 「アカウント」アプレットおよび「結果の検索」ページに表示される列を制御する (SummaryAttrNames)
- アイデンティティ配備内でユーザーが検索できる属性、つまりクエリー可能な属性を定義する (QueryableAttrNames)
- 検索を最適化するために userobj テーブルの個別の列に格納される属性を定義する (RepoIndexAttrs)

注 このオブジェクトは、基本的なレベルでシステムの動作を制御します。このオブジェクトを編集すると、システムのパフォーマンスに広範な影響が及びます。編集は慎重に行ってください。

このオブジェクトを表示および編集する

このオブジェクトは、Identity Manager IDE を使用すると、その他の設定および汎用システムオブジェクトとともに表示できます。Identity Manager IDE を使用してこのオブジェクトにアクセスする方法については、『Identity Manager Deployment Tools』の第 1 章「Using the Identity Manager IDE」を参照してください。

ユーザーを更新する

このオブジェクトの SummaryAttrName、QueryableAttrName、および RepoIndexAttrs セクションに属性を追加または削除した場合は、続いて次のようにしてすべてのユーザーを更新する必要があります。

このオブジェクトを編集したあとで、すべてのユーザーオブジェクトに対して refreshType インポートコマンドを実行して、概要の属性を利用可能にする必要があります。多数のユーザーを更新する必要がある場合、このプロセスには時間がかかることがあります。

ファイルは次のようにしてインポートすることができます。

```
<?xml version='1.0' encoding='UTF-8'?>

<!DOCTYPE Waveset PUBLIC 'waveset.dtd' 'waveset.dtd'>

<Waveset>

    <ImportCommand type='refreshType' targetType='User' />

</Waveset>
```

属性タイプ

概要の属性は、ユーザーがリストコマンドを使って取得できる情報を公開します。これらの属性は、SummaryAttrTypes および SummaryAttrNames セクションで設定できます。「結果の検索」列またはアプレットリスト列の表示に属性を含めるには、その属性をこのセクションで概要の属性として追加します。

クエリー可能な属性は、Identity Manager 内でユーザーが検索できる属性を定義します。これらの属性は、このオブジェクトの QueryableAttrNames セクションで定義されます。

クエリー可能なインライン属性は、関連付けられたテーブル (userattr) ではなくメインテーブル (userobj) に格納されます。これらの属性は単一値である必要があります。インライン属性に対するクエリーは、関連付けられた属性テーブルへのアクセスよりはるかに高速です。これらの属性は、このオブジェクトの RepoIndexAttr セクションに含まれています。

オブジェクト属性

ここで説明する属性は、デフォルトの UserUIConfig オブジェクト属性のサブセットで構成されます。実際の配備で見られる属性は異なることがあります。

SummaryAttrNames

SummaryAttrNames オブジェクトのメンバーである属性は、概要の属性として指定されています。Identity Manager は、製品リストの結果に概要の属性を表示します。これらの属性は、AppletColumns および「結果の検索」リストのスーパーセットである必要がありますが、QueryableAttrNames リストに含まれている必要はありません。

このオブジェクトを編集する際に、MemberObjectGroups 属性を削除しないでください。この属性は、承認を高速にするために使用されます。

次の属性は、**Identity Manager** によって提供されるデフォルトの概要の属性です。
name および id は組み込みの概要の属性であり、ここでは説明しません。このリストに属性を追加することができます。

role

Identity Manager のロールを識別します。ロール ID は縦棒 (|) で区切ります。

res

リソース名をコンマ区切り形式でリストします。このリストの要素数が `SummaryAttrResourceCountLimit` の値を超えると、このリストは切り捨てられ、省略記号 (...) が追加されます。

prov

プロビジョニングレベルを指定します。これにより、リソースで **Identity Manager** の割り当てによるリソースをプロビジョンした数が特定されます。(0 = なし、1 = 一部、2 = すべて)

dis

(ブール型) ユーザーが無効になっているかどうかを示します。

MemberObjectGroups

このメンバーが属する組織を指定します。この属性は削除しないでください。

fullname*, *lastname*, *firstname

それぞれユーザーの `fullname`、`lastname`、および `firstname` 属性を指定します。

QueryableAttrNames

ユーザーが **Identity Manager** で検索できる属性を指定します。このリストに属性を追加することができます。

デフォルトのクエリー可能な属性には次のものがあります。

- `correlationKey`
- `role`
- `email`
- `firstname`
- `lastname`
- `prov`
- `dis`

- name
- locked
- user_resources

AppletColumns

「アカウントのリスト」ページに表示する列の名前を指定します。「アカウントのリスト」ページに表示される列の内容を変更するには、このリストを編集します。このリストで名前を付けた列は、SummaryAttrNames に含める必要があります (そうしないと、製品インタフェースに表示される値は空白になる)。このリストは、各列が GenericObjects で構成されます。サポートされる属性は次のとおりです。

- width - (アプレットの実装にのみ有効) 列の初期幅を指定します。省略した場合 (またはゼロの場合)、アプレットはデフォルトの初期幅を列に割り当てます。
- sortBy - (アプレットの実装にのみ有効) 存在する場合は、アプレットが最初にソートするときの基準になる列を指定します。複数の列を指定した場合は、左側にある列が使用されます。
- label - (アプレットとツリーテーブルの両方の実装に有効) ローカライズされた列名に使用するメッセージキーを指定します。

ShowListCache

「Clear List Cache」ボタンを「アカウントのリスト」ページに表示するかどうかを指定します。

TemporarySummaryAttrResourceCountLimit

リソースの概要に示すリソースの数を指定します。数を大きくしすぎると、リソーススキーマ違反が発生します。デフォルト値は、3 です。

PolicyAccountAttributeNames

accountId ポリシーのポップアップに表示される属性を定義します。これらの属性は、ユーザーオブジェクトで検索可能な属性と一致している必要があります。ユーザーフォームに含まれている属性を追加できます。デフォルト値には、email、firstname、fullname、および lastname が含まれます。

PolicyPasswordAttributeNames

パスワードポリシーのポップアップに表示される属性を定義します。これらの属性は、ユーザーオブジェクトで検索可能な属性と一致している必要があります。ユーザーフォームに含まれている属性を追加できます。デフォルト値には、accountId、email、firstname、fullname、および lastname が含まれます。

PolicyOtherAttributeNames

その他のポリシータイプのポップアップに表示される属性を定義します。これらの属性は、ユーザーオブジェクトで検索可能な属性と一致している必要があります。ユーザーフォームに含まれている属性を追加できます。デフォルト値には、accountId、email、firstname、fullname、および lastname が含まれます。

PolicySpecialChars

ある文字が ID やパスワードの文字列に使用できるか否かを決定する文字列の品質ポリシーに含める特殊文字を指定します。パスワードとアカウント ID のポリシーによって、特殊文字の最大数と最小数に関する規則を指定することが可能になります。

TaskBarPages

ページの下部にタスクバーを表示する Identity Manager 管理者インタフェース内の JSP ページのパスを定義します。たとえば、Identity Manager で新しいユーザーを作成するときに、アカウントページの下部に「ユーザーの作成」が表示されます。これは、accounts/list.jsp がデフォルトで TaskBarPages 要素に含まれているためです。

RepolIndexAttr

検索を円滑にするために waveset.userobj テーブルにコピーされ、インデックスが生成される属性を定義します。ここで名前を付けた属性は、クエリー可能でもある必要があります。

検索パフォーマンスを向上させるには、このオブジェクトを編集します。このリストに含めることができる属性は 5 つだけであり、これらの属性は ATTR1-ATTR5 データベース列にマップされます。デフォルトでは、Identity Manager は firstname、lastname、および MemberObjectGroups のインデックスを生成します。ただし、高速検索の場合は、さらに 2 つの属性を追加できます。たとえば、配備に拡張属性 departmentNumber が含まれる場合は、ここでその属性を追加すると、その属性がすべてのリポジトリ検索に含まれるようになります。firstname、lastname、または MemberObjectGroups が必要でないことがわかっている場合は、これらの属性をその他の属性で置き換えることができます。

注	クエリー可能な属性のインデックスを生成しない場合でも、その属性を検索に使用できますが、データベース内のユーザー数、および同時に検索を実行しているユーザー数によって、パフォーマンスは大幅に低下することがあります。
---	---

国際化サポートを有効にする

このドキュメントでは、Identity Manager で複数の言語を使用する場合や英語以外の言語を表示する場合の設定に関する情報を提供します。

アーキテクチャーの概要

表 B-1 Identity Manager の国際化のコンポーネント	
ファイル	説明
WPMessages.properties	デフォルトのメッセージファイル。 \$WSHOME/idm/web/WEB-INF/classes/com/waveset/msgcat にあります。idmcommon.jar ファイルの一部として出荷されています。 メッセージテキストを英語で表示します。これは、IDM のインストール動作をカスタマイズして変更しないかぎり、デフォルトでロードされます。
Waveset.properties	\$WSHOME/config にあります。複数言語のサポートを有効にする場合に、編集します (Internationalization.enabled を true に設定する)。
System Configuration オブジェクト	カスタムメッセージカタログを指定します。
サポートされる各言語の追加のメッセージファイル	サポートされる追加の言語にはそれぞれ独自のメッセージファイル WPMessages_xx_XX.properties が必要です。ここで、xx は言語を表し、XX は国を表します。たとえば、WPMessages_en_US.properties には、アメリカ英語のメッセージが含まれています。各国際化カタログには独自の .jar があります。

注意の補足：

- 新しいカタログを /config にロードした場合、その新しいカタログはデフォルトより優先されます。これは、同じ名前のカタログに対してのみ適用されます。
- 複数のメッセージファイルがある場合は、catalogname:keyname を指定して、メッセージキーを取得するカタログを指定できます。

一般的なエントリ

メッセージは、キーとテキストの組み合わせに含まれており、次の 3 つの部分で構成されます。

- テキスト文字列、またはキー。データを取得するためにコードで使用する識別子です。この必須コンポーネントは翻訳しません。このコンポーネントは製品設定で使用され、翻訳用のプレースホルダとして機能します。
- キーとテキストを区切る等号 (「=」)。これは必須です。
- アプリケーションの実行時に表示されるデータを含む文字列。これは翻訳されたものであり、ブラウザでページがレンダリングされるときにキーの代わりに使用されます。

リソース配列の各行には 2 つの文字列が含まれています。各行の、引用符で囲まれた 2 番目の文字列を翻訳します。

翻訳対象の特定の文字列には、文字列が表示されるときに文字列に挿入されるデータ用の特殊コードが含まれています。たとえば、次の文字列を翻訳するとします。

```
UI_USER_CONNECT={0}, connected at 100 mbs
```

レンダリングされたものは、「jfaux、100 mb で接続」などとなります。

通常、翻訳文字列はブラウザ内に表示されるため、次に示すように、HTML タグを追加して文字列の書式を設定することをお勧めします。

```
_FM_ACCOUNT_ID_HELP=<b>Account ID</b><br>Enter a name for this user.  
This field is required.
```

複数言語のサポートの有効化

複数の言語カタログのサポートを有効にするには、次に詳しく説明されている手順を実行します。

- ローカライズされたファイルをダウンロードします。
- `Internationalization.enabled` を **true** に設定して、`Waveset.properties` ファイルを編集します。
- `¥sample¥i18n.xml` ファイルをインポート、編集して、ロードします。
- **Identity Manager** を再起動します
- ログイン画面で言語を選択します。また、デフォルト言語はユーザーのブラウザで選択されている言語と一致している必要があります。

手順 1: ローカライズされたファイルをダウンロードしてインストールする

インストールの前に

ローカライズされたファイルをインストールする前に、次のタスクを実行します。

1. **Identity Manager** をインストールします。詳細なインストール手順については、「**Identity Manager インストール**」を参照してください。
2. アプリケーションサーバーの次のロケールが **UTF-8** に設定されていることを確認します。
 - アプリケーションサーバーインスタンス
 - データベース
 - Java 仮想マシン (JVM)

ロケールの設定に関する情報については、これらの製品のマニュアルを参照してください。

メッセージカタログファイルのダウンロード

適切な msgcat.jar をダウンロードして、WEB-INF/lib に配置します。Identity Manager Image Server の Web サイトでは、ローカライズされた製品ファイルとマニュアルを含む、次の ZIP ファイルを提供しています。

表 B-2 メッセージカタログファイル

ファイル名 (.zip)	言語	ロケール
IDM_7-0_l10n_de	ドイツ語	de_DE
IDM_7-0_l10n_es	スペイン語	es_ES
IDM_7-0_l10n_fr	フランス語 (フランスおよびカナダ)	fr_FR
IDM_7-0_l10n_it	イタリア語	it_IT
IDM_7-0_l10n_ja	日本語	ja_JP
IDM_7-0_l10n_ko	韓国語	ko_KR
IDM_7-0_l10n_pt	ポルトガル語 (ブラジル)	pt_BR
IDM_7-0_l10n_zh	簡体字中国語	zh_CN
IDM_7-0_l10n_zh_TW	繁体字中国語	zh_TW

ZIP ファイルを一時的な場所にダウンロードします。デフォルトでは、ZIP ファイルの内容は FileName ディレクトリに抽出されます。ここで、FileName は、ZIP 拡張子を除いたダウンロードファイルの名前と一致します。

Zip ファイルの内容

抽出したすべての ZIP ファイルには次のものが含まれます。

- ローカライズされたメッセージカタログを含む JAR ファイル、ヘルプファイル、およびその他の重要なファイル。JAR ファイルの名前は IDM_7_0_l10n_Locale.jar です。
- Identity Manager のローカリゼーションに関する Readme

抽出したほとんどの ZIP ファイルには次のものが含まれます。

- 『Identity Manager 7.0 リリースノート』の翻訳版
- 『Identity Manager 管理ガイド』の翻訳版

その他の翻訳物が利用できることもあります。

ローカライズされたファイルのインストール

次の手順を使用して、ローカライズされたファイルをアプリケーションサーバーにインストールします。

1. JAR ファイルを、一時的な場所から `IdentityManagerInstallation/WEB-INF/lib` ディレクトリにコピーします。

手順 2: Waveset.Properties ファイルを編集する

1. テキストエディタで `IdentityManagerInstallation/config/Waveset.properties` ファイルを開きます。
2. `Internationalization.enabled` プロパティを `true` に変更します。
3. 変更を保存してファイルを閉じます。
4. この変更を有効にするには、**Identity Manager** を再起動するか、デバッグページで「**Reload Properties**」を選択します。

複数言語のリンクをログインページに作成する

複数言語のリンクをログインページに表示させるには、まず次の点を確認します。

- `waveset.properties` ファイルの `internationalization.enabled` プロパティが `true` に設定されていること。
- 関連する言語パックが `web-inf/lib` で利用できること。必要に応じて、それらのライブラリをロードしてください。

ログインページにリンクを作成するには、次を実行します。

1. **Identity Manager IDE** を使って、**System Configuration** オブジェクトを開き、関連する言語パッケージの名前で `LoadedLanguages` 属性を更新します。
2. **Identity Manager** からログアウトし、アプリケーションサーバーを再起動します。

コード例 B-1 LoadedLanguages 属性の更新

```
<?xml version='1.0' encoding='UTF-8'?>

<!DOCTYPE Waveset PUBLIC 'waveset.dtd' 'waveset.dtd'>
<Waveset>
<ImportCommand
class='com.waveset.session.SystemConfigurationUpdater'>
  <Object>
    <Attribute name='LoadedLanguages'>
      <List>
```

コード例 B-1 LoadedLanguages 属性の更新 (続き)

```
<Object name='en_US'>
  <Attribute name='lang' value='en' />
  <Attribute name='cntry' value='US' />
</Object>
<Object name='fr_CA'>
  <Attribute name='lang' value='fr' />
  <Attribute name='cntry' value='CA' />
</Object>
</List>
</Attribute>
</Object>
</ImportCommand>
</Waveset>
```

リソース上の 1 ユーザーに対して複数のアカウントを割り当てる

Identity Manager には、リソース上の 1 人のユーザーに複数のアカウントを割り当てて、管理する機能が用意されています。これは、目的または機能をアカウントに割り当てられるようにすることで実現します。これは、Identity Manager 内でアカウントのタイプと呼ばれます。

アカウントタイプを使用するには、アカウントタイプをユーザーに割り当てる前に、まずリソースのアカウントタイプを設定する必要があります。

1 つのリソース上の 1 ユーザーに複数のアカウントを割り当てる理由

場合によっては、Identity Manager ユーザーが 1 つのリソースで複数のアカウントを必要とすることがあります。1 人のユーザーは、リソースに関連した複数の異なる機能を持つことがあります。たとえば、1 人のユーザーがリソースのユーザーでもあり管理者でもあることがあります。最善の方法は、機能ごとに個別のアカウントを使用することです。これにより、あるアカウントの安全性が損なわれても、その他のアカウントによって許可されたアクセスのセキュリティは保護されます。

アカウントタイプを設定する

リソース上で複数のアカウントを持つユーザーをサポートするには、そのリソースで使用可能なアカウントのタイプを定義する必要があります。アカウントのタイプを定義するには、次の情報を把握する必要があります。

- **アカウントタイプの名前:** この名前によってこのリソースでのタイプが一意に識別されます。また、この名前は、Identity Manager 全体でタイプの識別に使用されます。

- **タイプごとに accountId を生成する方法**: accountId は明確に区別される必要があるため、特定のユーザーに対して、アカウントのタイプが異なれば生成される accountId も異なっている必要があります。このマッピングを指定するには、ID 規則を使用します。リソースのアイデンティティテンプレートと同様、Identity Manager はプロビジョニング中にアカウント ID を使用して accountId を生成します(必要な場合)。ID 規則のサンプルは、sample/identityRules.xml にあります。

アカウントタイプの設定と管理については、リソースのページに関連するオンラインヘルプを参照してください。

注 アカウントのタイプの管理を有効にするには、Identity Manager で 1 つ以上の ID 規則を作成する必要があります。

アカウントタイプを割り当てる

アカウントタイプを定義すれば、それらをリソースに割り当てることができます。Identity Manager は、アカウントタイプの各割り当てを個別のアカウントとして処理します。その結果、ロール内の個別の割り当てに異なる属性セットを設定することができます。

リソースごとに 1 つのアカウントがある場合と同様、割り当ての数に関係なく、特定のタイプのすべての割り当てで 1 つのアカウントだけが作成されます。

1 つのリソースで任意の数の異なるタイプのアカウントにユーザーを割り当てることができますが、各ユーザーには 1 つのリソースで特定のタイプのアカウントを 1 つ割り当てることができます。この規則の例外は、組み込みの「デフォルト」のタイプです。1 つのリソースで、任意の数のデフォルトタイプのアカウントをユーザーに割り当てることができます。ただし、この方法はお勧めできません。フォームと画面でアカウントを参照するときにあいまいになるためです。

アカウントタイプを削除する

アカウントタイプは、Identity Manager 内のほかのオブジェクトから参照されなくなるまで削除できません。アカウントタイプの名前を変更することもできません。

複数アカウントユーザーと調整

相関ロジックに、リソースアカウントのタイプが示されていることがあります。調整時に、自動リンクではアカウントタイプを把握する必要があります。これは、この操作の実行にフォームが使用されないためです。

調整で、あるリソースアカウントの **LINK** 応答を実行すると、通常はそのアカウントがデフォルトのアカウントタイプとしてユーザーに割り当てられます。ただし、複数アカウントユーザー用に設定されているリソースでは、これが常に適切だとは限りません。特に、検出されたアカウントは特定のアカウントタイプに属している可能性があります。そのようにユーザーにリンクする必要があります。適切なアカウントタイプを割り当てるには、使用するアカウントタイプを調整に通知する必要があります。**Identity Manager** は、この情報を相関規則の結果の一部として返すことにより、通知を行います。

CorrelationPlan は、相関規則の結果を拡張して、アカウントタイプが識別できるようにします。したがって、アカウントが特定のアカウントタイプである場合、相関規則は **CorrelationPlan** オブジェクトを返す必要があります。ただし、**CorrelationPlan** は標準的なリソース用に使用することもできます。特に設定しないかぎり、**CorrelationPlan** はデフォルトのアカウントタイプを示します。

CorrelationPlan を相関規則の結果として使用する例および詳細については、`sample/reconRules.xml` および **Javadoc** を参照してください。

アカウントタイプとユーザー用画面

アカウントタイプをユーザーに割り当てると、**Identity Manager** はそのアカウントタイプと `accountId` を使用できるようにします。「ユーザー」、「有効化」、「無効化」、「プロビジョン解除」画面などのユーザー用画面を操作するときは、対処方法を示した次のガイドラインに従ってください。

- デフォルトタイプのアカウントを示すときは、**NULL** 値を使用します。デフォルトタイプのアカウントを参照するときは、`accounts[corp-ad]` などのリソース名を使用します
- 特定のタイプのアカウントを参照するときは、リソース名の代わりにタイプ修飾した名前を使用します。タイプ修飾したリソース名の形式は次のとおりです。

<リソース名>|<アカウントのタイプ>

リソース **corp-ad** の **Admin** タイプのアカウントのアカウントデータを参照するには、`accounts[corp-ad|Admin]` と参照します。

Sun Communications Services 配備シナリオ

Sun Communications Services 配備シナリオでは、Sun Communications Services リソースにユーザーアカウントが定義される環境で Identity Manager を配備する方法を説明します。このシナリオは、Sun Communications Services が管理対象の唯一のリソースであることを前提としています。「[HR データベース / Active Directory 配備シナリオ](#)」では、複数のリソースに Identity Manager を配備するプロセスを説明します。

機能

Sun Communications Services シナリオには、Identity Manager にロードされる約 50 の XML ファイルが含まれています。これらのファイルによって、次のタイプのオブジェクトが作成および設定されます。

- ユーザーフォーム
- 規則
- アカウントポリシー
- 管理者ロール
- ロール
- タスク定義とスケジュール
- タスクテンプレート
- ログインアプリケーションとログインモジュールグループ
- Sun Communications Services リソース
- 電子メールテンプレート

さらに、シナリオでは、いくつかの JSP ファイルと、ディレクトリサーバーにロードすることができるサンプル LDIF ファイルも用意しています。LDIF ファイルにはいくつかのユーザーが定義され、その一部はマネージャーです。

シナリオの最も重要な機能には、次のものがあります。

- マネージャーは部下のアカウント属性データを編集できます。マネージャーは、管理者インタフェースへのアクセスに必要な機能を明示的に許可されてはいません。ただし、マネージャーは、リンクされた部下のリストをユーザーインタフェースから表示することができ、このリストからマネージャーは部下のアカウント属性にアクセスできます。
- ユーザーがユーザーインタフェースからアカウント属性データを編集すると、そのユーザーのマネージャーはその変更を承認して変更を有効にする必要があります。これはオプションであり、設定された場合のみ機能します。
- ユーザーはリソースへのアクセスをリクエストできます。利用可能なリソースのリストは動的に生成されます。直接またはロールを通じてすでにユーザーに割り当てられているリソースは、表示されません。
- デフォルトの秘密の質問に対する回答は、Sun Communications Services 属性から取得されます。

前提条件

Sun Communications Services シナリオでは、Identity Manager と次の製品との統合を説明します。

- Sun Java System Directory Server 5 2005Q1
- Sun Java System Messaging Server 6 2005Q1

ほかのディレクトリサーバーおよびメールサーバーは、提供されているスクリプトをカスタマイズしないかぎり、このシナリオでは動作しません。

また、Directory Server と Messaging Server では、同じ Sun Communications Services ベースコンテキストを使用する必要があります。

シナリオをロードおよび設定する

次の手順を実行して、環境を設定します。

警告 このシナリオは既存の本稼働環境にはロードしないでください。シナリオによっていくつかのシステム属性が再設定されます。

手順 1: Directory Server を設定する

1. **IdM User** という名前のユーザーをベースコンテキストに作成します。たとえば、`cn=IdM User,dc=example,dc=com` のようにします。このユーザーは、ディレクトリにプロビジョンするために **Sun Communications Services** リソースアダプタで使用されます。
2. ベースコンテキスト (`dc=example,dc=com`) のアクセス制御を変更して、**IdM User** にフルコントロール (読み取り、比較、検索、書き込み、削除、追加) を許可します。
3. **Directory Server** で更新履歴ログを有効にします。**Directory Server** 管理コンソールの「設定」タブから、「データ」をクリックし、右側にある「レプリケーション」タブをクリックします。「更新履歴ログの有効化」チェックボックスをチェックします。**Directory Server 5.0** 以降を使用している場合は、管理者コンソールの「旧バージョン形式の更新履歴ログプラグイン」も有効にします。

ここで、**Directory Server** を再起動します。

4. `cn=changelog` のアクセス制御を変更して、**IdM User** に読み取りと検索を許可します。
5. `$WSHOME/sample/scenarios/scenario1/schema.ldif` ファイルを **Directory Server** にインポートします。
6. `$WSHOME/sample/scenarios/scenario1/scenario1Data.ldif` ファイルを **Directory Server** にインポートします。

手順 2: Messaging Server を設定する

注

Sun Communications Services シナリオは、メールサーバー用のスキーマ 1 を使用してテストされています。このシナリオは、スキーマ 1.5 (互換モード) およびスキーマ 2 で使用できます。

MessageServerroot/config/option.dat ファイルの次のオプションを追加または修正して、メール転送と自動返信が機能するようにします。

```
DELIVERY_OPTIONS=*mailbox=$M%$Y$2I$_+$2S@ims-ms-daemon, ¥
&members=*, ¥
*native=$M@native-daemon, ¥
/hold=@hold-daemon:$A, ¥
*unix=$M@native-daemon, ¥
&file=+$F@native-daemon, ¥
&@members_offline=*, ¥
program=$M%$P@pipe-daemon, ¥
#forward=**, ¥
*^!autoreply=$M+$D@bitbucket
```

手順 3: シナリオをロードする

次の手順を使用して、Sun Communications Services シナリオを Identity Manager にロードします。

1. タスクテンプレートを表示および操作できるように Identity Manager を設定します。\$WSHOME/config/Waveset.properties ファイルを編集し、gui.enableTaskTemplateEditor=true を設定します。
2. Configurator で Identity Manager にログインします。
3. Identity Manager メニューバーから、「設定」をクリックし、「交換ファイルのインポート」をクリックします。

注

ブラウザは、Identity Manager サーバーから起動するか、または Identity Manager サーバーの idm/sample ディレクトリに含まれているファイルにアクセスする必要があります。

- 名前を入力するか、\$WSHOME/sample/scenarios/scenario1 /scenario1.xml ファイルを参照し、「インポート」をクリックします。

ファイルが正常にインポートされたことを示すメッセージが Identity Manager に表示されます。

- \$WSHOME/sample/scenarios/scenario1/jsp/ の内容を \$WSHOME/user/ にコピーします。これらの JSP ファイルは、マネージャーが部下のデータを編集できるようにし、ユーザーがリソースへのアクセスをリクエストするのに必要です。

手順 4: Sun Communications Services リソースアダプタを設定する

シナリオをロードすると、Sun JSCS リソースという名前のリソースが作成されます。このリソースは、環境に合わせて設定する必要があります。

次の手順を使用して、Sun JSCS リソースの最小限の設定を行います。追加の設定が必要になることがあります。

- Identity Manager メニューバーから、「リソース」をクリックします。

リソースはタイプ別にグループ化されており、名前の付いたフォルダごとにリスト内に表されています。階層表示を展開して現在定義されているリソースを表示するには、リソースタイプフォルダをダブルクリックするか、フォルダの隣にある「+」（プラス記号）をクリックします。

- Sun JSCS リソースアイコンを右クリックし、ポップアップメニューから「編集」>「リソースウィザード」オプションを選択します。「リソースパラメータ」ページが表示されます。

- 最小限、次のフィールドを環境に合わせて設定します。

- ホスト
- ユーザー DN
- パスワード
- ベースコンテキスト

注 「ユーザー DN」フィールドは、デフォルトの「cn=IdM User,dc=example,dc=com」になります。この値は、「[手順 1: Directory Server を設定する](#)」で作成したユーザーに変更してください。

「次へ」をクリックして、「アカウント属性」ページを表示します。

4. この時点では、「アカウント属性」ページの設定は必要ありません。アカウント属性の詳細については、「[Sun JSCS リソース属性](#)」を参照してください。

「次へ」をクリックして、「アイデンティティテンプレート」ページを表示します。

5. アイデンティティテンプレートは、ユーザーのアカウント名構文を定義します。アイデンティティテンプレートの値を更新します。Sun JSCS リソースの場合、この値の一般的な形式は次のようになります。

```
uid=$accountId$,ou=OrganizationUnit,BaseContext
```

「保存」をクリックします。

6. ここで、アダプタの **ActiveSync** 機能を無効にします。これを行うには、Sun JSCS リソースアイコンを右クリックし、「起動タイプ」パラメータの値を「無効」に変更して、「編集」>「Active Sync の動作設定の編集」を選択します。

また、「プロキシ管理者」を **Configurator** またはリソースで変更を処理するのに必要な機能を持つほかの管理者に変更します。

「保存」をクリックします。

7. **ActiveSync** 機能を無効にするために、アプリケーションサーバーを再起動します。

手順 5: 電子メールテンプレートを設定する

Sun Communications Services シナリオでは、Identity Manager は次のような場合に電子メールを送信します。

- ユーザーまたはマネージャーがアカウント属性の値を変更した場合。
- **ActiveSync** が Sun JSCS リソースでユーザーを検出した場合。

次の手順を使用して、Identity Manager 電子メールテンプレートを設定します。

1. メニューバーから、「設定」をクリックし、「電子メールテンプレート」をクリックします。
2. Sun JSCS **ActiveSync** アカウントの作成の通知のリンクをクリックします。
3. 「SMTP ホスト」および「From」フィールドの値を変更します。必要に応じて、その他のフィールドを編集します。
4. 「保存」をクリックします。
5. アカウントの更新の通知電子メールテンプレートおよび実装するその他のテンプレートで、手順 2 から 4 を繰り返します。

Sun Communications Services シナリオをロードして必要な設定を行ったあとは、実行する必要のある特定の手順はありません。ただし、この章の残りの部分に示されている手順を使用して、シナリオで提供される機能を調べてみることをお勧めします。特に指定がないかぎり、このシナリオに記載されるファイルはすべてデフォルトでロードされます。

サンプルディレクトリの概要

scenario1Data.ldif ファイルにはいくつかのデフォルトユーザーが定義され、その一部はマネージャーです。次の表には、各ユーザーを一覧し、存在する場合はユーザーのマネージャーと部下を示します。

表 D-1 Sun Communications Services シナリオのユーザー、マネージャー、および部下			
ユーザー ID	名前	マネージャー	部下
ah1340	Anthony Harris	John Thomas	Joseph Smith
js1260	Joseph Smith	Anthony Harris	Renee Lebec
tb926	Theodore Benjamin	Bill Cady	なし
ed1096	Eric D 'Angelo	Bill Cady	なし
rl891	Renee Lebec	Joseph Smith	Bill Cady
jt509	John Thomas	Josie Smith	Anthony Harris
rb945	Robert Blinn	なし	Josie Smith
js1436	Josie Smith	Robert Blinn	John Thomas
sp486	Stephen Perkins	Bill Cady	なし
bc1165	Bill Cady	Renee Lebec	Theodore Benjamin Eric D'Angelo Stephen Perkins

このシナリオのエンドユーザーの機能を表示するには、表に一覧されているユーザー ID を使用してユーザーインタフェースにログインします。このシナリオのすべてのユーザーのパスワードは **hello** です。

Identity Manager オブジェクトを作成および設定する

この項では、Sun Communications Services シナリオをロードしたあとで、選択した Identity Manager オブジェクトを作成および設定する方法について説明します。これらのオブジェクトは、このシナリオをサポートするのに必要です。

Sun JSCS リソース属性

scenario1Data.ldif ファイルで定義したユーザーには、inetOrgPerson などの標準的なオブジェクトクラスの属性が含まれます。追加の属性は、inetmailuser などの Messaging Server オブジェクトクラスで定義されます。Messaging Server 以外のメールサーバーで Sun Communications Services シナリオを使用する場合は、Sun JSCS リソースのスキーママップを更新する必要があります。

次の表では、Sun Communications Services シナリオ用に追加されたアカウント属性を一覧します。このシナリオ用にサポートされているサーバー以外のディレクトリサーバーやメールサーバーを使用する場合は、アカウント属性を変更する必要があることがあります。

表 D-2 Sun Communications Services シナリオのアカウント属性

Identity Manager 属性	Sun JSCS 属性	説明
manager	manager	ユーザーのマネージャーの識別名。
dn	entrydn	ユーザーの識別名。
employeeNumber	employeenumber	組織内の従業員を数値で示します
roomNumber	roomnumber	ユーザーのオフィスの電話番号。
building	extendedaddress	ユーザーのオフィスが入っている建物の名前またはコード。これは標準的な Sun JSCS 属性ではありません。
carLicense	carlicense	自動車の登録番号 (ナンバープレート)
ldapGroups	ldapGroups	Sun JSCS ユーザーのグループのリストを含む文字列。
email	mail	ユーザーの電子メールアドレス (主)。
alternateEmail	mailalternateaddress	1 つ以上の追加電子メールアドレス。
mailDeliveryOption	maildeliveryoption	1 つ以上の autoreply、hold、mailbox、native、または unix。

表 D-2 Sun Communications Services シナリオのアカウント属性 (続き)

Identity Manager 属性	Sun JSCS 属性	説明
mailHost	mailhost	この受信者に送信されたメッセージの最終宛先となる MTA の完全修飾ホスト名。
mailForwardingAddress	mailforwardingaddress	受け取ったメッセージを転送する宛先のアドレス。
inetUserStatus	inetuserstatus	グローバルサーバーアクセスに関するユーザーのアカウントのステータス。値は、active、inactive、および deleted です。
mailQuota	mailquota	ユーザーのメールボックスに許可されたディスク容量 (バイト単位)。
mailAutoReplySubject	mailautoreplysubject	自動返信応答の件名テキスト。
mailAutoReplyText	mailautoreplytext	受信者のドメイン内のユーザーを除くすべての送信者に送信された自動返信テキスト。
mailAutoReplyTextInternal	mailautoreplytextinternal	受信者のドメインから送信者に送信された自動返信テキスト。
vacationStartDate	vacationstartdate	不在返信開始日時 (YYYYMMDDHHMMSSZ 形式)。Z はリテラルです。
vacationEndDate	vacationenddate	不在返信終了日時。 vacationStartDate と同じ形式を使用します。
mailAutoReplyMode	mailautoreplymode	echo または reply。

また、Directory Server で次の Sun JSCS 属性にクエリーが行われますが、これらの属性は Identity Manager にマップされません。

- ismanager。この属性が存在し、「Y」の値を持つ場合、ユーザーはマネージャーと見なされます。
- manager。この属性の値は、その特定のエントリを管理するマネージャーのレコードの dn に設定されます。したがって、部下には、そのマネージャーの dn に設定されたマネージャー属性があります。

管理者ロール

AdminRole-User.xml ファイルは、2つの管理者ロールを作成します。

「User」管理者ロールは、AdminRole-User.xml ファイルで定義されます。この管理者ロールは、ユーザーアカウント管理者機能を「My Employees」組織に割り当て、マネージャーが自分の部下のデータを編集できるようにします。また、「User」管理者ロールは、承認者機能を最上位の組織に許可します。

ロール

「Mail Users」ロールは、Sun JSCS リソースをユーザーに割り当てます。このロールは、Role-MailUser.xml ファイルで定義されます。

ユーザーがディレクトリに存在し、そのディレクトリに inetuser、ipuser、inetmailuser、inetlocalmailrecipient、および userpresenceprofile オブジェクトクラスが含まれている場合、そのユーザーは、Identity Manager で「Mail Users」ロールが作成されるとそのロールを割り当てられます。1つ以上のこれらのオブジェクトクラスが存在しない場合、そのロールはユーザーに適用されません。

ユーザーが Identity Manager から作成され、「Mail Users」ロールが有効になっていると、調整時に電子メールアドレスが Messaging Server に作成されます。ユーザーにはメールアドレスがあるため、Is Mail User (Rule-IsMailUser.xml) の規則を渡します。

Sun Communications Services シナリオは、メールアドレスを設定してはいけないディレクトリで定義されるユーザーが存在する可能性があるということを前提としています。ディレクトリ内のすべてのユーザーに電子メールアドレスを設定する場合は、次のように操作します。

- inetuser、ipuser、inetmailuser、inetlocalmailrecipient、および userpresenceprofile オブジェクトクラスを、アダプタの管理対象のオブジェクトクラスのリストに追加します。
- 「Mail Users」ロールのプロパティを変更し、objectClass リソースアカウント属性を「None」に設定します。

オブジェクトグループ

シナリオでは、次のオブジェクトグループを作成します。

ユーザー - 調整したユーザーを配置する仮想組織。この組織は、ObjectGroup-Users.xml ファイルで定義されます。

My Employees - この仮想組織は、「User」管理者ロールに割り当てられます。この組織は、「Get My Subordinates」規則を呼び出します。この組織は、ObjectGroup-MyEmployees.xml ファイルで定義されます。

ログインアプリケーションとログインモジュールグループ

Identity Manager は、パススルー認証を使用して、エンドユーザーページの Sun JSCS パスワードの有効性を解決します。管理者インタフェースは、Identity Manager ユーザーのパスワードを使用します。この機能を有効にするために、シナリオでは、次のファイルを使用してユーザーインタフェースの Sun JSCS ログインモジュールグループを作成します。

- LoginApp-UserInterface.xml
- LoginModGroup-SunJSCSLoginModuleGroup.xml

一般的な設定

次のファイルは、Sun Communications Services シナリオのその他の点を設定します。

Configuration-UserUIConfig.xml - マネージャー属性を SummaryAttrNames および QueryableAttrNames 属性に追加し、管理者がマネージャーの識別名を指定することによってユーザーを検索できるようにします。

Configuration-UserExtendedAttributes.xml - マネージャーをユーザーオブジェクトの画面に追加します。

Configuration-CustomMessageCatalog.xml - ユーザーインタフェースに表示される文字列を定義します。

次の表では、フォームとプロセスのマップの変更点を一覧します。

表 D-3 フォームとプロセスのマッピング

プロセス / フォーム	タイプ	新しい値	元の値
プロセス	createUser	Create User Template	Create User
プロセス	updateUser	Update User Template	Update User
フォーム	endUserForm	Sun JSCS End User Form	End User Form
フォーム	endUserMenu	Custom End User Menu	End User Menu
フォーム	userForm	Custom Tabbed User Form	Tabbed User Form

ユーザーアカウントを作成する

ユーザーアカウントの作成は、管理者が実行する最も重要なアクティビティの1つです。Sun Communications Services シナリオでは、大部分のユーザーアカウントは Identity Manager が最初にディレクトリサーバーからアカウントを読み込むときに作成されます。ただし、ほかのアカウントは、ActiveSync によって読み込まれるか、または Identity Manager で管理者によって作成されます。この項では、これらのそれぞれの機構でユーザーを作成するプロセスについて説明します。

調整を使用してユーザーを作成する

調整を実行する前に、調整ポリシーを設定する必要があります。また、調整を実行するユーザーにユーザーフォームを割り当てる必要があります。

調整ポリシー

調整ポリシーによって、管理者は調整を実行するサーバーを選択し、調整の実行回数と実行時間を決定し、調整中に発生する状況ごとの応答を設定します。アカウント属性に対してネイティブに (Identity Manager を介せずに) 行われた変更を検出するように調整を設定することもできます。

Identity Manager にはデフォルトのグローバル調整ポリシーが用意されており、どのリソースにも継承できます。ただし、Sun Communications Services シナリオでは、デフォルトのポリシーにいくつかの変更を行う必要があります。これらの変更は Configuration-ReconcileConfiguration.xml ファイルで定義されています。

デフォルトのポリシーに対して行う重要な変更は2つあります。

- プロキシ管理者を調整管理者に設定します。調整応答が実行されるときは、この管理者として実行されます。調整管理者は、「User Form with Questions」ユーザーフォームでユーザーを作成するように設定されています。調整管理者としてログインする必要がある場合、パスワードは **reconadmin** です。

別のユーザーを使用してこの機能を実行する場合は、デフォルトのユーザーフォームを「User Form with Questions」に割り当てます。また、ユーザーが「Bulk Account Administrator」機能を持つようにしてください。

- UNMATCHED の状態で、リソースアカウントに基づいて新しいユーザーを作成するように設定します。Identity Manager は、対応するアカウントが Identity Manager にないリソースでアカウントを検出すると、新しい Identity Manager ユーザーを作成します。

ユーザーフォーム

調整に使用するユーザーフォーム（「User Form with Questions」）は、UserForm-UserFormWithQuestions.xml で定義されます。このユーザーフォームは、調整操作とリソースからの読み込み操作にのみ使用するように作られています。

デフォルトのユーザーフォームには、アカウントデータの読み込みには必要のないオーバーヘッドが含まれています。「User Form with Questions」フォームは、次の項目が定義されるように無駄な部分が省かれています。

- ユーザーのグローバルフルネーム。ユーザーの姓と名を連結することによって確定されます
- ユーザーの組織。Top:User に設定されます
- ユーザーの秘密の質問に対する回答。詳細については、「[秘密の質問に対する回答を設定する](#)」を参照してください。

リソースから読み込まれるアカウント情報はユーザーオブジェクトに格納され、ユーザー画面からアクセスできます。その結果、各ユーザーの詳細情報をユーザーフォームで参照する必要がなくなります。「User Form with Questions」フォームには、調整プロセスを高速化するのに役立つ最小限のデータが含まれています。何千ものユーザーが存在する本稼働環境では、調整プロセスに何時間もかかる可能性があります。したがって、不要なフィールドや属性をすべて削除すれば、調整プロセスが高速化されます。

フルネームと認証回答は取得された属性であるため、ユーザーフォームに残しておきます。組織は、ユーザーオブジェクトを配置する Identity Manager 仮想組織を指定します。

ワークフロー

Sun Communications Services シナリオでは、調整を介してユーザーアカウントを作成するときに、標準的なユーザー作成ワークフローを使用します。

テストの手順

Sun Communications Services アダプタを設定すると、調整を実行して、ユーザーデータを Identity Manager に移植できます。

Identity Manager に Configurator でログインし、Sun JSCS リソースを右クリックして「ただちに完全調整」オプションを選択し、完全調整を開始します。調整が完了すると、試行されたすべての調整のステータスが表示されます。

ActiveSync を使用してユーザーを作成する

ActiveSync は、リソースアカウントデータを Identity Manager に読み込む別の機構です。ActiveSync はリソースに対する変更を「待機」(ポーリング)し、差分変更をリアルタイムで検出します。ActiveSync は変更を検出するように設計されているため、アカウント情報ははじめて Identity Manager に読み込むときには使用しないでください。

Sun Communications Services シナリオでは、Sun JSCS からの変更を Identity Manager に反映させることができます。新しいユーザーが ActiveSync 機能で検出されると、その新しいユーザーはただちにエンドユーザーページにログインして自分のアカウントを管理できます。

シナリオでは、一時パスワードを作成して電子メールで新しいユーザーに送信することもできます。これは 2 つの理由で必要です。第 1 に、パスワードなしで Sun JSCS でアカウントを作成することが可能なためです。したがって、ユーザーは Identity Manager にログインできないことになります。第 2 に、パスワードは通常 Sun JSCS からは読み取られません。つまり、Sun JSCS ActiveSync をユーザーのパスワードの信頼性の高いアイデンティティ情報の源泉にする場合、Identity Manager はほかのリソースのパスワードを Sun JSCS パスワードと同期することができません。パスワードをすべてのリソース間で一貫性のあるものにするために、一時パスワードを使用します。ユーザーは一時パスワードを使用してログインし、すべてのリソースパスワードを変更します。

ユーザーフォーム

Sun JSCS ActiveSync フォームは、ActiveSync 属性を Identity Manager ユーザー属性に変換します。一時パスワード (change12345) はこのフォームで作成されます。

viewOptions.Process フィールド (下記参照) は、「Sun JSCS ActiveSync Create User」および「Sun JSCS ActiveSync Update User」ワークフローをそれぞれ作成および更新のために実行するようにします。その他の操作 (削除など) の場合は、通常のユーザーワークフローが実行されます。

コード例 D-1 viewOptions.Process フィールド

```

<Field name='viewOptions.Process'>
  <Comments>Use the optimized create and update workflows.</Comments>
  <Expansion>
    <switch>
      <ref>feedOp</ref>
      <case>
        <s>create</s>
        <s>Sun JSCS ActiveSync Create User</s>
      </case>
      <case>
        <s>update</s>
        <s>Sun JSCS ActiveSync Update User</s>
      </case>
      <case default='true'>
        <null/>
      </case>
    </switch>
  </Expansion>
</Field>

```

ワークフロー

TaskDefinition-SunJSCSActiveSyncCreateUser.xml ファイルは、ActiveSync を介してユーザーを作成するときに実行するワークフローを定義します。このワークフローは、リソースでユーザーが検出されると実行されます。

このワークフローは、2 つの操作にまとめられました。最初の操作は、Identity Manager でのアカウントのプロビジョンです。このワークフローでは次に、一時パスワードを記載した電子メール通知が新しく作成されたユーザーに送信されます。

また、resultLimit ワークフロー属性は 0 に設定されます。これも、リポジトリの結果のワークフローの速度を上げるのに役立ちます。

電子メールテンプレート

EmailTemplate-SunJSCSActiveSyncAccountCreationNotification.xml ファイルは、一時パスワードを新しく作成されたユーザーに送信する電子メールテンプレートを定義します。

テストの手順

前提条件

Sun JSCS リソースを ActiveSync 用に設定する必要があります。「手順 4: Sun Communications Services リソースアダプタを設定する」で、ActiveSync は無効にされました。Sun JSCS リソースを右クリックし、「編集」>「Active Sync ウィザード」を選択して、Active Sync ウィザードを実行します。

1. 「同期モード」ページで、「入力フォーム」フィールドを「Sun JSCS ActiveSync Form」に設定します。「次へ」をクリックします。
2. 「Active Sync の動作設定」ページで、次の操作を行います。
 - 「起動タイプ」を「自動」または「手動」に変更します。
 - 「プロキシ管理者」を **Configurator**、またはリソースで変更を処理するのに必要な機能を持つほかの管理者に設定します。
 - ログファイルを書き込むディレクトリを「ログファイルパス」で指定します。
 - 5 分などの値を「ポーリング間隔」フィールドに割り当てます。「次へ」をクリックします。
3. 「Active Sync の一般設定」ページで、「変更者フィルタ」フィールドを「cn=IdM User」に設定します。これにより、**Identity Manager** からリソースにプロビジョンされた変更が Active Sync イベントとして再処理されないようにします。
「保存」をクリックします。

Identity Manager を再起動して、変更を有効にします。

手順

次の手順を使用して、ActiveSync リソースでアカウントが検出されたときに **Identity Manager** が新しいユーザーを作成することをテストします。

1. ディレクトリサーバーのネイティブツールを使用して、新しい **inetOrgPerson** をベースコンテキストディレクトリに作成します。**Identity Manager** を使用してユーザーを作成しないでください。
2. ActiveSync 変更が **Identity Manager** に読み込まれるまで、「ポーリング間隔」フィールドで指定された時間待ちます。状況によっては、ActiveSync を起動する必要がある場合があります。
3. **Identity Manager** で、「アカウント」タブをクリックします。ユーザーが「ユーザー」組織に追加されています。
4. 新しいユーザーのアカウント ID と一時パスワードを使って、ユーザーインタフェースにログインします。
5. 「パスワードの変更」リンクを使って、ユーザーのパスワードを変更します。

最終的に、Sun JSCS の外部で作成されたユーザーに Identity Manager のアカウントが与えられます。一時パスワードは Sun JSCS リソースで設定され、その一時パスワードはユーザーに通知されます。

Identity Manager を使用してユーザーを作成する

Sun Communications Services シナリオには、カスタマイズしたユーザーフォーム、ワークフロー、タスクテンプレート、および通知が必要です。これらの点について詳細に説明します。

ユーザーフォーム

次のユーザーフォームは、新しいユーザーの作成に使用します。

- カスタマイズした Tabbed User Form
- カスタマイズした Dynamic SunJSCS User Form

カスタマイズした *Tabbed User Form*

Sun Communications Services シナリオでは、Tabbed User Form がカスタマイズした Tabbed User Form に置き換わります。標準的な Tabbed User Form は、「ユーザーの作成」および「ユーザーの編集」ページのレイアウトを制御します。これらのページにはいくつかのナビゲーションタブがあり、管理者はそれらのタブを使ってユーザーの ID、割り当て、機能、およびリソース属性を設定または表示できます。カスタマイズした Tabbed User Form も、MissingFields の代わりに

UserForm-CustomDynamicUserForms.xml ファイル (カスタマイズした動的ユーザーフォーム) の「リソースタブ」へのフィールド参照を使用することを除けば、基本的に同じです。これにより、直接またはロールを通じてユーザーに Sun JSCS リソースが割り当てられている場合に、Sun JSCS タブが生成されます。次に、カスタマイズした動的ユーザーフォームは、カスタマイズした Dynamic SunJSCS User Form を参照します。

置き換わった Tabbed User Form を登録するには、userForm フォームマッピングを変更する必要があります。シナリオでは、自動的に userForm 値がカスタマイズした Tabbed User Form に変更されます。

カスタマイズした *Dynamic SunJSCS User Form*

カスタマイズした Dynamic SunJSCS User Form は、「ユーザーの作成」または「ユーザーの編集」ページで Sun JSCS 固有の属性を制御します。このユーザーフォームでは、Sun JSCS 固有の属性が「User Attributes」と「Mail Attributes」の2つのセクションに分けられます。

「ユーザー属性」セクションでは、「Manager」フィールドにマネージャーのリストを含むドロップダウンメニューが含まれています。このリストには、「Get Managers」規則を呼び出すことによって値が入力されます。この規則は、`ismanager` 属性が Y に設定されている Sun JSCS リソースのアカウントをチェックします。この規則は、その条件を満たすアカウントごとに `entrydn` 属性を返します。

「Groups」複数選択ボックスは、「Get Sun JSCS Groups rule」規則によって値が入力されます。この規則はリソースにクエリーを行い、Sun JSCS グループのリストを返します。アカウントがすでに Sun JSCS グループのメンバーである場合、グループは「Current Groups」ボックスに配置されます。

「Assignments」ナビゲーションタブで、「Mail Users」ロールがユーザーに割り当てられている場合は、「Mail Account Created」フィールドに「Yes」の値が入ります。Sun JSCS リソースがロールを使わずに直接割り当てられている場合は、「Create Mail Account」ボタンが表示されます。このボタンをクリックすると、ユーザーに「Mail Users」ロールが割り当てられ、ボタンは消えます。

ユーザーフォームは、「Primary Email」および「他の電子メール」フィールドに入力された値の検証を実行します。ディレクトリサーバー上で一致する電子メールアドレスがチェックされます。一致するアドレスが見つかった場合は、エラーメッセージが表示されます。

ワークフロー

Sun JSCS シナリオでは、管理者インタフェースからユーザーが作成されるとタスクテンプレートが起動します。タスクテンプレートは、管理者がカスタマイズしたワークフローを記述してワークフロー動作を設定する代わりに、管理者インタフェースで同じ結果が得られるようにするコンポーネントと製品拡張機能のセットです。

タスクテンプレートには、「タスク」をクリックしてから「タスクテンプレート」サブタブをクリックすることによってアクセスできます。

タスクテンプレートには、次の要素が含まれています。

- **TaskDefinitionRef** - 操作を実行するため、この場合はユーザーを作成するためにどの **TaskDefinition** を起動するかを定義します。多くの場合、標準的なワークフローは **TaskDefinition** として定義されます。
- **FormRef - TaskTemplate** を設定するためのユーザーインタフェースを生成できるフォームへの参照。
- **Variables** - タスクの設定に使用する属性が含まれます。**Variables** についてはここでは説明しません。

次の XML フラグメントは、TaskTemplate-CreateUser.xml ファイルの冒頭を示しています。

コード例 D-2

TaskTemplate-CreateUser.xml ファイルの冒頭

```

<TaskTemplate id='#ID#TaskTemplate:CreateUser' name='Create User
Template' taskType='TaskConfiguration' visibility='invisible'>
  <TaskDefinitionRef>
    <ObjectRef type='ProvisioningTask' name='Create User'/>
  </TaskDefinitionRef>
  <FormRef>
    <ObjectRef type='UserForm' name='Create User Template Form'/>
  </FormRef>
  ...
</TaskTemplate>

```

TaskTemplate には、「Create User Template」という名前が付いています。TaskDefinitionRef は「Create User」という ProvisioningTask を指しています。これは、ユーザーを作成するための標準的なワークフローです。FormRef はユーザーの作成テンプレートフォームを参照します。これは、TaskTemplate でユーザーを作成するプロセスを制御する定義済みのフォームです。

承認

「Create User Template」タスクテンプレートは、Identity Manager でユーザーを作成するときに実行する承認ワークフローを定義します。このタスクテンプレートでは通知またはその他のテンプレート機能が設定されませんが、必要に応じて、このシナリオで実装することができます。

「Create User Template」を有効にするには、次の前提条件を満たす必要があります。

- 「Create User Template」を指すように createUser プロセスマップを更新する必要があります。「設定」をクリックしてから、「フォームおよびプロセスマッピング」サブタブをクリックします。「プロセスマッピング」セクションの下で、createUser の値を Create User から Create User Template に変更します。このマッピングは、Sun Communications Services シナリオですでに設定されています。
- 「Get Managers」規則 (Rule-GetManagers.xml) は、次の XML フラグメントに示すように、ismanager 属性を使用してアカウントがマネージャーに属するかどうかを決定します。

コード例 D-3 ismanager 属性の使用

```
<map>
  <s>searchAttrsToGet</s>
  <list>
    <s>cn</s>
    <s>dn</s>
    <s>ismanager</s>
  </list>
  <s>searchScope</s>
  <s>subTree</s>
  <s>searchFilter</s>
  <s>(&objectclass=inetorgperson)(ismanager=Y)</s>
</map>
```

別の属性を使用してこのステータスを通知する場合は、このカスタマイズした属性に合わせて規則を変更します。このタイプの属性を使用しない場合は、ismanager 属性へのすべての参照を削除します。この場合は、マップの最後の文字列も次のように変更します。

```
<s>(objectclass=inetorgperson)</s>
```

このシナリオでは、ユーザーの作成リクエストには、ユーザーのマネージャー (Identity Manager 管理者ではない) の承認が必要です。このマネージャーは、承認クエリーによって決定されます。このシナリオでは、次のフィールドは事前に設定されています。

- 追加の承認者を決定する方法** - 「クエリー」オプションは、承認者のアカウント ID が Sun JSCS リソースにクエリーを行うことによって決定されることを示します。クエリーのパラメータは、「問い合わせ先のリソース属性」および「比較対象の属性」フィールドから作成されます。
- 問い合わせ先のリソース属性** - リソースの entrydn 属性にクエリーを行います。このクエリーにより、ユーザーのマネージャーが検索されます。
- 比較対象の属性** - user.accounts[LDAP Resource].manager の値を entrydn リソース属性の値と比較します。一致するものがあれば、新しいユーザーの承認者が見つかったことを示します。
- 承認がタイムアウトになるまでの時間** - デフォルト値は 1 日です。マネージャーがユーザー作成のリクエストを承認しない場合、承認リクエストはそのマネージャーのマネージャーに送信されます (「エスカレーション管理者クエリー」は、「承認の管理者のクエリー」と同じ)。承認リクエストをエスカレーションできるのは 1 度だけであることに注意してください。指定された時間内に 2 番目のマネージャーがリクエストを承認しなくても、そのリクエストは 3 番目のマネージャーには転送されません。

- **承認の属性** - 承認情報に表示される属性。新しいユーザーの姓、名、電子メールアドレス、および Sun JSCS グループが承認情報に表示されますが、これらの値は承認者によって変更される可能性があります。

Sun JSCS リソースで承認者が要求されるようにリソースが設定されている場合、その承認はマネージャーが承認リクエストを受け取る前に行う必要があります。

通知

Sun Communications Services シナリオでは、標準的なワークフロープロセスである「Notification Evaluator」を使って通知を処理します。「Notification Evaluator」プロセスは、「Process User Manager」という名前のアクティビティを追加することによって拡張されています。

「Process User Manager」アクティビティは2つの規則を呼び出します。

- **Get Sun JSCS Resource Name** - ユーザーのマネージャーを決定するために接続するリソースを決定します。
- **Get Manager Account Id - Sun JSCS** リソースにクエリーを行い、ユーザーのマネージャー属性の値を取得します。

マネージャーのアカウント ID の値をフェッチするときに、**Identity Manager** は通知の電子メールをマネージャーに送信します。この電子メールテンプレートは、正しく設定する必要があります。

ユーザーアカウントの更新

一般に、アカウントに変更を加えるときに発生するワークフロープロセスは、アカウントを作成するときに発生するワークフロープロセスに似ています。これらのプロセスは似ていますが、既存のユーザーのプロビジョニングは新しいユーザーのプロビジョニングほど複雑ではありません。

調整によって検出される更新

調整で検出されるデータ変更は、ユーザーの作成時と同じ一般的なプロセスで行います。「User Form with Questions」フォームでプロセスを制御します。

詳細については、「[調整を使用してユーザーを作成する](#)」を参照してください。

ActiveSync によって検出される更新

Sun JSCS ActiveSync のユーザー更新ワークフローは、ActiveSync 更新を実行する方法を定義します。ActiveSync を介して変更が検出されると、Identity Manager は reProvision ワークフローサービスをトリガーします。

詳細については、「[ActiveSync を使用してユーザーを作成する](#)」を参照してください。

Identity Manager で実行された更新

Sun Communications Services シナリオでは、ユーザーデータは、管理者が管理者インタフェースから、またはエンドユーザーかマネージャーがユーザーインタフェースから編集できます。このデータの変更方法に関係なく、Identity Manager は同じワークフローを実行して、承認と通知を同じ方法で処理します。ワークフロー、承認、および通知の詳細については、「[Identity Manager を使用してユーザーを作成する](#)」を参照してください。

「Update User Template」タスクテンプレートは、ワークフローを定義します。これは、「Create User Template」タスクテンプレートの省略バージョンです。ほとんどの場合、既存ユーザーの属性の変更は、新しいユーザーの作成ほど急を要しません。したがって、「Update User Template」タスクテンプレートには、これらの機能は含まれていません。

通知は、ユーザーの作成時と同様に機能します。

updateUser プロセスマップは、「Update User Template」を指すように更新する必要があります。このマッピングは Sun Communications Services シナリオですでに設定されていますが、これを複製するには、「設定」をクリックして、「フォームおよびプロセスマッピング」サブタブをクリックします。「プロセスマッピング」セクションの下で、updateUser の値を Update User から Update User Template に変更します。

変更をユーザーインターフェースに実装する

ユーザーインターフェースには、特に管理機能のないユーザー向けの Identity Manager システム画面が制限付きで表示されます。デフォルトでは、ユーザーはユーザーインターフェースを使ってパスワードの変更とアカウント属性の修正ができます。

Sun Communications Services シナリオでは、次の機能を標準的なユーザーインターフェースに追加します。

- デフォルトの秘密の質問を表示します。秘密の質問に対する回答は、inetOrgPerson オブジェクトクラスの carLicense 属性から読み取られます。
- ユーザーは、Sun JSCS グループメンバーシップなどのさまざまなアカウント属性を変更できます。どの値の変更にも、ユーザーのマネージャーの承認が必要です。
- ユーザーは、リソースへのアクセスをリクエストできます。
- ユーザーがマネージャーである場合は、その部下の属性を変更できます。

これらの機能について、次に詳しく説明します。

秘密の質問に対する回答を設定する

ユーザーがパスワードを忘れたり、パスワードをリセットした場合、1つ以上のアカウント秘密の質問に答えて Identity Manager にアクセスすることができます。これらの質問、および質問を管理する規則は、アカウントポリシーで定義されます。

このシナリオの「Default Lighthouse Account Policy」には、デフォルトの秘密の質問「Car license plate number?」が含まれています。最初にアカウントを調整すると、「User Form with Questions」フォームは、inetOrgPerson オブジェクトクラスの carLicense 属性の値を読み込みます。ユーザーフォームは、いくつかの追加の処理を実行して、carLicense の値が秘密の質問の回答として使用されるようにします。

ただし、ナンバープレートの番号は、企業の Sun JSCS サーバーから省略されることがよくあるため、秘密の質問に使う属性として最適ではない場合があります。また、ナンバープレートの番号はセキュリティ保護された情報ではないため、承認されていない人物が簡単に別の人の Sun JSCS パスワードをリセットできます。

読み取りアクセスを IDM ユーザーやディレクトリ管理者などの所有者や高位レベルの管理者に制限する ACI がディレクトリにある属性のほうが適切です。それらの属性が利用可能かどうかは、ディレクトリサーバーによって異なります。

それらの属性が利用できない場合は、ユーザーがナンバープレートの番号に加えて従業員 ID も指定しなければならないようにシナリオを拡張できます。これを行うには、リソースのスキーママップ、アカウントポリシー、および「User Form with Questions」フォームを更新する必要があります。このシナリオも、従業員 ID が公開されている場合は安全ではありません。

リソース

スキーママップは、ユーザーフォーム、規則、またはその他の Identity Manager オブジェクトで使用するすべての属性を定義する必要があります。このシナリオでは、`employeenumber` Sun JSCS 属性はすでに `employeeNumber` ユーザー属性にマップされています。

アカウントポリシー

アカウントポリシーは、「設定」タブをクリックし、「ポリシー」サブタブをクリックして、管理者インターフェースから修正できます。編集するアカウントポリシーを選択して、「従業員 ID を入力してください」などの新しい質問を「二次認証ポリシーオプション」の見出しの下に入力します。

質問を直接 `Policy-DefaultLighthouseAccountPolicy.xml` ファイルに追加するには、次のように質問属性リストを修正します。

```
<Attribute name='questions'>
  <List>
    <Question>Car license plate number?</Question>
    <Question>Enter Employee ID</Question>
  </List>
</Attribute>
```

ユーザーフォーム

「質問付きユーザーフォーム」フォームの `FieldLoop` ブロックを次のように置き換えます。

コード例 D-4 FieldLoop ブロックの置き換え

```
<FieldLoop for='questionName' in='waveset.questions[*].name'>
  <Field name='waveset.questions[%(questionName)].answer'>
    <Expansion>
      <cond>
        <eq>
          <ref>waveset.questions[%(questionName)].question</ref>
          <s>Car license plate number?</s>
        </eq>
      </cond>
    </Expansion>
  </Field>
</FieldLoop>
```

コード例 D-4 FieldLoop ブロックの置き換え (続き)

```

        <ref>global.carLicense</ref>
        <cond>
            <eq>
                <ref>waveset.questions[%(questionName)].question</ref>
                <s>Employee number?</s>
            </eq>
            <ref>global.employeeNumber</ref>
        </cond>
    </cond>
</Expansion>
<Disable>
    <and>
        <neq>
            <ref>waveset.questions[%(questionName)].question</ref>
            <s>Car license plate number?</s>
        </neq>
        <neq>
            <ref>waveset.questions[%(questionName)].question</ref>
            <s>Enter Employee ID</s>
        </neq>
    </and>
</Disable>
</Field>
</FieldLoop>

```

このコードブロックは、認証ポリシーで定義されたそれぞれの質問が <s> 要素で指定されたテキストに一致するかどうかを判別します。質問が一致しない場合、そのフィールドは無効になります。

アカウント属性を変更する

Sun JSCS エンドユーザーフォームは、どのアカウント属性を表示するか、および属性を編集可能にするかどうかを決定します。属性が編集されると、ユーザーのマネージャーはその変更を通知されます。マネージャーは、その変更を承認または却下する必要があります。この機能の実装の詳細については、「[Identity Manager を使用してユーザーを作成する](#)」を参照してください。

Sun JSCS エンドユーザーフォームを使うと、ユーザーは自分がメンバーになっている Sun JSCS グループに対する変更をリクエストできます。Sun JSCS グループを管理するには、ldapGroups 属性をアダプタのスキーママップに追加する必要があります。このシナリオでは、ユーザーのマネージャーがだれであるかを追跡するために、マネージャーと inetOrgPerson オブジェクトクラスの employeeNumber 属性もスキーママップに追加されています。

ユーザーフォームは、「Get Sun JSCS Groups rule」規則を呼び出します。この規則は、ディレクトリサーバーで利用可能な Sun JSCS グループのリストを取得します。

Sun JSCS エンドユーザーフォームには、「マネージャー」および「Manager Name」という名前のフィールドも含まれています。「マネージャー」フィールドには、manager Sun JSCS 属性から返された値が含まれています。ただし、「Manager Name」フィールドは、「Get Manager Name」規則によって取得されます。この規則は、「マネージャー」フィールドに一覧されている Sun JSCS ユーザーを検索し、このユーザーの名 (givenName) と姓 (sn) を取得して、これらの値を連結します。

ユーザーがマネージャーである場合は、部下のテーブルに、部下であるユーザーのユーザー ID、名前、および電子メールアドレスが入力されます。「[マネージャーが部下のデータを編集する](#)」を参照してください。

マネージャーが部下のデータを編集する

Sun Communications Services シナリオでは、マネージャーには管理者インターフェースにログインする機能はありません。ただし、マネージャーは自分の部下の属性を編集できます。これは、シナリオで仮想組織、つまり組織を管理する管理者ロールを作成し、マネージャーと部下の関係が存在するかどうかを判別する規則をいくつか提供しているためです。

仮想組織

「My Employees」組織には、manager Sun JSCS 属性がログインしたユーザーの識別名と一致するユーザーが含まれます。ユーザーがマネージャーでない場合、組織は空になり、そのユーザーはほかのアカウントを操作できません。マネージャーが操作できるのは直属の部下だけであることに注意してください。2 番目のマネージャーは最初のマネージャーのデータを編集できますが、最初のマネージャーの部下のデータは編集できません。

この組織には、「Get My Subordinates」規則が割り当てられています。この規則には、UserMembersRule の authType があり、ユーザーがこの組織のメンバーになれるかどうかを決定します。この規則は次に、「Get Sun JSCS Resource Name」規則と「Get Subordinate Account IDs」規則を呼び出します。「Get Subordinate Account IDs」規則は、「Get Sun JSCS Resource Name」規則に返されたリソースを使用して、そのマネージャーが現在のユーザーである Sun JSCS ユーザーを検索する場所を決定します。「Get Subordinate Account IDs」規則は、ディレクトリで cn、givenname、および dn 属性値を検索し、dn の値を返します。

管理者ロール

AdminRole-User.xml ファイルは、「User」管理者ロールを作成します。この管理者ロールは、自動的にすべての **Identity Manager** ユーザーに与えられる点に特徴があります。さらに、このロールには次の特性があります。

- ユーザーアカウント管理者が割り当てられている。
- ObjectGroup-MyEmployees.xml ファイルで作成される MyEmployees 仮想組織を制御する
- 最上位の組織で利用できる。

ユーザーフォーム

「End User Subordinate」フォーム - このフォームは、部下のセクションが追加されている点を除いて、Sun JSCS エンドユーザーフォームと同様です。このフォームには、マネージャーが部下の名前をクリックしたあとアクセスされます。

JSP ファイル

changesSubUser.jsp ファイルは、「End User Subordinate」フォームの表示を制御します。このファイルは、Sun JSCS エンドユーザーフォームで参照されます。

リソースへのアクセスをリクエストする

Sun Communications Services シナリオでは、リソースアカウントをリクエストする機能をエンドユーザーに与えます。ユーザーがリソースを選択すると、プロセスを進めるためにユーザーが入力できる属性のセットが **Identity Manager** に表示されます。この機能を有効にするために、シナリオにはカスタマイズしたフォーム、規則、および JSP ページが含まれています。

カスタムエンドユーザーメニューユーザーフォーム

シナリオを最初にロードすると、設定スクリプトによって endUserMenu フォームマッピングがエンドユーザーメニューからカスタムエンドユーザーメニューに変更されます。UserForm-CustomEndUserMenu.xml ファイルは、カスタムエンドユーザーメニューフォームを定義します。このフォームは、エンドユーザーがユーザーインターフェースにログインしたときに表示される内容を制御します。

カスタムエンドユーザーフォームは、ユーザーがリソースへのアクセスをリクエストできるようにするページへのリンクを追加する点で、デフォルトエンドユーザーメニューとは異なります。次のフィールドは、「Request Resource Access」リンクを追加します。

```
<Field>
```

```
<Display class='Link'>
    <Property name='name' value='Request Resource Access' />
    <Property name='URL' value='user/requestResources.jsp' />
</Display>
</Field>
```

「Resource Request User」フォーム

デフォルトの管理者インタフェースでは、ユーザーを作成または編集するときに **Tabbed User Form** が表示されます。「割り当て」ナビゲーションタブで、管理者は「個別リソース割り当て」複数選択ボックスを使って1つ以上のリソースをユーザーに割り当てることができます。「Resource Request User」フォームは、この複数選択ボックスで主に構成されています。

ただし、エンドユーザーには、利用可能なリソースのリストを表示する機能はありません。この制限を回避するために、フォームは「Get Unassigned Resources」規則を呼び出します。この規則には、次の2つの重要な要素があります。

- **RunAsUser** 要素は、利用可能なリソースのクエリーに使用するユーザーアカウントを指定します。例では、**Configurator** を使用してこのタスクを実行します。本稼働環境では、リソースのクエリーを行う最小限の機能を持つ管理者アカウントを作成します。
- この規則は、**FormUtil** メソッド `getUnassignedResources` を呼び出します。このメソッドは、ロールを通じてすでにユーザーに割り当てられているリソースの名前を除くすべての利用可能なリソースのリストを作成します。

JSP ファイル

ユーザーがリソースへのアクセスをリクエストできるようにする機能を有効にするには、次の JSP ファイルを `$WSHOME/sample/scenarios/scenario1/jsp/` から `$WSHOME/user/` にコピーする必要があります。

- `requestResources.jsp` - `user/changeAll.jsp` から取得されます。
`requestResources.jsp` ファイルに対して行われた唯一の変更として、結果およびポスト URL の設定といくつかのメッセージカタログ文字列の変更があります。
- `requestResourcesResults.jsp` - `user/changeAllResults.jsp` から修正なしでコピーされます。

変更を管理者インタフェースに実装する

承認ページを修正する

マネージャーがユーザーインタフェースにログインすると、マネージャーに保留中の承認リクエストがある場合は、「インボックスの表示」というラベルの付いたリンクが表示されます。UserForm-ApprovalForm.xml は、変更された属性を表示することでデフォルトの承認フォームを拡張します。

「Find User Form」を修正する

シナリオでは、新しい検索オプションを「ユーザーの検索」ページに追加します。このオプションにより管理者は、そのマネージャーが、指定されたテキストで始まるか、そのテキストを含むか、またはそのテキストに一致するユーザーを検索できます。

この機能を有効にするには、UserUIConfig 設定オブジェクトと 2 つのユーザーフォームの修正が必要です。

UserUIConfig 設定オブジェクト

UserUIConfig 設定オブジェクトは、アカウント検索機能と結果の表示方法を制御します。マネージャーに基づいてユーザーを検索するには、manager 属性をQueryableAttrNames および SummaryAttrNames 要素に追加する必要があります。QueryableAttrNames は検索可能な属性を定義します。SummaryAttrNames はリスト結果に表示可能な属性のことです。シナリオでは、これらの属性が自動的に追加されます。

「User Search Library」ユーザーフォーム

「User Search Library」ユーザーフォームは、manager フィールドを定義します。このマルチパートフィールドには、次のファセットが含まれます。

- 管理者が複数のタイプで検索を実行できるようにするチェックボックス。
- 「マネージャー」のラベル。
- マネージャー属性の参照
- 演算子を選択するためのドロップダウンメニュー（「が次の文字列で始まる」、「含む」、または「は」）。
- 検索対象の文字列を入力するためのテキストボックス。

次のフラグメントは、マネージャーフィールドを定義しています。

コード例 D-5 マネージャーフィールドを定義するフラグメント

```
<Field name='manager'>
  <Field name='query.clauses[clause1].conditions[manager].active'>
    <Display class='Checkbox' />
  </Field>
  <Field>
    <Display class='Label'>
      <Property name='value' value='Manager' />
      <Property name='rowHold' value='true' />
    </Display>
  </Field>
  <Field name='query.clauses[clause1].conditions[manager].attribute'>
    <Expansion>
      <s>manager</s>
    </Expansion>
  </Field>
  <Field name='query.clauses[clause1].conditions[manager].operator'>
    <Display class='Select'>
      <Property name='valueMap'>
        <ref>displayHints.searchOperators</ref>
      </Property>
      <Property name='rowHold' value='true' />
    </Display>
  </Field>
  <Field name='query.clauses[clause1].conditions[manager].value'>
    <Display class='Text'>
      <Property name='rowHold' value='true' />
      <Property name='size' value='20' />
      <Property name='maxLength' value='48' />
    </Display>
  </Field>
</Field>
```

「Find User Form」 ユーザーフォーム

UserForm-FindUserForm.xml ファイルは、デフォルトの「Find User Form」を修正して、フィールド参照を manager フィールドに追加します。このフィールドは、「User Search Library」フォームで定義されます。

HR データベース /Active Directory 配備 シナリオ

このシナリオでは、ユーザーアカウントが人事データベース (たとえば、PeopleSoft) で定義され、IT インフラストラクチャーへのアクセスが Active Directory によって制御されている環境に Identity Manager を配備する方法について説明します。また、このシナリオでは複数のリソースに Identity Manager を配備するプロセスも示します。

この項では、次の内容について説明します。

- 機能
- 前提条件
- シナリオをロードおよび設定する
- 配備の確認

注	このシナリオでは、人事データベースと Active Directory の知識があることを前提としています。追加情報については、ベンダーのマニュアルを参照してください。Identity Manager の追加情報については、『Identity Manager 管理ガイド』を参照してください。
---	--

機能

このシナリオは、多くの一般的な配備状況に適用されます。ユーザーは PeopleSoft で作成され (PeopleSoft は信頼性の高いアイデンティティ情報の源泉)、IT インフラストラクチャーの承認は Active Directory グループメンバーシップに基づいています。

このシナリオには、次の機能が含まれています。

- PeopleSoft は権限 Active Sync ソースです。
- Identity Manager ロールは、Active Directory グループのメンバーシップを決定します。
- PeopleSoft Active Sync イベントは、Identity Manager ロールを割り当てます。このイベントは、Active Directory ユーザーを作成し、PeopleSoft で定義されたユーザーの部署に基づいてグループメンバーシップを割り当てます。このシナリオでは、デフォルトで次のユーザーグループを認識します。
 - 人事
 - エンジニアリング
 - マーケティング
 - 販売部門
- Active Directory グループの所有者は、グループへの新しいメンバーの追加を承認する必要があります。
- Active Directory は、次の属性の信頼性の高いアイデンティティ情報の源泉として機能します。
 - 説明
 - 会社の電話番号
 - 携帯電話番号
 - ファックス番号
- ユーザーの作成と更新は、サービスレベル契約 (SLA) の追跡のために監査されます。このシナリオには、ユーザーの部署に基づいて SLA のタイミングを示すためのデフォルトのレポートが含まれています。
- Active Directory で作成されたアカウントで既存の Identity Manager ユーザー (たとえば、まだ PeopleSoft で作成されていないユーザー) に一致しないアカウントは、承認ページを使って Identity Manager ユーザーにリンクさせることができます。この状況は、新しい従業員が仕事を始めて、人事部が PeopleSoft でアカウントを作成する前にネットワークリソースにアクセスする必要がある場合によく生じます。

内容のサンプル

このシナリオには、Identity Manager に読み込まれる約 30 の XML ファイルが含まれています。これらのファイルは次の場所にあります。

<IDM_HOME>/sample/scenarios/scenario2/

これらのファイルによって、次のタイプのオブジェクトが作成および設定されます。

- タスクテンプレート
- 前提条件 ListGroups および ADUserForm
- 拡張ユーザー属性
- ユーザーを保持する組織
- Active Sync 設定に必要なオブジェクト
 - タスク定義
 - 規則
 - ユーザーフォーム
- サービスレベル契約の監査レポートのサンプル
- リソース
- 規則
- フォーム
- ロール
- システム設定の更新

前提条件

このシナリオを実行するには、環境に次のものが含まれている必要があります。

- PeopleSoft リソースアダプタ -
`com.waveset.adapter.PeopleSoftComponentActiveSyncAdapter`
- Active Directory リソースアダプタ -
`com.waveset.adapter.ActiveDirectoryActiveSyncAdapter`

これらのアダプタの詳細については、『Identity Manager Resources Reference』を参照してください。

シナリオをロードおよび設定する

次の手順を実行して、環境を設定します。

- [Active Directory の設定](#)
- [PeopleSoft の設定](#)
- [Identity Manager の設定](#)

注 このシナリオは既存の本稼働環境にはロードしないでください。シナリオによっていくつかのシステム属性が再設定されます。

Active Directory の設定

1. Active Directory ユーザーとコンピュータアプリケーションを使って、Scenario2 と呼ばれる組織単位 (OU) を Active Directory ドメインに作成します。
2. 上の OU を右クリックし、「新規」>「グループ」を選択して、OU にグループを作成します。例：
 - 人事
 - エンジニアリング
 - マーケティング
 - 販売部門
3. adresowner と呼ばれる OU にユーザーを作成します。
4. 「管理方法」タブを選択し、名前の値を次のように設定して、前述した各グループのプロパティを修正します。

<AD Domain>/Scenario2/adresowner

PeopleSoft の設定

1. PeopleSoft 環境で部署と部署 ID を作成します。たとえば、次のようにします。
 - 人事 (KU001)
 - エンジニアリング (KU017)
 - マーケティング (KU015)
 - 販売とサービス (KU014)
2. 自分の環境で前述の名前または部署番号を変更する必要がある場合は、次の規則のマッピングを修正します。

<IDM_HOME>/sample/scenario2/GetPeopleSoftDepartmentName.xml

Identity Manager の設定

次の手順の詳細については、『Identity Manager 管理ガイド』を参照してください。

1. Identity Manager をインストールし、次のファイルをインポートします。

<IDM_HOME>/sample/init.xml
2. psjoa.jar ファイルを PeopleSoft インストールメディアから次の場所にコピーします。

<IDM_HOME>/WEB-INF/lib ディレクトリ
3. 次の設定ファイルをインポートします。

<IDM_HOME>/sample/scenarios/scenario2/scenario2.xml
4. 自分の環境に合わせて、次の PeopleSoft リソースと Active Directory リソースを設定します。
 - ホスト名
 - ポート
 - ユーザー名
 - パスワード
 - コンテナ
 - LDAP ホスト名
 - アイデンティティテンプレート

5. 必要に応じて、Active Sync のログを有効にします。

Active Sync ウィザードを使って、PeopleSoft リソースと Active Directory リソースの両方の「ログファイルパス」と「ログレベル」を変更します。

6. 各ロールを修正して、Active Directory 設定に一致するように AD グループロール属性を変更します。

たとえば、ロール属性値は、「[Active Directory の設定](#)」の手順 2 で作成したグループの識別名 (DN) に一致させる必要があります。Active Directory リソースの AD グループ属性は、ロールで設定した AD グループ属性を介して設定されます。

7. 「アカウントの作成の承認」電子メールテンプレートの SMTP ホストを有効な SMTP サーバーに変更します。

- a. 「設定 > 「サーバー」に移動して、「デフォルトサーバーの編集」をクリックします。

- b. 「電子メールテンプレート」タブで「デフォルト SMTP サーバー」を選択します。

8. アプリケーションサーバーを再起動します。

9. Identity Manager のすべての PeopleSoft ユーザーを初期化します。PeopleSoft リソースのすべてのアカウントに対して Identity Manager ユーザーを作成するには、「リソースから読み込み」を実行します。

「アカウント」> 「リソースから読み込み」に移動し、PeopleSoft リソースを選択します。「アカウントの読み込み」をクリックして、すべての PeopleSoft アカウントを Identity Manager に読み込みます。

10. Active Sync が PeopleSoft ユーザーへの変更を検出することを確認します。

PeopleSoft リソース用の Active Sync は、コンテナを再起動したときに自動的に開始される必要があります。自動的に開始しない場合は、「リソース」- 「リソースのリスト」ページから、PeopleSoft リソースの「Active Sync の開始」を選択します。

11. Active Directory リソースで完全調整を開始します。

「リソースのリスト」ページで、Active Directory リソースを選択し、「リソースアクション」メニューから「ただちに完全調整」をクリックします。

12. accountId adresowner を使って Identity Manager アカウントを作成します。このアカウントを「ユーザー」組織に割り当てます。承認者機能を割り当て、Top:Users を管理する組織にします。

13. このアカウントを、ここで作成した Active Directory リソースアカウントにリンクさせます。

- a. 「リソース」- 「アカウントインデックスの検査」- 「Windows Active Directory」に移動します。

- b. 次の adresowner アカウントを選択します。

```
(cn=adresowner,ou=Scenario2,dc=example,dc=com)
```

このアカウントは UNMATCHED になり、__UNKNOWN__ を所有者としてリストします。

- c. 右クリックして、「所有者の指定」を選択します。上の手順 12 で作成した Identity Manager アカウントが表示されます。
- d. adresowner をクリックし、アカウントインデックスアプレットで状況が「確認済み」に変わり、所有者が adresowner になっていることを確認します。
- Active Directory グループのメンバーシップの承認は、この Identity Manager アカウントに送られます。

14. Active Directory リソースで Active Sync を開始して、Active Directory からアイデンティティ情報の源泉として信頼性が高い属性を設定します。

Active Directory の Active Sync ウィザードに移動し、スタートアップモードを自動に変更します。Active Sync 設定を保存します。

配備の確認

次の手順を実行してユーザーを設定し、シナリオが正常にインストールおよび設定されたことを確認します。

- [ユーザーの作成および反映](#)
- [ユーザーの修正](#)
- [監査の設定](#)
- [リンクの確認](#)

ユーザーの作成および反映

1. PeopleSoft 環境で新しいテストユーザーを作成します。
たとえば、「ホーム」>「従業員の管理」>「使用」>「雇用」をクリックします。
ユーザーの関連アドレス、プロフィール、ID 情報などを入力します。
2. 「[Active Directory の設定](#)」の手順 2 で作成した Active Directory グループにマップされる部署の 1 つに新しいユーザーを割り当てます。

PeopleSoft の Active Sync で変更が検出されるのを待ちます。
3. Identity Manager 管理者インタフェースに adresowner ユーザーとしてログインします。
4. 「承認」-「承認待ち」に移動します。

リストには、[手順 1](#) で PeopleSoft リソースに作成したユーザーの承認リクエストが含まれています。このリクエストを承認します。
5. Identity Manager からログアウトし、Configurator でログインし直します。
6. 「アカウント」をクリックします。PeopleSoft アカウントの Identity Manager ユーザーが「アカウントリスト」に表示されていることを確認します。
7. Active Directory サーバーにログインします。
8. Active Directory ユーザーとコンピュータツールを使い、[手順 1](#) で PeopleSoft 環境で入力した情報を使って Active Directory アカウントが作成されたことを確認します。
9. 割り当てられた部署に対応する Active Directory グループを表示して、このアカウントがグループのメンバーであることを確認します。

ユーザーの修正

1. Active Directory ユーザーとコンピュータツールを使って、テストユーザーの次の項目を修正します。
 - 説明
 - 会社の電話番号
 - 携帯電話番号
 - ファックス番号
2. Configurator で Identity Manager にログインします。
3. 「アカウント」をクリックします。[手順 1](#) で修正した Active Directory テストユーザー情報が Identity Manager ユーザーデータベースに格納されていることを確認します。

注 次の Active Sync ポーリングが実行されるまで、ユーザーは Identity Manager ユーザーデータベースに表示されません。

監査の設定

このシナリオには、次の監査ワークフローファイルの例が用意されています。これは、エンジニアリンググループのパフォーマンスタイミングを測定するように設計されています。

Report-WorkflowEngineering.xml

この監査には、サービスレベル契約をサポートするシステムパフォーマンスデータが用意されており、必要に応じて修正できます。

この監査例を設定および実行するには、次の手順を実行します。

1. メニューバーから「設定」を選択し、「監査」を選択します。
「監査設定」ページに監査設定グループのリストが表示されます。各グループには、1つ以上のイベントが含まれていることがあります。グループごとに、成功イベント、失敗イベント、またはその両方を記録できます。
2. リストから **WorkflowEngineering** 設定グループをクリックし、「監査設定グループの編集」ページを表示します。このページを使って、システム監査ログに監査設定グループの一部として記録する監査イベントのタイプを選択できます。
3. 「実行」>「レポート」をクリックします。

監査ログレポートを定義するには、「レポートの実行」ページのレポートオプションのリストから「監査ログレポート」を選択します。

4. 「レポートの実行」リストページから **WorkflowEngineering** 監査ログレポートを実行します。

「実行」をクリックすると、保存した条件に一致するすべての結果のレポートが生成されます。レポートには、イベントが発生した日付、実行した操作、および操作の結果が含まれます。

注 レポートをダウンロードして保存し、Excel などの別のアプリケーションで保存したり、開いたりすることができます。レポートをダウンロードするには、「ダウンロード」をクリックします。

リンクの確認

1. Active Directory サーバーにログインします。

2. OU に新しいテストユーザーを作成します。
3. Active Directory Active Sync ポーリングを待ちます。
4. Configurator で Identity Manager にログインします。
5. [手順 2](#) で作成したアカウントの新しい Identity Manager ユーザーを作成します。
目的の PeopleSoft 従業員 ID を新しい Identity Manager ユーザーアカウント ID として使用します。
6. Active Directory で作成したアカウントを、PeopleSoft データベースで作成する前にリンクします。
「承認」ページに移動し、AD リンク管理者ユーザーの承認を表示します。 [手順 5](#) で作成したテストユーザーにアカウントをリンクします。
7. PeopleSoft にログインします。
8. [手順 5](#) で割り当てたのと同じ従業員 ID でテストユーザーを作成します。
9. PeopleSoft Active Sync ポーリングを待ちます。
10. Identity Manager にログインします。
11. テストユーザーに有効な Active Directory および PeopleSoft アカウントがあることを確認します。

索引

A

Active Directory [46, 63, 64](#)
ユーザーとコンピュータ MMC [49](#)
Active Sync
アカウントデータの読み込み [49](#)

C

com.waveset.adapter.
ActiveDirectoryActiveSyncAdapter クラス [160](#)
CSV ファイル [49, 51, 73](#)
customStyle.css [88, 89](#)

D

Directory Server [129](#)

I

Identity Manager
パスワードポリシー [53](#)
ロゴ、置換 [97](#)

J

JSP ファイル [90](#)

L

LDAP [46, 68, 70](#)
LDAP シナリオ
ActiveSync [140](#)
Directory Server [129](#)
Find User Form [155](#)
Messaging Server [130](#)
Tabbed User Form [143](#)
アカウント属性 [151](#)
アカウントの更新 [147](#)
オブジェクトグループ [137](#)
概要 [133](#)
カスタマイズした動的ユーザーフォーム [143](#)
仮想組織 [152](#)
管理者インタフェースの変更 [155](#)
管理者ロール [136](#)
機能 [127](#)
承認 [145](#)
設定 [131](#)
前提条件 [128, 160](#)
調整 [138](#)
通知 [147](#)
電子メールテンプレート [132](#)
秘密の質問 [149](#)
マネージャーによるデータの編集 [152](#)

M

ユーザー ID [133](#)
ユーザーインターフェースの変更 [149](#)
ユーザーフォーム [139](#)
リソースアクセス [153](#)
リソース属性 [134](#)
ロード [130](#)
ロール [136](#)
ログインアプリケーション [137](#)
ワークフロー [144](#)

M

Messaging Server [130](#)
MMC [49](#)

P

PeopleSoft [46, 68, 69](#)

Q

QueryableAttrNames [114](#)

R

reconRules.xml [60](#)
Remedy [68, 72](#)
RepoIndexAttrs [116](#)

S

SAP [46](#)
SecurID [63, 65](#)
Solaris [63, 66](#)
サポート [xiii](#)

パッチ [xiii](#)

style.css [88](#)

SummaryAttrNames オブジェクト [113](#)

System Configuration オブジェクト
カスタマイズ [103](#)
国際化 [117](#)

U

UserUIConfig オブジェクト [112](#)

W

waveset.accountId [56](#)
waveset.organization [56](#)
Waveset.properties ファイル [117, 121](#)
WPMessages.properties ファイル [117](#)
WPMessages_ja.properties [90, 93](#)

X

XML ファイル [49](#)

あ

アカウント ID と一致するユーザー名 [50, 51](#)
アカウント ID ポリシー [53](#)
アカウントインデックス [17](#)
 アカウントのリンク [61](#)
 一括操作 [52](#)
 調整 [52](#)
 ファイルから読み込み [50](#)
 リソースから読み込み [51](#)
アカウント調整 [21](#)
アカウントのリンク
 アカウントインデックスの使用 [61](#)

- 概要 [57](#)
- 自己検索の使用 [61](#)
- 手動 [61](#)
- アダプタ
 - アカウントの無効化の例 [41](#)
 - フォーム処理 [36](#)
- アダプタ、設定 [53](#)
- アダプタの設定 [53](#)

い

- 一括操作、作成 [51,73](#)
- 一括操作の作成 [51,73](#)

お

- オブジェクトグループ [137](#)
- オブジェクト属性 [113](#)

か

- 概要の属性 [113](#)
- 拡張ユーザー属性設定オブジェクト [58](#)
- 確認規則
 - アカウントのリンク [57](#)
 - カスタム [60](#)
 - ファイルから読み込み [50](#)
 - リソースから読み込み [51](#)
- カスタマイズ
 - System Configuration オブジェクト [103](#)
 - ヘッダーとフッター [90](#)
 - ロゴ [97](#)
- カスタマイズファイル [89](#)
- カスタム規則 [60](#)
- カスタム関連キー [58](#)
- 環境、評価 [45](#)
- 管理者ロール [136](#)

き

- 規則
 - カスタム [60](#)

く

- クイックリンク、ログインページに追加 [91](#)

け

- 言語サポート、有効化 [119](#)

こ

- コンマ区切り値ファイル、「CSV ファイル」を参照

さ

- サポート
 - Solaris [xiii](#)

し

- 自己検索 [61](#)
- 承認 [145](#)

す

- スタイルシート、修正 [88](#)
- スタイル設定、デフォルト [89](#)

そ

そ

相関キー、カスタム 58

相関規則

アカウントのリンク 57

カスタム 60

ファイルから読み込み 50

リソースから読み込み 51

属性

オブジェクト 113

概要 113

クエリー可能 114

タイプ 113

ユーザー画面 50, 55

ち

調整 21

LDAP シナリオ 138

概要 49, 52

確認規則 25

相関規則 25

デモンタスク 30

ネイティブ変更の監査 29

ポリシー設定 24, 138

リソーススケジュール 30

ワークフロー 27

調整設定オブジェクト 31

つ

通知 147

て

データの読み込み、シナリオの例 62

データ読み込み

アカウント、作成 55

準備 53

プロセス 48

テキスト属性 89

テキスト、デフォルト 89

デフォルトスタイル設定 89

デフォルトテキスト 89

は

パスワードポリシー 53

ひ

ビュー属性 55

ふ

ファイルから読み込み 48, 49

フォント特性、変更 96

フッター

カスタマイズ 90

バーの色の変更 97

へ

ページタイトルとサブタイトル、変更 92

ヘッダー

カスタマイズ 90

バーの色の変更 97

め

メッセージカタログファイル 120

メッセージ、国際化 118

ゆ

ユーザー画面 [55](#)

 アカウントタイプの参照 [125](#)

 属性 [50](#)

ユーザーフォーム [55, 139](#)

ら

ラベリング演習、サンプル [96](#)

ラベリング演習の例 [96](#)

り

リソース

 最初に読み込むものの選択 [46](#)

 複数アカウントユーザーの割り当て [123](#)

リソースから読み込み [48, 50](#)

リソースタイムアウト設定 [31](#)

ろ

ロール [136](#)

ログインアプリケーション [137](#)

ロゴ、カスタマイズ [97](#)

わ

ワークフロー

 LDAP シナリオ [144](#)

