Novell Identity Manager

3

www.novell.com

2006 年 7 月 21 日

ポリシービルダとドライバのカスタマイズ ガイド

Novell®

保証と著作権

米国 Novell, Inc. およびノベル株式会社は、本書の内容または本書に起因する結果に関して、いかなる表示も行いません。また、本書の商品性、および特定用途への適合性について、いかなる黙示の保証も行いません。米国 Novell, Inc. およびノベル株式会社は、本書の内容を改訂または変更する権利を常に留保します。米国 Novell, Inc. およびノベル株式会社は、このような改訂または変更を個人または事業体に通知する義務を負いません。

米国 Novell, Inc. およびノベル株式会社は、ノベル製ソフトウェアの使用に起因する結果に関して、いかなる表示も行いません。また、商品性、および特定目的への適合性について、いかなる黙示の保証も行いません。米国 Novell, Inc. およびノベル株式会社は、ノベル製ソフトウェアの内容を変更する権利を常に留保します。米国 Novell, Inc. およびノベル株式会社は、このような変更を個人または事業体に通知する義務を負いません。

本契約の締結に基づいて提供されるすべての製品または技術情報には、米国の輸出管理規定およびその他の国の貿易関連法規が適用されます。お客様は、取引対象製品の輸出、再輸出または輸入に関し、国内外の輸出管理規定に従うこと、および必要な許可、または分類に従うものとします。お客様は、現在の米国の輸出除外リストに掲載されている企業、および米国の輸出管理規定で指定された輸出禁止国またはテロリスト国に本製品を輸出または再輸出しないものとします。お客様は、取引対象製品を、禁止されている核兵器、ミサイル、または生物化学兵器を最終目的として使用しないものとします。本ソフトウェアの輸出については、www.novell.co.jp/info/exports/expmtx.html または www.novell.com/ja-jp/company/exports/ もあわせてご参照ください。弊社は、お客様が必要な輸出承認を取得しなかったことに対し如何なる責任も負わないものとします。

Novell, Inc.
404 Wyman Street, Suite 500
Waltham, MA 02451
U.S.A.
www.novell.com

オンラインマニュアル : 本製品とその他の Novell 製品のオンラインヘルプにアクセスする場合や、アップデート版を入手する場合は、www.novell.com/documentation をご覧ください。

Novell の商標

DirXML は、米国 Novell, Inc. の米国およびその他の国々における登録商標です。
eDirectory は、米国 Novell, Inc. の商標です。
Novell は、米国 Novell, Inc. の米国およびその他の国々における登録商標です。
Nsure は、米国 Novell, Inc. の商標です。

第三者の商標

第三者の商標は、それぞれの所有者に属します。

目次

このガイドについて

Novell® Identity Manager 3.0 は、アプリケーション、ディレクトリ、およびデータベース間で情報を共有するためのデータ共有および同期サービスです。このサービスは、分散された情報をまとめてリンクし、ユーザは識別情報の変更時に指定システムを自動的に更新するポリシーを設定できます。

Identify Manager は、アカウントプロビジョニング、セキュリティ、シングルサインオン、ユーザセルフサービス、認証、認可、自動化されたワークフロー、および Web サービスの基盤となります。Identify Manager を使用すると、分散された識別情報を統合、管理、および制御できるため、適切なユーザに適切なリソースを安全に提供できます。

このガイドでは、Identity Manager 3.0 のポリシービルダおよびドライバ環境設定について詳しく説明します。

対象読者

このガイドは、Identity Manager の管理者を対象にしています。

ご意見やご要望

このマニュアルおよび本製品に含まれるその他のマニュアルに関するご意見やご要望をお聞かせください。オンラインヘルプの各ページの下部にあるユーザコメント機能を使用するか、または www.novell.com/documentation/feedback.html にアクセスして、ご意見をお寄せください。

最新のマニュアル

このマニュアルの最新のバージョンについては、Identity Manager のマニュアルの Web サイト (http://www.novell.com/documentation/idm) を参照してください。

Identity Manager 2.0 のマニュアルについては、Identity Manager のマニュアルの Web サイト (http://www.novell.com/documentation/idm) を参照してください。

その他のマニュアル

Identity Manager ドライバの使用に関するマニュアルについては、Identity Manager Driver のマニュアルの Web サイト (http://www.novell.com/documentation/idmdrivers/index.html) を参照してください。

表記規則

本マニュアルでは、手順に含まれる複数の操作および相互参照パス内の項目を分けるために、大なり記号 (>) を使用しています。

商標記号 (®、™ など) は、Novell の商標を示します。アスタリスク (*) は第三者の商標を示します。

ポリシーとフィルタ

1

この節では、Identity Manager 環境でのポリシーとフィルタ、およびそれらの機能の概要について解説します。次のトピックについて説明しています。

1.1 ポリシーおよびフィルタとは

ポリシーの主な機能は、Identity Manager で更新を送受信する方法をカスタマイズすることです。

ポリシーを理解するには、ドライバシムの記述内容をある程度詳しく理解することが有用です。

ドライバシムが記述されている場合は、ドライバを展開している企業が使用する可能性のあるすべての情報を同期するための機能を組み込もうとします。開発者は、ドライバシムを、接続システム内で関連する変更を検出し、それをアイデンティティボールトに送るように記述します。

この変更は、Identity Manager の仕様に応じた形式で、XML ドキュメントに保存されます。次に、このような XML ドキュメントの抜粋を示します。

```
<nds dtdversion="2.0" ndsversion="8.7.3">
<source>
   <product version="2.0">DirXML</product>
   <contact>Novell, Inc.</contact>
</source>

<input>
   <add class-name="User" event-id="0" src-dn="\ACME\Sales\Smith"
   src-entry-id="33071">
      <add-attr attr-name="Surname">
         <value timestamp="1040071990#3" type="string">Smith</value>
      </add-attr>
      <add-attr attr-name="Telephone Number">
         <value timestamp="1040072034#1" type="teleNumber">111-1111</
value>
      </add-attr>
   </add>
</input>
</nds>
```

ドライバは、関連する変更をすべてレポートするように設計されているので、ユーザが情報をフィルタできるようになっています。フィルタは情報をブロックするように設計されています。フィルタを変更して、必要な情報だけが自分の環境に送られるようにします。重要な変更の指定とそれらの処理方法のロジックは、ドライバシムではなくエンジンで扱います。

企業内でグループがあまり重要ではない場合は、アイデンティティボールトまたは接続システムで、グループに関するすべての操作をブロックするフィルタを実装できます。企業内でユーザおよびグループを考慮する場合は、両方のタイプのオブジェクトを、アイデンティティボールトおよび接続システム間で同期するフィルタを実装できます。

必要なオブジェクトだけを同期できるようにフィルタを定義するのは、ドライバのカスタマイズの第 1 歩です。

次の段階として、フィルタを通過したオブジェクトに対して、Identity Manager が何を行うかを定義します。例として、前に挙げた XML ドキュメントの追加操作を参照してください。接続システムに、名前が「Smith」、電話番号が「111-1111」のユーザが追加されています。この操作が許可されているとすると、Identity Manager では、このユーザに対する処理内容を決定する必要があります。

この決定のために、Identity Manager は、一連のポリシーを特定の順序で適用します。

まず、一致ポリシーが、「このオブジェクトはすでにデータストア内にあるか ?」という質問に答えます。このためには、オブジェクトに固有の特性を定義する必要があります。一般的にチェックされる属性は、電子メールアドレスです。これは通常、固有であるためです。「2 つのオブジェクトの電子メールアドレスが同じ場合、それらは同じオブジェクトである」というポリシーを定義できます。

一致するものがあった場合、Identity Manager は、これを関連付けと呼ばれる属性に記録します。関連付けは、Identity Manager が接続システム内のオブジェクトを関連付けられるようにする一意の値です。

一致しているものがない場合は、作成ポリシーが呼び出されます。作成ポリシーは、オブジェクトの作成条件を Identity Manager に通知します。作成ルールでは特定の属性の存在を必須にすることができます。これらの属性が存在しない場合、Identity Manager は必要な情報が提供されるまで、オブジェクトの作成をブロックします。

オブジェクトが作成されると、配置ポリシーによって、オブジェクトの配置場所が Identity Manager に通知されます。オブジェクトを、それが元々あったシステムと同じ階層構造で作成するように指定できます。また、オブジェクトを属性値に基づいてまったく別の場所に配置することもできます。

オブジェクトの位置属性に従ってユーザを階層に配置し、フルネーム属性に従って名前を付ける場合は、作成ポリシーでこれらの属性を必須にすることができます。これによって、属性は確実に存在するようになり、配置の方針どおりに正しく機能します。

ポリシーの利用法は、他にも多数あります。ポリシービルダを使用すると、一意の値の生成、属性の追加および削除、イベントおよびコマンドの生成、電子メールの送信など、多くのことを簡単に実行できます。XSLT を使用して XML ドキュメントを直接変換することによって、より高度な変換も実行できます (変更は XML ドキュメントでアイデンティティボールトとやりとりできます)。

基本的には、ポリシーによって Identity Manager による更新の処理方法を制御できることを覚えておいてください。

さまざまなタイプのポリシーについては、12 ページのセクション 1.2「ポリシーの概要」を参照してください。次に、ポリシービルダの使用方法について、35 ページの第 2 章「Designer でポリシービルダを使用したポリシーの定義」、または 215 ページの第 3 章「iManager のポリシービルダを使用したポリシーの定義」を参照してください。

1.1.1 以前のバージョンからの用語の変更

DirXML® 1.1a では、「ルール」という用語は、文脈に応じて、一連のルール、そのセットに含まれる個々のルール、および個々のルールに含まれる条件やアクションを指すために使用されていました。しかし、このようにさまざまな状況で同じ用語が使われているために、文脈がわかりにくい場合に混乱を招いていました。

Identity Manager 2 では、記述されている高レベルな変換を説明する場合、以前使用していた「ルール」という用語の代わりに「ポリシー」という用語を使用します。この変更に伴い、一連のポリシーを定義し、各ポリシーに 1 つ以上のルールが含まれるようになりました。「ルール」という用語は、複数の条件とアクションからなる個々のセットのみを指すようになりました。

次の表は、DirXML 1.1a から Identity Manager 2.x. への用語の変更を示しています。

表 *1-1* *DirXML 1.1a* から *Identity Manager 2.x* への用語の変更

概念	DirXML 1.1a の用語	Identity Manager 2.x の用語
変換セット	ルール	ポリシーセット
セットに含まれる個々の変換	ルール	ポリシー
個々の変換に含まれる条件とアクション	ルール	ルール

次の表は、Identity Manager 2.x から Identity Manager 3.0 への用語の変更を示しています。

表 *1-2* *Identity Manager 2.x* から *Identity Manager 3.0* への用語の変更

概念	Identity Manager 2.x の用語	Identity Manager 3 の用語
製品	DirXML	Identity Manager
製品がインストールされているサーバ	DirXML サーバ	メタディレクトリサーバ
データの同期先のアプリケーションまたはデータベースのサーバ	DirXML 接続システムサーバ	接続システムサーバ
オブジェクトが保存される場所	eDirectory™	アイデンティティボールト
処理コンポーネント	DirXML エンジン	メタディレクトリエンジン

1.1.2 DirXML スクリプト

DirXML スクリプトは、Identity Manager ポリシーを実装する主要な方法です。このスクリプトでは、順序が指定された一連のルールによって実装されるポリシーを記述します。ルールにはテストする一連の条件と、その条件を満たしたときに順次実行される一連のアクションが含まれています。

DirXML スクリプトの作成には、使いやすい GUI を備えたポリシービルダを使用します。

1.2 ポリシーの概要

この節では、使用できるポリシーのタイプ、Identity Manager におけるポリシーの役割、および独自のポリシーを定義する方法の概要を説明します。次のトピックについて説明します。

- 12 ページのセクション 1.2.1「ポリシー」
- 32 ページのセクション 1.2.2「ポリシーの定義」

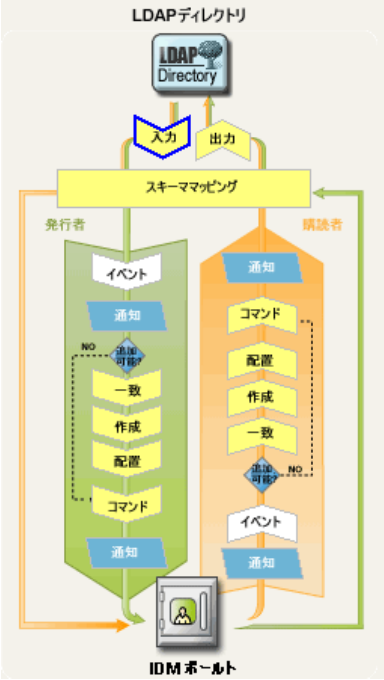
1.2.1 ポリシー

購読者チャネルと発行者チャネルの両方で定義できるポリシーにはさまざまなタイプがあります。各ポリシーは、データ変換のさまざまな段階で適用されます。また、一部のポリシーは特定のアクションが発生した場合にのみ適用されます。たとえば、作成ポリシーは新しいオブジェクトが作成される場合にのみ適用されます。

チャネルでのポリシーの実行順序を次に示します。

- 13 ページの 「イベント変換ポリシー」
- 15 ページの 「一致ポリシー」
- 17 ページの 「作成ポリシー」
- 19 ページの 「配置ポリシー」
- 22 ページの 「コマンド変換ポリシー」
- 25 ページの 「スキーママッピングポリシー」
- 28 ページの 「出力変換ポリシー」
- 30 ページの 「入力変換ポリシー」

図 *1-1* ポリシーの実行順序



イベント変換ポリシー

イベント変換ポリシーは、アイデンティティボールトまたは接続アプリケーションで発生したイベントの、メタディレクトリエンジンでのビューを変更します。イベント変換ポリシーで実行される最も一般的なタスクは、スコープフィルタリングやイベントタイプフィルタリングなどのカスタムフィルタリングです。

スコープフィルタリングは、イベントの位置または属性値に基づいて、不要なイベントを削除します。たとえば、部署属性が特定の値と一致しないか、特定のグループのメンバではない場合に、イベントを削除します。

イベントタイプフィルタリングは、イベントタイプに基づいて、不要なイベントを削除します。たとえば、すべての削除イベントを削除します。

例 :

- スコープフィルタリング
- タイプフィルタリング

スコープフィルタリング **:** 次の DirXML スクリプトポリシーの例では、Users サブツリー内に含まれ、無効になっておらず、役職属性に「Consultant」または「Manager」という語が含まれていないユーザに対するイベントだけを許可します。操作がブロックされていることを示すステータスドキュメントも生成します。

```
<policy>
      <rule>
            <description>Scope Filtering</description>
            <conditions>
                     <or>
                        <if-class-name op="equal">User</if-class-name>
                     </or>
                     <or>
                           <if-src-dn op="not-in-subtree">Users</if-
src-dn>
                            <if-attr name="Login Disabled"
op="equal">True</if-attr>
                        <if-attr mode="regex" name="Title"
op="equal">.*Consultant.*</if-attr>
                        <if-attr mode="regex" name="Title"
op="equal">.*Manager.*</if-attr>
                  </or>
            </conditions>
            <actions>
                  <do-status level="error">
                        <arg-string>
                              <token-text>User doesn’t meet required
conditions</token-text>
                        </arg-string>
                  </do-status>
                  <do-veto/>
               </actions>
      </rule>
</policy>
```

次の DirXML スクリプトポリシーは、ユーザオブジェクトに対する変更操作を拒否します。ただし、すでに関連付けられているオブジェクトの変更を除きます。

```
<policy>
      <rule>
             <description>Veto all operation on User except modifies
of already associated objects</description>
            <conditions>
                  <or>
                        <if-class-name op="equal">User</if-class-name>
                  </or>
                  <or>
                        <if-operation op="not-equal">modify</if-
operation>
                         <if-association op="not-associated"/>
                  </or>
            </conditions>
            <actions>
                     <do-veto/>
            </actions>
      </rule>
</policy>
```

タイプフィルタリング **:** 次の DirXML スクリプトポリシーの例では、最初のルールで、Employee および Contractor コンテナ内のオブジェクトの同期だけを許可します。2 つ目のルールでは、すべての名前変更操作および移動操作をブロックします。

```
<policy>
       <rule>
            <description>Only synchronize the Employee and Contractor
subtrees</description>
            <conditions>
                        <and>
                              <if-src-dn op="not-in-
container">Employees</if-src-dn>
                              <if-src-dn op="not-in-
container">Contractors</if-src-dn>
                        </and>
            </conditions>
            <actions>
                        <do-status level="warning">
                           <arg-string>
                             <token-text>Change ignored: Out of
scope.</token-text>
                           </arg-string>
                         </do-status>
                          <do-veto/>
            </actions>
       </rule>
       <rule>
            <description>Don’t synchronize moves or renames</
description>
```

```
            <conditions>
                        <or>
                            <if-operation op="equal">move</if-
operation>
                            <if-operation op="equal">rename</if-
operation>
                        </or>
            </conditions>
             <actions>
                        <do-status level="warning">
                                <arg-string>
                                          <token-text>Change ignored:
We don’t like you to do that.</token-text>
                                </arg-string>
                    </do-status>
                    <do-veto/>
             </actions>
      </rule>
</policy>
```

次の DirXML スクリプトポリシーは、すべての追加イベントをブロックします。

```
<policy>
      <rule>
              <description>Type Filtering</description>
              <conditions>
                        <and>
                                 <if-operation op="equal">add</if-
operation>
                        </and>
              </conditions>
              <actions>
                         <do-status level="warning">
                                  <arg-string>
                                        <token-text>Change ignored:
Adds are not allowed.</token-text>
                                  </arg-string>
                         </do-status>
                         <do-veto/>
               </actions>
        </rule>
</policy>
```

一致ポリシー

購読者一致および発行者一致などの一致ポリシーは、ソースデータストア内の関連付けられていないオブジェクトに対応するオブジェクトをターゲットデータストアで検索します。重要な点は、一致ポリシーが常に必要または望ましいとは限らないということです。

たとえば、次の状況では一致ポリシーが望ましくない可能性があります。

- 既存のオブジェクトまたは対応するオブジェクトが存在しない初期移行時

一致ポリシーは、誤った一致が検出されないように十分に注意して作成する必要があります。

例 :

- インターネット電子メールアドレスによる一致
- 共通名による一致

ID による一致 **:** 次の DirXML スクリプトポリシーの例は、インターネット電子メールアドレスに基づいてユーザを照合します。

```
<policy>
        <rule>
            <description>Match Users based on email address</
description>
            <conditions>
                      <and>
                        <if-class-name op="equal">User</if-class-name>
                      </and>
            </conditions>
            <actions>
                   <do-find-matching-object>
                         <arg-dn>
                           <token-text>ou=people,o=novell</token-text>
                         </arg-dn>
                      <arg-match-attr name="Internet EMail Address"/>
                        </do-find-matching-object>
            </actions>
       </rule>
</policy>
```

名前による一致 **:** 次の DirXML スクリプトポリシーの例では、その共通名属性に基づいて、グループオブジェクトを照合します。

```
<?xml version="1.0" encoding="UTF-8"?>
<policy>
        <rule>
                <description>Match Group by Common Name</description>
                <conditions>
                          <or>
                                   <if-class-name op="equal">Group</
if-class-name>
                           </or>
                </conditions>
                <actions>
                        <do-find-matching-object scope="subtree">
                               <arg-match-attr name="CN"/>
                         </do-find-matching-object>
                </actions>
         </rule>
</policy>
```

作成ポリシー

購読者作成ポリシーおよび発行者作成ポリシーなどの作成ポリシーは、新しいオブジェクトを作成するときに満たす必要がある条件を定義します。作成ポリシーがないことは、オブジェクトを作成できることを意味します。

たとえば、アイデンティティボールトに新しいユーザを作成しますが、新しいユーザオブジェクトには名前と ID だけを指定するとします。この作成は eDirectory ツリーでミラーリングされますが、追加はアイデンティティボールトに接続されているアプリケーションでは直ちに反映されません。これは、より完全な定義が指定されたユーザオブジェクトだけを許可するように、作成ポリシーで指定されているためです。

作成ポリシーは、発行者と購読者の両方で同じにすることも別にすることもできます。

テンプレートオブジェクトを指定して、オブジェクトが eDirectory で作成される場合に作成プロセスで使用することができます。

作成ポリシーは、通常、次の目的で使用されます。

- 属性がないなどの理由で条件を満たさないオブジェクトの作成を拒否する。
- デフォルト属性値を設定する。
- デフォルトパスワードを設定する。

例 :

- 必須属性
- デフォルト属性値
- デフォルトパスワード
- テンプレートの指定

必須属性 **:** 次に示す DirXML スクリプトポリシー例の最初のルールでは、ユーザを作成するにはユーザオブジェクトに CN、名前、名字、インターネット電子メールアドレスの各属性があらかじめ含まれている必要があります。2 つ目のルールでは、すべての部門オブジェクトで OU 属性が必要になります。最後のルールでは、名前が「Fred」であるすべてのユーザオブジェクトが拒否されます。

```
<policy>
      <rule>
            <description>Veto if required attributes CN, Given Name,
Surname and Internet EMail Address not available</description>
            <conditions>
                  <or>
                        <if-class-name op="equal">User</if-class-
name>
                  </or>
            </conditions>
            <actions>
             <do-veto-if-op-attr-not-available name="CN"/>
             <do-veto-if-op-attr-not-available name="Given Name"/>
                  <do-veto-if-op-attr-not-available name="Surname"/>
                  <do-veto-if-op-attr-not-available name="Internet
EMail Address"/>
            </actions>
```

```
      </rule>
    <rule>
            <description>Organizational Unit Required Attributes</
description>
            <conditions>
                <or>
                        <if-class-name op="equal">Organizational
Unit</if-class-name>
                  </or>
            </conditions>
            <actions>
                  <do-veto-if-op-attr-not-available name="OU"/>
            </actions>
      </rule>
</policy>
```

デフォルト属性値 **:** 次の DirXML スクリプトポリシーの例では、ユーザの説明属性のデフォルト値を追加します。

```
<policy>
      <rule>
            <description>Default Description of New Employee</
description>
            <conditions>
                     <or>
                       <if-class-name op="equal">User</if-class-name>
                     </or>
            </conditions>
            <actions>
                   <do-set-default-attr-value name="Description">
                         <arg-value type="string">
                                 <token-text>New Employee</token-text>
                         </arg-value>
                   </do-set-default-attr-value>
            </actions>
       </rule>
</policy>
```

デフォルトパスワード **:** 次の DirXML スクリプトポリシーの例では、名前の最初の 2 文字と名字の最初の 6 文字 (すべて小文字) で構成したパスワード値を作成します。

```
<policy>
       <rule>
                        <description>Default Password of [2]FN+[6]LN</
description>
            <conditions>
                  <and>
                        <if-class-name op="equal">User</if-class-name>
                         <if-password op="not-available"/>
                  </and>
            </conditions>
            <actions>
```

```
                  <do-set-dest-password>
                        <arg-string>
                              <token-lower-case>
                                    <token-substring length="2">
                                          <token-op-attr name="Given
Name"/>
                                    </token-substring>
                                  <token-substring length="6">
                                          <token-op-attr
name="Surname"/>
                                    </token-substring>
                                     </token-lower-case>
                           </arg-string>
                  </do-set-dest-password>
             </actions>
      </rule>
</policy>
```

テンプレートの指定 **:** 次の DirXML スクリプトポリシーの例では、ユーザの役職属性で、そのユーザがマネージャであることが示されている (「Manager」を含んでいる) 場合のテンプレートオブジェクトを指定します。

```
<policy>
         <rule>
                   <description>Assign Manager Template if Title
contains Manager</description>
                   <conditions>
                          <and>
                             <if-class-name op="equal">User</if-class-
name>
                              <if-op-attr name="Title" op="available"/
>
                               <if-op-attr mode="regex" name="Title"
op="equal">.*Manager.*</if-op-attr>
                          </and>
                     </conditions>
                     <actions>
                            <do-set-op-template-dn>
                                    <arg-dn>
                                          <token-text>Users\Manager
Template</token-text>
                                     </arg-dn>
                             </do-set-op-template-dn>
                   </actions>
           </rule>
</policy>
```

配置ポリシー

配置ポリシーは、新しいオブジェクトを配置する場所と、アイデンティティボールトおよび接続されたアプリケーションでのオブジェクトの名前を決定します。

配置ポリシーはアイデンティティボールトでオブジェクト作成を行う場合に、発行者チャネルで必要です。配置ポリシーは、購読者チャネルでは必須でない場合もあります。これはターゲットデータストアの特性に応じて、接続されたアプリケーションでオブジェクト作成を行う場合も同じです。たとえば、リレーショナルデータベースへの同期では、リレーショナルデータベースの行には位置も名前もないので配置ポリシーは不要です。

例 :

- 属性値による配置
- 名前による配置

属性値による配置 **:** 次の DirXML スクリプトポリシーの例では、部署属性の値に基づいて、特定のコンテナにユーザを作成します。

```
<policy>
        <rule>
             <description>Department Engineering</description>
             <conditions>
                  <and>
                        <if-class-name op="equal">User</if-class-name>
                        <if-op-attr mode="regex" name="Department"
op="equal">.*Engineering.*</if-op-attr>
                  </and>
            </conditions>
            <actions>
                  <do-set-op-dest-dn>
                        <arg-dn>
                              <token-text>Eng</token-text>
                          <token-text>\</token-text>
                                      <token-op-attr name="CN"/>
                        </arg-dn>
                  </do-set-op-dest-dn>
            </actions>
       </rule>
       <rule>
            <description>Department HR</description>
            <conditions>
                  <and>
                         <if-class-name op="equal">User</if-class-
name>
                        <if-op-attr mode="regex" name="Department"
op="equal">.*HR.*</if-op-attr>
                  </and>
            </conditions>
            <actions>
                  <do-set-op-dest-dn>
                        <arg-dn>
                              <token-text>HR</token-text>
                          <token-text>\</token-text>
                           <token-op-attr name="CN"/>
                        </arg-dn>
                  </do-set-op-dest-dn>
            </actions>
      </rule>
```

```
</policy>
```

次の DirXML スクリプトポリシーでは、入力ドキュメントの src-dn によって、ユーザまたは部門の配置を決定します。

```
<policy>
       <rule>
             <description>PublisherPlacementRule</description>
             <conditions>
                       <or>
                            <if-class-name op="equal">User</if-class-
name>
                             <if-class-name op="equal">Organizational
Unit</if-class-name>
                       </or>
                       <or>
                             <if-src-dn op="in-subtree">o=people,
o=novell</if-src-dn>
                      </or>
               </conditions>
               <actions>
                    <do-set-op-dest-dn>
                        <arg-dn>
                              <token-text>People</token-text>
                              <token-text>\</token-text>
                              <token-unmatched-src-dn convert="true"/>
                        </arg-dn>
                     </do-set-op-dest-dn>
                  </actions>
        </rule>
</policy>
```

名前による配置 **:** 次の DirXML スクリプトポリシーの例では、ユーザの名字の最初の文字に基づいて、特定のコンテナにユーザを配置します。名字が A ～ I で始まるユーザは Users1 コンテナ、J ～ R のユーザは Users2 コンテナ、S ～ Z は Users3 コンテナに配置されます。

```
<policy>
       <rule>
            <description>Surname - A to I in Users1</description>
            <conditions>
                        <and>
                              <if-class-name op="equal">User</if-
class-name>
                              <if-op-attr mode="regex" name="Surname"
op="equal">[A-I].*</if-op-attr>
                         </and>
            </conditions>
            <actions>
                <do-set-op-dest-dn>
                        <arg-dn>
                              <token-text>Users1</token-text>
```

```
                        <token-text>\</token-text>
                              <token-op-attr name="CN"/>
                        </arg-dn>
                  </do-set-op-dest-dn>
            </actions>
       </rule>
       <rule>
            <description>Surname - J to R in Users2</description>
            <conditions>
                     <and>
                              <if-class-name op="equal">User</if-class-
name>
                               <if-op-attr mode="regex" name="Surname"
op="equal">[J-R].*</if-op-attr>
                        </and>
            </conditions>
            <actions>
               <do-set-op-dest-dn>
                    <arg-dn>
                              <token-text>Users2</token-text>
                        <token-text>\</token-text>
                         <token-op-attr name="CN"/>
                        </arg-dn>
                  </do-set-op-dest-dn>
            </actions>
        </rule>
        <rule>
             <description>Surname - S to Z in Users3</description>
             <conditions>
                       <and>
                             <if-class-name op="equal">User</if-class-
name>
                              <if-op-attr mode="regex" name="Surname"
op="equal">[S-Z].*</if-op-attr>
                        </and>
             </conditions>
             <actions>
                  <do-set-op-dest-dn>
                     <arg-dn>
                              <token-text>Users3</token-text>
                              <token-text>\</token-text>
                              <token-op-attr name="CN"/>
                        </arg-dn>
                  </do-set-op-dest-dn>
            </actions>
      </rule>
</policy>
```

コマンド変換ポリシー

コマンド変換ポリシーは、コマンドを置き換えるか追加することによって、Identity Manager がターゲットデータストアに送信するコマンドを変更します。削除コマンドを行わずに、それを変更、移動、または無効化コマンドに置き換えるのは、コマンド変換ポリ

シーのコマンド置き換えの例です。コマンド変換ポリシーでのコマンドの追加の一般的な例として、追加コマンドの属性値に基づいた変更コマンドの作成があります。

基本的には、コマンド変換ポリシーは、メタディレクトリエンジンに送信されたイベントのデフォルト処理の結果として Identity Manager が実行するコマンドを変更するために使用されます。

また、他のポリシーの説明に合致しないポリシーをここで含めるのも一般的な使用法です。

例 :

- 削除から変更および移動への変換
- 追加操作の作成
- パスワード期限の時刻の設定

削除から変更への変換 **:** 次の DirXML スクリプトポリシーでは、「ログインの無効化」属性の削除操作を変更操作に変換します。

```
<policy>
      <rule>
            <description>Convert User Delete to Modify</description>
            <conditions>
                  <and>
                        <if-operation op="equal">delete</if-
operation>
                  <if-class-name op="equal">User</if-class-name>
                  </and>
            </conditions>
            <actions>
                  <do-set-dest-attr-value name="Login Disabled">
                        <arg-value type="state">
                              <token-text>true</token-text>
                        </arg-value>
                  </do-set-dest-attr-value>
                  <do-veto/>
            </actions>
      </rule>
</policy>
```

追加操作の作成 **:** 次の DirXML スクリプトポリシーでは、ユーザのターゲットコンテナがすでに存在するかどうかを判断します。コンテナがない場合、このポリシーでは、コンテナオブジェクトを作成する追加操作を作成します。

```
<policy>
      <rule>
            <description>Check if destination container already
exists</description>
            <conditions>
                  <and>
                        <if-operation op="equal">add</if-operation>
                  </and>
            </conditions>
```

```
    <actions>
                  <do-set-local-variable name="target-container">
                        <arg-string>
                              <token-dest-dn length="-2"/>
                        </arg-string>
                  </do-set-local-variable>
                  <do-set-local-variable name="does-target-exist">
                        <arg-string>
                              <token-dest-attr class-
name="OrganizationalUnit" name="objectclass">
                                    <arg-dn>
                                          <token-local-variable
name="target-container"/>
                                    </arg-dn>
                              </token-dest-attr>
                        </arg-string>
                  </do-set-local-variable>
            </actions>
      </rule>
      <rule>
            <description>Create the target container if necessary</
description>
            <conditions>
                  <and>
                        <if-local-variable name="does-target-exist"
op="available"/>
                        <if-local-variable name="does-target-exist"
op="equal"/>
                  </and>
               </conditions>
            <actions>
                  <do-add-dest-object class-name="organizationalUnit"
direct="true">
                        <arg-dn>
                              <token-local-variable name="target-
container"/>
                        </arg-dn>
                  </do-add-dest-object>
                  <do-add-dest-attr-value direct="true" name="ou">
                        <arg-dn>
                              <token-local-variable name="target-
container"/>
                        </arg-dn>
                        <arg-value type="string">
                              <token-parse-dn dest-dn-format="dot"
length="1" src-dn-format="dest-dn" start="-1">
                                    <token-local-variable
name="target-container"/>
                              </token-parse-dn>
                                    </arg-value>
                  </do-add-dest-attr-value>
            </actions>
       </rule>
```

```
</policy>
```

パスワード期限の時刻の設定 : 次の DirXML スクリプトポリシーでは、eDirectory ユーザの「パスワード期限の時刻」属性を変更します。

```
<?xml version="1.0" encoding="UTF-8"?>
<policy xmlns:jsystem="http://www.novell.com/nxsl/java/
java.lang.System">
    <rule>
            <description>Set password expiration time for a given
interval from current day</description>
            <conditions>
                        <and>
                    <if-operation op="equal">modify-password</if-
operation>
                </and>
            </conditions>
            <actions>
                        <do-set-local-variable name="interval">
                    <arg-string>
                              <token-text>30</token-text>
                    </arg-string>
                </do-set-local-variable>
                <do-set-dest-attr-value class-name="User"
name="Password Expiration Time" when="after">
                                              <arg-association>
                           <token-association/>
                    </arg-association>
                    <arg-value type="string">
                                                                <token-
xpath expression="round(jsystem:currentTimeMillis() div 1000 +
(86400*$interval))"/>
                    </arg-value>
                </do-set-dest-attr-value>
            </actions>
      </rule>
</policy>
```

スキーママッピングポリシー

スキーママッピングポリシーは、アイデンティティボールトと接続システム間のスキーママッピングの定義を保持します。

アイデンティティボールトスキーマは、eDirectory から読み込まれます。接続されているアプリケーションのスキーマは、接続システムの Identity Manager ドライバから提供されます。2 つのスキーマが特定された後、アイデンティティボールトとターゲットアプリケーション間で単純なマッピングが作成されます。

スキーママッピングポリシーを Identity Manager ドライバ環境設定に定義した後は、対応するデータをマップできます。

次の点に十分注意してください。

- 同じポリシーが両方向で適用されます。
- メタディレクトリエンジンとアプリケーションシム間で渡されるドキュメントは、いずれのチャネルでも、またいずれの方向でも、すべてのドキュメントがスキーママッピングポリシーを通過します。

管理情報については、423 ページの第 7 章「スキーママッピングポリシーの管理」を参照してください。

例 :

- 基本的なスキーママッピングポリシー
- カスタムスキーママッピングポリシー

基本的なスキーママッピングポリシー : 次の DirXML スクリプトポリシーの例では、基本的なスキーママッピングポリシーの XML ソースをそのまま表示しています。ただし、Identity Manager の Designer を使用してポリシーを編集する場合、デフォルトのスキーママッピングエディタでは、ポリシーをグラフィカルに表示および編集できます。

```
<?xml version="1.0" encoding="UTF-8"?><attr-name-map>
      <class-name>
              <app-name>WorkOrder</app-name>
              <nds-name>DirXML-nwoWorkOrder</nds-name>
      </class-name>
      <class-name>
               <app-name>PbxSite</app-name>
               <nds-name>DirXML-pbxSite</nds-name>
      </class-name>
      <attr-name class-name="DirXML-pbxSite">
               <app-name>PBXName</app-name>
               <nds-name>DirXML-pbxName</nds-name>
      </attr-name>
      <attr-name class-name="DirXML-pbxSite">
                   <app-name>TelephoneNumber</app-name>
                  <nds-name>Telephone Number</nds-name>
      </attr-name>
      <attr-name class-name="DirXML-pbxSite">
               <app-name>LoginName</app-name>
               <nds-name>DirXML-pbxLoginName</nds-name>
       </attr-name>
       <attr-name class-name="DirXML-pbxSite">
               <app-name>Password</app-name>
               <nds-name>DirXML-pbxPassword</nds-name>
       </attr-name>
       <attr-name class-name="DirXML-pbxSite">
               <app-name>Nodes</app-name>
               <nds-name>DirXML-pbxNodesNew</nds-name>
       </attr-name>
      </attr-name-map>
```

カスタムスキーママッピングポリシー : 次の DirXML スクリプトポリシーの例は、DirXML スクリプトを使用してカスタムスキーママッピングを実行します。

```
<?xml version="1.0" encoding="UTF-8"?><policy>
      <rule>
            <!--
            The Schema Mapping Policy can only handle one-to-one
mappings.
            That Mapping Policy maps StudentPersonal addresses.
            This rule maps StaffPersonal addresses.
      -->
            <description>Publisher Staff Address Mappings</
description>
            <conditions>
                  <and>
                        <if-local-variable name="fromNds"
op="equal">false</if-local-variable>
                        <if-xpath op="true">@original-class-name =
'StaffPersonal'</if-xpath>
                  </and>
            </conditions>
            <actions>
                  <do-rename-op-attr dest-name="SA" src-name="Address/
Street/Line1"/>
                  <do-rename-op-attr dest-name="Postal Office Box"
src-name="Address/Street/Line2"/>
                  <do-rename-op-attr dest-name="Physical Delivery
Office Name" src-name="Address/City"/>
                  <do-rename-op-attr dest-name="S" src-name="Address/
StatePr"/>
                  <do-rename-op-attr dest-name="Postal Code" src-
name="Address/PostalCode"/>
            </actions>
      </rule>
      <rule>
            <description>Subscriber Staff Address Mappings</
description>
            <!--
            The Schema Mapping Policy has already mapped addresses to
StudentPersonal.
            This rule maps StudentPersonal to StaffPersonal.
      -->
            <conditions>
                  <and>
                        <if-local-variable name="fromNds"
op="equal">true</if-local-variable>
                        <if-op-attr name="DirXML-sifIsStaff"
op="equal">true</if-op-attr>
                  </and>
            </conditions>
            <actions>
                  <do-rename-op-attr dest-name="Address/Street/Line1"
src-name="StudentAddress/Address/Street/Line1"/>
                  <do-rename-op-attr dest-name="Address/Street/Line2"
src-name="StudentAddress/Address/Street/Line2"/>
                  <do-rename-op-attr dest-name="Address/City" src-
```

```
name="StudentAddress/Address/City"/>
                  <do-rename-op-attr dest-name="Address/StatePr" src-
name="StudentAddress/Address/StatePr"/>
                  <do-rename-op-attr dest-name="Address/PostalCode"
src-name="StudentAddress/Address/PostalCode"/>
            </actions>
      </rule>
</policy>
```

出力変換ポリシー

出力変換ポリシーは主に、メタディレクトリエンジンが提供するデータからアプリケーションシムで求められるデータへのデータ形式の変換を行います。次に変換の例を示します。

- 属性値の形式の変換
- XML ボキャブラリの変換
- 出力変換ポリシーでは、メタディレクトリエンジンからアプリケーションシムに返されるステータスメッセージのカスタム処理も提供できます

いずれかのチャネルでメタディレクトリエンジンがアプリケーションシムに提供するドキュメントはすべて出力変換ポリシーを通過します。出力変換はスキーママッピングの後に実行されるので、すべてのスキーマ名はアプリケーションネームスペース内にあります。

例 :

- 属性値の形式の変換
- ステータスメッセージのカスタム処理

属性値の変換 **:** 次の DirXML スクリプトポリシーの例は、電話番号を「(nnn) nnn-nnnn」形式から「nnn.nnn.nnnn」形式に再フォーマットします。逆の変換については、入力変換ポリシーの例を参照してください。

```
<policy>
      <rule>
            <description>Reformat all telephone numbers from (nnn)
nnn-nnnn to nnn.nnn.nnnn</description>
            <conditions/>
            <actions>
                        <do-reformat-op-attr name="telephoneNumber">
                            <arg-value type="string">
                                        <token-replace-first
regex="^\((\d\d\d)\) *(\d\d\d)-(\d\d\d\d)$" replace-with="$1.$2.$3">
                                                    <token-local-
variable name="current-value"/>
                                    </token-replace-first>
                            </arg-value>
                         </do-reformat-op-attr>
            </actions>
     </rule>
```

```
</policy>
```

ステータスメッセージのカスタム処理 **:** 次の DirXML スクリプトポリシーの例では、レベルが「success」と等しくなく、また操作データ内に子の password-publish-status 要素を含むステータスドキュメントを検出し、DoSendEmailFromTemplate(テンプレートから電子メールを送信) アクションを使用して電子メールを生成します。

```
<?xml version="1.0" encoding="UTF-8"?>
      <policy>
                <description>Email notifications for failed password
publications</description>
            <rule>
                        <description>Send e-mail for a failed publish
password operation</description>
                      <conditions>
                              <and>
                                        <if-global-variable
mode="nocase" name="notify-user-on-password-dist-failure"
op="equal">true</if-global-variable>
                                        <if-operation
op="equal">status</if-operation>
                                        <if-xpath
op="true">self::status[@level != 'success']/operation-data/password-
publish-status</if-xpath>
                                  </and>
                     </conditions>
                     <actions>
                     <!-- generate email notification -->
                      <do-send-email-from-template notification-
dn="\cn=security\cn=Default Notification Collection" template-
dn="\cn=security\cn=Default Notification Collection\cn=Password Sync
Fail">
                            <arg-string name="UserFullName">
                                  <token-src-attr name="Full Name">
                                      <arg-association>
                                                  <token-xpath
expression="self::status/operation-data/password-publish-status/
association"/>
                                      </arg-association>
                                  </token-src-attr>
                             </arg-string>
                             <arg-string name="UserGivenName">
                                   <token-src-attr name="Given Name">
                                          <arg-association>
                                                    <token-xpath
expression="self::status/operation-data/password-publish-status/
association"/>
                                          </arg-association>
                                  </token-src-attr>
                            </arg-string>
                            <arg-string name="UserLastName">
                                 <token-src-attr name="Surname">
                                                  <arg-association>
```

```
                                                <token-xpath
expression="self::status/operation-data/password-publish-status/
association"/>
                                        </arg-association>
                                </token-src-attr>
                        </arg-string>
                        <arg-string name="ConnectedSystemName">
                                <token-global-variable
name="ConnectedSystemName"/>
                        </arg-string>
                        <arg-string name="to">
                                <token-src-attr name="Internet Email
Address">
                                 <arg-association>
                                        <token-xpath
expression="self::status/operation-data/password-publish-status/
association"/>
                                   </arg-association>
                                </token-src-attr>
                         </arg-string>
                         <arg-string name="FailureReason">
                                  <token-text/>
                                     <token-xpath
expression="self::status/child::text()"/>
                           </arg-string>
                </do-send-email-from-template>
            </actions>
      </rule>
</policy>
```

入力変換ポリシー

入力変換ポリシーは主に、アプリケーションシムが提供するデータからメタディレクトリエンジンで求められるデータへのデータ形式の変換を行います。次に変換の例を示します。

- 属性値の形式の変換
- XML ボキャブラリの変換
- ドライバハートビート
- 入力変換ポリシーでは、アプリケーションシムからメタディレクトリエンジンに返されるステータスメッセージのカスタム処理も提供できます。

いずれかのチャネルでアプリケーションシムがメタディレクトリエンジンに提供するドキュメントはすべて入力変換ポリシーを通過します。

例 :

- 属性値の形式の変換
- ドライバハートビート

属性値の形式の変換 **:** 次の DirXML スクリプトポリシーの例では、電話番号を「nnn.nnn.nnnn」形式から「(nnn) nnn-nnnn」形式に再フォーマットします。逆の変換については、出力変換ポリシーの例を参照してください。

```
<policy>
      <rule>
            <description>Reformat all telephone numbers from
nnn.nnn.nnnn to (nnn) nnn-nnnn</description>
            <conditions/>
            <actions>
                  <do-reformat-op-attr name="telephoneNumber">
                        <arg-value type="string">
                                          <token-replace-first
regex="^(\d\d\d)\.(\d\d\d)\.(\d\d\d\d)$" replace-with="($1) $2-$3">

<token-local-variable name="current-value"/>
                                       </token-replace-first>
                        </arg-value>
                  </do-reformat-op-attr>
            </actions>
      </rule>
</policy>
```

ドライバハートビート **:** 次の DirXML スクリプトポリシーでは、ステータスハートビートイベントを作成します。ドライバのハートビート機能は、各ハートビート間隔の成功メッセージ (HEARTBEAT: $driver) の送信に使用されます。メッセージは、Novell Audit によって監視されます。Identity Manager ドライバは、ハートビートをサポートしている必要があります。また、ハートビートがドライバ環境設定ページで有効になっている必要があります。

```
<?xml version="1.0" encoding="UTF-8" ?>
 <policy>
      <rule>
      <description>Heartbeat Rule, v1.01, 040126, by Holger Dopp</
description>
            <conditions>
                  <and>
                        <if-operation op="equal">status</if-operation>
                        <if-xpath op="true">@type="heartbeat"</if-
xpath>
                  </and>
            </conditions>
            <actions>
                  <do-set-xml-attr expression="." name="text1">
                        <arg-string>
                              <token-global-variable
name="dirxml.auto.driverdn" />
             </arg-string>
                  </do-set-xml-attr>
                  <do-set-xml-attr expression="." name="text2">
                        <arg-string>
                              <token-text>HEARTBEAT</token-text>
```

```
                        </arg-string>
                    </do-set-xml-attr>
                </actions>
  </rule>
</policy>
```

1.2.2 ポリシーの定義

すべてのポリシーは、次の 2 つの方法のいずれかで定義されます。

- ポリシービルダインタフェースを使用して DirXML スクリプトを作成する。既存の XSLT 以外のルールは、インポート時に自動的に DirXML スクリプトに変換されます。
- XSLT スタイルシートを使用する。

スキーママッピングポリシーは、スキーママッピングテーブルを使用して定義することもできます (通常はこちらが使用されます)。

ポリシービルダおよび DirXML スクリプト

ポリシービルダインタフェースは、実装するポリシーのほとんどの定義に使用されます。ユーザがポリシーを簡単に定義および管理できるように、ポリシービルダインタフェースでは、グラフィック環境が使用されます。

ポリシービルダ内のルール作成の基になる機能は、DirXML スクリプトと呼ばれるカスタムスクリプト言語によって提供されます。

DirXML スクリプトには、テスト可能な多岐にわたる条件、実行するアクション、およびポリシーに追加する動的な値が含まれます。これらのオプションはいずれも、それぞれの場所で有効な選択肢のみを提示するインテリジェントドロップダウンリスト、および共通の値へのクイックリンクを使用して表示されます。

ポリシービルダを使用すれば、DirXML スクリプトを直接変更する必要はありません。

ポリシービルダの詳細については、35 ページの第 2 章「Designer でポリシービルダを使用したポリシーの定義」および 215 ページの第 3 章「iManager のポリシービルダを使用したポリシーの定義」を参照してください。DirXML スクリプトの詳細については、11 ページのセクション 1.1.2「DirXML スクリプト」を参照してください。

ヒント **:** ポリシービルダは使用しなくてもかまいません。完全な DirXML スクリプトについては、DirXML Driver Developer Kit Documentation (http://developer.novell.com/ndk/doc/dirxml/dirxmlbk/ref/index.html) の Web サイトを参照してください。

XSLT スタイルシート

より複雑なポリシーを定義するには、XSLT スタイルシートを使用して、XML ドキュメントを、必要な変更を含む別の XML ドキュメントに直接変換します。

スタイルシートでは柔軟な指定ができるので、ポリシービルダのルール作成を使用して利用できる定義済みの条件およびアクションに合わない場合に使用されます。

XSLT スタイルシートを作成するには、XSLT、nds.dtd、およびメタディレクトリエンジンとの間で変換するコマンドおよびイベントについて十分に理解しておく必要があります。nds.dtd の詳細については、NDS DTD (http://developer.novell.com/ndk/doc/dirxml/dirxmlbk/ref/ndsdtd/index.html) を参照してください。

XSLT スタイルシートの詳細については、383 ページの第 5 章「XSLT スタイルシートを使用したポリシーの定義」を参照してください。

ダウンロード可能な Identity Manager ポリシー

Novell では、ダウンロードして自分の環境で使用できるサンプルポリシーが用意されています。ポリシーは、Patches (Novell のサポート Web サイト) (http://support.novell.com/filefinder/20607/index.html) で入手できます。ファイルをダウンロードし、それを解凍します。How_To_Install.rtf ファイルには、インストールの手順が含まれています。

Designer を使用してファイルをインポートするには、57 ページの 「XML ファイルからのポリシーのインポート」を参照してください。iManager を使用してファイルをインポートするには、227 ページのセクション 3.2.9「XML ファイルからのポリシーのインポート」を参照してください。

1.3 フィルタ

フィルタでは、メタディレクトリエンジンがイベントを処理するオブジェクトクラスや属性、およびそれらのクラスおよび属性に対する変更の処理方法を指定します。

フィルタは、ベースクラスがフィルタで指定されたクラスのいずれかと一致するオブジェクトで発生するイベントだけを通過させます。フィルタで指定されたクラスのサブオーディネートクラスであるオブジェクトで発生するイベントは通過させません。ただし、サブオーディネートクラスも指定されている場合を除きます。チャネルごとに個別のフィルタを設定して、各クラスおよび属性の同期方向および信頼されるデータソースを制御できます。

注 **:** eDirectory では、エントリの作成に使用されるオブジェクトクラスがベースクラスです。フィルタでは、ベースクラスの継承元であるスーパークラス、または追加属性の取得元である補助クラスを指定するよりもむしろ、このクラスを指定する必要があります。

たとえば、名字属性および名前属性を持つユーザクラスがフィルタで同期対象として設定されている場合、メタディレクトリエンジンはこれらの属性に対するすべての変更を渡します。しかし、エントリの電話番号属性が変更されている場合は、電話番号属性がフィルタに設定されていないため、メタディレクトリエンジンはこのイベントをドロップします。

フィルタの設定には次を含める必要があります。

- 同期する属性
- まだ同期されていないが、何らかのアクションを実行するポリシーをトリガするために使用される属性

フィルタの定義については、397 ページの第 6 章「フィルタの管理」を参照してください。

2 Designer でポリシービルダを使用したポリシーの定義

ポリシービルダは、接続システム間でのデータのやりとりを定義するポリシーを作成、および管理するための機能を完備したグラフィカルインタフェースです。

この章では、次のようなポリシーおよびポリシービルダの使用方法について説明します。

- 35 ページのセクション 2.1「ポリシー」
- 36 ページのセクション 2.2「Designer におけるポリシービルダーのタスク」

この章では、次の節についても詳しく説明します。

- 124 ページのセクション 2.3「正規表現」
- 125 ページのセクション 2.4「XPath 1.0 の式」
- 126 ページのセクション 2.5「条件」
- 143 ページのセクション 2.6「アクション」
- 190 ページのセクション 2.7「名詞トークン」
- 204 ページのセクション 2.8「動詞トークン」

2.1 ポリシー

ポリシーの動作を理解するには、まず、ポリシーのコンポーネントを理解する必要があります。

- ポリシーは複数のルールで構成されています。
- ルールとは、定義したアクション (143 ページの「アクション」を参照) が実行されるために満たされていなければならない条件 (126 ページの「条件」を参照) のセットです。
- アクションは実行時に展開されるトークンから派生する動的な引数を持つことができます。
- トークンは、2 つに分類することができます。名詞 (190 ページの「名詞トークン」を参照) と動詞 (204 ページの「動詞トークン」を参照) です。
  - 名詞トークンは現在の操作、ソースやターゲットのデータストア、または外部ソースなどから派生する値を展開します。
  - 動詞トークンは、そのトークンのサブオーディネイトにある他のトークンの連結された結果を変更します。
- 正規表現 (124 ページの「正規表現」を参照) および XPath 1.0 の式 (125 ページの「XPath 1.0 の式」を参照) は、一般的にには、ポリシーに対し適した結果を作成するためにルールで使用されます。
- ポリシーとは XDS ドキュメント上で操作を実行するもので、その主な目的はドキュメントを調べて変更を加えることです。
- 操作とは XDS ドキュメント内の要素のことで、入力要素と出力要素の子になります。これらの要素は Novell の nds.dtd の一部です。詳細については、「NDS DTD (http://

developer.novell.com/ndk/doc/dirxml/dirxmlbk/ref/ndsdtd/index.html)」を参照してください。

- 通常、1 つの操作は 1 つのイベント、コマンドまたはステータスを表します。
- ポリシーは、操作ごとに個別に適用されます。ポリシーが各操作に順番に適用されるので、その操作が現在の操作になります。各ルールは現在の操作に順次適用されます。直前のルールによって実行されたアクションが原因で、ルールがそれ以降適用されなくなる場合を除き、すべてのルールが現在の操作に適用されます。
- ポリシーはドキュメント外のコンテキストを取得して、結果のドキュメントに反映されない副次的動作を発生させることもできます。

2.2 Designer におけるポリシービルダーのタスク

この節では、ポリシービルダの一般的なタスクを実行する手順を説明します。

- 36 ページのセクション 2.2.1「ポリシービルダの起動」
- 40 ページのセクション 2.2.2「ポリシーの作成」
- 50 ページのセクション 2.2.3「ルールの作成」
- 59 ページのセクション 2.2.4「引数の作成」
- 69 ページのセクション 2.2.5「ポリシーの編集」
- 72 ページのセクション 2.2.6「事前定義されたルールの使用」
- 107 ページのセクション 2.2.7「ポリシーシミュレータを使用したポリシーのテスト」
- 117 ページのセクション 2.2.8「DirXML スクリプトの編集」

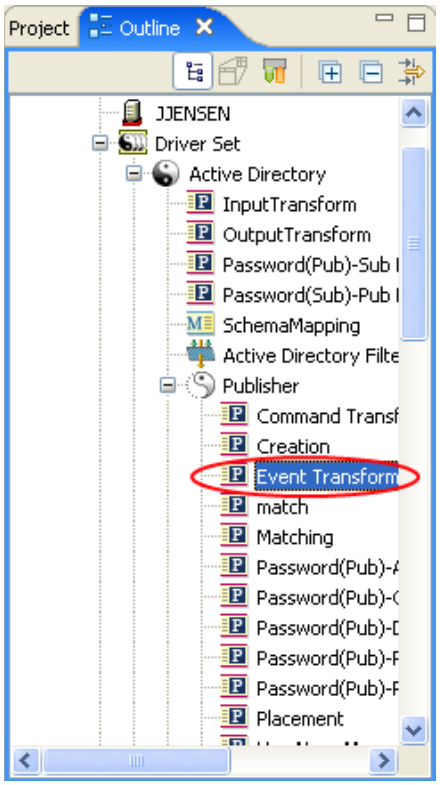
2.2.1 ポリシービルダの起動

ポリシービルダは［Model Outline (モデルアウトライン)］ビュー、［Policy Flow (ポリシーフロー)］ビュー、またはポリシーセットから起動できます。

［Model Outline (モデルアウトライン)］ビュー

1. Designer でプロジェクトを開きます。
2. ［Outline (アウトライン)］タブをクリックし、［Show Model Outline (モデルアウトラインの表示)］アイコンを選択します。> >

3 ［Model Outline (モデルアウトライン)］ビューに一覧表示されているポリシーをダブルクリックするか、右クリックして［編集］を選択します。> >

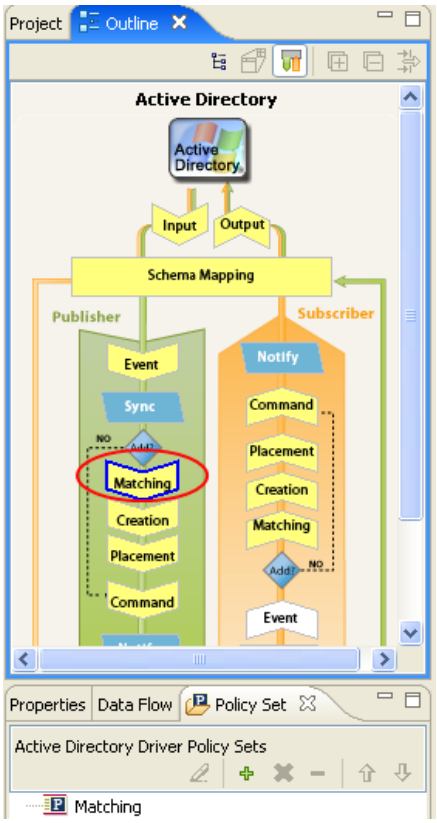


［Policy Flow (ポリシーフロー)］ビュー

1 Designer でプロジェクトを開きます。

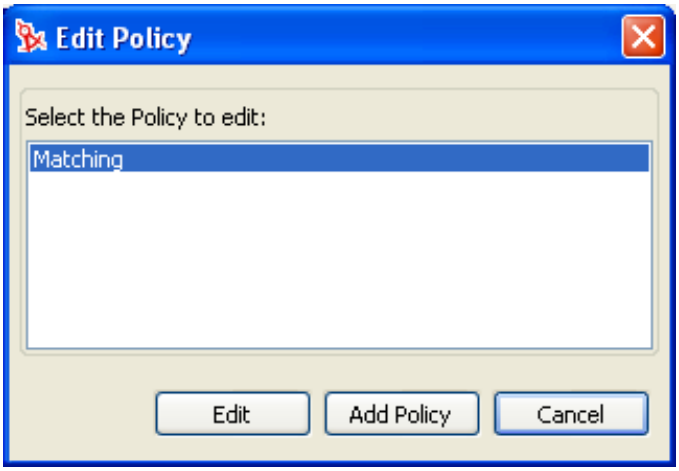
2 ［Outline (アウトライン)］タブを選択し、［Show Policy Flow (ポリシーフローの表示)］アイコンを選択します。> >

3 ［Policy Flow (ポリシーフロー)］ビューでポリシー (たとえば、一致ポリシー) を右クリックし、［Edit Policy (ポリシーの編集)］を選択します。



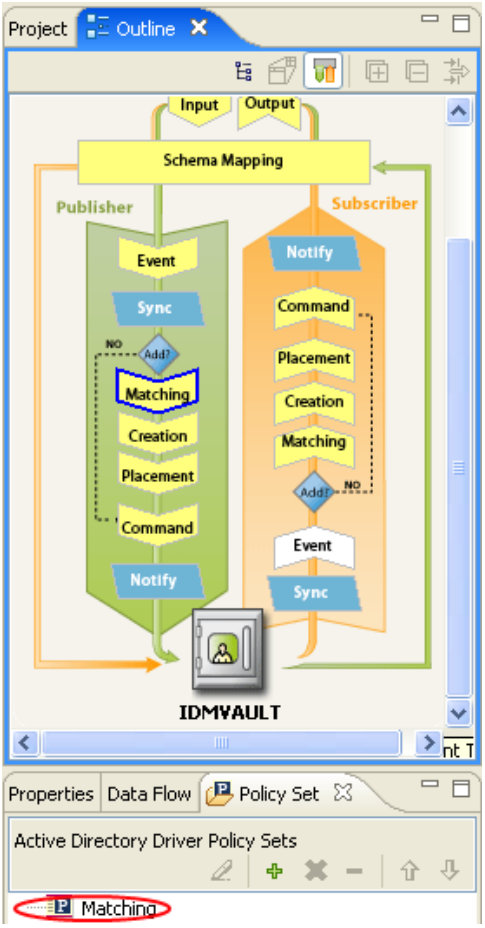
4 ポリシーフローで一致ポリシーをダブルクリックすることもできます。

5 ポリシーを選択してから、［編集］をクリックします。



ポリシーセット

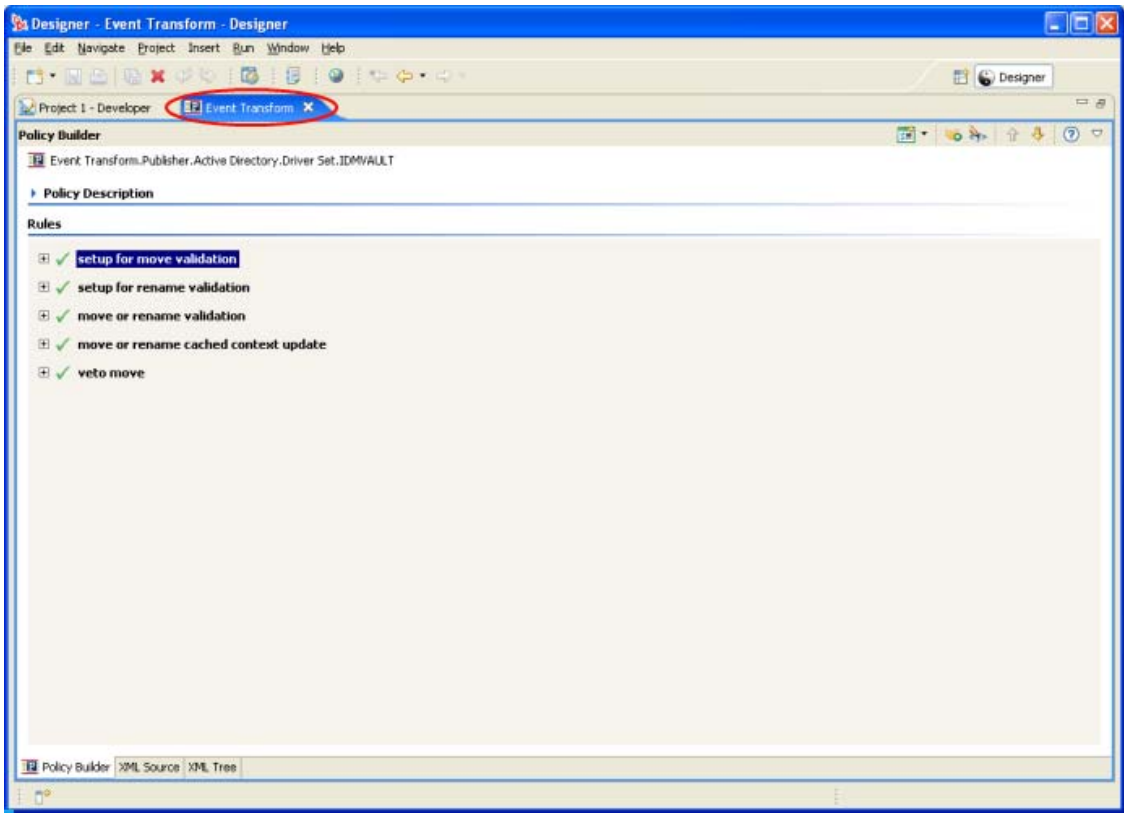
1 ポリシーセットでポリシーを右クリックし、［編集］をクリックします。



2 ポリシーセットでポリシーを選択して、［*Edit the policy (* ポリシーの編集 *)*］アイコンをクリックすることもできます。

［ポリシービルダ］ウィンドウに情報をすべて表示してスクロールしなくて済むようにするには、ポリシータブをダブルクリックして、ポリシービルダのウィンドウを最大化します。ウィンドウを最小化するには、ポリシータブをダブルクリックします。

図 *2-1* ポリシービルダのフルスクリーン表示



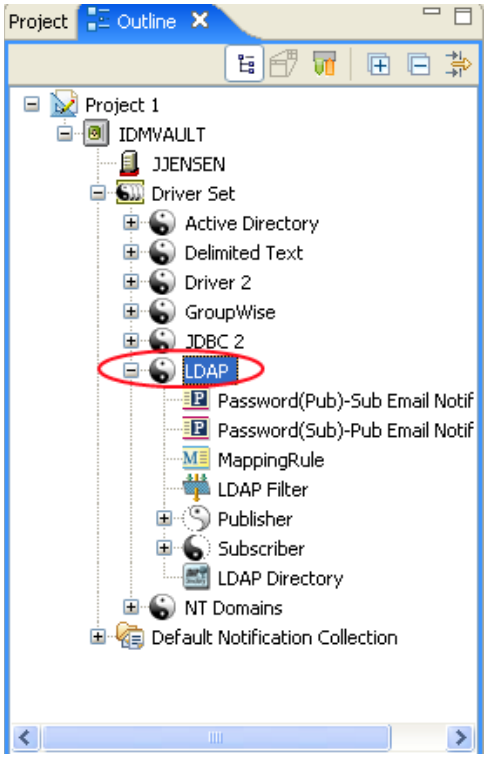
2.2.2 ポリシーの作成

ポリシーは接続システムにデータを送信します。ポリシーはポリシーセットを通じて作成されます。

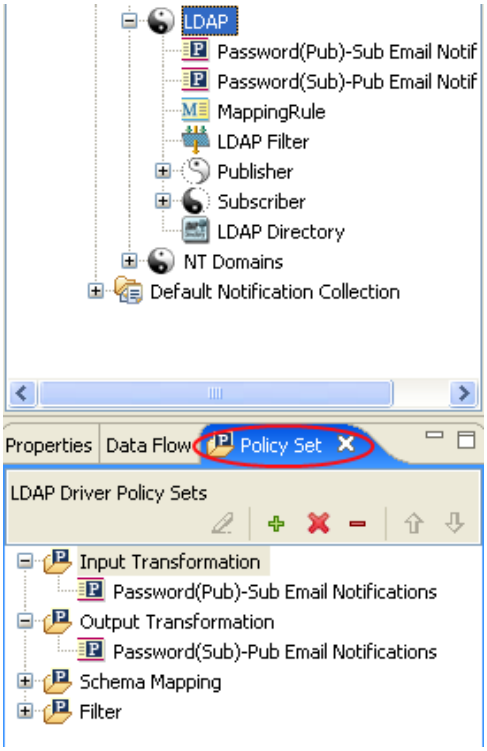
- 41 ページの 「ポリシーセットへのアクセス」
- 43 ページの 「ポリシーセットの使用」
- 45 ページの 「Add Policy Wizard (ポリシー追加ウィザード) の使用」

ポリシーセットへのアクセス

1 開いているプロジェクトの［Outline (アウトライン)］ビューからドライバオブジェクトを選択します。

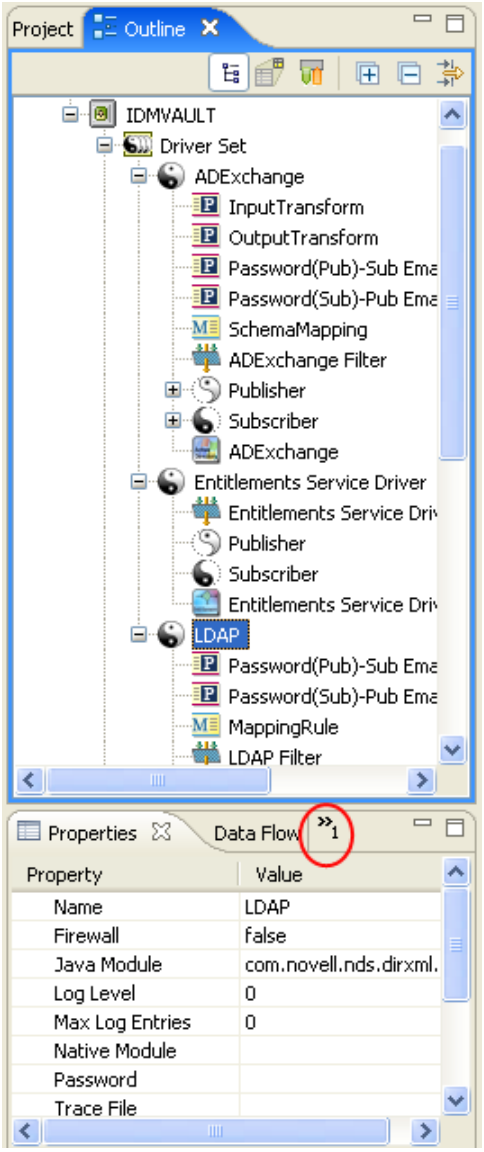


2［Policy Set (ポリシーセット)］タブを選択します。

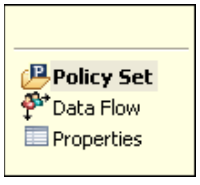


［Policy Set (ポリシーセット)］タブが表示されない場合は

1 二重矢印をクリックします。



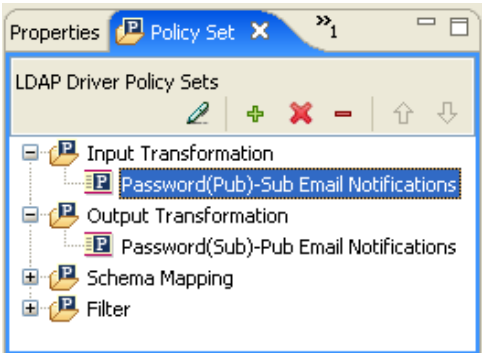
2 ［Policy Set (ポリシーセット)］を選択します。



ポリシーセットの使用

ポリシーセットには、ツールバーおよびポリシーリストが含まれます。

ポリシーリストには、選択したポリシーセットに含まれるすべてのポリシーが表示されます。変換中にリスト内のポリシーが上から順に実行されます。ツールバーには、ボタンおよびドロップダウンメニューがあります。これを使用して、リストに表示するポリシーの管理 (ポリシーの編集、追加、削除、名前変更、および処理順の変更など) ができます。



ポリシーセットのツールバー

ポリシーセットには、ポリシーのコピーが表示されます。ツールバーのボタンは、選択している項目に応じて有効または無効になります。さまざまなアイコンについての説明を次に示します。

表 *2-1* ポリシーセットのツールバー

操作	説明
Edit a policy (ポリシーの編集)	ポリシービルダを起動します。
Create or add a new policy to the Policy Set (新しいポリシーの作成またはポリシーセットへの追加)	Add Policy Wizard (ポリシー追加ウィザード) を起動します。
Remove and delete the selected policy (選択したポリシーを除外して削除)	プロジェクトからポリシーを削除します。
Remove the selected policy from the Policy Set, do not delete (選択したポリシーをポリシーセットから除外するがポリシー自体は削除しない)	選択したポリシーセットオブジェクトからポリシーを除外しますが、そのポリシー自体は削除しません。
Move the policy up the policy chain (ポリシーチェーン内でポリシーを上に移動)	ポリシーの処理順序を上に移動します。
Move the policy down the policy chain (ポリシーチェーン内でポリシーを下に移動)	ポリシーの処理順序を下に移動します。

キーボード操作

ポリシーセット内では、マウスを使用するのと同じようにキー操作でも移動できます。サポートされているキー操作を次に示します。

表 2-2 キーボード操作

キー操作	説明
上矢印	選択したポリシーの処理順序を上に移動します。
下矢印	選択したポリシーの処理順序を下に移動します。
Delete	プロジェクトからポリシーを削除します。
マイナス記号	選択したポリシーセットからポリシーを除外しますが、そのポリシー自体は削除されません。
プラス記号	Add Policy Wizard (ポリシー追加ウィザード) を起動します。
Ctrl+Z	最後の操作を元に戻します。
Ctrl+Y	最後の操作をやり直します。

Add Policy Wizard (ポリシー追加ウィザード) の使用

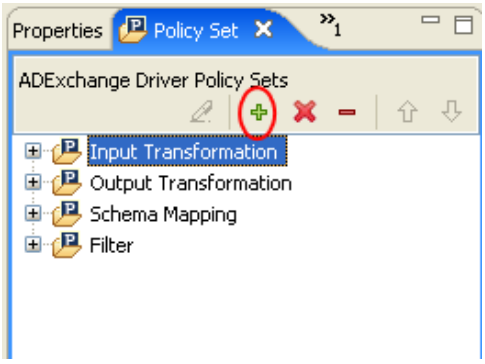
Add Policy Wizard (ポリシー追加ウィザード) は、ツールバーの［Create or add a new policy to the Policy Set (新しいポリシーの作成またはポリシーセットへの追加)］アイコンをクリックすると起動します。Add Policy Wizard (ポリシー追加ウィザード) で実行できる作業は、次のとおりです。

- 46 ページの 「ポリシーの作成」
- 48 ページの 「ポリシーのコピー」
- 49 ページの 「ポリシーへのリンク」

Add Policy Wizard (ポリシー追加ウィザード) を起動するには

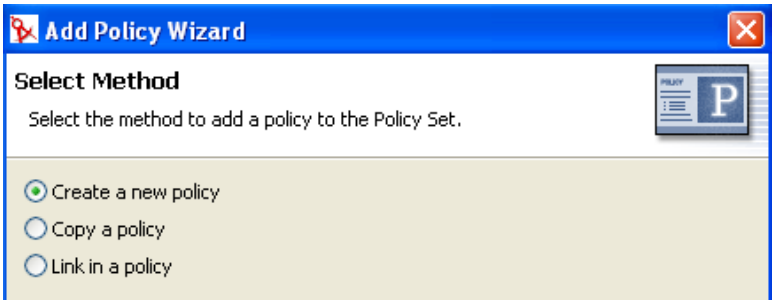
1 ［Outline (アウトライン)］ビューでドライバを選択します。

2 ポリシーセットでポリシーセット項目を選択し、ツールバーの［Create or add a new policy to the Policy Set (新しいポリシーの作成またはポリシーセットへの追加)］アイコンをクリックします。

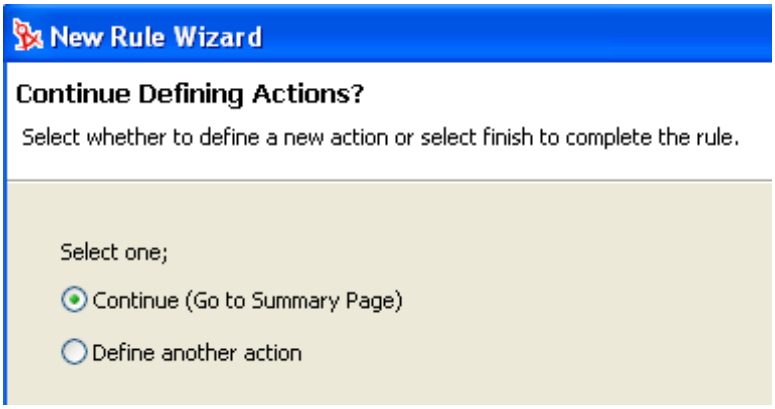


ポリシーの作成

1 Add Policy Wizard (ポリシー追加ウィザード) で、［新しいポリシーの作成］を選択し、［次へ］をクリックします。



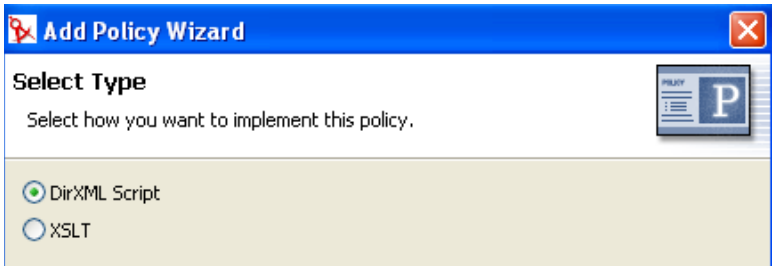
2 ポリシー名を入力します。



3 デフォルトコンテナをそのまま使用するか、ポリシーの作成先にするドライバオブジェクト、発行者オブジェクト、または購読者オブジェクトを参照し、選択します。

作成先は、ポリシーを整理する方法に応じて決定します。デフォルトでは、ポリシーは、Add Policy Wizard (ポリシー追加ウィザード) が起動されたときに［Outline (アウトライン)］タブで選択されていたコンテナオブジェクトに配置されます。たとえば、［Outline (アウトライン)］タブで発行者オブジェクトに移動し、その後でポリシーをポリシーセットに追加する場合、ポリシーはデフォルトで発行者コンテナに配置されます。別のコンテナにポリシーを作成する場合は、この設定を変更できます。たとえば、ダミードライバの下にポリシーライブラリを設定し、共通ポリシーをすべてこのドライバの下に配置して、他のドライバからポリシーを簡単に参照できるようにできます。これでポリシーを共用できます。ポリシーの変更が必要な場合は、1 回だけ変更すれば済みます。ポリシーが複数のドライバで再利用されない場合は、通常、そのポリシーをドライバ、またはそれを使用しているチャネルの下に作成します。

4 実装するポリシーのタイプを選択します。ポリシーのタイプは、デフォルトで［DirXML スクリプト］になります。DirXML® スクリプトを使用しない場合は、［XSLT］または［スキーマのマッピング］を選択できます。



5 ［終了］をクリックします。

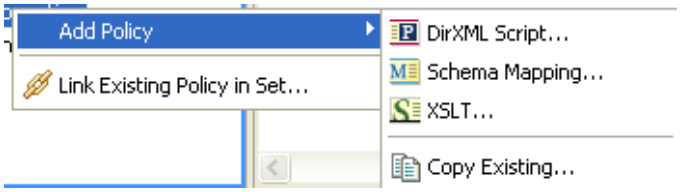
スキーママッピングポリシーセットが選択されている場合は、スキーママッピング用の追加オプションを利用できます。拡張されたポリシーセットに新しいポリシーが表示されます。

ポリシーセットを右クリックしてポリシーを追加することもできます。

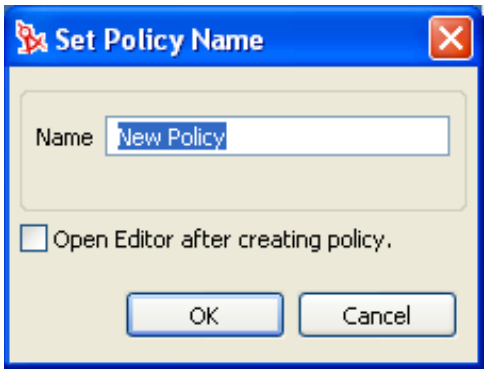
1 ポリシーセット (たとえば、入力変換セット) を右クリックします。

2 ［Add Policy (ポリシーの追加)］を選択します。

3 ポリシーを実装する方法 (［DirXML スクリプト］、［スキーマのマッピング］、［XSLT］、または［Copy Existing (既存のポリシーをコピー)］) を選択します。



4 ポリシーに名前を付けます。

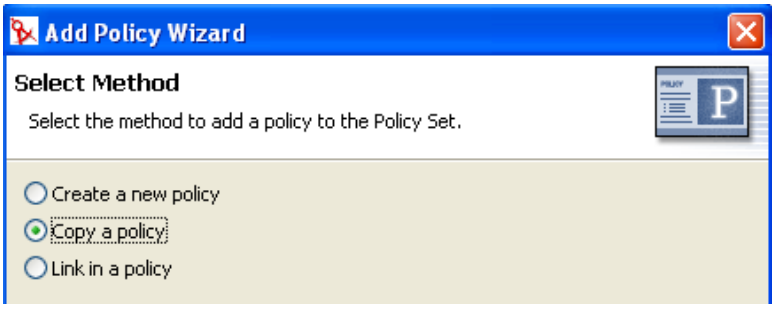


5 ［Open Editor after creating policy (ポリシーの作成後にエディタを開く)］をクリックします。

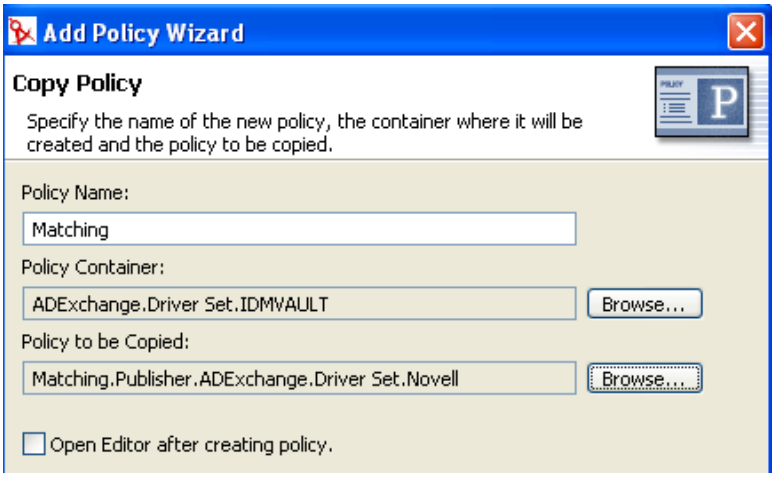
6 ［OK］をクリックします。

ポリシーのコピー

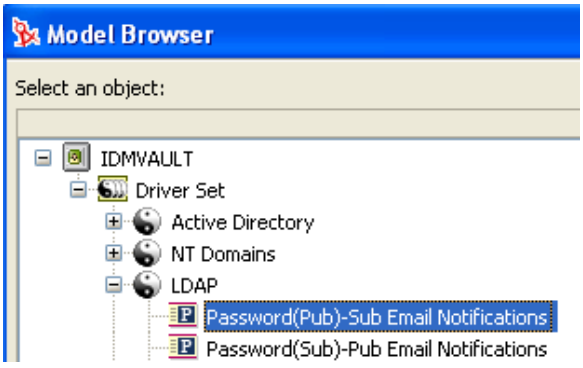
1 Add Policy Wizard (ポリシー追加ウィザード) で、［Copy a policy (ポリシーのコピー)］を選択し、［次へ］をクリックします。



2 ポリシーに名前を付けます。



3 デフォルトコンテナをそのまま使用するか、ポリシーの作成先にするドライバオブジェクト、発行者オブジェクト、または購読者オブジェクトを選択します。

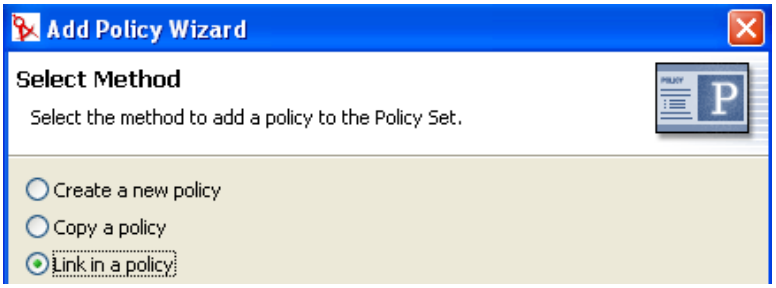


4 コピーするポリシーを参照して選択し、［OK］をクリックします。

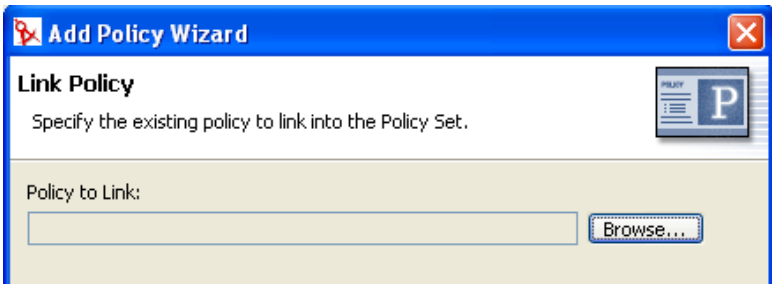
5 ［終了］をクリックして、選択したポリシーをコピーします。

ポリシーへのリンク

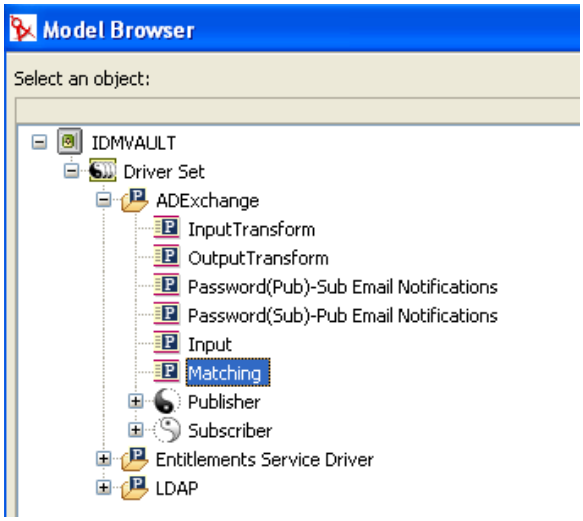
1 Add Policy Wizard (ポリシー追加ウィザード) で、[Link in a policy (ポリシーへのリンク)] を選択し、[次へ] をクリックします。



2 [参照] をクリックし、モデルブラウザを起動します。



3 ポリシーセットにリンクするポリシーオブジェクトをブラウズして選択し、[OK] をクリックします。



ポリシーをポリシーセットにリンクしても、新しいポリシーオブジェクトは作成されません。代わりに、既存のポリシーへの参照が追加されます。現在のアイデンティティボールト内の既存のポリシーであればどれでも参照できます。ポリシーは、現在のドライバオブジェクト内に含まれている必要はありませんが、ポリシーのタイプは、リンク先のポリシーセットで有効でなければなりません。たとえば、スキーママッピングポリシーを入力ポリシーセットにリンクすることはできません。

すべてのポリシーを表示しているときに、ポリシーをポリシーセットにリンクすることはできません。

4 ［終了］をクリックして、選択したポリシーにリンクします。

2.2.3 ルールの作成

ルールは、定義したアクションが実行されるために満たされていなければならない条件のセットです。ルールは、条件グループ、条件、およびアクションから作成されます。

ルールは、4 つの方法で作成できます。

- 50 ページの 「新しいルールの作成」
- 55 ページの 「事前定義されたルールの使用」
- 56 ページの 「既存のルールの包含」
- 57 ページの 「XML ファイルからのポリシーのインポート」

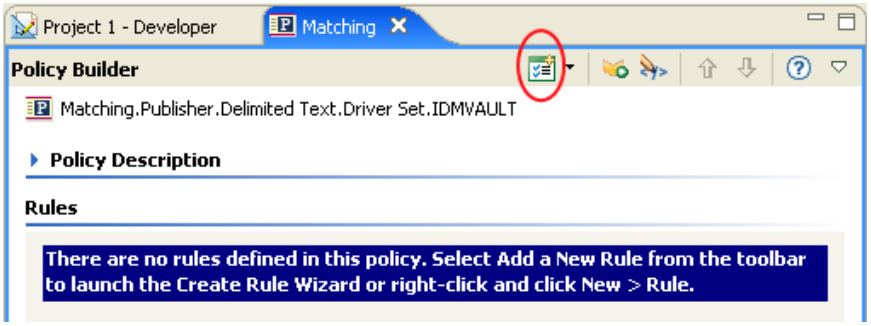
新しいルールの作成

ルールを作成する場合は、条件グループ、条件、およびアクションを作成します。個々のルールは、条件、アクション、および引数で構成されます。詳細については、各項目を作成するときに［ヘルプ］アイコン をクリックします。ヘルプファイルには、使用されている項目の定義および例が含まれます。

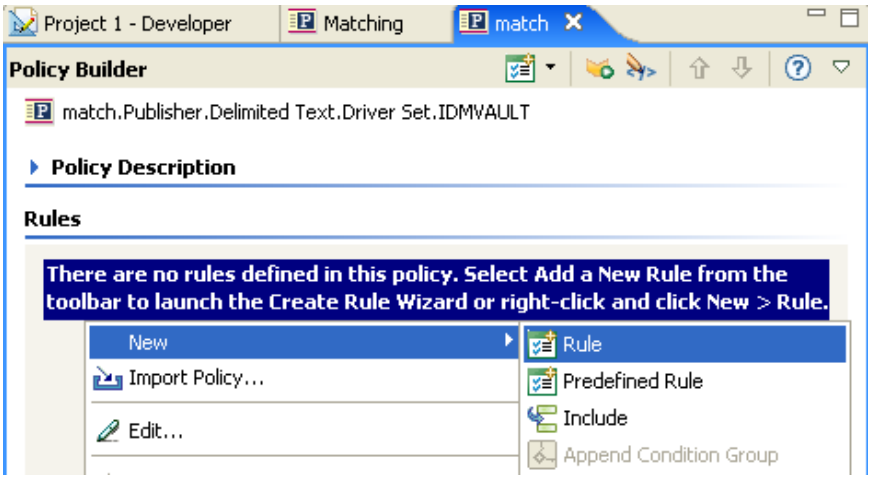
- 50 ページの 「ルールの作成」
- 54 ページの 「条件グループの作成」
- 54 ページの 「条件の作成」
- 55 ページの 「アクションの作成」

ルールの作成

1 ポリシービルダのツールバーから、［Rule (ルール)］を選択します。

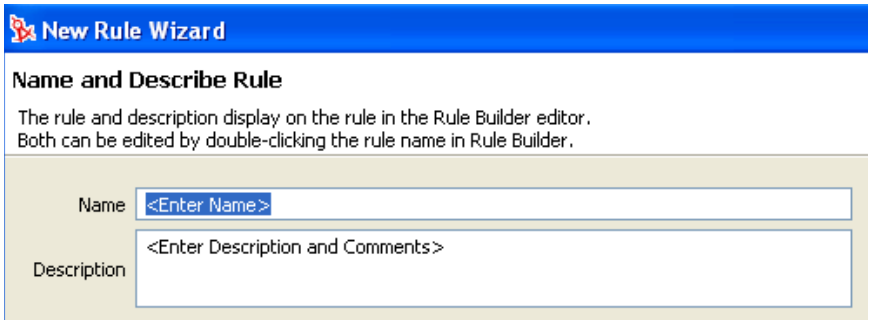


右クリックして、[New (新規作成)] > [Rule (ルール)] の順にクリックすることもできます。

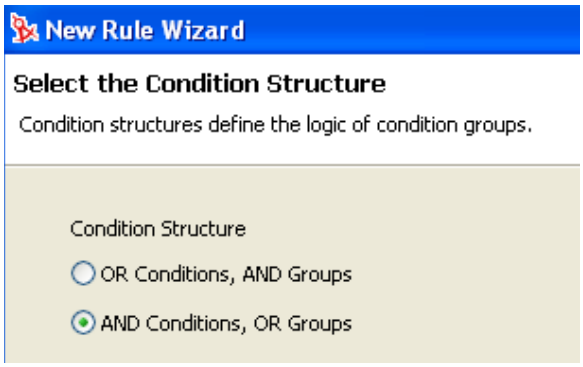


いずれかのオプションで、Create Rule Wizard (ルール作成ウィザード) を起動します。

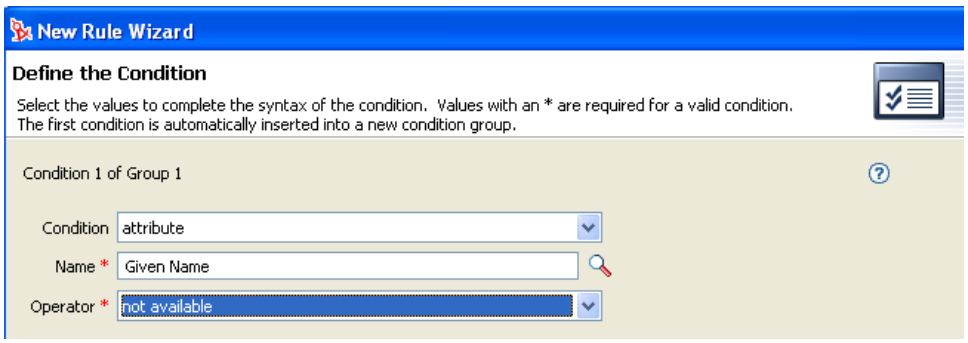
2 ルールの名前を指定して、[次へ] をクリックします。



3 条件構造 ([OR Conditions, AND Groups (OR 条件、AND グループ)]、または [AND Conditions, OR Groups (AND 条件、OR グループ)]) を選択し、[次へ] をクリックします。

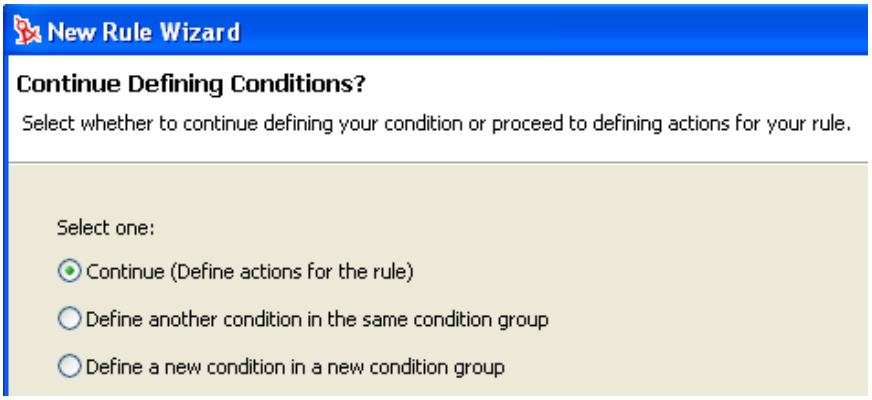


4 必要な条件を選択し、適切な情報を指定し、［次へ］をクリックします。

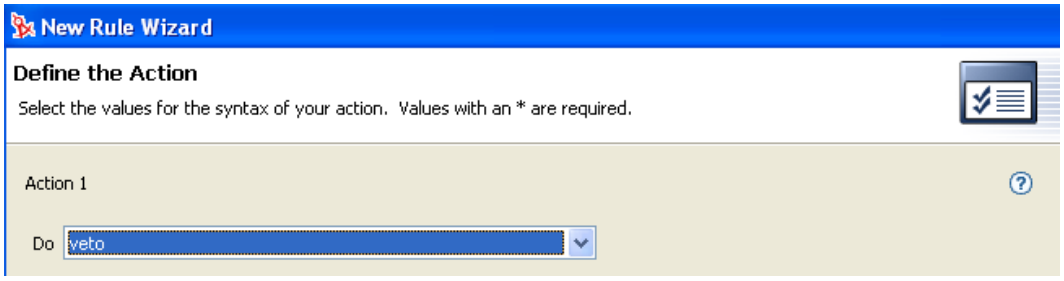


作成できる個々の条件については、［ヘルプ］アイコン をクリックします。

5 この時点で、追加条件または条件グループを定義できます。この例では、条件は 1 つだけにします。［続行］を選択し、［次へ］をクリックします。

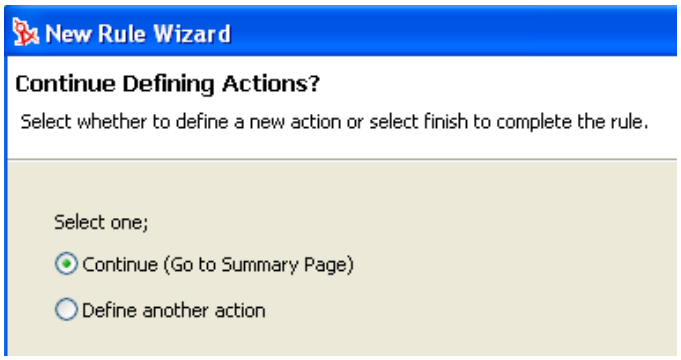


6 必要なアクションを選択し、［次へ］をクリックします。

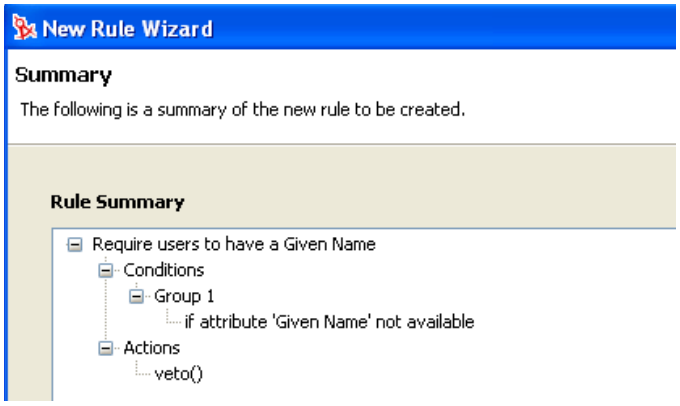


作成できる個々のアクションについては、［ヘルプ］アイコン をクリックします。

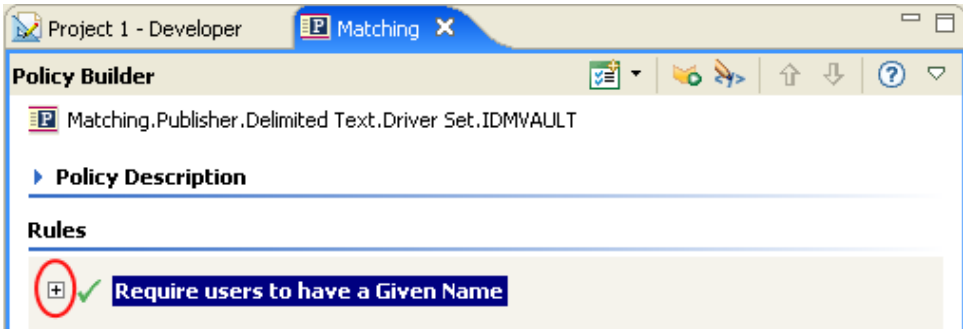
7 この時点で、追加アクションを定義できます。この例では、アクションは 1 つだけにします。［続行］を選択し、［次へ］をクリックします。



8 概要のページに、作成したルールが表示されます。［終了］をクリックし、ルールの作成を完了します。

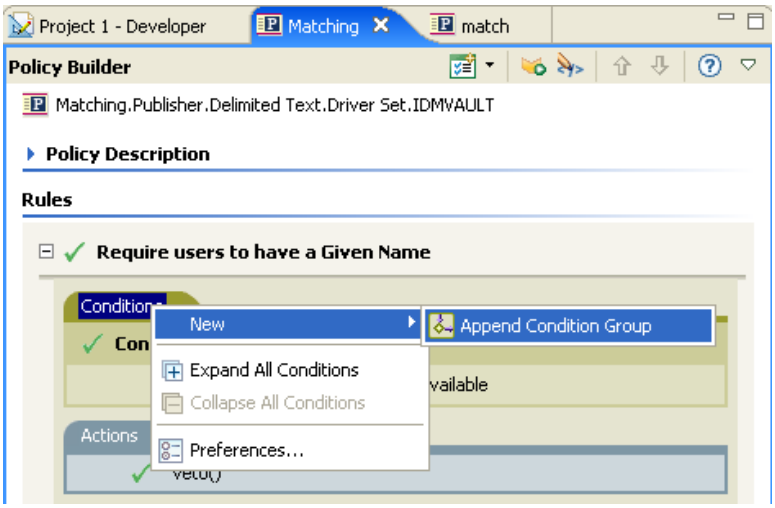


ルールのビューを展開または縮小表示するには、プラスまたはマイナス記号をクリックします。



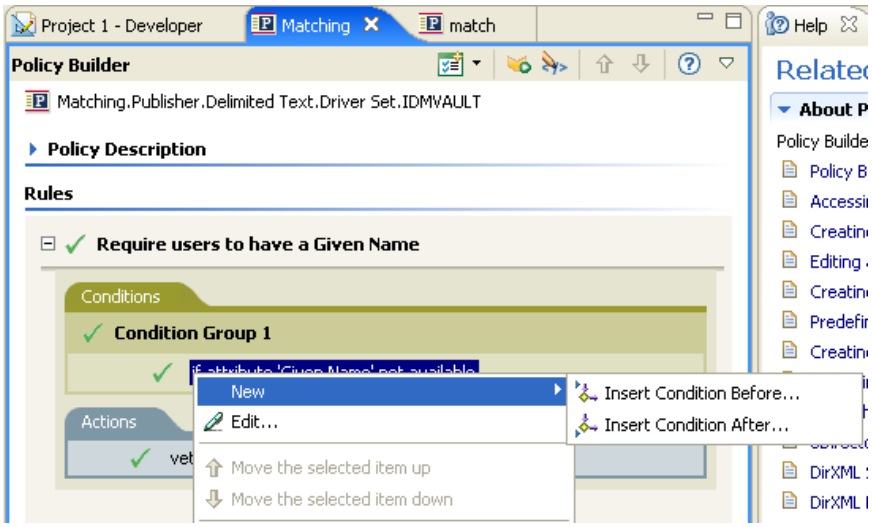
条件グループの作成

1 ［条件］タブを右クリックするか、［Conditional Group (条件グループ)］を右クリックし、［New (新規作成)］> ［Append Condition Group (条件グループの追加)］の順にクリックします。



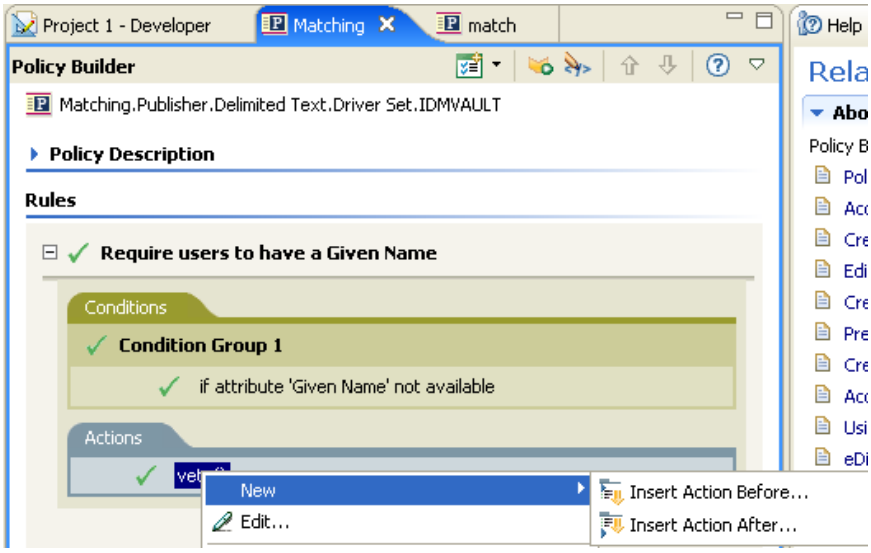
条件の作成

1 条件を右クリックし、［New (新規作成)］> ［Insert Condition Before (条件を前に挿入)］または ［Insert Condition After (条件を後に挿入)］の順に選択します。



アクションの作成

1 アクションを右クリックし、［New (新規作成)］> ［Insert Action Before (アクションを前に挿入)］または［Insert Action After (アクションを後に挿入)］の順にクリックします。

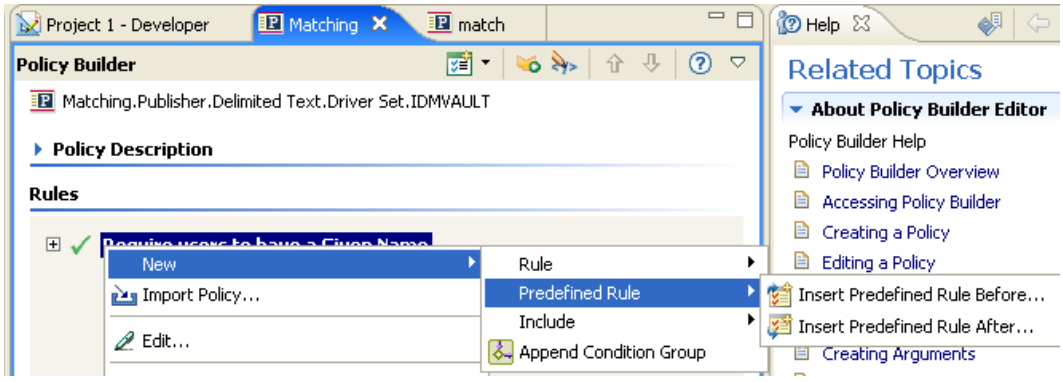


事前定義されたルールの使用

Designer には、事前定義されたルールのリストが用意されています。これらのルールをインポートすることで、ルールを自分で作成する場合と同様に使用できます。

1 ポリシービルダ内を右クリックし、［New (新規作成)］> ［Predefine Rules (事前定義されたルール)］> ［Insert Predefined Rule Before (事前定義されたルールを前に挿入)］または［Insert Predefined Rule After (事前定義されたルールを後に挿入)］の順に選択します。

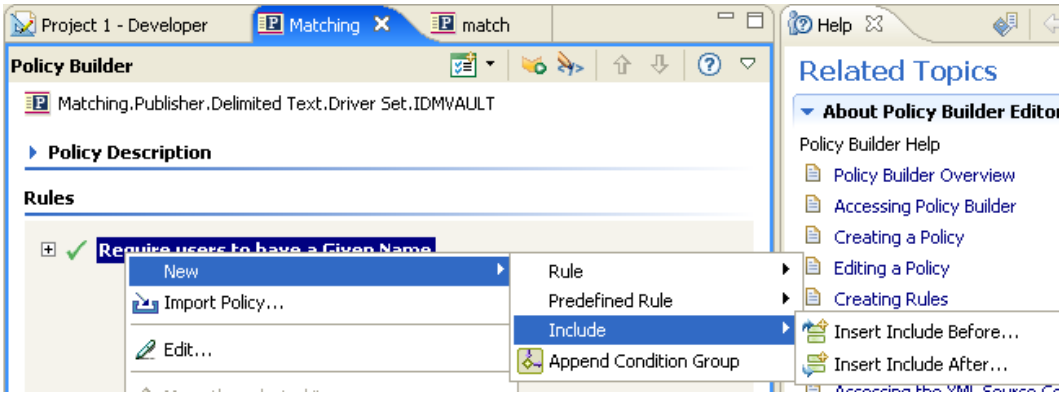
詳細については、72 ページのセクション 2.2.6「事前定義されたルールの使用」を参照してください。



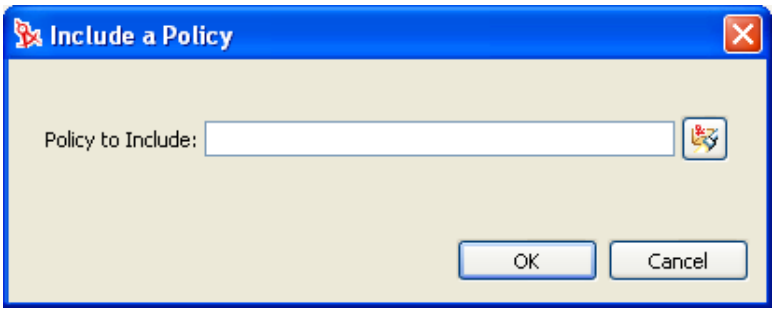
既存のルールの包含

Designer では、別のポリシーのルールを含めることができます。

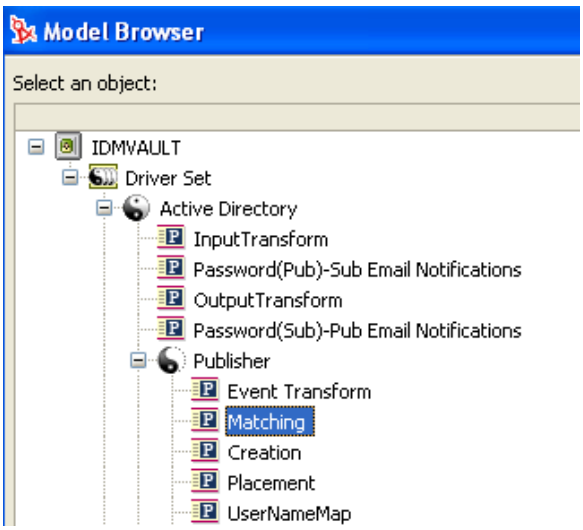
1 ポリシービルダ内を右クリックし、［New (新規作成)］> ［Include (包含)］> ［Insert Include Before (包含を前に挿入)］または［Insert Include After (包含を後に挿入)］の順にクリックします。



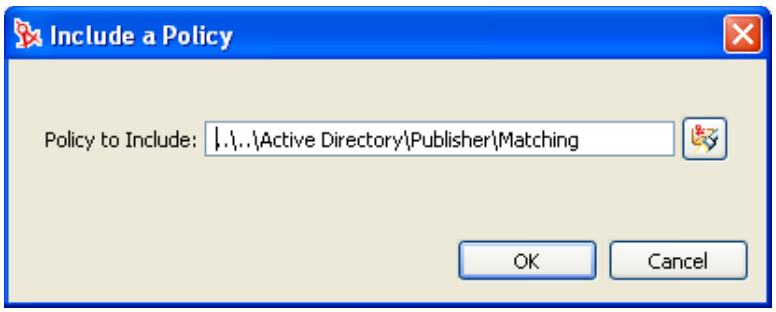
2 参照アイコンをクリックします。



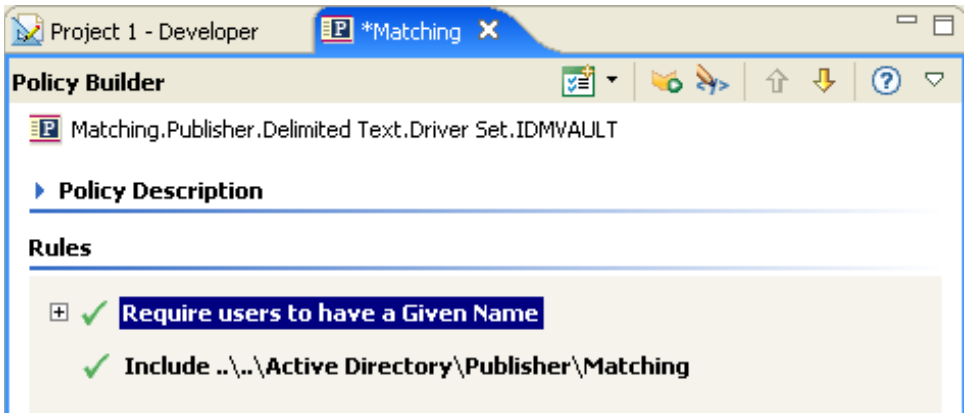
3 含めるポリシーを参照して選択し、［OK］をクリックします。



4 フィールドに、ポリシーへのパスが挿入されます。［OK］をクリックします。



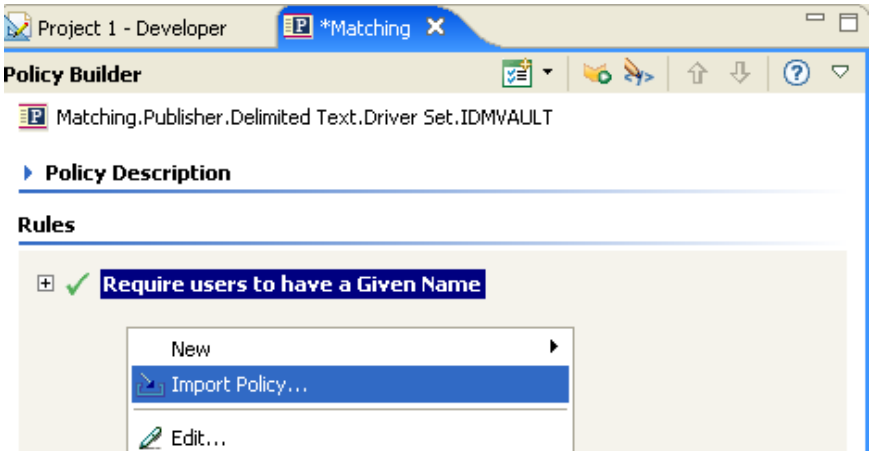
ルールは、元のルールへのリンクです。この場所でルールを編集することはできません。変更する場合は元のルールにアクセスします。



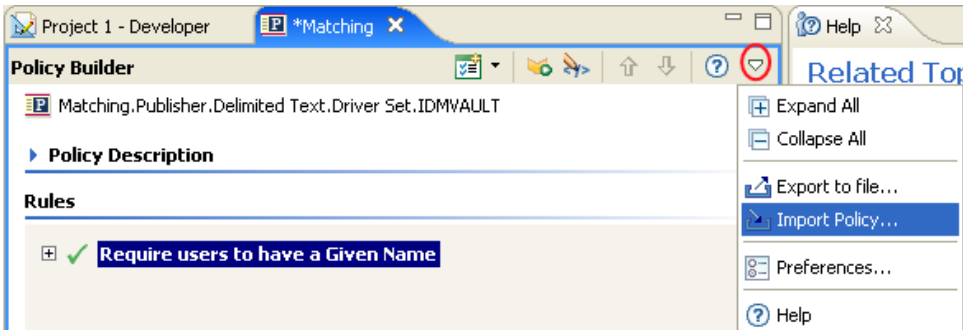
XML ファイルからのポリシーのインポート

ルールおよびポリシーは、XML ファイルとして保存できます。使用するルールまたはポリシーを含むファイルがある場合、ポリシービルダでファイルをインポートできます。

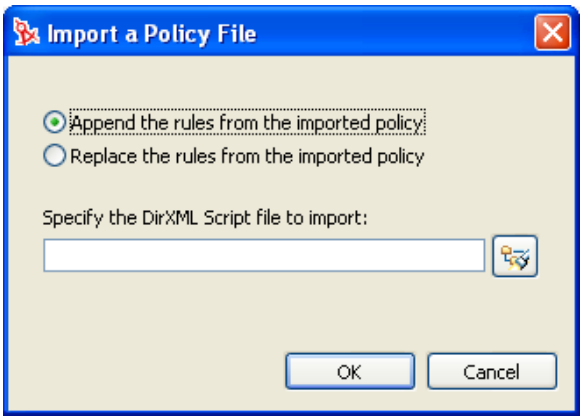
1 ポリシービルダで、右クリックして［ポリシーのインポート］を選択します。



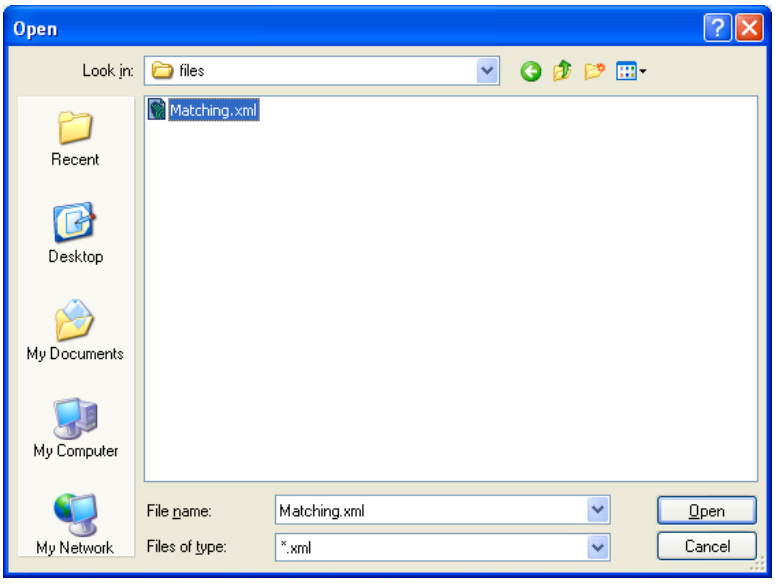
ツールバーのドロップダウンリストから［ポリシーのインポート］アイコンを選択することもできます。



2 ［インポートされたポリシーからルールを追加する］または［Replace the rules from the imported policy (インポートされたポリシーのルールと置き換える)］の 2 つのオプションのいずれかを選択します。



3 参照アイコンをクリックして、DirXML スクリプトを含むファイルを選択し、［開く］をクリックします。



4［OK］をクリックします。

2.2.4 引数の作成

引数ビルダでは、動的なグラフィカルインタフェースによって、ポリシービルダで使用する複雑な引数の式を作成できます。引数ビルダにアクセスするには、61 ページの 「引数ビルダ」を参照してください。

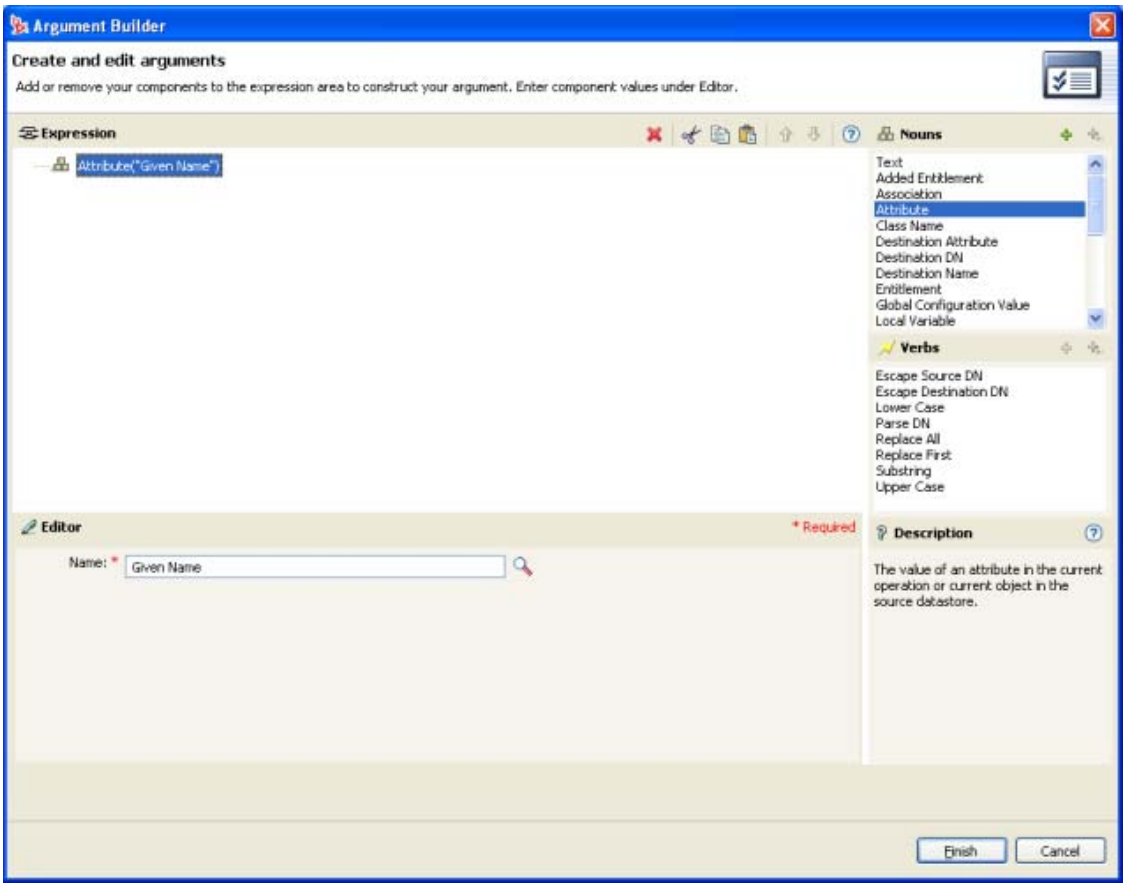
引数はアクションによって動的に使用されるもので、実行時に展開されるトークンから派生します。

トークンは、名詞と動詞の 2 つに分類できます。名詞トークンは、現在の操作、ソースやターゲットのデータストア、または外部ソースなどから派生する値を展開します。動詞トークンは、そのトークンのサブオーディネイトにある他のトークンの連結された結果を変更します。

式を定義するには、値、オブジェクト、変数などの名詞トークンを 1 つ以上選択し、これらを「部分文字列」、「エスケープ」、「大文字」および「小文字」などの動詞トークンと組み合わせ、引数を作成します。複数のトークンを組み合わせることで、複雑な引数を作成できます。

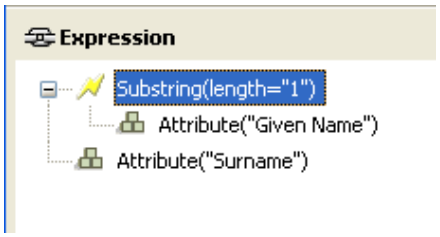
たとえば、引数を属性値に設定する場合は、属性の名詞を選択してから、属性名を選択します。

図 2-2 引数ビルダ



属性を一部だけ使用する場合は、属性の名詞を「部分文字列」動詞と組み合わせます。

図 *2-3* 式



名詞または動詞を追加した後は、エディタを使用して値を入力してから、次の名詞または動詞を追加できます。変更を適用するために［式］ペインを更新する必要はなく、次の操作を実行すれば変更は反映されます。

引数ビルダで参照できるトークンの詳細については、190 ページの 「名詞トークン」および204 ページの 「動詞トークン」を参照してください。

ほとんどの引数は引数ビルダで定義できますが、ポリシービルダ内の条件エディタおよびアクションエディタで使用されるビルダが他にもいくつかあります。各ビルダは、次に示すどのビルダでも再帰的に呼び出すことができます。

- 60 ページの 「アクションビルダ」
- 61 ページの 「引数ビルダ」
- 62 ページの 「一致属性ビルダ」
- 64 ページの 「アクションの引数コンポーネントビルダ」
- 65 ページの 「引数値リストビルダ」
- 66 ページの 「名前付き文字列ビルダ」
- 66 ページの 「条件の引数コンポーネントビルダ」
- 68 ページの 「パターン文字列ビルダ」

次に示すのは、各ビルダのアクセス方法です。

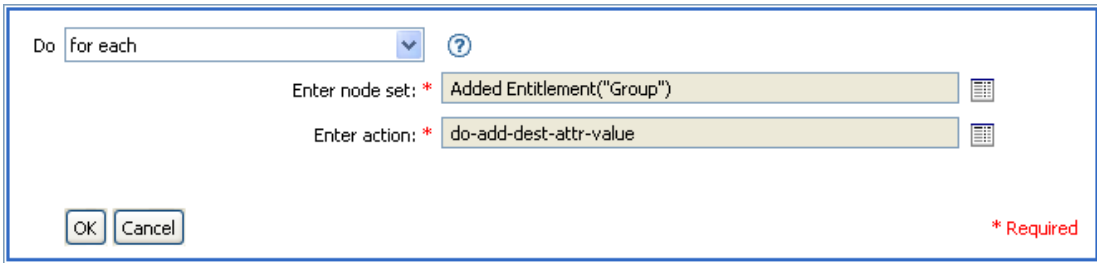
アクションビルダ

アクションビルダを起動するには、次のいずれかのアクションを選択し、［引数を編集する］アイコン をクリックします。

- 繰り返し (For Each)
- エンタイトルメントの実装

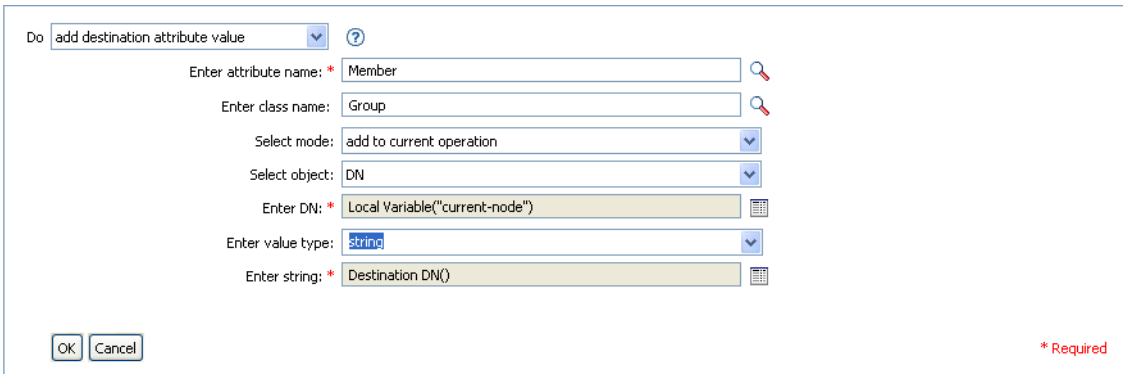
次の例では、ターゲット属性値の追加アクションが Group エンタイトルメントごとに実行され、現在の操作に追加されます。

図 2-4 *For Each* アクション



ターゲット属性値の追加アクションを定義するには、アクションビルダを起動するアイコンをクリックします。アクションビルダで、目的のアクションを定義します。次の例では、メンバー属性が、追加された各 Group エンタイトルメントのターゲットオブジェクトに追加されます。

図 2-5 引数アクションビルダ



引数ビルダ

引数ビルダを起動するには、次のいずれかのアクションを選択し、［引数を編集する］アイコン をクリックします。

- 144 ページの 「関連付けの追加」
- 145 ページの 「ターゲット属性値の追加」
- 146 ページの 「ターゲットオブジェクトの追加」
- 148 ページの 「ソース属性値の追加」
- 151 ページの 「XML テキストの追加」
- 152 ページの 「ターゲット属性値のクリア」(選択されたオブジェクトが［DN］または［関連付け］である場合)。
- 153 ページの 「ソース属性値のクリア」(選択されたオブジェクトが［DN］または［関連付け］である場合)。
- 156 ページの 「ターゲットオブジェクトの削除」(選択されたオブジェクトが［DN］または［関連付け］である場合)。

- 157 ページの 「ソースオブジェクトの削除」(選択されたオブジェクトが［DN］または［関連付け］である場合)。
- 157 ページの 「一致オブジェクトの検索」
- 159 ページの 「繰り返し (For Each)」
- 163 ページの 「ターゲットオブジェクトの移動」
- 164 ページの 「ソースオブジェクトの移動」
- 165 ページの 「操作属性の再フォーマット」
- 166 ページの 「関連付けを削除」
- 167 ページの 「ターゲット属性値の削除」
- 168 ページの 「ソース属性値の削除」
- 169 ページの 「ターゲットオブジェクトの名前変更」(選択されたオブジェクトが［DN］または［関連付け］および［文字列を入力］である場合)。
- 170 ページの 「ソースオブジェクトの名前変更」(選択されたオブジェクトが［DN］または［関連付け］および［文字列を入力］である場合)。
- 174 ページの 「ターゲット属性値の設定」(選択されたオブジェクトが［DN］または［関連付け］であり、［値タイプを入力］が指定されていない場合)。
- 176 ページの 「ターゲットパスワードの設定」
- 176 ページの 「ローカル変数の設定」
- 178 ページの 「操作関連付けの設定」
- 178 ページの 「操作クラス名の設定」
- 178 ページの 「操作ターゲット DN の設定」
- 179 ページの 「操作プロパティの設定」
- 180 ページの 「操作ソース DN の設定」
- 180 ページの 「操作テンプレート DN の設定」
- 181 ページの 「ソース属性値の設定」
- 182 ページの 「ソースパスワードの設定」
- 184 ページの 「XML 属性の設定」
- 185 ページの 「ステータス」
- 187 ページの 「メッセージのトレース」

1 名詞および動詞を使用して引数を作成します。

名詞および動詞を組み合わせることで、目的の引数を作成できます。

2 ［終了］をクリックします。

一致属性ビルダ

一致属性ビルダでは、データストアに一致するオブジェクトが存在するかどうかを判断するために、157 ページの 「一致オブジェクトの検索」のアクションによって使用される属性および値を選択できます。

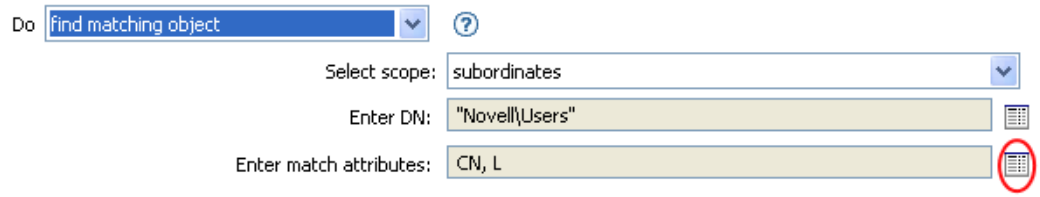
たとえば、共通名と場所に基づいてユーザを一致させるには

1 ［一致オブジェクトの検索］のアクションを選択します。

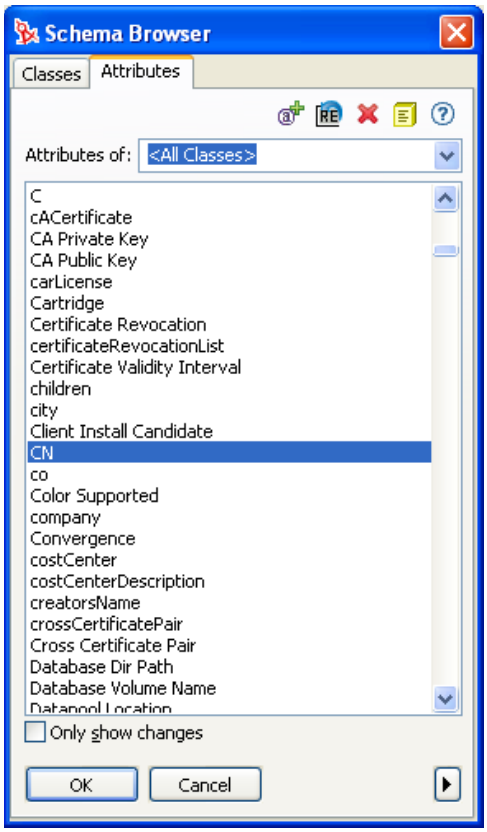
2 一致オブジェクトの検索の範囲を選択します。［エントリ］、［サブオーディネート］、または［サブツリー］から選択します。

3 検索の開始点となる DN を指定します。

4 ［Edit match attributes (一致属性の編集)］アイコン をクリックして、一致属性ビルダを起動します。



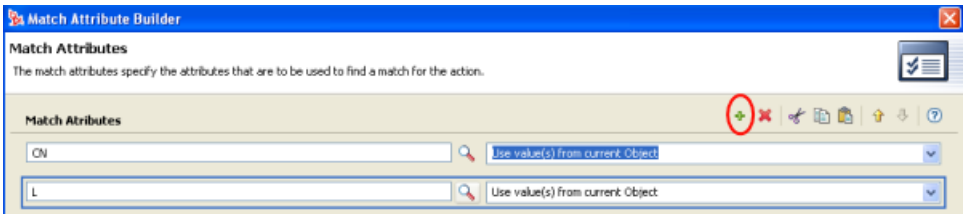
5 ［属性の参照］アイコン をクリックして、Schema Browser (スキーマブラウザ) を起動します。



6 ［属性］タブをクリックして、属性を参照して選択します。

7 ［OK］をクリックします。

属性を複数追加する場合は、［新規アイテムの追加］アイコン ✚ をクリックして新しい行を追加します。



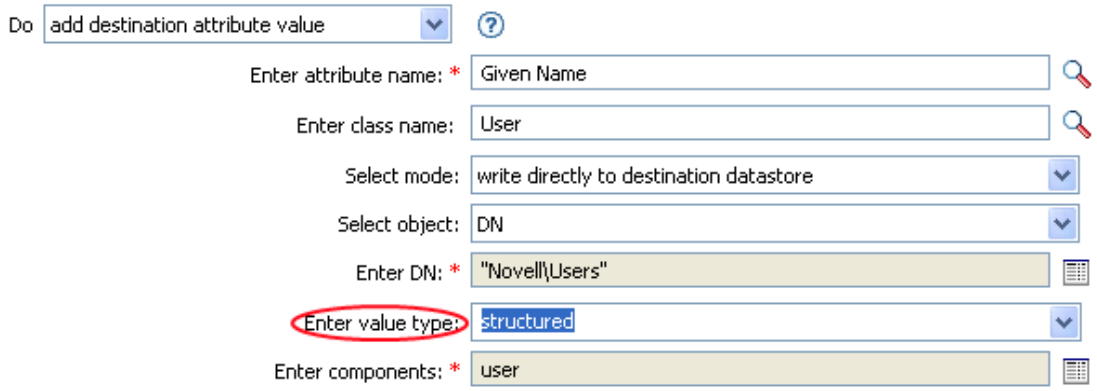
8 ［終了］をクリックします。

アクションの引数コンポーネントビルダ

アクションの引数コンポーネントビルダを起動するには、［値タイプを入力］で［構造］が選択されている状態で、次のいずれかのアクションを選択し、［Edits components (コンポーネントの編集)］アイコン をクリックします。

- 145 ページの 「ターゲット属性値の追加」
- 148 ページの 「ソース属性値の追加」
- 165 ページの 「操作属性の再フォーマット」
- 167 ページの 「ターゲット属性値の削除」
- 168 ページの 「ソース属性値の削除」
- 174 ページの 「ターゲット属性値の設定」
- 181 ページの 「ソース属性値の設定」

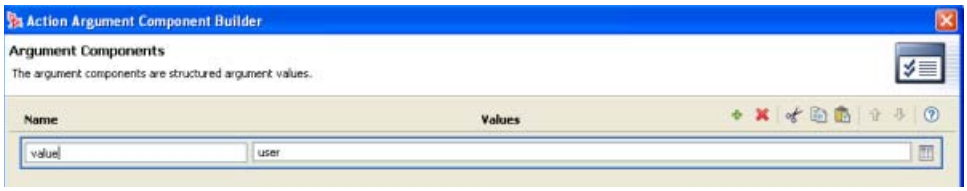
図 *2-6* ターゲット属性値の追加アクション



1 値タイプが［構造］に設定されている状態で、［コンポーネントを編集する］アイコン をクリックします。

2 アクションコンポーネントの値を作成します。

引数ビルダでは、値を入力するか、［引数を編集する］アイコン をクリックして値を作成できます。



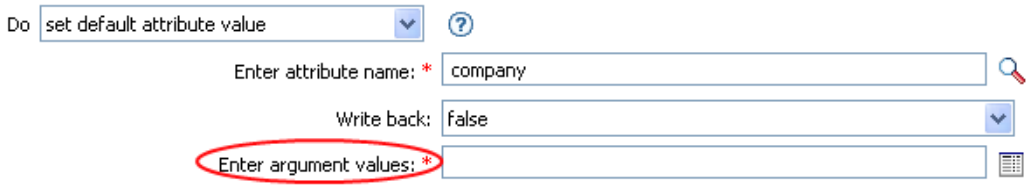
3 ［終了］をクリックします。

引数値リストビルダ

引数値リストビルダを起動するには、次のアクションを選択し、［引数を編集する］アイコン をクリックします。

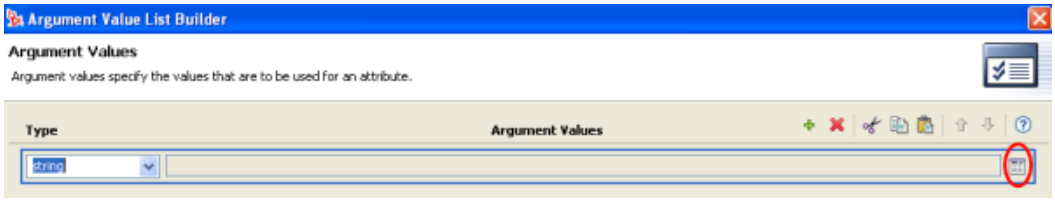
- デフォルト属性値の設定

図 2-7 デフォルト属性値の設定



1 値のタイプ (［counter (カウンタ)］、［DN］、［int (整数)］、［間隔］、［octet (オクテット)］、［state (状態)］、［文字列］、［構造］、［teleNumber (電話番号)］、［時間］) を選択します。

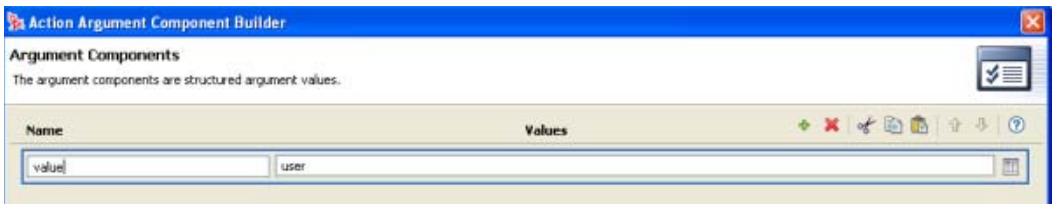
2 ［Edit the value lists (値リストを編集する)］アイコン をクリックします。



3 ［引数を編集する］アイコン をクリックします。

4 アクションコンポーネントの値を作成します。

引数ビルダでは、値を入力するか、［引数を編集する］アイコン をクリックして値を作成できます。



5 ［終了］をクリックします。

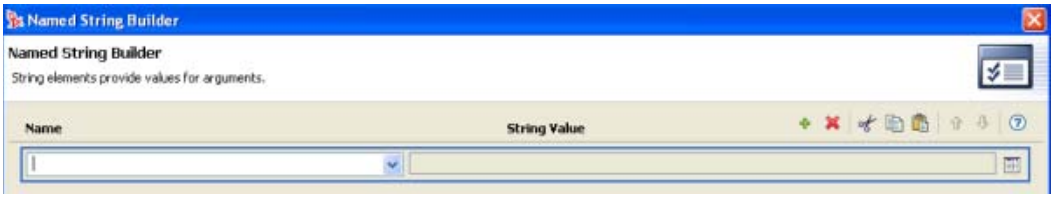
名前付き文字列ビルダ

名前付き文字列ビルダを起動するには、次のいずれかのアクションを選択し、［文字列を編集する］アイコン をクリックします。

- イベントの生成
- 電子メールの送信
- テンプレートから電子メールを送信

1 ドロップダウンリストから、文字列の名前を選択します。

2 ［引数を編集する］アイコン をクリックして引数ビルダを起動し、文字列の値を作成します。



3 ［終了］をクリックします。

電子メールの送信アクションの場合、名前付き文字列は電子メールの要素に対応します。

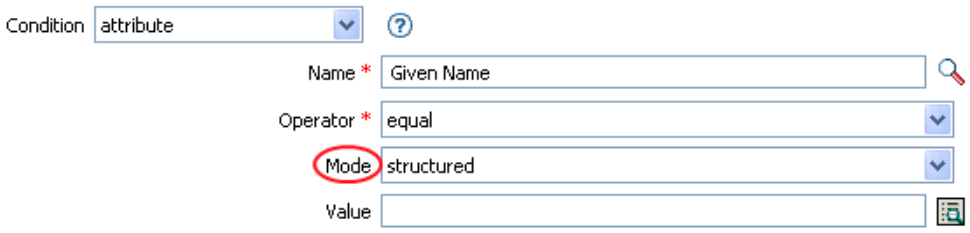
図 *2-8* 電子メールの送信アクションの電子メール要素

指定できる値の完全なリストは、名前付き文字列ビルダを起動するアクションに対応するヘルプファイルに含まれています。

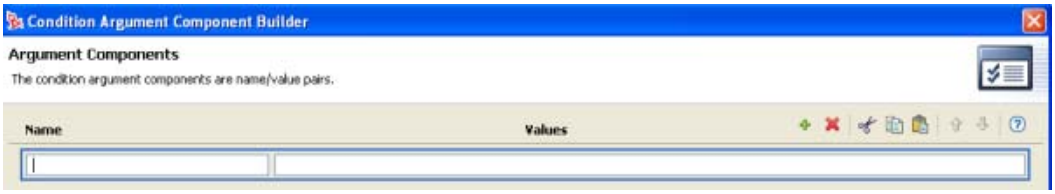
条件の引数コンポーネントビルダ

条件の引数コンポーネントビルダを起動するには、次のいずれかの条件を選択し、次に、［Launch ArgComponent Builder (引数コンポーネントビルダの起動)］アイコン を表示するために、［Mode］で［構造］を選択する必要があります。

- 「属性」条件
- 「ターゲット属性」条件
- 「ソース属性」条件
- 「操作属性」条件



1 条件コンポーネントの名前および値を指定します。

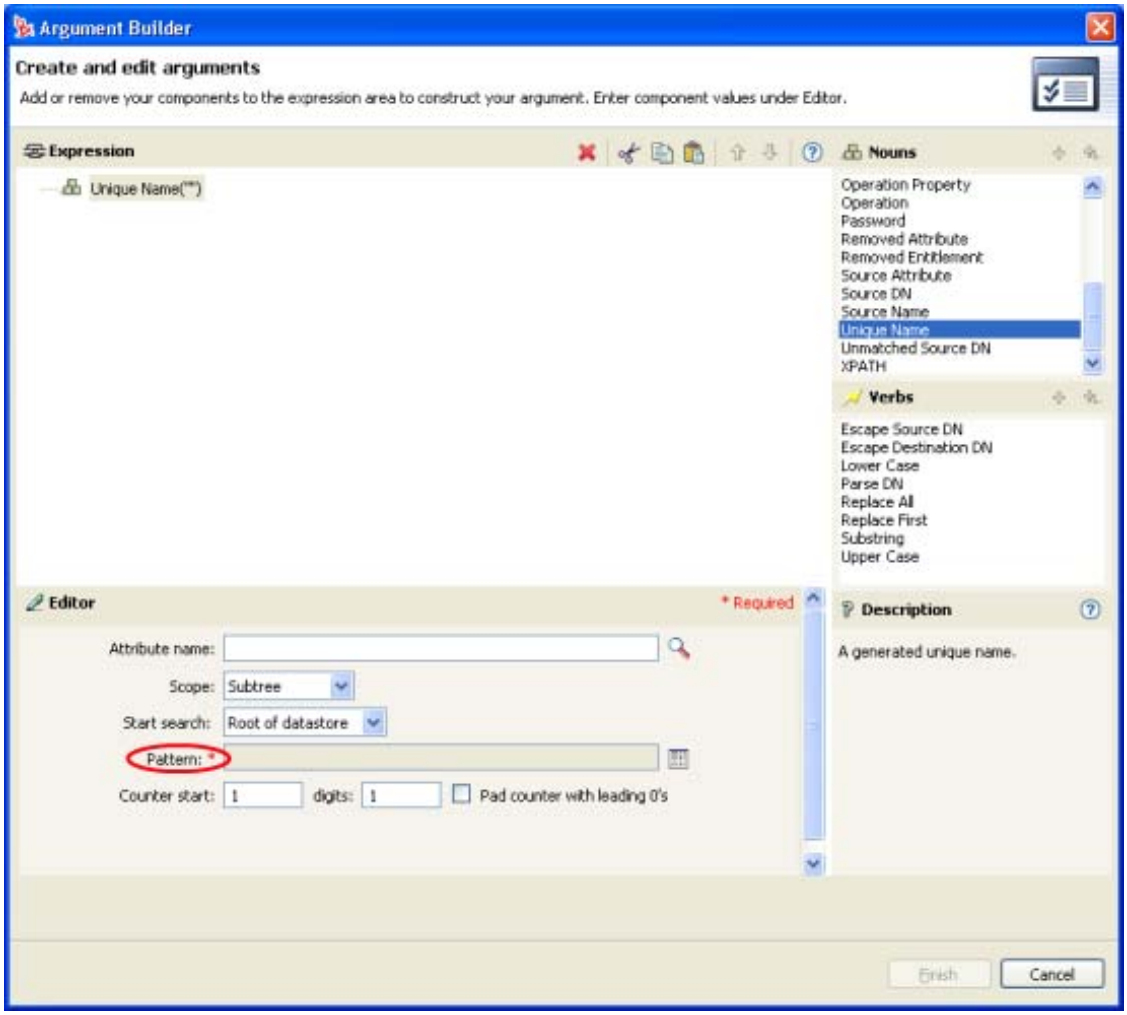


2 ［終了］をクリックします。

パターン文字列ビルダ

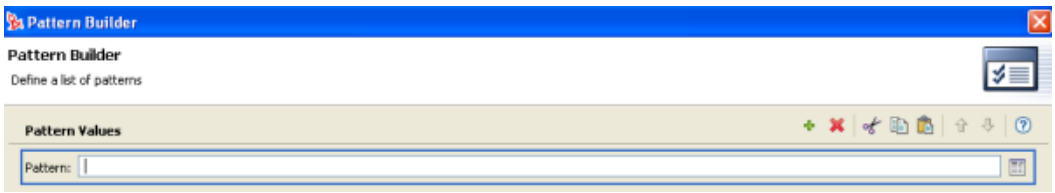
パターン文字列ビルダは、一意の名前トークンが選択されているときに引数ビルダのエディタから起動できます。引数ビルダのエディタペインに［パターン］フィールドが表示されるので、このフィールドをクリックして、パターン文字列ビルダを起動します。

図 2-9 引数ビルダ内の一意の名前トークン



1 ［Edit patterns (パターンを編集する)］アイコン をクリックして、パターンビルダを起動します。

2 パターンを指定するか、［引数を編集する］アイコン をクリックして引数ビルダを使用してパターンを作成します。

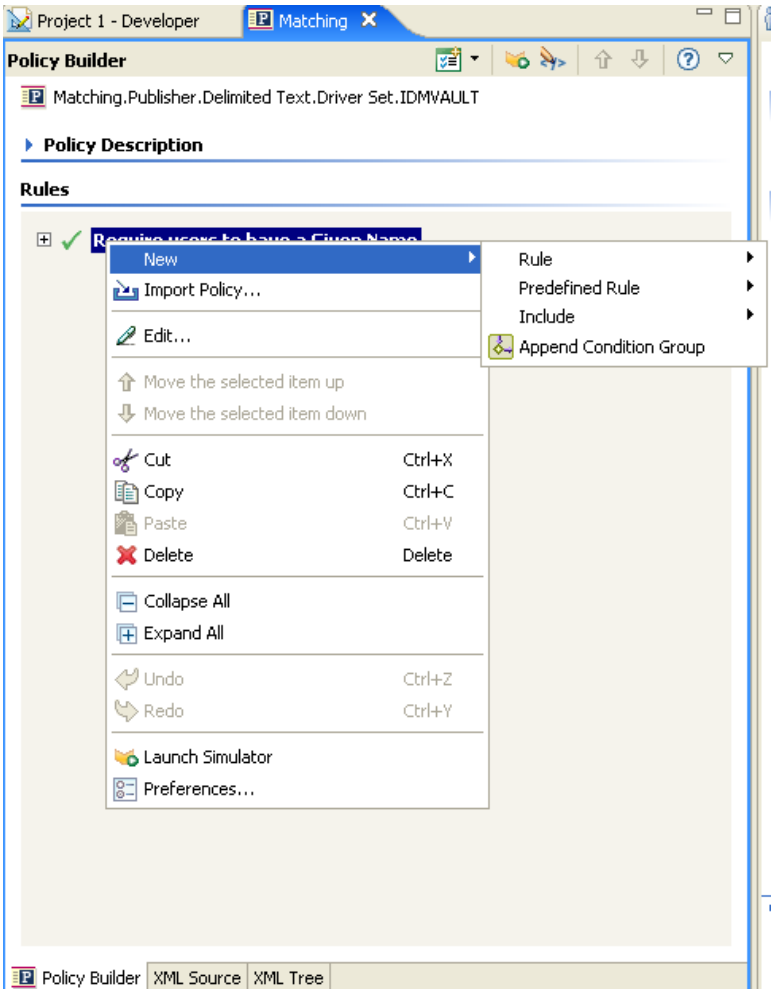


3 ［終了］をクリックします。

2.2.5 ポリシーの編集

ポリシービルダでは、ポリシーを作成および編集できます。ルール、条件、およびアクションは、ドラッグアンドドロップできます。その他の操作は、ポリシービルダのツールバーからアクセスします。コンテキストに応じたメニューを表示するには、項目を右クリックします。

図 *2-10* ポリシービルダのコンテキストメニューおよびツールバー



ポリシービルダのアクションおよびメニュー項目

次の表は、ポリシービルダのさまざまなアクションおよびメニュー項目のリストを示しています。

表 *2-3* ポリシービルダのアクションおよびメニュー項目

操作	説明
Collapse All (すべて縮小)	展開されているルールをすべて縮小します。
コピー	選択された項目をクリップボードにコピーします。

操作	説明
Copy and drop (コピーしてドロップ)	項目を選択し、キーを押しながらドラッグします。
切り取り	選択された項目を切り取り、クリップボードにコピーします。
削除	選択された項目を削除します。
Disable (無効)	ルール、条件、またはアクションを無効にします。✔ アイコンをクリックします。
Drag and drop (ドラッグアンドドロップ)	項目を選択し、再配置できます。項目を選択し、それを新しい場所にドラッグします。
編集	選択された項目を編集できます。ルールビルダを開くには、ルールを選択し、［編集］をクリックします。
有効	ルール、条件、またはアクションを有効にします。⛔ アイコンをクリックします。
すべて展開	各ルールの条件およびアクションを表示できるように、ルールをすべて展開します。
ポリシーのインポート	ファイルシステムからポリシーをインポートし、それをポリシーに追加するか、ポリシーのルールをすべて置き換えます。
Launch Simulator (シミュレータの起動)	ポリシーシミュレータを起動します。
Move and drop (移動してドロップ)	項目を選択し、移動できます。項目を選択して、ドラッグします。
Move the selected item down (選択された項目を下に移動)	項目をポリシーのリスト内で下に移動します。
Move the selected item down (選択された項目を上に移動)	項目をポリシーのリスト内で上に移動します。
［New (新規作成)］> ［条件グループ］	選択された項目の後に、新しい条件グループを作成します。
［New (新規作成)］> ［Include (対象項目)］	選択した項目の後に新しい対象項目を作成します。
［New (新規作成)］> ［Predefined Rule (事前定義されたルール)］	事前定義されたルールを挿入します。
［New (新規作成)］> ［Rule (ルール)］	選択した項目の後に新しいルールを作成します。
貼り付け	選択された項目の後に、クリップボードの内容を貼り付けます。
初期設定	情報の表示方法を変更できます。
選択	項目をクリックして選択します。

キーボード操作

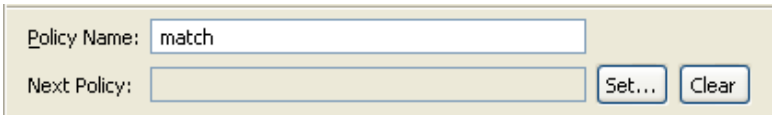
ポリシービルダ内では、マウスを使用するのと同じようにキー操作で移動できます。サポートされているキー操作を次に示します。

表 2-4 ポリシービルダでのキーボード操作

キー操作	説明
Ctrl+C	選択された項目をクリップボードにコピーします。
Ctrl+X	選択された項目を切り取り、クリップボードに追加します。
Ctrl+V	選択された項目の後に、クリップボードの内容を貼り付けます。
Delete	選択された項目を削除します。
左矢印	ルールノードを縮小します。
右矢印	ルールノードを展開します。
上矢印	上に移動します。
下矢印 >	下に移動します。
Ctrl+Z	元に戻す
Ctrl+Y	やり直し

ポリシーの名前変更

1. ［Outline (アウトライン)］ビューで、名前を変更するポリシーを選択します。
2. 右クリックして［プロパティ］を選択します。
3. ［ポリシー名］フィールドでポリシーの名前を変更します。



4. ［OK］をクリックします。

自分の作業の保存

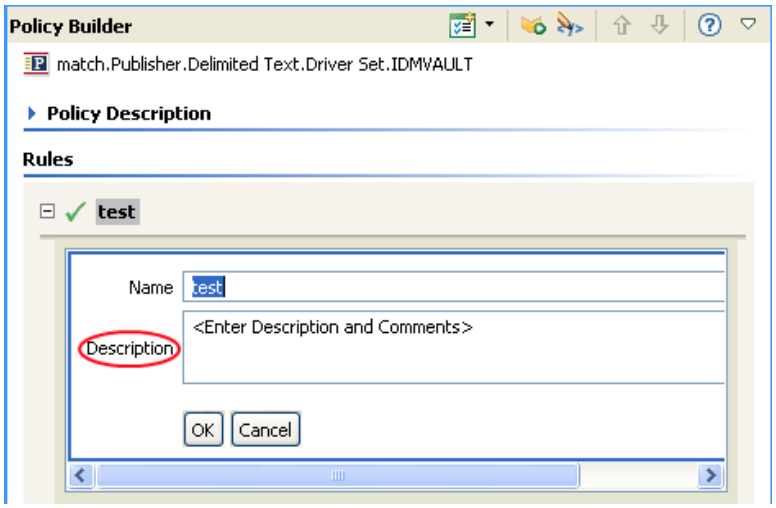
次のいずれかを実行します。

- メインメニューで、［ファイル］>［保存］(または［すべて保存］) の順にクリックします。
- エディタのタブで［X］をクリックして、エディタを閉じます。
- メインメニューの［ファイル］メニューで、［閉じる］を選択します。
- Ctrl+S キーを押します。

ポリシーの説明

［説明］フィールドは、ポリシーの機能についてのメモを追加する場所です。

図 *2-11* ポリシーの説明



2.2.6 事前定義されたルールの使用

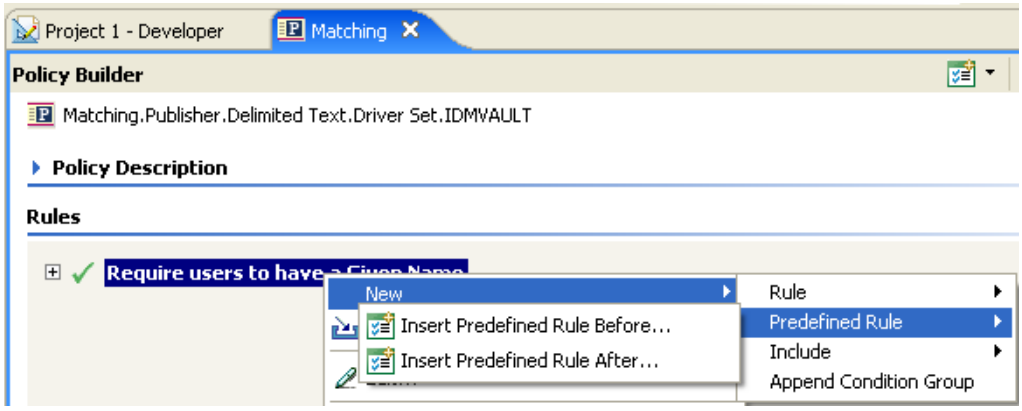
Designer には、20 の事前定義されたルールが備わっています。これらのルールをインポートすることで、ルールを自分で作成する場合と同様に使用できます。これらのルールには、管理者が使用する一般的なタスクが含まれています。ルールをカスタマイズするには、各自の環境に合わせた情報を指定する必要があります。

- 74 ページの 「コマンド変換 - 部署別コンテナの作成 - パート 1 とパート 2」
- 76 ページの 「コマンド変換 - 無効にする発行者の削除」
- 78 ページの 「作成 - 属性が必要」
- 79 ページの 「作成 - 発行者 - テンプレートの使用」
- 81 ページの 「作成 - デフォルト属性値の設定」
- 83 ページの 「作成 - デフォルトパスワードの設定」
- 85 ページの 「イベント変換 - スコープフィルタリング - サブツリーの組み込み」
- 86 ページの 「イベント変換 - スコープフィルタリング - サブツリーの除外」
- 88 ページの「入出力変換 - 電話番号の形式を (nnn) nnn-nnnn から nnn-nnn-nnnn に変更」
- 89 ページの「入出力変換 - 電話番号の形式を nnn-nnn-nnnn から (nnn) nnn-nnnn に変更」
- 91 ページの 「一致 - 発行者 (ミラーリング)」
- 92 ページの 「一致 - 購読者 (ミラーリング) - LDAP 形式」
- 94 ページの 「一致 - 属性値別」
- 96 ページの 「配置 - 発行者 (ミラーリング)」
- 98 ページの 「配置 - 購読者 (ミラーリング)-LDAP 形式」
- 99 ページの 「配置 - 発行者 (フラット)」
- 101 ページの 「配置 - 購読者 (フラット)-LDAP 形式」

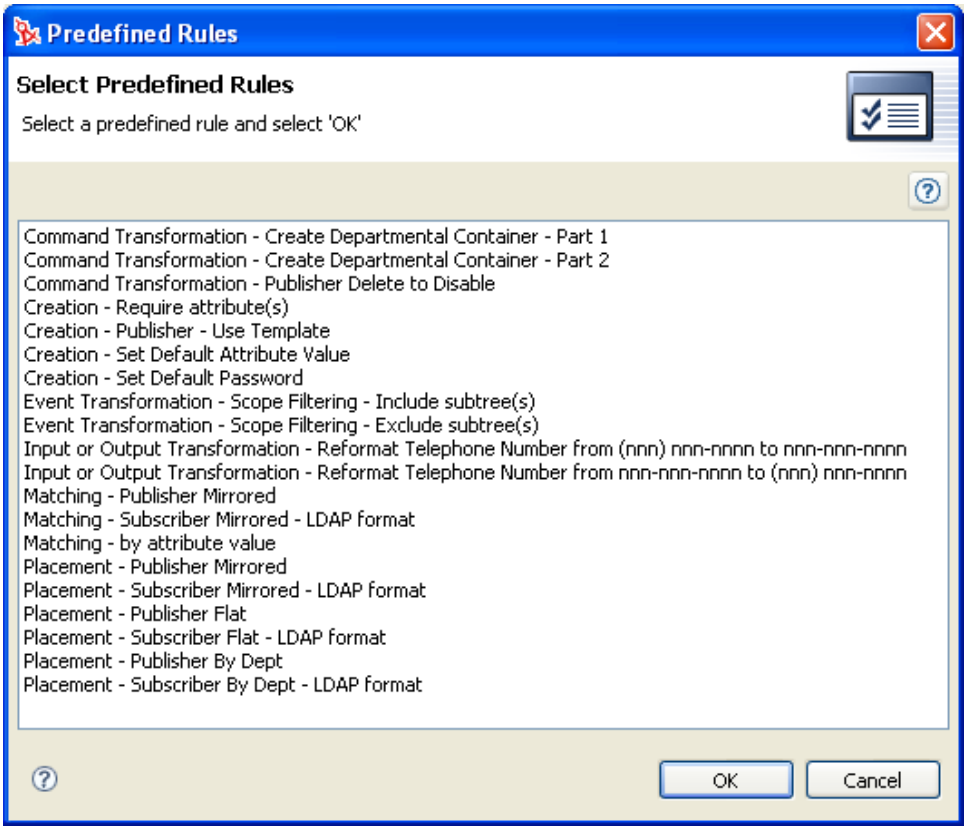
- 103 ページの 「配置 - 部署別発行者」
- 105 ページの 「配置 - 部署別購読者 -LDAP 形式」

事前定義されたルールにアクセスするには

1 ポリシービルダ内で、右クリックして［New (新規作成)］>［事前定義されたルール］>［Insert Predefined Rule Before (事前定義されたルールを前に挿入)］または［Insert Predefined Rule After (事前定義されたルールを後に挿入)］の順に選択します。



［事前定義されたルール］ダイアログボックスに、使用可能なルールのリストが表示されます。



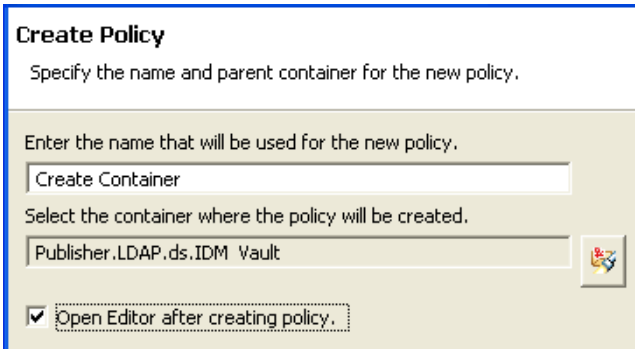
コマンド変換 - 部署別コンテナの作成 - パート 1 とパート 2

ターゲットデータストア内に部署別コンテナがない場合に、そのコンテナを作成します。このルールは、ドライバ内のコマンド変換ポリシーに実装します。ルールは、発行者と購読者のどちらのチャネルにも、また両方のチャネルにも設定できます。

この事前定義されたルールを使用する手順には、コマンド変換ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つがあります。このルールを追加するコマンド変換ポリシーがすでにある場合は、74 ページの 「事前定義されたルールのインポート」へ進みます。

ポリシーの作成

1. ［Outline (アウトライン)］ビューまたは［Policy Flow (ポリシーフロー)］ビューから発行者チャネルまたは購読者チャネルを選択します。
2. ［Policy Set (ポリシーセット)］ビューでコマンド変換ポリシーセットを選択し、［Create or add a new policy to the Policy Set (新しいポリシーの作成またはポリシーセットへの追加)］アイコン ✚ をクリックして、新しいポリシーを作成します。
3. ［新しいポリシーの作成］をクリックし、［次へ］をクリックします。
4. ポリシーに名前を付けます。
5. 表示されている場所を使用して、ポリシーをドライバに配置します。



6. ［Open Editor after creating policy (ポリシーの作成後にエディタを開く)］を選択し、［次へ］をクリックします。
7. ポリシーのタイプとして、［DirXML スクリプト］を選択し、［終了］をクリックします。
8. ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか ?)」というメッセージが表示されます。［はい］をクリックします。ポリシービルダが起動され、新しいコマンド変換ポリシーが保存されます。

事前定義されたルールのインポート

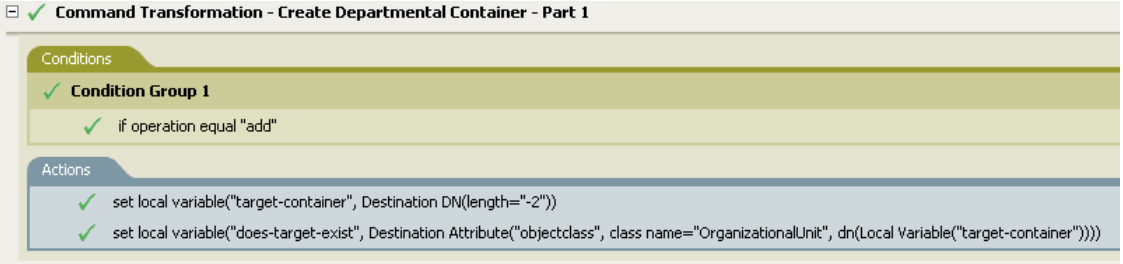
1. ポリシービルダ内で、右クリックして［New (新規作成)］>［Predefined Rule (事前定義されたルール)］>［Insert Predefined Rule Before (事前定義されたルールを前に挿入)］または［Insert Predefined Rule After (事前定義されたルールを後に挿入)］の順に選択します。

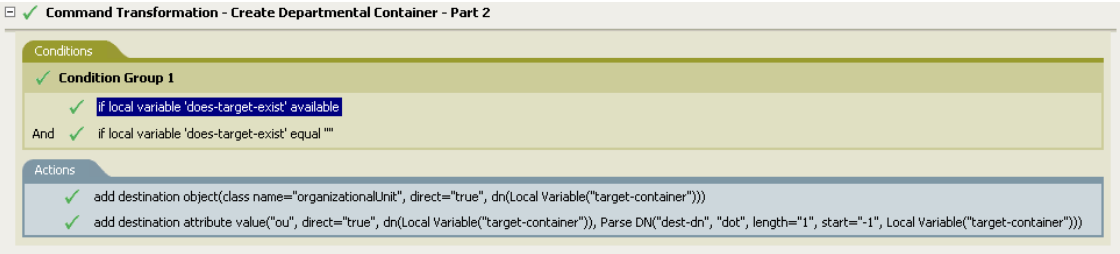
2 ［コマンド変換 - 部署別のコンテナの作成 - パート 1］を選択し、［OK］をクリックします。

3 ポリシービルダ内で、右クリックして［New (新規作成)］>［Predefined Rule (事前定義されたルール)］>［Insert Predefined Rule Before (事前定義されたルールを前に挿入)］または［Insert Predefined Rule After (事前定義されたルールを後に挿入)］の順に選択します。

4 ［コマンド変換 - 部署別のコンテナの作成 - パート 2］を選択し、［OK］をクリックします。

5 ［ファイル］>［保存］の順にクリックして、ルールを保存します。





このルールには、環境に応じて変更すべき情報はありません。

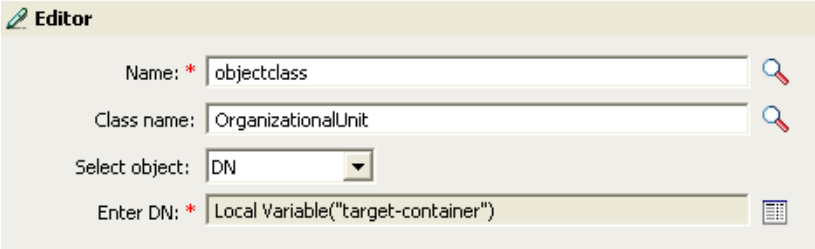
重要 : ルールが順序どおりに表示されていることを確認します。パート 1 は 2 よりも先に実行する必要があります。

ルールの動作

このルールは、オブジェクトのターゲットの場所が存在しない場合に使用されます。このルールでは、オブジェクトが配置できない場合、作成を拒否する代わりにコンテナが作成され、その中にオブジェクトが配置されます。

パート 1 では「追加」イベントが想定されます。「追加」イベントが発生すると、2 つのローカル変数が設定されます。最初のローカル変数は、target-container という名前になります。target-container の値が、ターゲット DN に設定されます。2 つ目のローカル変数は、does-target-exist という名前になります。does-target-exist の値は、objectclass のターゲット

属性値に設定されます。クラスは OrganizationalUnit に設定されます。OrganizationalUnit の DN は、ローカル変数 target-container に設定されます。



パート 2 では、ローカル変数 does-target-exist が使用可能かどうかが確認されます。また、ローカル変数 does-target-exist の値が空白に設定されているかどうかも確認されます。値が空白である場合、部門オブジェクトが作成されます。部門の DN は、ローカル変数 target-container の値に設定されます。また、OU 属性の値も追加されます。OU 属性の値は、ローカル変数 target-container に設定されます。これは、ソース形式をターゲット DN として使用します。ターゲット形式はドット形式です。

コマンド変換 - 無効にする発行者の削除

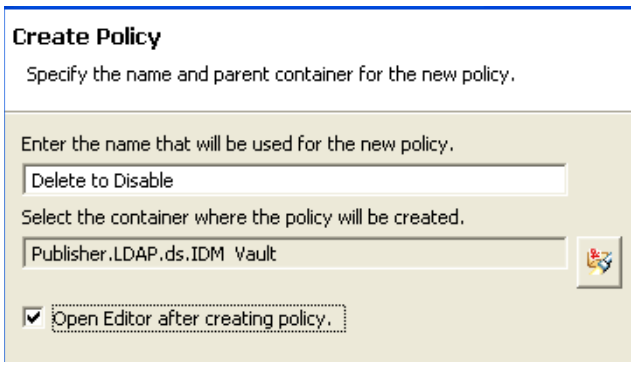
ユーザオブジェクトの「削除」イベントを、ユーザオブジェクトの無効化に変換します。このルールは、ドライバ内のコマンド変換ポリシーに実装します。ルールは、発行者チャネルに実装する必要があります。

この事前定義されたルールを使用する手順には、コマンド変換ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つがあります。このルールを追加するコマンド変換ポリシーがすでにある場合は、77 ページの 「事前定義されたルールのインポート」へ進みます。

ポリシーの作成

1. ［Outline (アウトライン)］ビューまたは［Policy Flow (ポリシーフロー)］ビューから発行者チャネルを選択します。
2. ［Policy Set (ポリシーセット)］ビューでコマンド変換ポリシーセットを選択し、［Create or add a new policy to the Policy Set (新しいポリシーの作成またはポリシーセットへの追加)］アイコン をクリックして、新しいポリシーを作成します。
3. ［新しいポリシーの作成］をクリックし、［次へ］をクリックします。
4. ポリシーに名前を付けます。

5 ポリシーのドライバでの配置先として自動的に挿入される場所を使用します。



6 [Open Editor after creating policy (ポリシーの作成後にエディタを開く)] を選択し、[次へ] をクリックします。

7 ポリシーのタイプとして、[DirXML スクリプト] を選択し、[終了] をクリックします。

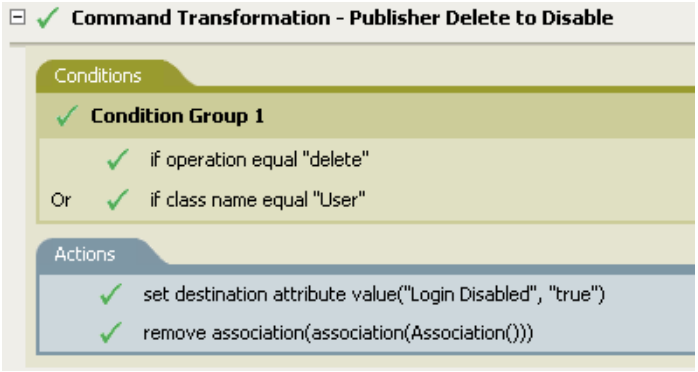
8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか ?)」というメッセージが表示されます。[はい] をクリックします。ポリシービルダが起動され、新しいコマンド変換ポリシーが保存されます。

事前定義されたルールのインポート

1 ポリシービルダ内で、右クリックして [New (新規作成)] > [Predefined Rule (事前定義されたルール)] > [Insert Predefined Rule Before (事前定義されたルールを前に挿入)] または [Insert Predefined Rule After (事前定義されたルールを後に挿入)] の順に選択します。

2 [コマンド変換 - 無効にする発行者の削除] を選択し、[OK] をクリックします。

3 [ファイル] > [保存] の順にクリックして、ルールを保存します。



このルールには、環境に応じて変更すべき情報はありません。

ルールの動作

このルールは、接続データストアで「削除」イベントが発生したときに使用されます。ユーザオブジェクトはアイデンティティボールトで削除される代わりに、無効になりま

す。ユーザオブジェクトに対して削除イベントが発生するときはいつでも、「ログインの無効化」のターゲット属性値が True に設定され、ユーザオブジェクトから関連付けが削除されます。ユーザオブジェクトは、Novell eDirectory ツリーへはログインできなくなりますが、ユーザオブジェクトは削除されません。

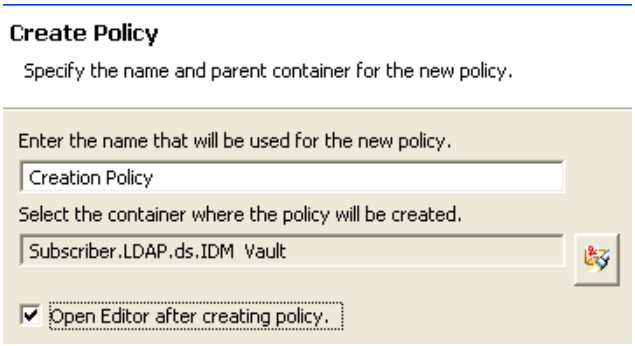
作成 - 属性が必要

このルールは、必要な属性が入力されない場合にユーザオブジェクトを作成できないようにします。このルールは、ドライバ内の作成ポリシーに実装します。ルールは、発行者と購読者のどちらのチャネルにも、また両方のチャネルにも設定できます。

この事前定義されたルールを使用するには、作成ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する作成ポリシーがすでにある場合は、79 ページの 「事前定義されたルールのインポート」へ進みます。

ポリシーの作成

1. ［Outline (アウトライン)］ビューまたは［Policy Flow (ポリシーフロー)］ビューから発行者チャネルまたは購読者チャネルを選択します。
2. ［Policy Set (ポリシーセット)］ビューで作成ポリシーセットを選択し、［Create or add a new policy to the Policy Set (新しいポリシーの作成またはポリシーセットへの追加)］アイコン ✚ をクリックして、新しいポリシーを作成します。
3. ［新しいポリシーの作成］をクリックし、［次へ］をクリックします。
4. ポリシーに名前を付けます。
5. ポリシーのドライバでの配置先として自動的に挿入される場所を使用します。



6. ［Open Editor after creating policy (ポリシーの作成後にエディタを開く)］を選択し、［次へ］をクリックします。
7. ポリシーのタイプとして、［DirXML スクリプト］を選択し、［終了］をクリックします。
8. ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか ?)」というメッセージが表示されます。［はい］をクリックします。ポリシービルダが起動され、新しい作成ポリシーが保存されます。

事前定義されたルールのインポート

1 ポリシービルダ内で、右クリックして［New (新規作成)］>［Predefined Rule (事前定義されたルール)］>［Insert Predefined Rule Before (事前定義されたルールを前に挿入)］または［Insert Predefined Rule After (事前定義されたルールを後に挿入)］の順に選択します。

2 ［作成 - 属性が必要］を選択し、［OK］をクリックします。

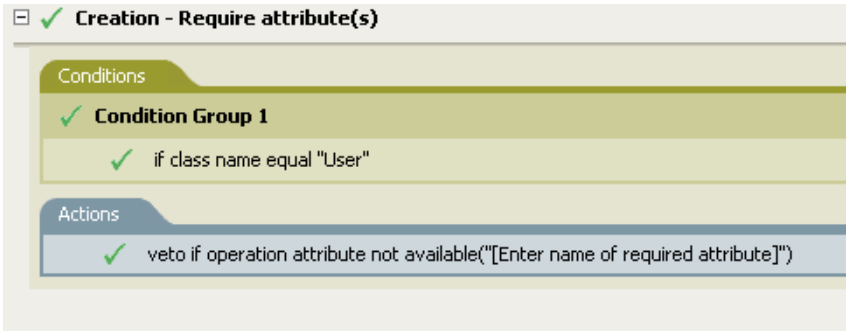
3 ［アクション］タブをダブルクリックし、アクションを編集します。

4 ［名前を入力］フィールドから、［必要な属性の名前を入力してください］を削除します。

5 作成するユーザオブジェクトに必要な属性を参照し、［OK］をクリックします。

6 ［OK］をクリックします。

7 ［ファイル］>［保存］の順に選択して、ルールを保存します。



ルールの動作

このルールは、ビジネスプロセスにおいて、ユーザオブジェクトが作成されるときに特別な属性が必要な場合に使用されます。ユーザオブジェクトを作成する場合、このルールでは、必須属性が入力されないと、オブジェクトの作成が拒否されます。必須属性は複数指定できます。

複数の必須属性を指定する場合は、アクションを右クリックして［New (新規)］>［Append Action (アクションの追加)］の順に選択します。［操作属性値がない場合は拒否］を選択し、必須属性を参照します。

作成 - 発行者 - テンプレートの使用

ユーザオブジェクトの作成時に、Novell eDirectory のテンプレートオブジェクトを使用できるようにします。このルールは、ドライバ内の発行者作成ポリシーに実装します。このルールは、発行者チャネルにのみ実装できます。

この事前定義されたルールを使用するには、作成ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する作成ポリシーがすでにある場合は、80 ページの 「事前定義されたルールのインポート」へ進みます。

ポリシーの作成

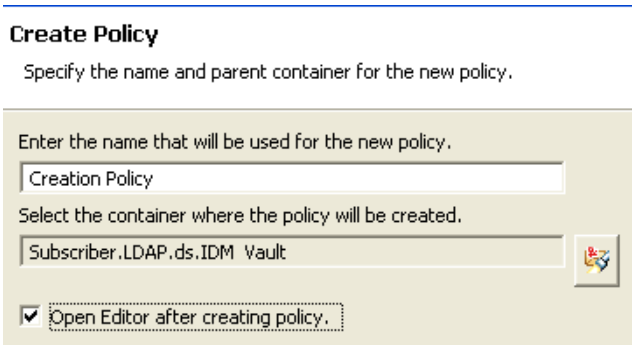
1 ［Outline (アウトライン)］ビューまたは［Policy Flow (ポリシーフロー)］ビューから発行者チャネルを選択します。

2 ［Policy Set (ポリシーセット)］ビューで作成ポリシーセットを選択し、［Create or add a new policy to the Policy Set (新しいポリシーの作成またはポリシーセットへの追加)］アイコン をクリックして、新しいポリシーを作成します。

3 ［新しいポリシーの作成］をクリックし、［次へ］をクリックします。

4 ポリシーに名前を付けます。

5 ポリシーのドライバでの配置先として自動的に挿入される場所を使用します。



6 ［Open Editor after creating policy (ポリシーの作成後にエディタを開く)］を選択し、［次へ］をクリックします。

7 ポリシーのタイプとして、［DirXML スクリプト］を選択し、［終了］をクリックします。

8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか ?)」というメッセージが表示されます。［はい］をクリックします。ポリシービルダが起動され、新しい作成ポリシーが保存されます。

事前定義されたルールのインポート

1 ポリシービルダ内で、右クリックして［New (新規作成)］> ［Predefined Rule (事前定義されたルール)］> ［Insert Predefined Rule Before (事前定義されたルールを前に挿入)］または［Insert Predefined Rule After (事前定義されたルールを後に挿入)］の順に選択します。

2 ［作成 - 発行者 - テンプレートの使用］を選択し、［OK］をクリックします。

3 ［アクション］タブをダブルクリックし、アクションを編集します。

4 ［DN を入力］フィールドから、[テンプレートオブジェクトの DN を入力してください] を削除します。

5 ［引数の編集］アイコン をクリックして、引数ビルダを起動します。

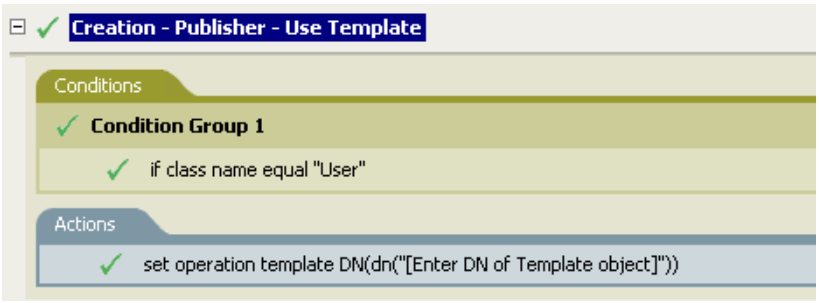
6 ［Noun (名詞)］リストの［テキスト］を選択します。

7 ［テキスト］をダブルクリックして、引数に追加します。

8 エディタで、参照アイコンをクリックしてテンプレートオブジェクトを参照して選択し、［OK］をクリックします。

9 ［OK］をクリックします。

10 ［ファイル］>［保存］の順にクリックして、ルールを保存します。



ルールの動作

このルールは、テンプレートオブジェクトを使用してアイデンティティボールト内にユーザを作成する場合に使用されます。異なるユーザに共通の属性がある場合、テンプレートを使用することで時間を節約できます。テンプレートオブジェクトに情報を入力してユーザオブジェクトを作成すると、Identity Manager はテンプレートを呼び出して、それをユーザオブジェクトの作成に使用します。

ユーザオブジェクトの作成中に、ルールは、操作テンプレート DN の設定アクションを実行します。アクションは、テンプレートオブジェクトを呼び出し、テンプレート内の情報を使用してユーザオブジェクトを作成します。

作成 - デフォルト属性値の設定

ユーザオブジェクトの作成時に割り当てられる属性のデフォルト値を設定できます。このルールは、ドライバ内の購読者作成ポリシーまたは発行者作成ポリシーに実装します。

この事前定義されたルールを使用するには、作成ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する作成ポリシーがすでにある場合は、82 ページの 「事前定義されたルールのインポート」へ進みます。

ポリシーの作成

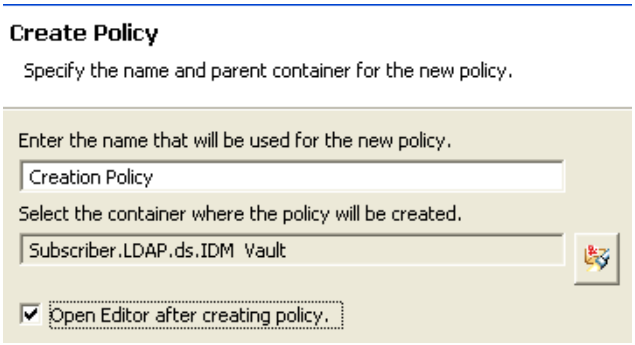
1 ［Outline (アウトライン)］ビューまたは［Policy Flow (ポリシーフロー)］ビューから発行者チャネルまたは購読者チャネルを選択します。

2 ［Policy Set (ポリシーセット)］ビューで作成ポリシーセットを選択し、［Create or add a new policy to the Policy Set (新しいポリシーの作成またはポリシーセットへの追加)］アイコン ✚ をクリックして、新しいポリシーを作成します。

3 ［新しいポリシーの作成］をクリックし、［次へ］をクリックします。

4 ポリシーに名前を付けます。

5 ポリシーのドライバでの配置先として自動的に挿入される場所を使用します。



6 ［Open Editor after creating policy (ポリシーの作成後にエディタを開く)］を選択し、［次へ］をクリックします。

7 ポリシーのタイプとして、［DirXML スクリプト］を選択し、［終了］をクリックします。

8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか ?)」というメッセージが表示されます。［はい］をクリックします。ポリシービルダが起動され、新しい作成ポリシーが保存されます。

事前定義されたルールのインポート

1 ポリシービルダ内で、右クリックして［New (新規作成)］> ［Predefined Rule (事前定義されたルール)］> ［Insert Predefined Rule Before (事前定義されたルールを前に挿入)］または［Insert Predefined Rule After (事前定義されたルールを後に挿入)］の順に選択します。

2 ［作成 - デフォルト属性値の設定］を選択し、［OK］をクリックします。

3 ［アクション］タブをダブルクリックし、アクションを編集します。

4 ［属性名を入力してください］フィールドから、［属性名を入力してください］を削除します。

5 参照アイコンをクリックして、作成する属性を参照して選択します。

6 ［引数値を入力］フィールドから、［デフォルト属性値を入力してください］を削除します。

7 ［引数の編集］アイコン をクリックして、引数値リストビルダを起動します。

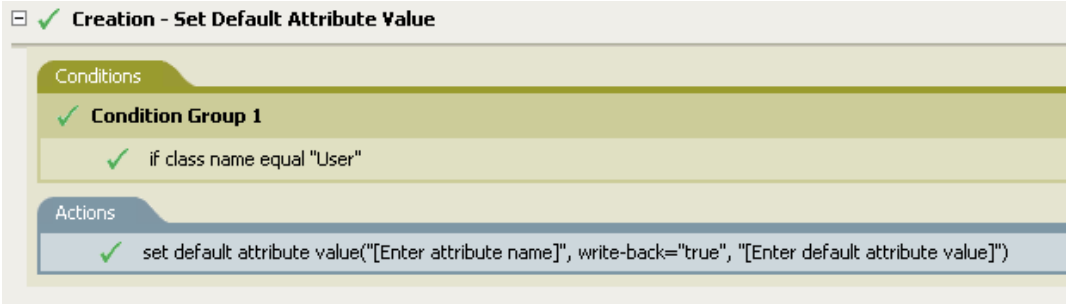
8 デフォルト値にするデータのタイプを選択します。

9 ［引数の編集］アイコン をクリックして、引数ビルダを起動します。

10 引数ビルダで、属性の値を作成し、［OK］をクリックします。

11 ［OK］をクリックします。

12 ［ファイル］>［保存］の順にクリックして、ルールを保存します。



ルールの動作

このルールは、デフォルト属性および値を指定したユーザオブジェクトを作成する場合に使用されます。ユーザオブジェクトが作成される場合、ルールは属性とその属性の値を設定します。

複数の属性値を定義する場合は、アクションを右クリックして［New (新規)］>［Append Action (アクションの追加)］の順にクリックします。アクションを選択し、デフォルト属性値を設定し、82 ページのステップ 1 から 83 ページのステップ 12 の手順を実行して属性に値を割り当てます。

作成 - デフォルトパスワードの設定

ユーザオブジェクトの作成中に、ユーザオブジェクトのデフォルトパスワードが設定されます。このルールは、ドライバ内の作成ポリシーに実装します。ルールは、発行者と購読者のどちらのチャネルにも、また両方のチャネルにも設定できます。

この事前定義されたルールを使用するには、作成ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する作成ポリシーがすでにある場合は、84 ページの 「事前定義されたルールのインポート」へ進みます。

ポリシーの作成

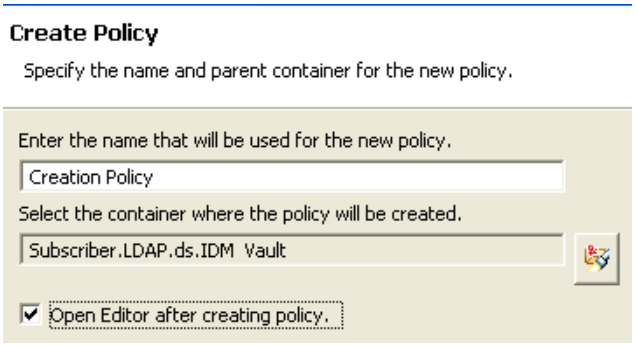
1 ［Outline (アウトライン)］ビューまたは［Policy Flow (ポリシーフロー)］ビューから発行者チャネルまたは購読者チャネルを選択します。

2 ［Policy Set (ポリシーセット)］ビューで作成ポリシーセットを選択し、［Create or add a new policy to the Policy Set (新しいポリシーの作成またはポリシーセットへの追加)］アイコン ✚ をクリックして、新しいポリシーを作成します。

3 ［新しいポリシーの作成］をクリックし、［次へ］をクリックします。

4 ポリシーに名前を付けます。

5 ポリシーのドライバでの配置先として自動的に挿入される場所を使用します。



6 [Open Editor after creating policy (ポリシーの作成後にエディタを開く)] を選択し、[次へ] をクリックします。

7 ポリシーのタイプとして、[DirXML スクリプト] を選択し、[終了] をクリックします。

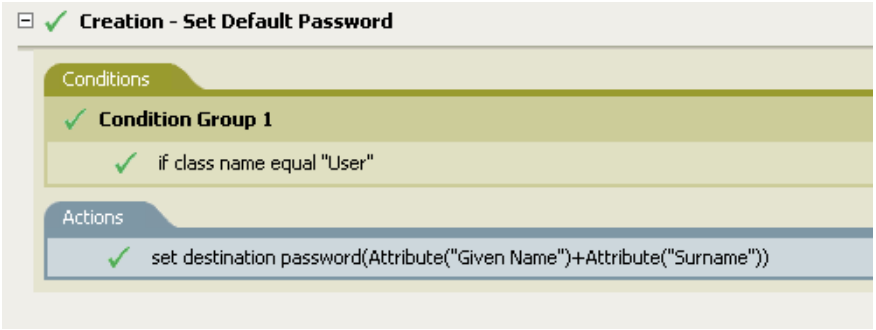
8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか ?)」というメッセージが表示されます。[はい] をクリックします。ポリシービルダが起動され、新しい作成ポリシーが保存されます。

事前定義されたルールのインポート

1 ポリシービルダ内で、右クリックして [New (新規作成)] > [Predefined Rule (事前定義されたルール)] > [Insert Predefined Rule Before (事前定義されたルールを前に挿入)] または [Insert Predefined Rule After (事前定義されたルールを後に挿入)] の順に選択します。

2 [作成 - デフォルトパスワードの設定] を選択し、[OK] をクリックします。

3 [ファイル] > [保存] の順にクリックして、ルールを保存します。



このルールには、環境に応じて変更すべき情報はありません。

ルールの動作

このルールは、デフォルトパスワードを指定してユーザオブジェクトを作成する場合に使用されます。ユーザオブジェクトの作成時に、ユーザオブジェクトに設定されるパスワードは、そのユーザオブジェクトの名前属性に名字属性を加えたものになります。

デフォルトパスワードの値は、引数を編集することで変更できます。パスワードは、引数ビルダを使用して任意の値に設定できます。

イベント変換 - スコープフィルタリング - サブツリーの組み込み

特定のサブツリー以外で発生するすべてのイベントを除外します。このルールは、ドライバ内のイベント変換ポリシーに実装します。ルールは、発行者と購読者のどちらのチャネルにも、また両方のチャネルにも設定できます。

この事前定義されたルールを使用するには、イベント変換ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加するイベント変換ポリシーがすでにある場合は、(86 ページ) 事前定義されたルールのインポートへ進みます。

ポリシーの作成

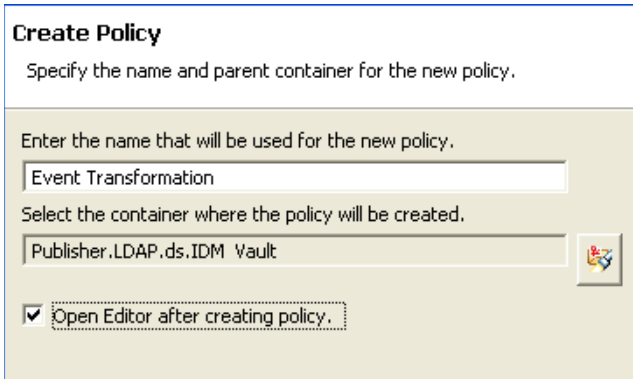
1 ［Outline (アウトライン)］ビューまたは［Policy Flow (ポリシーフロー)］ビューから発行者チャネルまたは購読者チャネルを選択します。

2 ［Policy Set (ポリシーセット)］ビューでイベント変換ポリシーセットを選択し、［Create or add a new policy to the Policy Set (新しいポリシーの作成またはポリシーセットへの追加)］アイコン をクリックして、新しいポリシーを作成します。

3 ［新しいポリシーの作成］をクリックし、［次へ］をクリックします。

4 ポリシーに名前を付けます。

5 ポリシーのドライバでの配置先として自動的に挿入される場所を使用します。



6 ［Open Editor after creating policy (ポリシーの作成後にエディタを開く)］を選択し、［次へ］をクリックします。

7 ポリシーのタイプとして、［DirXML スクリプト］を選択し、［終了］をクリックします。

8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか ?)」というメッセージが表示されます。［はい］をクリックします。ポリシービルダが起動され、新しいイベント変換ポリシーが保存されます。

事前定義されたルールのインポート

1 ポリシービルダ内で、右クリックして［New (新規作成)］>［Predefined Rule (事前定義されたルール)］>［Insert Predefined Rule Before (事前定義されたルールを前に挿入)］または［Insert Predefined Rule After (事前定義されたルールを後に挿入)］の順にクリックします。

2 ［イベント変換 - スコープフィルタリング - サブツリーの組み込み］を選択し、［OK］をクリックします。

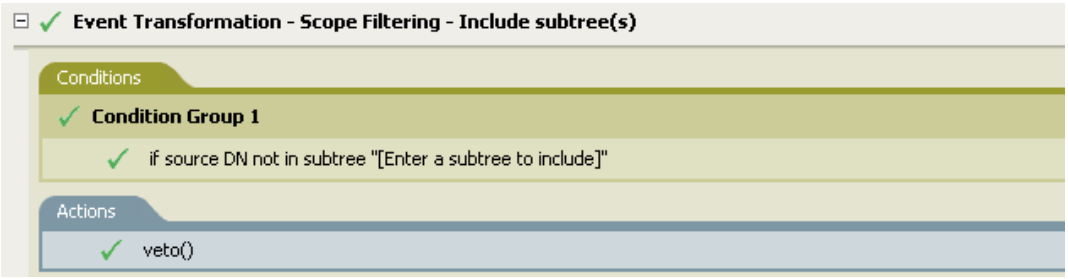
3 ［条件］タブをダブルクリックし、条件を編集します。

4 ［値］フィールドの［組み込むサブツリーを入力してください］を削除します。

5 ［参照］ボタンをクリックしてアイデンティティボールトを参照し、イベントを同期させるツリーの部分を選択し、［OK］をクリックします。

6 ［OK］をクリックします。

7 ［ファイル］>［保存］の順にクリックして、ルールを保存します。



ルールの動作

このルールは、アイデンティティボールトの一部を同期から除外する場合に使用されます。これによって、フィルタを使用しなくても、同期するオブジェクトと同期しないオブジェクトを区別できます。アイデンティティボールトの特定部分以外でイベントが発生した場合、そのイベントは拒否されます。

イベント変換 - スコープフィルタリング - サブツリーの除外

特定のサブツリー内で発生するすべてのイベントを除外します。このルールは、ドライバ内のイベント変換ポリシーに実装します。ルールは、発行者と購読者のどちらのチャネルにも、また両方のチャネルにも設定できます。

この事前定義されたルールを使用するには、イベント変換ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加するイベント変換ポリシーがすでにある場合は、87 ページの 「事前定義されたルールのインポート」へ進みます。

ポリシーの作成

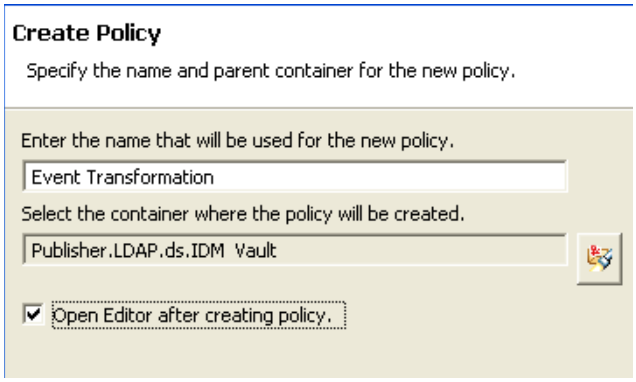
1 ［Outline (アウトライン)］ビューまたは［Policy Flow (ポリシーフロー)］ビューから発行者チャネルまたは購読者チャネルを選択します。

2 ［Policy Set (ポリシーセット)］ビューでイベント変換ポリシーセットを選択し、［Create or add a new policy to the Policy Set (新しいポリシーの作成またはポリシーセットへの追加)］アイコン ✚ をクリックして、新しいポリシーを作成します。

3 ［新しいポリシーの作成］をクリックし、［次へ］をクリックします。

4 ポリシーに名前を付けます。

5 表示されている場所を使用して、ポリシーをドライバに配置します。



6 [Open Editor after creating policy (ポリシーの作成後にエディタを開く)] を選択し、[次へ] をクリックします。

7 ポリシーのタイプとして、[DirXML スクリプト] を選択し、[終了] をクリックします。

8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか ?)」というメッセージが表示されます。[はい] をクリックします。ポリシービルダが起動され、新しいイベント変換ポリシーが保存されます。

事前定義されたルールのインポート

1 ポリシービルダを右クリックし、[New (新規作成)] > [Predefined Rule (事前定義されたルール)] の順にクリックします。

2 [イベント変換 - スコープフィルタリング - サブツリーの除外] を選択し、[OK] をクリックします。

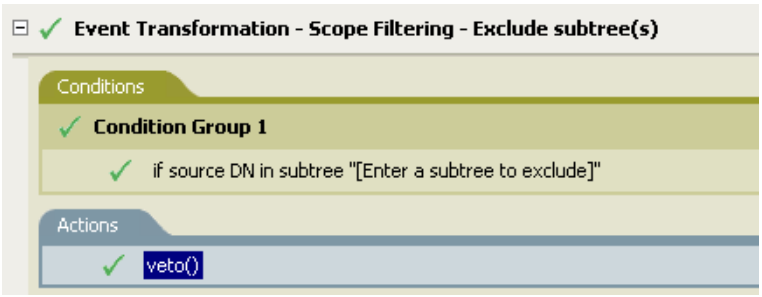
3 [条件] タブをダブルクリックし、条件を編集します。

4 [値] フィールドの [除外するサブツリーを入力してください] を削除します。

5 [参照] ボタンをクリックしてアイデンティティボールトを参照し、イベントを同期から除外するツリーの部分を選択し、[OK] をクリックします。

6 [OK] をクリックします。

7 [ファイル] > [保存] の順にクリックして、ルールを保存します。



ルールの動作

このルールは、アイデンティティボールトの一部を同期から除外する場合に使用されます。これによって、フィルタを使用しなくても、同期するオブジェクトと同期しないオブジェクトを区別できます。アイデンティティボールトの特定部分でイベントが発生するたびに、そのイベントは拒否されます。

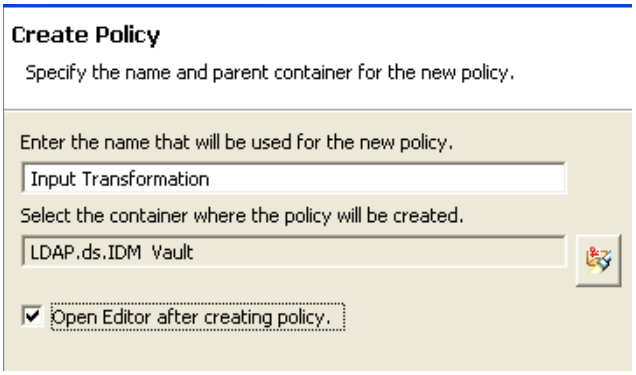
入出力変換 - 電話番号の形式を (nnn) nnn-nnnn から nnn-nnn-nnnn に変更

条件を満たした場合に電話番号の形式を変換します。このルールは、ドライバ内の入出力変換ポリシーに実装します。ルールは、発行者と購読者のどちらのチャネルにも、また両方のチャネルにも設定できます。

この事前定義されたルールを使用するには、入出力変換ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する入出力変換ポリシーがすでにある場合は、89 ページの 「事前定義されたルールのインポート」へ進みます。

ポリシーの作成

1. ［Outline (アウトライン)］ビューまたは［Policy Flow (ポリシーフロー)］ビューから発行者チャネルまたは購読者チャネルを選択します。
2. ［Policy Set (ポリシーセット)］ビューで入出力変換ポリシーセットを選択し、［Create or add a new policy to the Policy Set (新しいポリシーの作成またはポリシーセットへの追加)］アイコン ✚ をクリックして、新しいポリシーを作成します。
3. ［新しいポリシーの作成］をクリックし、［次へ］をクリックします。
4. ポリシーに名前を付けます。
5. 表示されている場所を使用して、ポリシーをドライバに配置します。



6. ［Open Editor after creating policy (ポリシーの作成後にエディタを開く)］を選択し、［次へ］をクリックします。
7. ポリシーのタイプとして、［DirXML スクリプト］を選択し、［終了］をクリックします。
8. ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか ?)」というメッセージが表示されます。［はい］をクリックします。ポリシービルダが起動され、新しい入出力変換ポリシーが保存されます。

事前定義されたルールのインポート

1 ポリシービルダ内で、右クリックして［New (新規作成)］>［Predefined Rule (事前定義されたルール)］>［Insert Predefined Rule Before (事前定義されたルールを前に挿入)］または［Insert Predefined Rule After (事前定義されたルールを後に挿入)］の順に選択します。

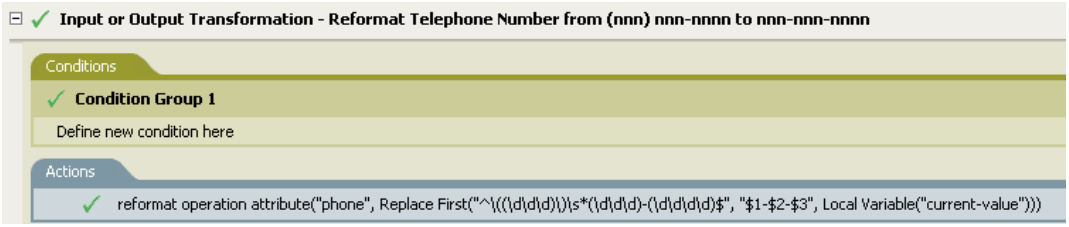
2 入出力変換 - 電話番号の形式を (nnn) nnn-nnnn から nnn-nnn-nnnn に変更］を選択し、［OK］をクリックします。

3 ［条件］タブをダブルクリックし、条件を編集します。

4 電話番号の形式変更が実行されるときの条件を定義します。

5 ［OK］をクリックします。

6 ［ファイル］>［保存］の順にクリックして、ルールを保存します。



ルールの動作

このルールは、電話番号の形式を変更する場合に使用されます。電話番号の形式変更が実行されるときに指定する条件を定義します。

入出力変換 - 電話番号の形式を nnn-nnn-nnnn から (nnn) nnn-nnnn に変更

条件を満たした場合に電話番号の形式を変換します。このルールは、入出力変換ポリシーに実装します。ルールは、発行者と購読者のどちらのチャネルにも、また両方のチャネルにも設定できます。

この事前定義されたルールを使用するには、入出力変換ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する入出力変換ポリシーがすでにある場合は、90 ページの 「事前定義されたルールのインポート」へ進みます。

ポリシーの作成

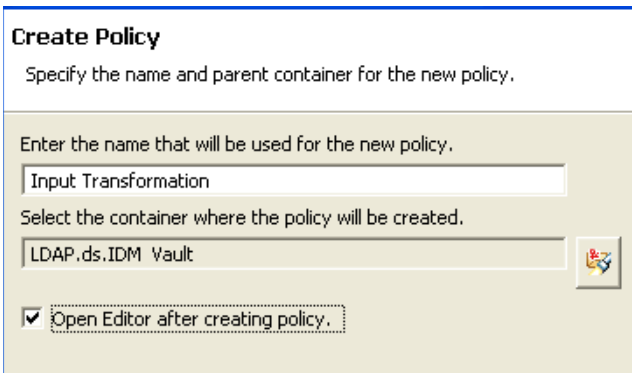
1 ［Outline (アウトライン)］ビューまたは［Policy Flow (ポリシーフロー)］ビューから発行者チャネルまたは購読者チャネルを選択します。

2 ［Policy Set (ポリシーセット)］ビューで入出力変換ポリシーセットを選択し、［Create or add a new policy to the Policy Set (新しいポリシーの作成またはポリシーセットへの追加)］アイコン ✚ をクリックして、新しいポリシーを作成します。

3 ［新しいポリシーの作成］をクリックし、［次へ］をクリックします。

4 ポリシーに名前を付けます。

5 表示されている場所を使用して、ポリシーをドライバに配置します。



6 [Open Editor after creating policy (ポリシーの作成後にエディタを開く)] を選択し、[次へ] をクリックします。

7 ポリシーのタイプとして、[DirXML スクリプト] を選択し、[終了] をクリックします。

8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか ?)」というメッセージが表示されます。[はい] をクリックします。ポリシービルダが起動され、新しい入出力変換ポリシーが保存されます。

事前定義されたルールのインポート

1 ポリシービルダ内で、右クリックして [New (新規作成)] > [Predefined Rule (事前定義されたルール)] > [Insert Predefined Rule Before (事前定義されたルールを前に挿入)] または [Insert Predefined Rule After (事前定義されたルールを後に挿入)] の順に選択します。

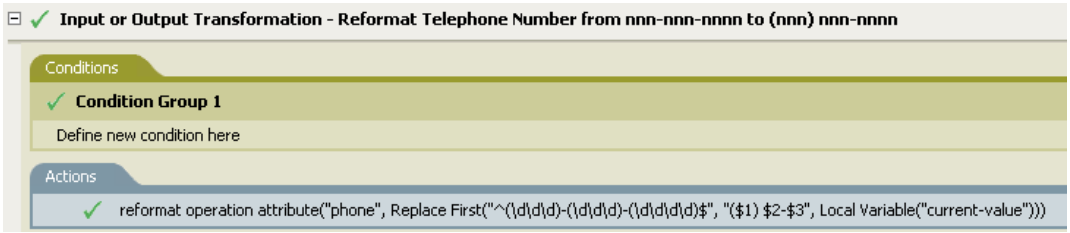
2 [入出力変換 - 電話番号の形式を nnn-nnn-nnnn から (nnn) nnn-nnnn に変更] をクリックし、[OK] をクリックします。

3 [条件] タブをダブルクリックし、条件を編集します。

4 電話番号の形式変更が実行されるときの条件を定義します。

5 [OK] をクリックします。

6 [ファイル] > [保存] の順にクリックして、ルールを保存します。



ルールの動作

このルールは、電話番号の形式を変更する場合に使用されます。電話番号の形式変更が実行されるときに指定する条件を定義します。

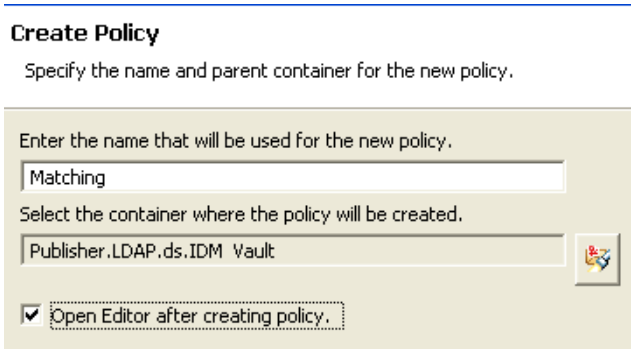
一致 - 発行者 (ミラーリング)

指定したポイントからデータストア内のミラー化された構造を使用して、アイデンティティボールト内のオブジェクトを照合します。このルールは、ドライバ内の一致ポリシーに実装します。このルールは、発行者チャネルにのみ実装できます。

この事前定義されたルールを使用するには、一致ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する一致ポリシーがすでにある場合は、(91 ページ) 事前定義されたルールのインポートへ進みます。

ポリシーの作成

1. ［Outline (アウトライン)］ビューまたは［Policy Flow (ポリシーフロー)］ビューから発行者チャネルを選択します。
2. ［Policy Set (ポリシーセット)］ビューで一致ポリシーセットを選択し、［Create or add a new policy to the Policy Set (新しいポリシーの作成またはポリシーセットへの追加)］アイコン ✚ をクリックして、新しいポリシーを作成します。
3. ［新しいポリシーの作成］をクリックし、［次へ］をクリックします。
4. ポリシーに名前を付けます。
5. 表示されている場所を使用して、ポリシーをドライバに配置します。



6. ［Open Editor after creating policy (ポリシーの作成後にエディタを開く)］を選択し、［次へ］をクリックします。
7. ポリシーのタイプとして、［DirXML スクリプト］を選択し、［終了］をクリックします。
8. ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか ?)」というメッセージが表示されます。［はい］をクリックします。ポリシービルダが起動され、新しい一致ポリシーが保存されます。

事前定義されたルールのインポート

1. ポリシービルダ内で、右クリックして［New (新規作成)］>［Predefined Rule (事前定義されたルール)］>［Insert Predefined Rule Before (事前定義されたルールを前に挿入)］または［Insert Predefined Rule After (事前定義されたルールを後に挿入)］の順に選択します。
2. ［一致 - 発行者 (ミラーリング)］を選択し、［OK］をクリックします。

3 ［条件］タブをダブルクリックし、条件を編集します。

4 ［値］フィールドから、［ソース階層のベースを入力してください］を削除します。

5 一致作業を開始するソース階層内のコンテナを参照して選択し、［OK］をクリックします。

6 ［OK］をクリックします。

7 ［アクション］タブをダブルクリックし、アクションを編集します。

8 ［文字列を入力］フィールドから、[宛先階層のベースを入力してください] を削除します。

9 ［引数の編集］アイコン をクリックして、引数ビルダを起動します。

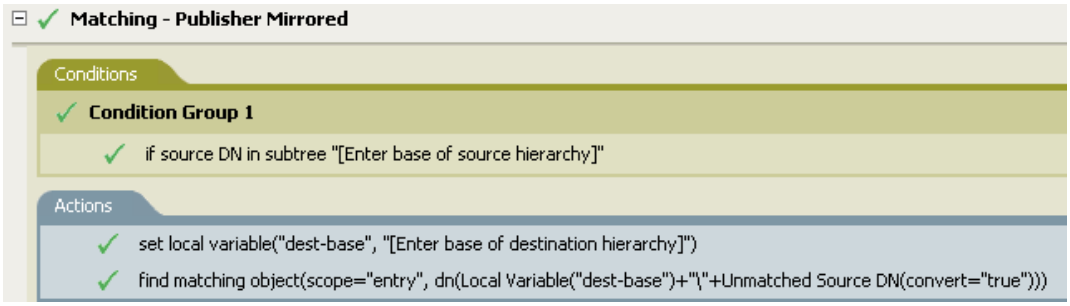
10 ［Noun (名詞)］リストで［テキスト］を選択します。

11 ［テキスト］をダブルクリックして、引数に追加します。

12 エディタで、参照アイコンをクリックして、ソース構造を一致させる宛先階層内のコンテナを参照し、［OK］をクリックします。

13 ［OK］をクリックします。

14 ［ファイル］>［保存］の順にクリックして、ルールを保存します。



ルールの動作

指定したポイントからデータストア内のミラー化された構造を使用して、アイデンティティボールト内のオブジェクトを照合します。「追加」イベントが発生し、ドライバがオブジェクトの存在を確認する場合に、データストアの特定の DN から確認を開始します。ドライバは、次に、データストアで構造がミラー化されるアイデンティティボールトの開始点になるようにローカル変数 dest-base を設定します。その後、ドライバは、ローカル変数 dest-base に \ およびオブジェクトのソース DN を追加して、検索するコンテキストを作成します。検索するパスは、スラッシュ形式で作成されます。

一致 - 購読者 (ミラーリング) - LDAP 形式

指定したポイントからアイデンティティボールト内のミラー化された構造を使用して、データストア内のオブジェクトを照合します。このルールは、ドライバ内の一致ポリシーに実装します。このルールは、購読者チャネルにのみ実装できます。

この事前定義されたルールを使用するには、一致ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する一致ポリシーがすでにある場合は、93 ページの 「事前定義されたルールのインポート」へ進みます。

ポリシーの作成

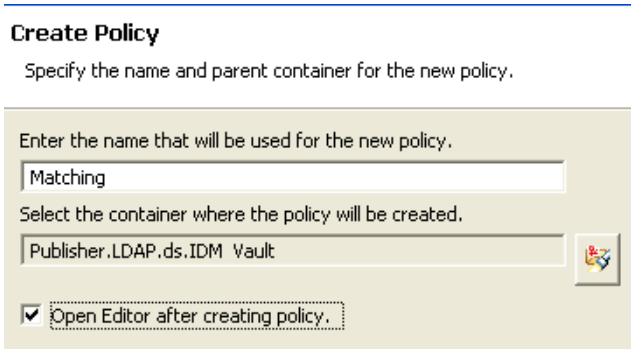
1 ［Outline (アウトライン)］ビューまたは［Policy Flow (ポリシーフロー)］ビューから発行者チャネルを選択します。

2 ［Policy Set (ポリシーセット)］ビューで一致ポリシーセットを選択し、［Create or add a new policy to the Policy Set (新しいポリシーの作成またはポリシーセットへの追加)］アイコン ✚ をクリックして、新しいポリシーを作成します。

3 ［新しいポリシーの作成］をクリックし、［次へ］をクリックします。

4 ポリシーに名前を付けます。

5 表示されている場所を使用して、ポリシーをドライバに配置します。



6 ［Open Editor after creating policy (ポリシーの作成後にエディタを開く)］を選択し、［次へ］をクリックします。

7 ポリシーのタイプとして、［DirXML スクリプト］を選択し、［終了］をクリックします。

8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか ?)」というメッセージが表示されます。［はい］をクリックします。ポリシービルダが起動され、新しい一致ポリシーが保存されます。

事前定義されたルールのインポート

1 ポリシービルダ内で、右クリックして［New (新規作成)］>［Predefined Rule (事前定義されたルール)］>［Insert Predefined Rule Before (事前定義されたルールを前に挿入)］または［Insert Predefined Rule After (事前定義されたルールを後に挿入)］の順に選択します。

2 ［一致 - 購読者 (ミラーリング)-LDAP 形式］を選択し、［OK］をクリックします。

3 ［条件］タブをダブルクリックし、条件を編集します。

4 ［値］フィールドから、［ソース階層のベースを入力してください］を削除します。

5 照合作業を開始するソース階層内のコンテナを参照して選択し、［OK］をクリックします。

6 ［OK］をクリックします。

7 ［アクション］タブをダブルクリックし、アクションを編集します。

8 ［文字列を入力］フィールドから、[宛先階層のベースを入力してください] を削除します。

9 ［引数の編集］アイコン をクリックして、引数ビルダを起動します。

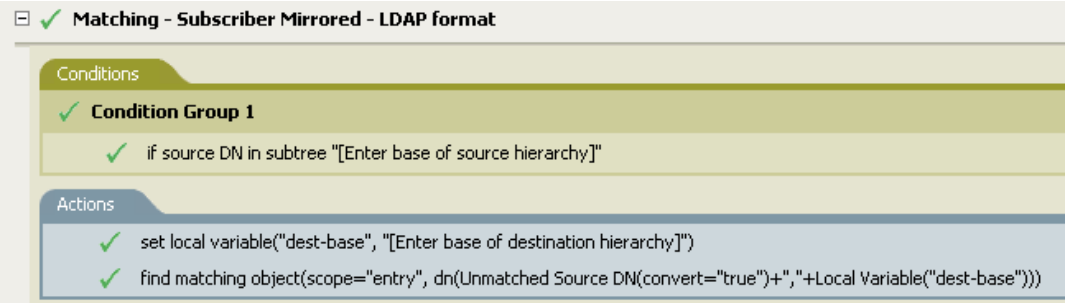
10 ［Noun (名詞)］リストの［テキスト］を選択します。

11 ［テキスト］をダブルクリックして、引数に追加します。

12 エディタで、参照アイコンをクリックして、ソース構造を照合する宛先階層内のコンテナを参照し、選択したら［OK］をクリックします。

13 ［OK］をクリックします。

14 ［ファイル］>［保存］の順にクリックして、ルールを保存します。



ルールの動作

指定したポイントからアイデンティティボールト内のミラー化された構造を使用して、データストア内のオブジェクトを照合します。「追加」イベントが発生し、ドライバがオブジェクトの存在を確認する場合に、アイデンティティボールトの特定の DN で確認を開始します。ドライバは、次に、アイデンティティボールトで構造がミラー化されるデータストアの開始点になるようにローカル変数 dest-base を設定します。その後、ドライバは、オブジェクトのソース DN に、カンマ (,)、ローカル変数 dest-base を追加して、検索するコンテキストを作成します。LDAP 形式で、検索するパスを作成します。

一致 - 属性値別

特定の属性値でオブジェクトを照合します。このルールは、ドライバ内の一致ポリシーに実装します。ルールは、発行者と購読者のどちらのチャネルにも、また両方のチャネルにも設定できます。

この事前定義されたルールを使用するには、一致ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する一致ポリシーがすでにある場合は、95 ページの 「事前定義されたルールのインポート」へ進みます。

ポリシーの作成

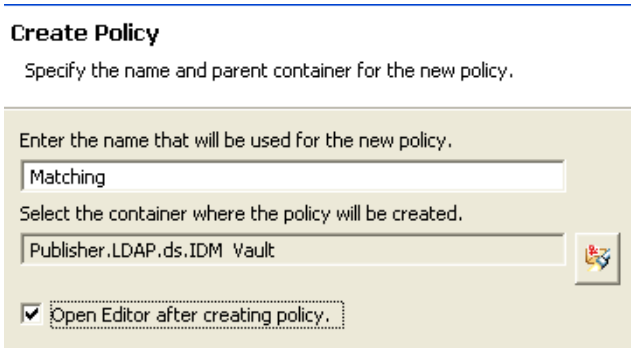
1 ［Outline (アウトライン)］ビューまたは［Policy Flow (ポリシーフロー)］ビューから発行者チャネルを選択します。

2 ［Policy Set (ポリシーセット)］ビューで一致ポリシーセットを選択し、［Create or add a new policy to the Policy Set (新しいポリシーの作成またはポリシーセットへの追加)］アイコン をクリックして、新しいポリシーを作成します。

3 ［新しいポリシーの作成］をクリックし、［次へ］をクリックします。

4 ポリシーに名前を付けます。

5 表示されている場所を使用して、ポリシーをドライバに配置します。



6 ［Open Editor after creating policy (ポリシーの作成後にエディタを開く)］を選択し、［次へ］をクリックします。

7 ポリシーのタイプとして、［DirXML スクリプト］を選択し、［終了］をクリックします。

8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか ?)」というメッセージが表示されます。［はい］をクリックします。ポリシービルダが起動され、新しい一致ポリシーが保存されます。

事前定義されたルールのインポート

1 ポリシービルダ内で、右クリックして［New (新規作成)］> ［Predefined Rule (事前定義されたルール)］> ［Insert Predefined Rule Before (事前定義されたルールを前に挿入)］または［Insert Predefined Rule After (事前定義されたルールを後に挿入)］の順に選択します。

2 ［一致 - 属性値別］を選択し、［OK］をクリックします。

3 ［アクション］タブをダブルクリックし、アクションを編集します。

4 ［DN を入力］フィールドから、［検索を開始するベース DN を入力してください］を削除します。

5 ［引数の編集］アイコン をクリックして、引数ビルダを起動します。

6 ［Noun (名詞)］リストの［テキスト］を選択します。

7 ［テキスト］をダブルクリックして、引数に追加します。

8 エディタで、参照アイコンをクリックして、検索を開始するコンテナを参照し、選択したら［OK］をクリックします。

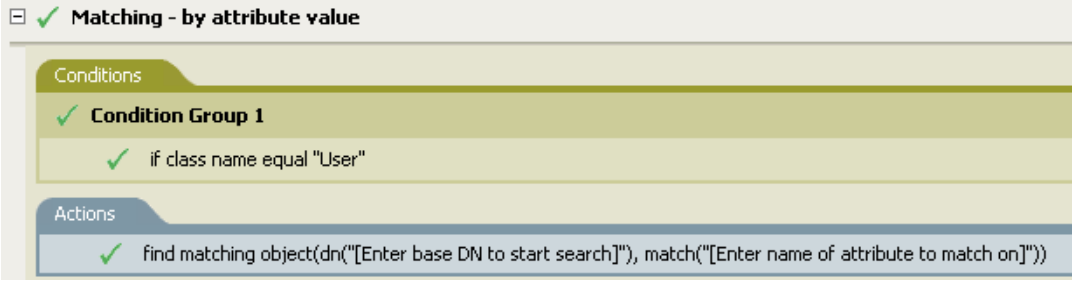
9 ［一致属性を入力］フィールドから、［Enter name of attribute to match on (一致させる属性名を入力してください)］を削除します。

10 ［引数の編集］アイコン をクリックして、一致属性ビルダを起動します。

11 参照アイコンをクリックして、一致させる属性を選択します。一致させる属性を 1 つ以上選択したら、［OK］をクリックします。

12 ［OK］をクリックします。

13 ［ファイル］>［保存］の順にクリックして、ルールを保存します。



ルールの動作

属性でユーザオブジェクトを照合します。ユーザオブジェクトを同期する場合、ドライバはルールを使用して指定した属性が存在するかどうかを確認します。属性が存在しない場合、新しいユーザオブジェクトが作成されます。

配置 - 発行者 (ミラーリング)

指定したポイントからデータストア内のミラー化された構造を使用して、オブジェクトをアイデンティティボールト内に配置します。このルールは、ドライバ内の配置ポリシーに実装します。このルールは、発行者チャネルにのみ実装できます。

この事前定義されたルールを使用するには、配置ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する配置ポリシーがすでにある場合は、97 ページの 「事前定義されたルールのインポート」へ進みます。

ポリシーの作成

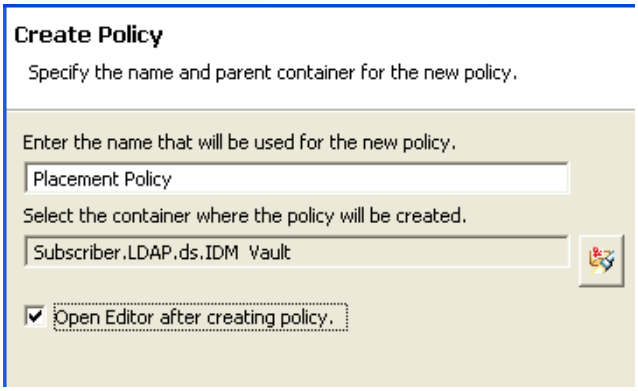
1 ［Outline (アウトライン)］ビューまたは［Policy Flow (ポリシーフロー)］ビューから発行者チャネルを選択します。

2 ポリシーセット内で配置ポリシーセットを選択し、［Create or add a new policy to the Policy Set (新しいポリシーの作成またはポリシーセットへの追加)］アイコン ✚ をクリックして、新しいポリシーを作成します。

3 ［新しいポリシーの作成］をクリックし、［次へ］をクリックします。

4 ポリシーに名前を付けます。

5 表示されている場所を使用して、ポリシーをドライバに配置します。



6 ［Open Editor after creating policy (ポリシーの作成後にエディタを開く)］を選択し、［次へ］をクリックします。

7 ポリシーのタイプとして、［DirXML スクリプト］を選択し、［終了］をクリックします。

8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか ?)」というメッセージが表示されます。［はい］をクリックします。ポリシービルダが起動され、新しい配置ポリシーが保存されます。

事前定義されたルールのインポート

1 ポリシービルダ内で、右クリックして［New (新規作成)］>［Predefined Rule (事前定義されたルール)］>［Insert Predefined Rule Before (事前定義されたルールを前に挿入)］または［Insert Predefined Rule After (事前定義されたルールを後に挿入)］の順に選択します。

2 ［配置 - 発行者 (ミラーリング)］を選択し、［OK］をクリックします。

3 ［条件］タブをダブルクリックし、条件を編集します。

4 ［値］フィールドから、［ソース階層のベースを入力してください］を削除します。

5 オブジェクトをイベントの対象にするソース階層でコンテナを参照し、選択したら［OK］をクリックします。

6 ［アクション］タブをダブルクリックし、アクションを編集します。

7 ［文字列を入力］フィールドから、[宛先階層のベースを入力してください] を削除します。

8 ［引数の編集］アイコン をクリックして、引数ビルダを起動します。

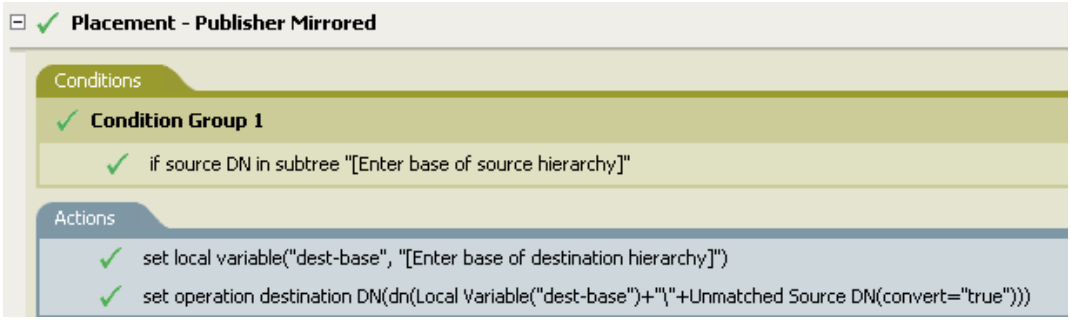
9 ［Noun (名詞)］リストの［テキスト］を選択します。

10 ［テキスト］をダブルクリックして、引数に追加します。

11 エディタで、参照アイコンをクリックして、オブジェクトを配置する宛先階層内のコンテナを参照し、選択したら［OK］をクリックします。

12 ［OK］をクリックします。

13 ［ファイル］>［保存］の順にクリックして、ルールを保存します。



ルールの動作

ユーザオブジェクトがソース階層内にある場合、オブジェクトは、データストアからミラー化された構造に配置されます。配置はローカル変数 dest-base が定義されているポイ

ントから開始します。ユーザオブジェクトは「dest-base\ 一致しないソース DN」に配置されます。このルールではスラッシュ形式を使用します。

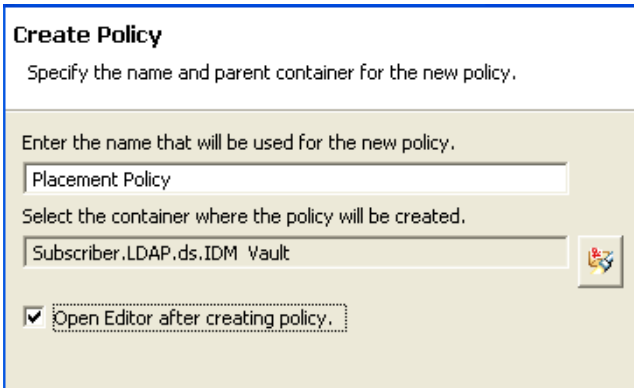
配置 - 購読者 (ミラーリング)-LDAP 形式

指定したポイントからアイデンティティボールト内のミラー化された構造を使用して、オブジェクトをデータストア内に配置します。このルールは、ドライバ内の配置ポリシーに実装します。このルールは、購読者チャネルにのみ実装できます。

この事前定義されたルールを使用するには、配置ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する配置ポリシーがすでにある場合は、99 ページの 「事前定義されたルールのインポート」へ進みます。

ポリシーの作成

1. ［Outline (アウトライン)］ビューまたは［Policy Flow (ポリシーフロー)］ビューから発行者チャネルを選択します。
2. ［Policy Set (ポリシーセット)］ビューで配置ポリシーセットを選択し、［Create or add a new policy to the Policy Set (新しいポリシーの作成またはポリシーセットへの追加)］アイコン をクリックして、新しいポリシーを作成します。
3. ［新しいポリシーの作成］をクリックし、［次へ］をクリックします。
4. ポリシーに名前を付けます。
5. 表示されている場所を使用して、ポリシーをドライバに配置します。



6. ［Open Editor after creating policy (ポリシーの作成後にエディタを開く)］を選択し、［次へ］をクリックします。
7. ポリシーのタイプとして、［DirXML スクリプト］を選択し、［終了］をクリックします。
8. ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか ?)」というメッセージが表示されます。［はい］をクリックします。ポリシービルダが起動され、新しい配置ポリシーが保存されます。

事前定義されたルールのインポート

1 ポリシービルダ内で、右クリックして［New (新規作成)］>［Predefined Rule (事前定義されたルール)］>［Insert Predefined Rule Before (事前定義されたルールを前に挿入)］または［Insert Predefined Rule After (事前定義されたルールを後に挿入)］の順に選択します。

2 ［配置 - 購読者 (ミラーリング) - LDAP 形式］を選択し、［OK］をクリックします。

3 ［条件］タブをダブルクリックし、条件を編集します。

4 ［値］フィールドから、［ソース階層のベースを入力してください］を削除します。

5 オブジェクトをイベントの対象にするソース階層でコンテナを参照し、選択したら［OK］をクリックします。

6 ［アクション］タブをダブルクリックし、アクションを編集します。

7 ［文字列を入力］フィールドから、[宛先階層のベースを入力してください] を削除します。

8 ［引数の編集］アイコン をクリックして、引数ビルダを起動します。

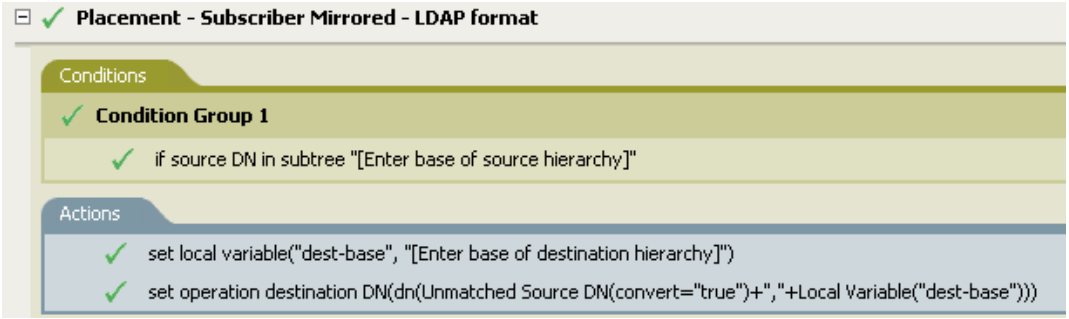
9 ［Noun (名詞)］リストの［テキスト］を選択します。

10 ［テキスト］をダブルクリックして、引数に追加します。

11 エディタで、参照アイコンをクリックして、オブジェクトを配置する宛先階層内のコンテナを参照し、選択したら［OK］をクリックします。

12 ［OK］をクリックします。

13 ［ファイル］>［保存］の順にクリックして、ルールを保存します。



ルールの動作

ユーザオブジェクトがソース階層内にある場合、オブジェクトは、アイデンティティボールトからミラー化された構造に配置されます。配置はローカル変数 dest-base が定義されているポイントから開始します。ユーザオブジェクトは一致しないソース DN「dest-base」に配置されます。このルールでは LDAP 形式を使用します。

配置 - 発行者 (フラット)

データストアのオブジェクトをアイデンティティボールトの 1 つのコンテナ内に配置します。このルールは、ドライバ内の配置ポリシーに実装します。このルールは、発行者チャネルにのみ実装できます。

この事前定義されたルールを使用するには、配置ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加

する配置ポリシーがすでにある場合は、100 ページの 「事前定義されたルールのインポート」へ進みます。

ポリシーの作成

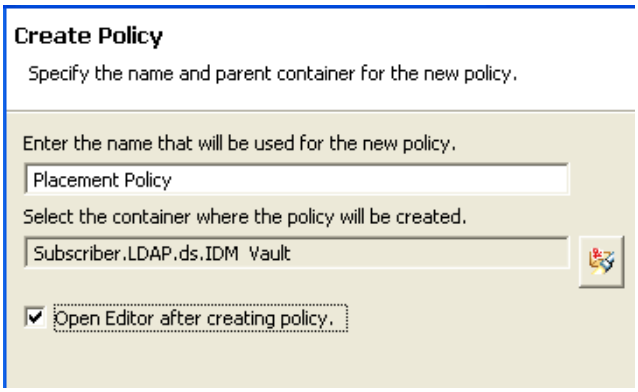
1 ［Outline (アウトライン)］ビューまたは［Policy Flow (ポリシーフロー)］ビューから発行者チャネルを選択します。

2 ［Policy Set (ポリシーセット)］ビューで配置ポリシーセットを選択し、［Create or add a new policy to the Policy Set (新しいポリシーの作成またはポリシーセットへの追加)］アイコン をクリックして、新しいポリシーを作成します。

3 ［新しいポリシーの作成］をクリックし、［次へ］をクリックします。

4 ポリシーに名前を付けます。

5 表示されている場所を使用して、ポリシーをドライバに配置します。



6 ［Open Editor after creating policy (ポリシーの作成後にエディタを開く)］を選択し、［次へ］をクリックします。

7 ポリシーのタイプとして、［DirXML スクリプト］を選択し、［終了］をクリックします。

8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか ?)」というメッセージが表示されます。［はい］をクリックします。ポリシービルダが起動され、新しい配置ポリシーが保存されます。

事前定義されたルールのインポート

1 ポリシービルダ内で、右クリックして［New (新規作成)］> ［Predefined Rule (事前定義されたルール)］> ［Insert Predefined Rule Before (事前定義されたルールを前に挿入)］または［Insert Predefined Rule After (事前定義されたルールを後に挿入)］の順に選択します。

2 ［配置 - 発行者 (フラット)］を選択し、［OK］をクリックします。

3 ［アクション］タブをダブルクリックし、アクションを編集します。

4 ［文字列を入力］フィールドから、［宛先コンテナの DN を入力してください］を削除します。

5 ［引数の編集］アイコン をクリックして、引数ビルダを起動します。

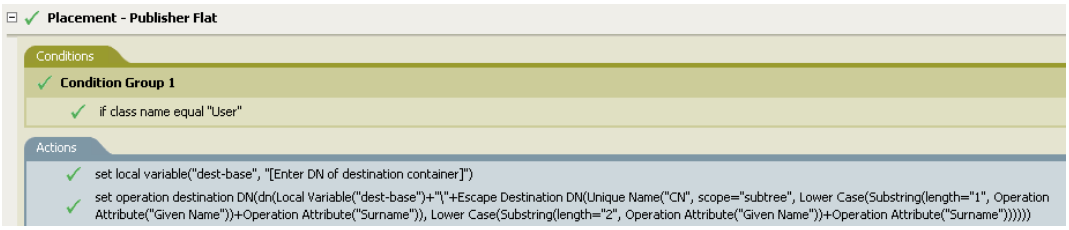
6 ［Noun (名詞)］リストの［テキスト］を選択します。

7 ［テキスト］をダブルクリックして、引数に追加します。

8 エディタで、参照アイコンをクリックして、すべてのユーザオブジェクトを配置するターゲットコンテナを参照し、選択したら［OK］をクリックします。

9 ［OK］をクリックします。

10 ［ファイル］>［保存］の順にクリックして、ルールを保存します。



ルールの動作

このルールは、すべてのユーザオブジェクトをターゲット DN に配置します。このルールでは、ターゲットコンテナの DN をローカル変数 dest-base として設定します。その後で、ターゲット DN を dest-base\CN 属性に設定します。ユーザオブジェクトの CN 属性は、名前属性および名字属性の最初の 2 文字 (小文字) になります。このルールではスラッシュ形式を使用します。

配置 - 購読者 (フラット)-LDAP 形式

アイデンティティボールトのオブジェクトをデータストア内の 1 つのコンテナに配置します。このルールは、ドライバ内の購読者配置ポリシーに実装します。

この事前定義されたルールを使用するには、配置ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する配置ポリシーがすでにある場合は、102 ページの 「事前定義されたルールのインポート」へ進みます。

ポリシーの作成

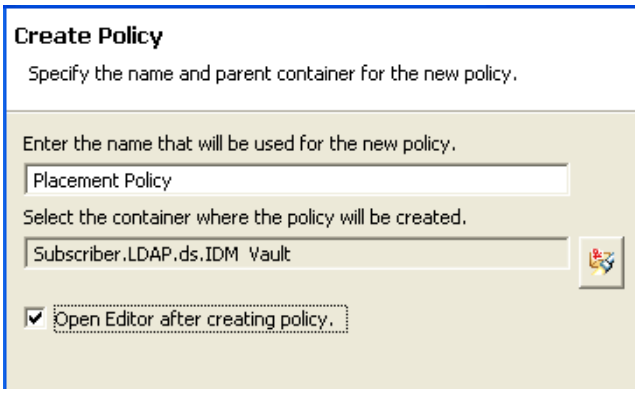
1 ［Outline (アウトライン)］ビューまたは［Policy Flow (ポリシーフロー)］ビューから発行者チャネルを選択します。

2 ［Policy Set (ポリシーセット)］ビューで配置ポリシーセットを選択し、［Create or add a new policy to the Policy Set (新しいポリシーの作成またはポリシーセットへの追加)］アイコン ✚ をクリックして、新しいポリシーを作成します。

3 ［新しいポリシーの作成］をクリックし、［次へ］をクリックします。

4 ポリシーに名前を付けます。

5 表示されている場所を使用して、ポリシーをドライバに配置します。



6 ［Open Editor after creating policy (ポリシーの作成後にエディタを開く)］を選択し、［次へ］をクリックします。

7 ポリシーのタイプとして、［DirXML スクリプト］を選択し、［終了］をクリックします。

8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか ?)」というメッセージが表示されます。［はい］をクリックします。ポリシービルダが起動され、新しい配置ポリシーが保存されます。

事前定義されたルールのインポート

1 ポリシービルダ内で、右クリックして［New (新規作成)］> ［Predefined Rule (事前定義されたルール)］> ［Insert Predefined Rule Before (事前定義されたルールを前に挿入)］または［Insert Predefined Rule After (事前定義されたルールを後に挿入)］の順に選択します。

2 ［配置 - 購読者 (フラット)-LDAP 形式］を選択し、［OK］をクリックします。

3 ［アクション］タブをダブルクリックし、アクションを編集します。

4 ［文字列を入力］フィールドから、［宛先コンテナの DN を入力してください］を削除します。

5 ［引数の編集］アイコン をクリックして、引数ビルダを起動します。

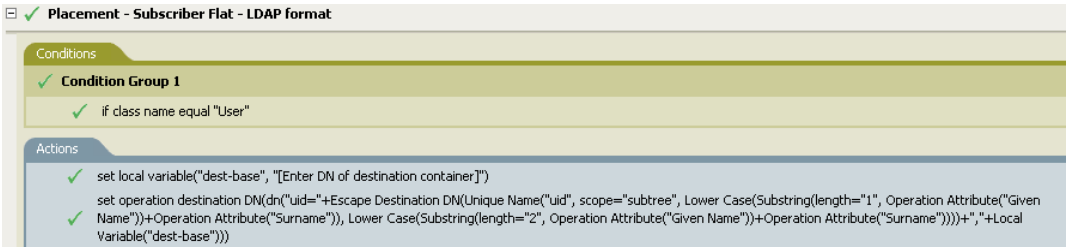
6 ［Noun (名詞)］リストの［テキスト］を選択します。

7 ［テキスト］をダブルクリックして、引数に追加します。

8 エディタで、すべてのユーザオブジェクトを配置するターゲットコンテナを追加します。コンテナが LDAP 形式で指定されていることを確認し、［OK］をクリックします。

9 ［OK］をクリックします。

10 ［ファイル］>［保存］の順にクリックして、ルールを保存します。



ルールの動作

このルールは、すべてのユーザオブジェクトをターゲット DN に配置します。このルールでは、ターゲットコンテナの DN をローカル変数 dest-base として設定します。その後で、ターゲット DN を uid= 一意の名前、dest-base に設定します。ユーザオブジェクトの uid 属性は、名前属性および名字属性の最初の 2 文字 (小文字) になります。このルールでは LDAP 形式を使用します。

配置 - 部署別発行者

オブジェクトを、データストア内の 1 つのコンテナからアイデンティティボールト内の複数のコンテナ内に配置します。このルールは、ドライバ内の配置ポリシーに実装します。このルールは、発行者チャネルにのみ実装できます。

この事前定義されたルールを使用するには、配置ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する配置ポリシーがすでにある場合は、104 ページの 「事前定義されたルールのインポート」へ進みます。

ポリシーの作成

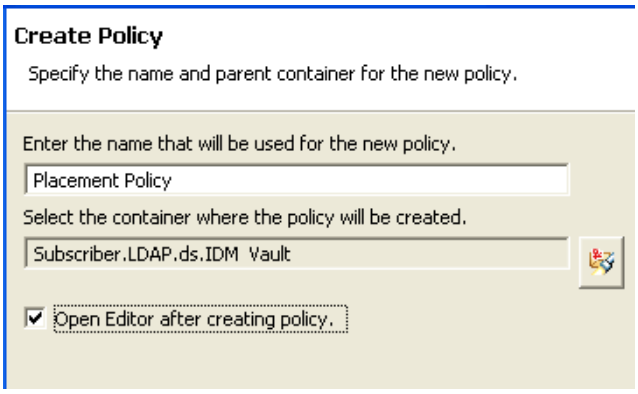
1 ［Outline (アウトライン)］ビューまたは［Policy Flow (ポリシーフロー)］ビューから発行者チャネルを選択します。

2 ［Policy Set (ポリシーセット)］ビューで配置ポリシーセットを選択し、［Create or add a new policy to the Policy Set (新しいポリシーの作成またはポリシーセットへの追加)］アイコン ✚ をクリックして、新しいポリシーを作成します。

3 ［新しいポリシーの作成］をクリックし、［次へ］をクリックします。

4 ポリシーに名前を付けます。

5 表示されている場所を使用して、ポリシーをドライバに配置します。



6 ［Open Editor after creating policy (ポリシーの作成後にエディタを開く)］を選択し、［次へ］をクリックします。

7 ポリシーのタイプとして、［DirXML スクリプト］を選択し、［終了］をクリックします。

8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか ?)」というメッセージが表示されます。［はい］をクリックします。ポリシービルダが起動され、新しい配置ポリシーが保存されます。

事前定義されたルールのインポート

1 ポリシービルダ内で、右クリックして［New (新規作成)］> ［Predefined Rule (事前定義されたルール)］> ［Insert Predefined Rule Before (事前定義されたルールを前に挿入)］または［Insert Predefined Rule After (事前定義されたルールを後に挿入)］の順に選択します。

2 ［配置 - 部署別発行者］を選択し、［OK］をクリックします。

3 ［アクション］タブをダブルクリックし、アクションを編集します。

4 ［文字列を入力］フィールドから、［宛先組織の DN を入力してください］を削除します。

5 ［引数の編集］アイコン をクリックして、引数ビルダを起動します。

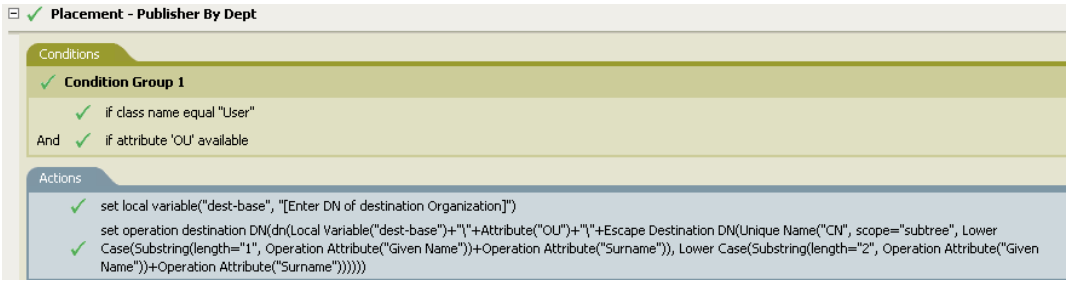
6 ［Noun (名詞)］リストの［テキスト］を選択します。

7 ［テキスト］をダブルクリックして、引数に追加します。

8 エディタで、参照アイコンをクリックして、アイデンティティボールト内の親コンテナを参照し、選択します。すべての部署別コンテナがこの DN の子コンテナであることを確認し、［OK］をクリックします。

9 ［OK］をクリックします。

10 ［ファイル］>［保存］の順にクリックして、ルールを保存します。



ルールの動作

このルールでは、OU 属性に格納された値に基づいて、ユーザオブジェクトを適切な部署別コンテナに配置します。配置する必要があり、使用可能な OU 属性を持っているユーザオブジェクトの場合は、「dest-base\OU 属性 \CN 属性の値」に配置されます。

dest-base はローカル変数です。DN は、部署別コンテナのルートの相対パスである必要があります。このパスは組織または部門になります。OU 属性に格納された値は、ローカル変数 dest-base の子コンテナ名である必要があります。

子コンテナは、配置されるユーザオブジェクトに関連付けられている必要があります。OU 属性の値は、子コンテナ名である必要があります。OU 属性が存在しない場合、このルールは実行されません。

ユーザオブジェクトの CN 属性は、名前属性および名字属性の最初の 2 文字 (小文字) になります。このルールではスラッシュ形式を使用します。

配置 - 部署別購読者 -LDAP 形式

OU 属性に基づいて、オブジェクトを、アイデンティティボールト内の 1 つのコンテナからデータストア内の複数のコンテナ内に配置します。このルールは、ドライバ内の配置ポリシーに実装します。このルールは、購読者チャネルにのみ実装できます。

この事前定義されたルールを使用するには、配置ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する配置ポリシーがすでにある場合は、106 ページの 「事前定義されたルールのインポート」へ進みます。

ポリシーの作成

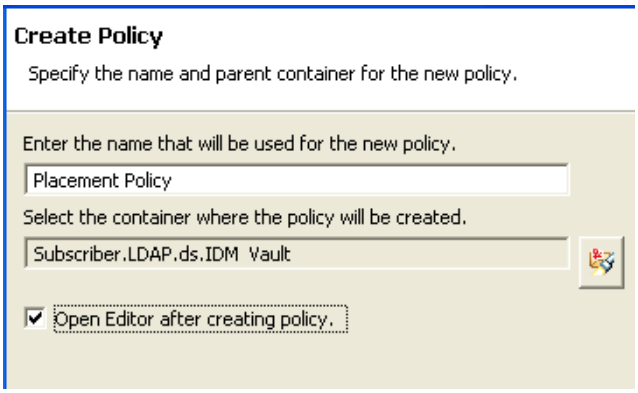
1 ［Outline (アウトライン)］ビューまたは［Policy Flow (ポリシーフロー)］ビューから発行者チャネルを選択します。

2 ［Policy Set (ポリシーセット)］ビューで配置ポリシーセットを選択し、［Create or add a new policy to the Policy Set (新しいポリシーの作成またはポリシーセットへの追加)］アイコン ✚ をクリックして、新しいポリシーを作成します。

3 ［新しいポリシーの作成］をクリックし、［次へ］をクリックします。

4 ポリシーに名前を付けます。

5 表示されている場所を使用して、ポリシーをドライバに配置します。



6 ［Open Editor after creating policy (ポリシーの作成後にエディタを開く)］を選択し、［次へ］をクリックします。

7 ポリシーのタイプとして、［DirXML スクリプト］を選択し、［終了］をクリックします。

8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか ?)」というメッセージが表示されます。［はい］をクリックします。ポリシービルダが起動され、新しい配置ポリシーが保存されます。

事前定義されたルールのインポート

1 ポリシービルダ内で、右クリックして［New (新規作成)］> ［Predefined Rule (事前定義されたルール)］> ［Insert Predefined Rule Before (事前定義されたルールを前に挿入)］または［Insert Predefined Rule After (事前定義されたルールを後に挿入)］の順に選択します。

2 ［配置 - 部署別購読者 -LDAP 形式］を選択し、［OK］をクリックします。

3 ［アクション］タブをダブルクリックし、アクションを編集します。

4 ［文字列を入力］フィールドから、［宛先組織の DN を入力してください］を削除します。

5 ［引数の編集］アイコン をクリックして、引数ビルダを起動します。

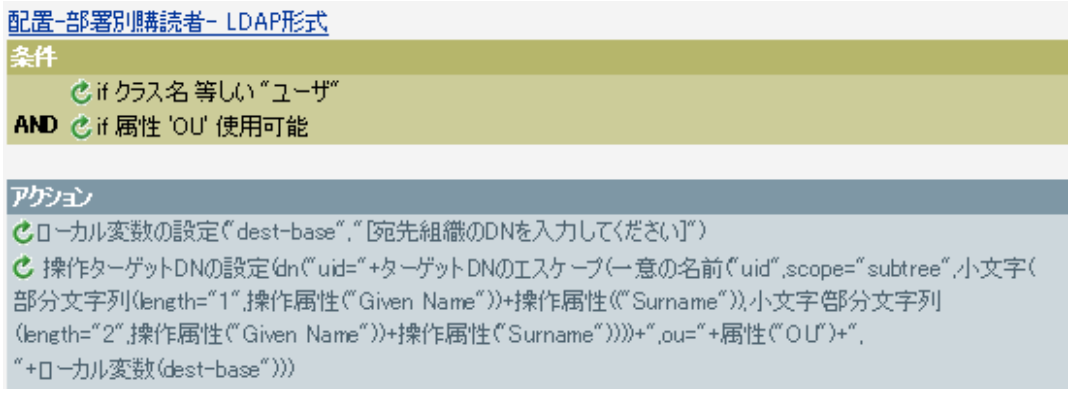
6 ［Noun (名詞)］リストの［テキスト］を選択します。

7 ［テキスト］をダブルクリックして、引数に追加します。

8 エディタで、データストアに親コンテナを追加します。この親コンテナは、LDAP 形式で指定する必要があります。すべての部署別コンテナがこの DN の子コンテナであることを確認し、［OK］をクリックします。

9 ［OK］をクリックします。

10 ［ファイル］>［保存］の順にクリックして、ルールを保存します。



ルールの動作

このルールでは、OU 属性に格納された値に基づいて、ユーザオブジェクトを適切な部署別コンテナに配置します。配置する必要があり、使用可能な OU 属性を持っているユーザオブジェクトの場合は、「uid= 一意の名前 ,ou=OU 属性の値 ,dest-base」に配置されます。

dest-base はローカル変数です。DN は、部署別コンテナのルートの相対パスである必要があります。このパスは組織または部門になります。OU 属性に格納された値は、ローカル変数 dest-base の子コンテナ名である必要があります。

子コンテナは、配置されるユーザオブジェクトに関連付けられている必要があります。OU 属性の値は、子コンテナ名である必要があります。OU 属性が存在しない場合、このルールは実行されません。

ユーザオブジェクトの uid 属性は、名前属性および名字属性の最初の 2 文字 (小文字) になります。このルールでは LDAP 形式を使用します。

2.2.7 ポリシーシミュレータを使用したポリシーのテスト

ポリシーシミュレータを使用すると、アイデンティティボールトにポリシーを実装しなくても、ドライバのフローの任意の時点でポリシーを実行して、結果を確認できます。また、運用環境、または接続システムに影響を与えずにポリシーをテストできます。

ポリシーシミュレータで一般的なタスクについては、次の節を参照してください。

- 108 ページの 「ポリシーシミュレータへのアクセス」
- 109 ページの 「ポリシーシミュレータの使用」

ポリシーシミュレータは、XML を使用します。eDirectory ドキュメントタイプ定義ファイル (nds.dtd) では、メタディレクトリエンジンが処理できる XML ドキュメントのスキーマを定義します。このスキーマに準拠しない XML ドキュメントでは、エラーが発生します。ドキュメントが nds.dtd に準拠しているかどうかを確認し、エラー発生の理由についての情報を調べるには、「eDirectory DTD Commands and Events (http://developer.novell.com/ndk/doc/dirxml/index.html?page=/ndk/doc/dirxml/dirxmlbk/data/a36pjzu.html)」参照してください。

ポリシーシミュレータでは、SOAP および区切りテキストなどのアプリケーションドライバの最初のポリシーセットのシミュレートはできません。これらのドライバは、カンマ区

切りのファイルまたはテキストファイルを入力として使用し、XML または XDS をポリシーチェーン内のポリシーから派生します。現在、ポリシーシミュレータが入力として受け付けるのは、有効な XML または XDS だけです。将来のリリースで、機能が追加される可能性があります。

ポリシーシミュレータへのアクセス

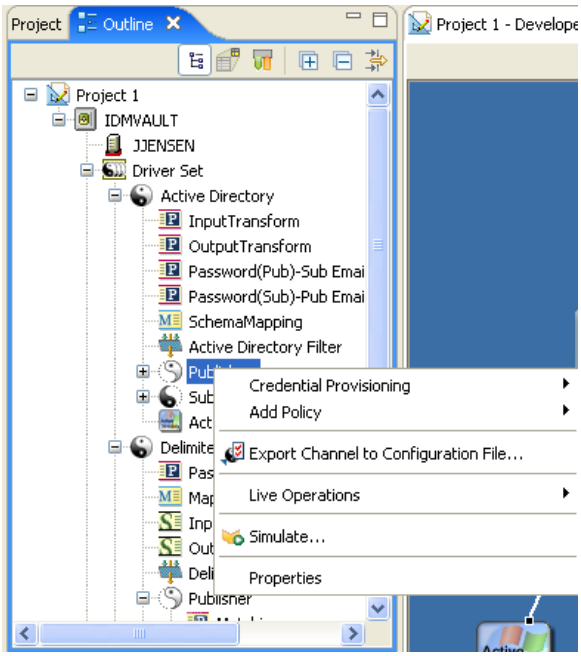
ポリシーシミュレータには、次の 3 つの方法でアクセスできます。

- 108 ページの 「[Outline (アウトライン)] ビュー」
- 108 ページの 「ポリシーフロー」
- 109 ページの 「エディタ」

[Outline (アウトライン)] ビュー

1 [Show Model Outline (モデルアウトラインの表示)] アイコン をクリックします。

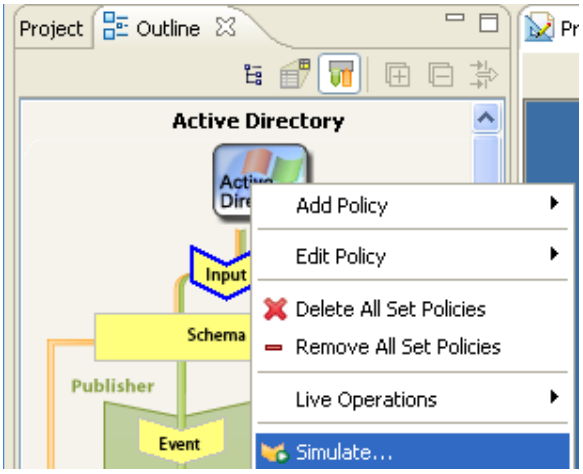
2 シミュレートするドライバ、発行者、購読者、マッピングルール、フィルタ、または任意のポリシーを右クリックし、[Simulate (シミュレート)] をクリックします。



ポリシーフロー

1 [Show Policy Flow (ポリシーフローの表示)] アイコン をクリックします。

2 シミュレートする入力、出力、スキーママッピング、フィルタ、および任意のポリシーセットのアイコンを右クリックし、［Simulate (シミュレート)］をクリックします。



エディタ

ポリシーシミュレータには、ポリシービルダ、スキーママッピングエディタ、またはフィルタエディタから、ツールバーの［Policy Simulator (ポリシーシミュレータ)］アイコン を選択してアクセスできます。

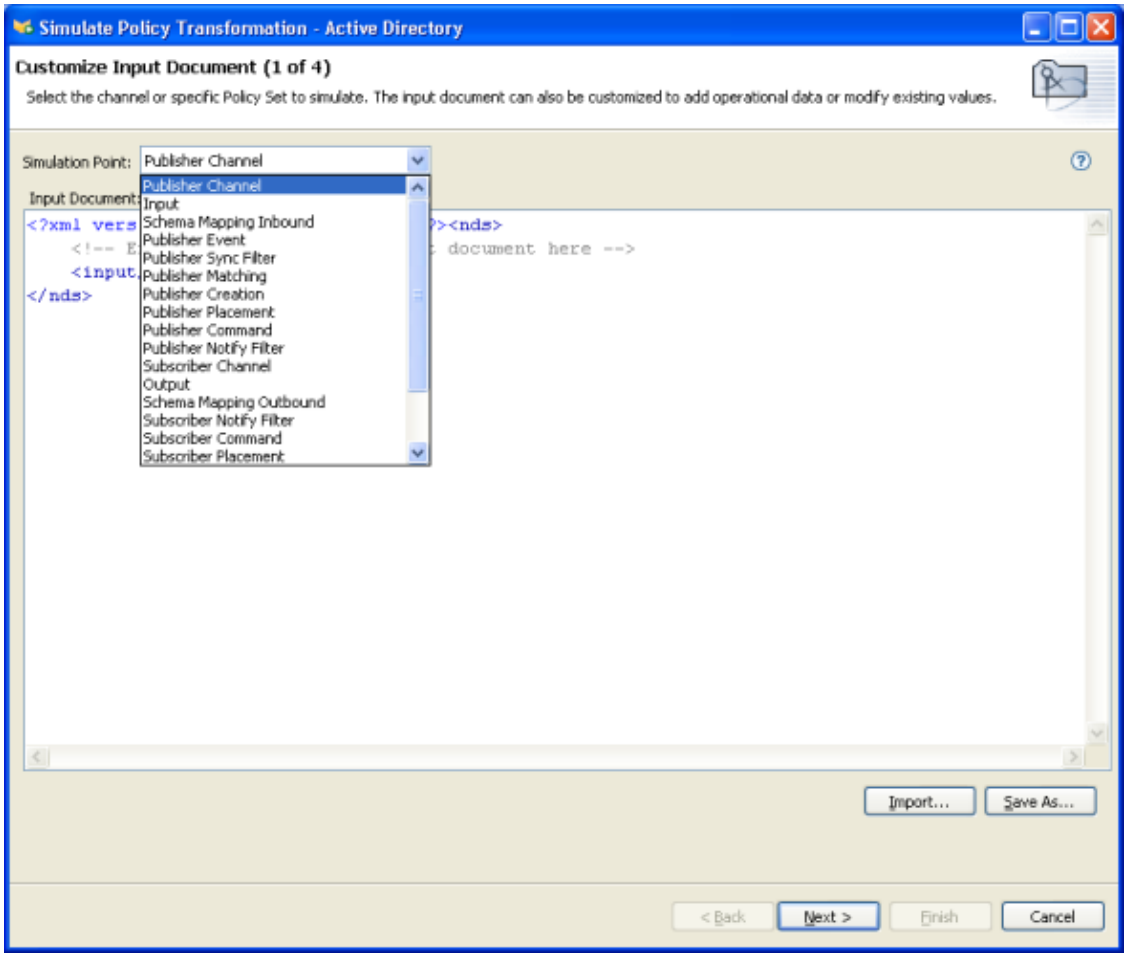
ポリシーシミュレータの使用

ポリシーシミュレータでは、ドライバフロー内のポイントを選択して、ポリシーを特定の操作でテストできます。テスト中に、入力ドキュメントおよび出力ドキュメントを編集できます。変更を保持する場合は、［名前を付けて保存］アイコンを選択して、ドキュメントを XML ファイルとして保存します。

ポリシーシミュレータを使用するには

1 ［Simulation Point (シミュレーションポイント)］ドロップダウンリストで、ポリシーをテストするドライバフロー内の場所を選択します。［発行者チャネル］、［購読者チャネル］、［入力］、［スキーマのマッピング］、［イベント］、［Sync Filter (同期フィルタ)］、［一致］、［作成］、［配置］、［コマンド］、および［Notify Filter (通知フィルタ)］の任意の項目を選択できます。

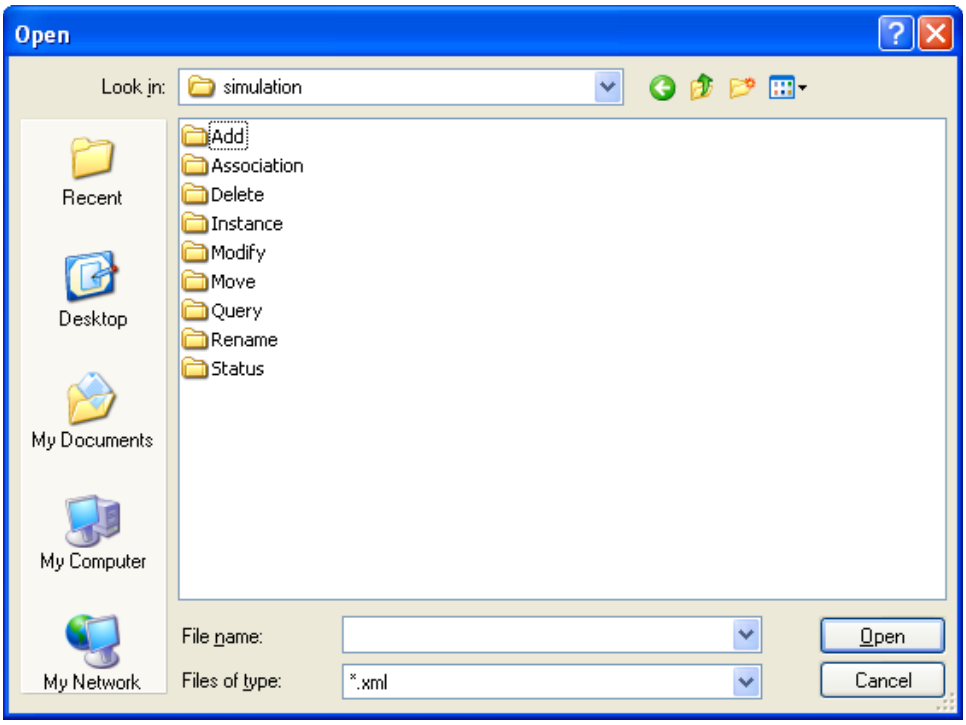
テストする特定のポリシーまたはルールを選択する場合は、［Simulation Point (シミュレーションポイント)］オプションには、［To NDS (NDS へ)］または［From NDS (NDS から)］だけが表示されます。



2 ［Import (インポート)］を選択し、テストするファイルを参照して選択します。

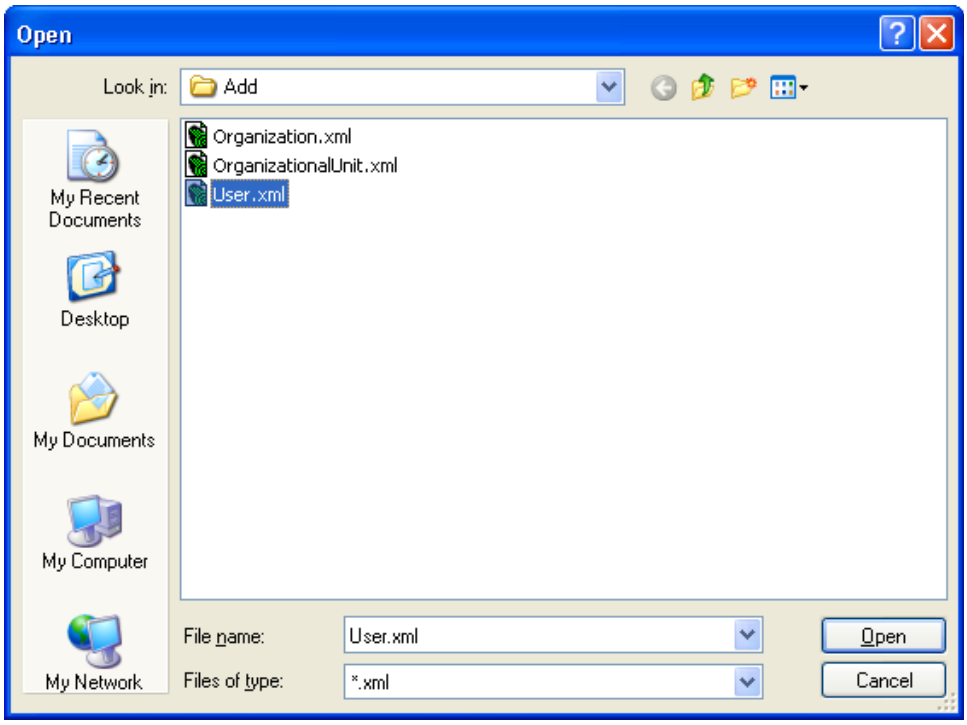
Designer にはサンプルのイベントファイルが付属します。ファイルは、プラグインの com.novell.designer.idm.policy\simulation にあります。イベントは、Add (追加)、Association (関連付け)、Delete (削除)、Instance (インスタンス)、Modify (変更)、

Move (移動)、Query (クエリ)、Rename (名前変更)、および Status (ステータス) です。



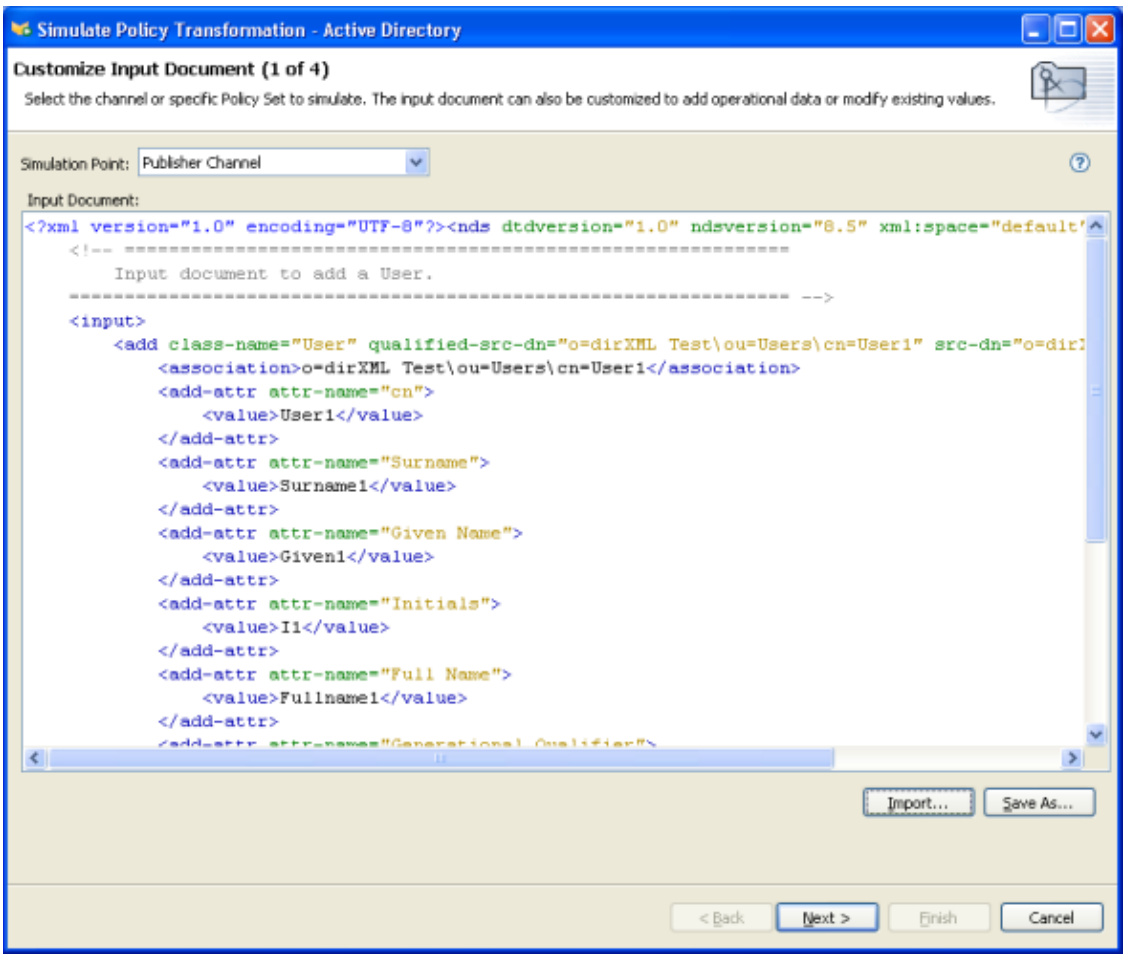
3 フォルダをダブルクリックして、使用可能なイベントを表示します。選択できるファイルは、イベントごとに異なります。たとえば、[Add (追加)] を選択した場合は、Organization.xml、OrganizationalUnit.xml、および User.xml の 3 つのオプションがあり

ます。ファイルはイベントを示します。User.xml を選択した場合、これはユーザオブジェクトの追加イベントになります。

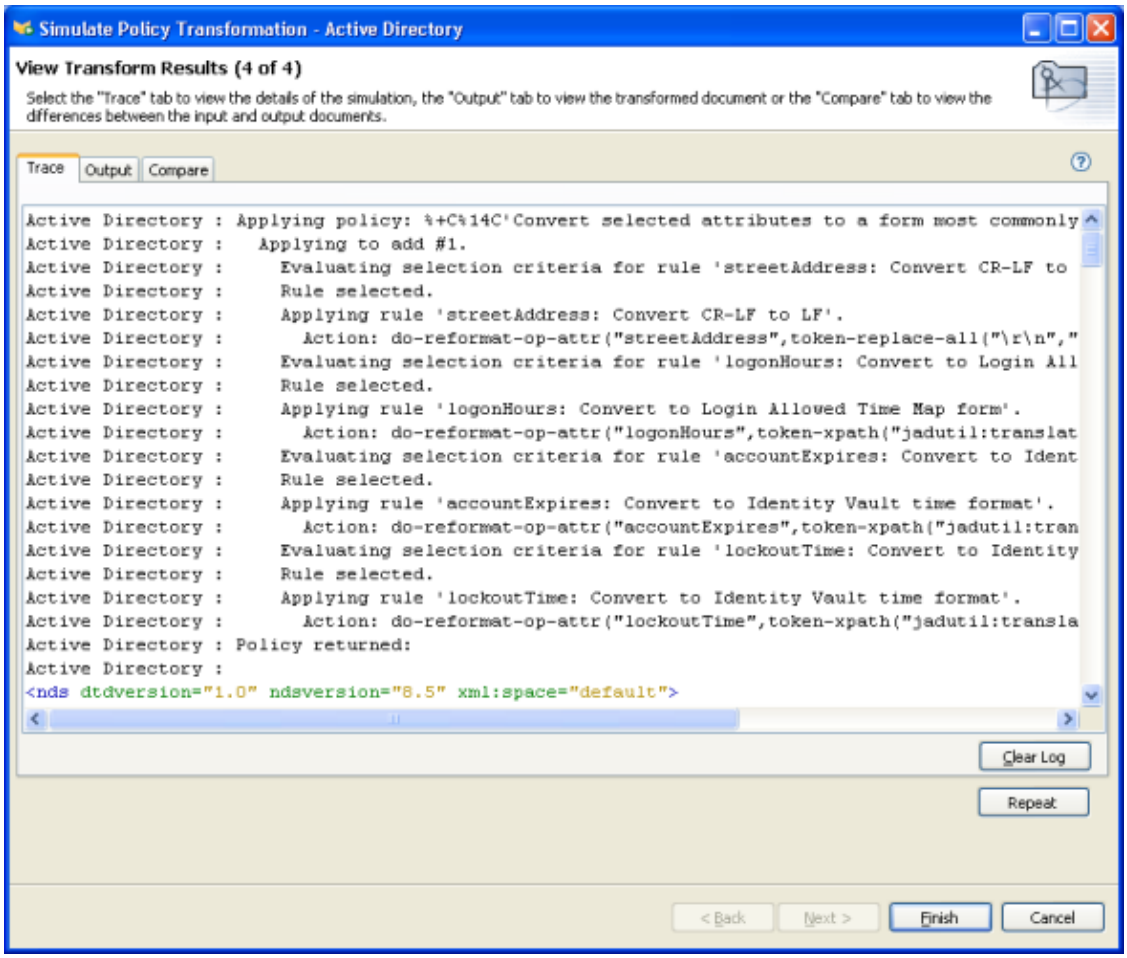


4 ファイルを選択し、［開く］をクリックして、ウィンドウに入力ドキュメントを表示します。

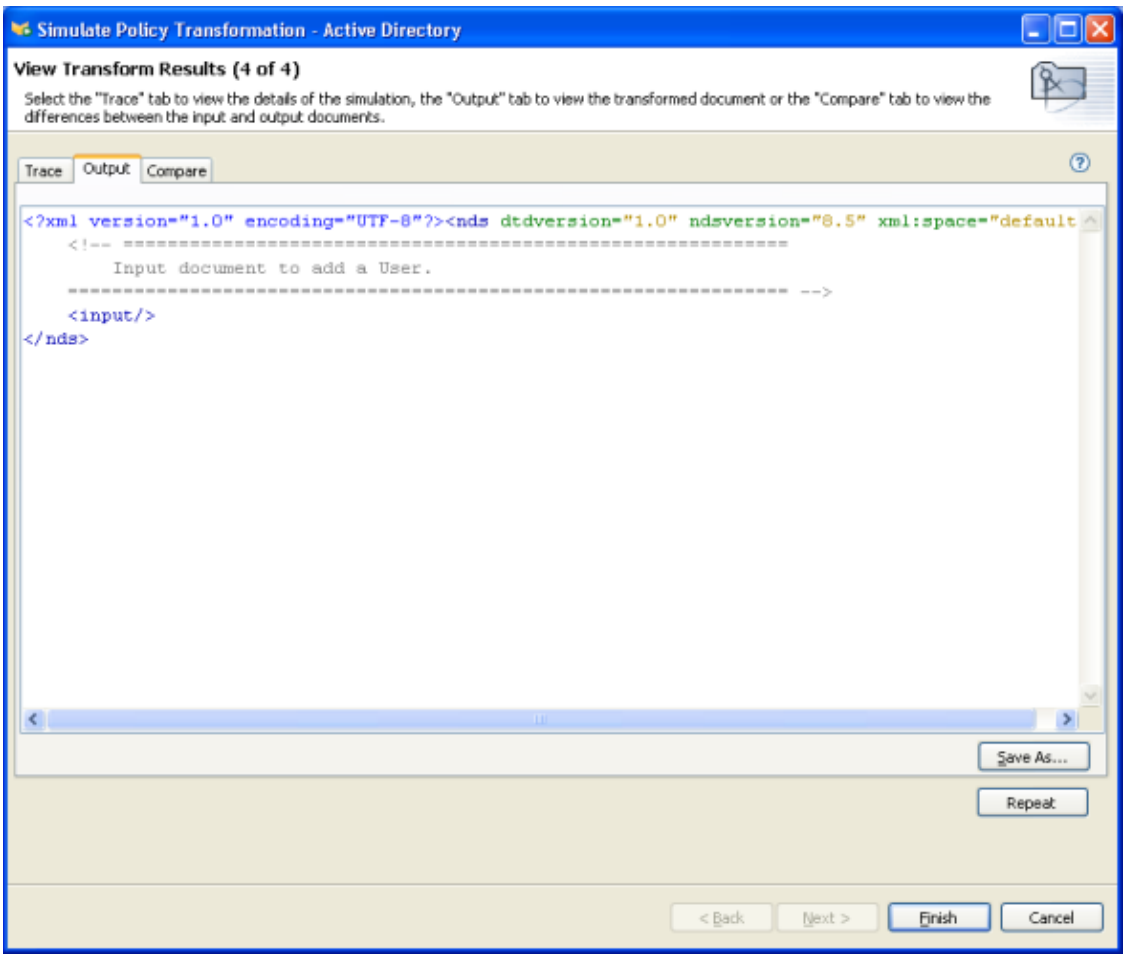
5［次へ］をクリックします。



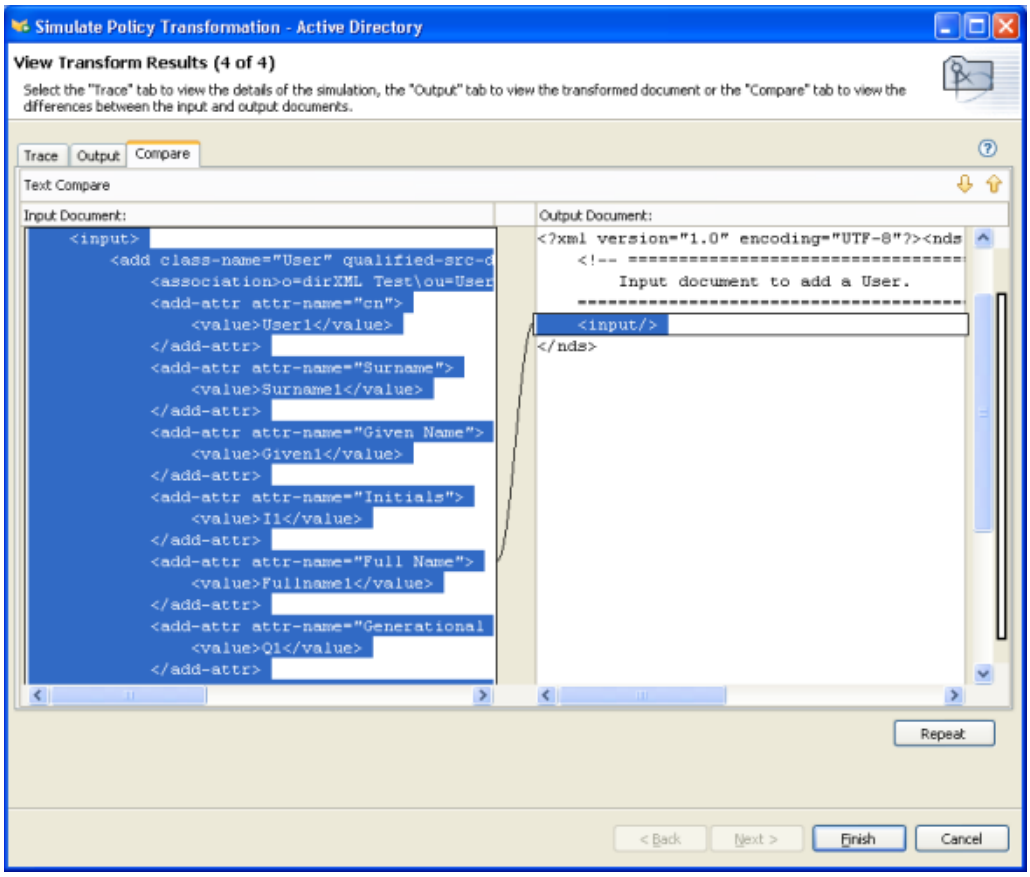
6［トレース］タブを選択して、ポリシーが処理されたときのイベントの結果を表示します。このウィンドウ内の情報は、DSTRACE に表示される情報と同じです。



7［出力］タブを選択して、生成された出力ドキュメントを表示します。



8 ［Compare (比較)］タブを選択して、出力ドキュメントを入力ドキュメントと比較します。



9 結果を確認したら、［Repeat (繰り返し)］をクリックして、同じポリシーで別のイベントをテストします。

10 テストが済んだら、［終了］をクリックして、ポリシーシミュレータを閉じます。

Java 拡張を使用したポリシーのシミュレート

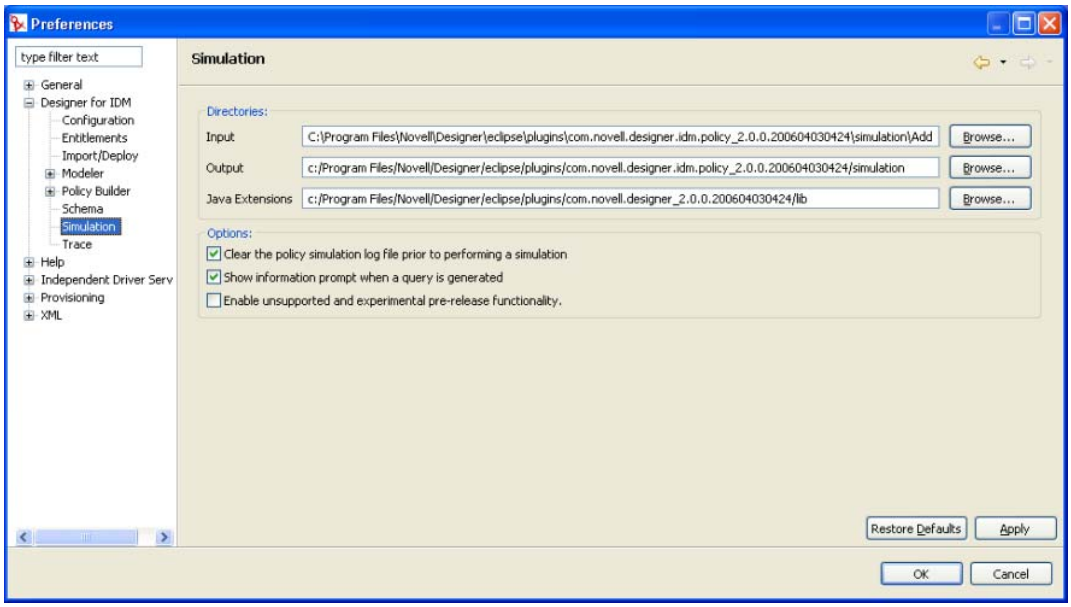
jar ファイルが置かれているディレクトリを指定することによって、外部の Java 拡張への参照を含むポリシーをシミュレートできるようになりました。

拡張ディレクトリを決定または変更するには

1 ツールバーで［Windows (ウィンドウ)］>［初期設定］の順に選択します。

2 ［Designer for IDM (IDM の Designer)］>［Simulation (シミュレーション)］の順に移動します。

3 Java クラスを含む jar ファイルを、指定したディレクトリにコピーし、ポリシーをシミュレートします。



注 : [Enable unsupported and experimental pre-release functionality (サポートされていないか、事前公開されている実験的な機能を有効にする)] オプションを使用すると、ポリシーシミュレータで、ライブのアイデンティティボールトまたは接続システムに対するポリシーをテストできます。このオプションは、Designer 1.2 ではサポートされていないため、説明はありません。

2.2.8 DirXML スクリプトの編集

Designer では、XML エディタまたはテキストエディタを使用して、XML を表示、編集、および検証できます。

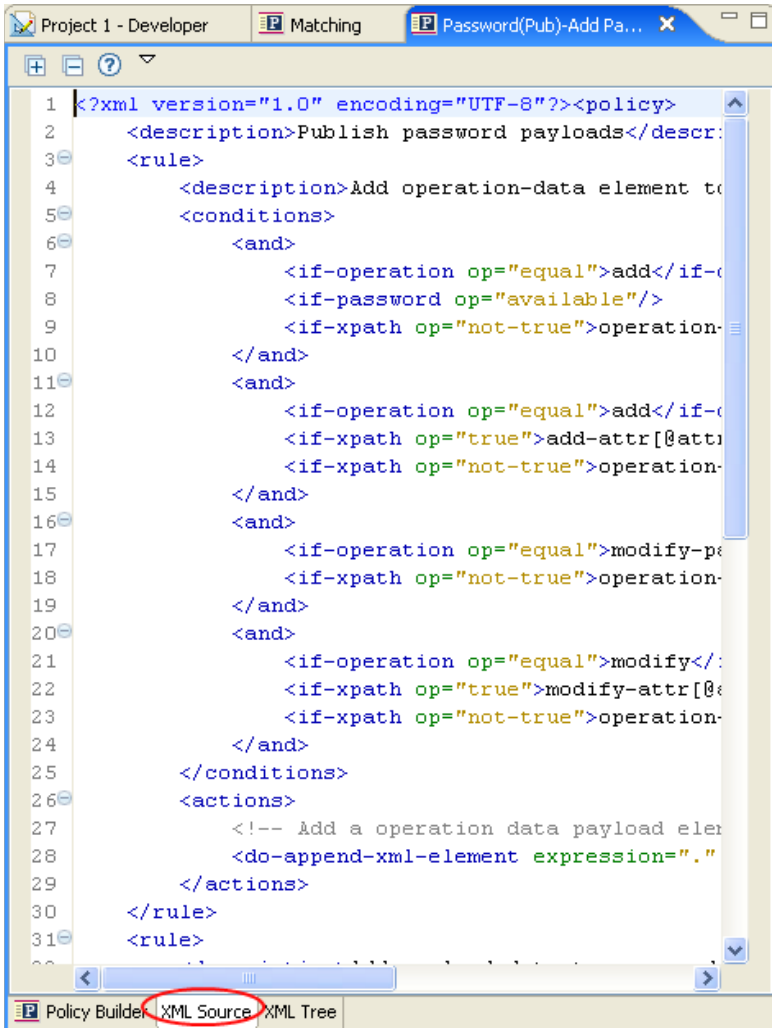
- 117 ページの 「XML ソースの表示」
- 121 ページの 「XML ソースの編集」
- 124 ページの 「XML ソースの検証」

XML ソースの表示

XML ソースは、XML 形式または XML ツリー形式で表示できます。

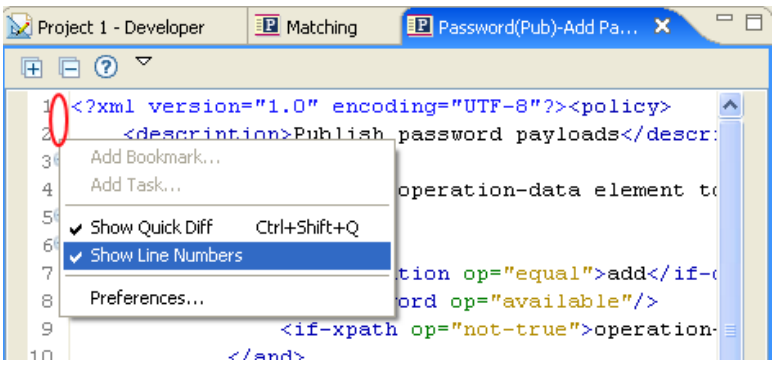
XML ソースビューを開くには

1 ポリシービルダのワークスペースの下部にある［XML Source (XML ソース)］をクリックします。



XML エディタに行番号が表示されます。

2 行番号を表示するには、左の余白を右クリックし、［Show Line Numbers (行番号の表示)］を選択します。



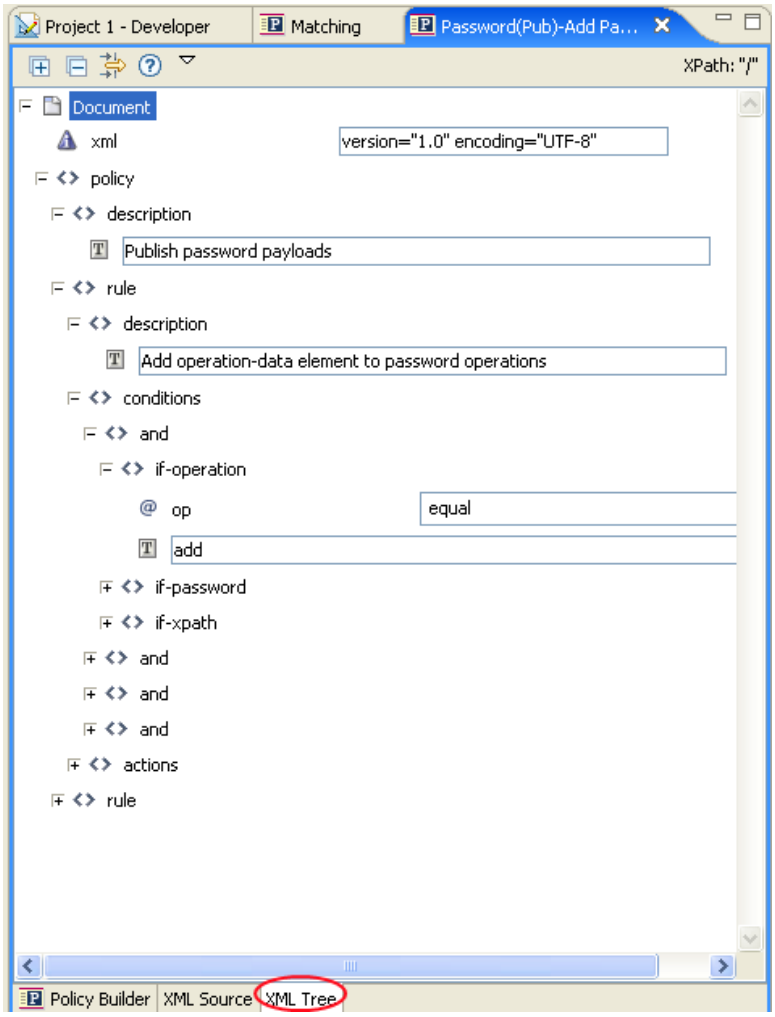
XML エディタは、XML を機能別に展開または縮小します。多くの XML を含む機能が複数ある場合は、左上隅のマイナスアイコンをクリックして、XML を縮小できます。

3 XML 機能をすべて展開するには、左隅のプラスアイコンをクリックします。

各要素には、左の余白にそれぞれのプラスまたはマイナスアイコンがあります。

XML をツリー形式で表示するには

1 ポリシービルダのワークスペースの下部にある［XML Tree (XML ツリー)］をクリックします。

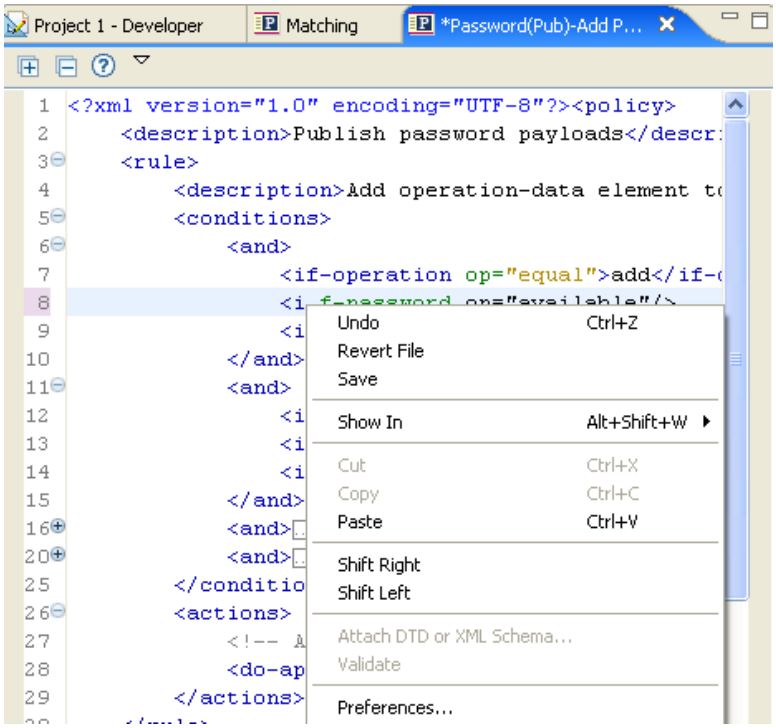


ツリー全体を表示するには、一覧表示されている個々の項目を展開します。

XML ソースの編集

XML は XML エディタで編集できます。GUI を使用する場合と同様、XML エディタで変更することもできます。

図 2-12 *XML ソースの編集*



ロードされるデフォルトエディタは、.xml ファイルのタイプに関連付けられています。デフォルトエディタが見つからない場合は、システムのテキストエディタがロードされます。XML ソースビューの機能は、ロードされるエディタに基づきます。

右クリックすると、XML エディタに含まれる機能のリストが表示されます。

表 2-5 *XML エディタのオプション*

機能	説明
元に戻す	最後のアクションを元に戻します。
Revert File (ファイルを戻す)	ファイルを、保存されていたバージョンに戻します。
保存	ファイルを保存します。
切り取り	選択された情報を切り取ります。
貼り付け	情報をドキュメントに貼り付けます。
Shift Right (右にシフト)	行を右にインデントします。
Shift Left (左にシフト)	行を左にインデントします。

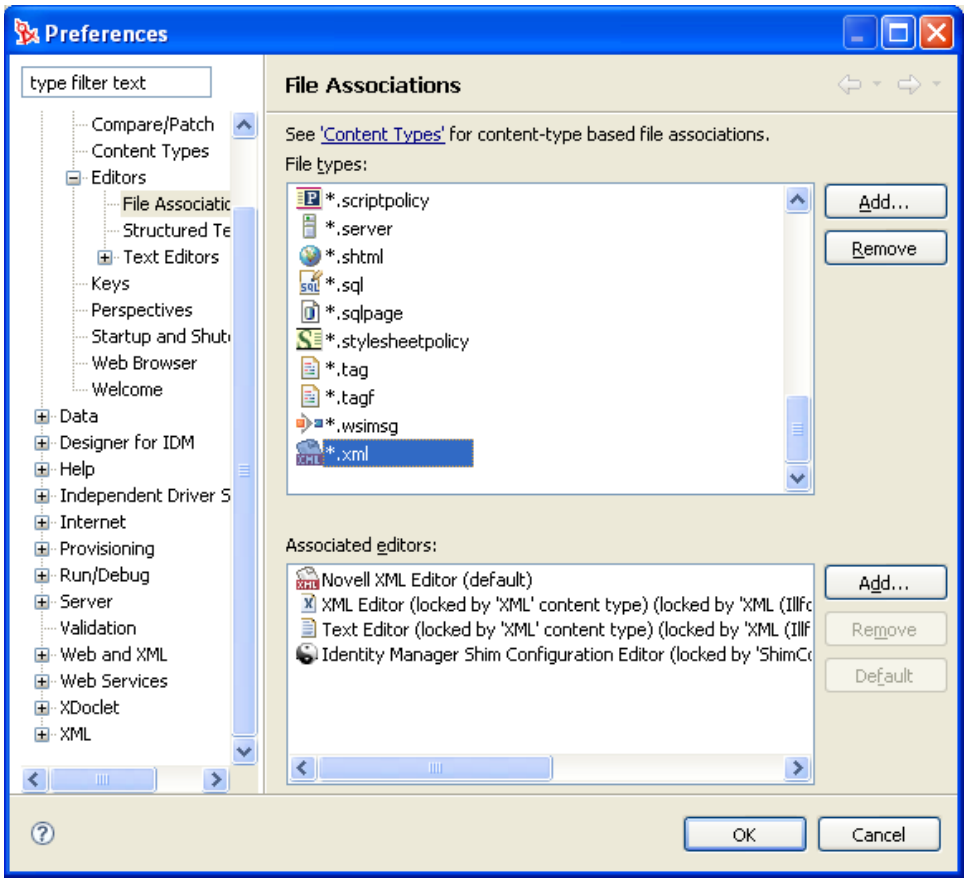
機能	説明
Attach DTD or XML Schema (DTD または XML スキーマを添付)	ポリシーの検証のために、DTD または XML スキーマファイルを添付します。
検証	XML コードを検証します。
初期設定	XML エディタの初期設定を指定します。

ソースビュー用に、別の XML エディタを選択するには

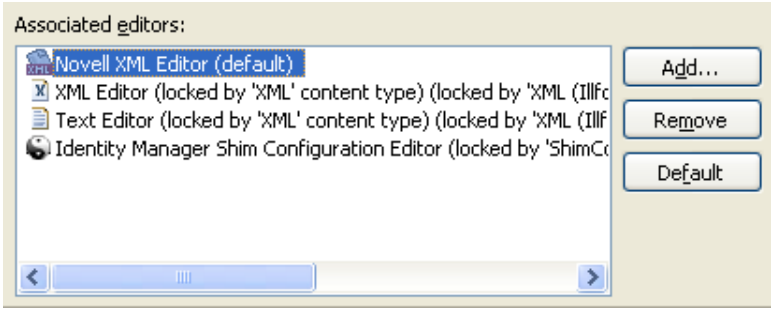
1 メインメニューの［Window (ウィンドウ)］>［初期設定］の順に選択します。

2 ［一般］>［Editors (エディタ)］>［File Associations (ファイルの関連付け)］の順に選択します。

3 ［ファイルタイプ］の下のリストから［*.xml］を選択します。



4 ［Associated editors (関連付けられているエディタ)］ペインで、エディタ (たとえば、［Novell XML Editor (Novell XML エディタ)］) を選択します。(適切なエディタがリストにない場合は、［追加］をクリックしてリストに追加します)。



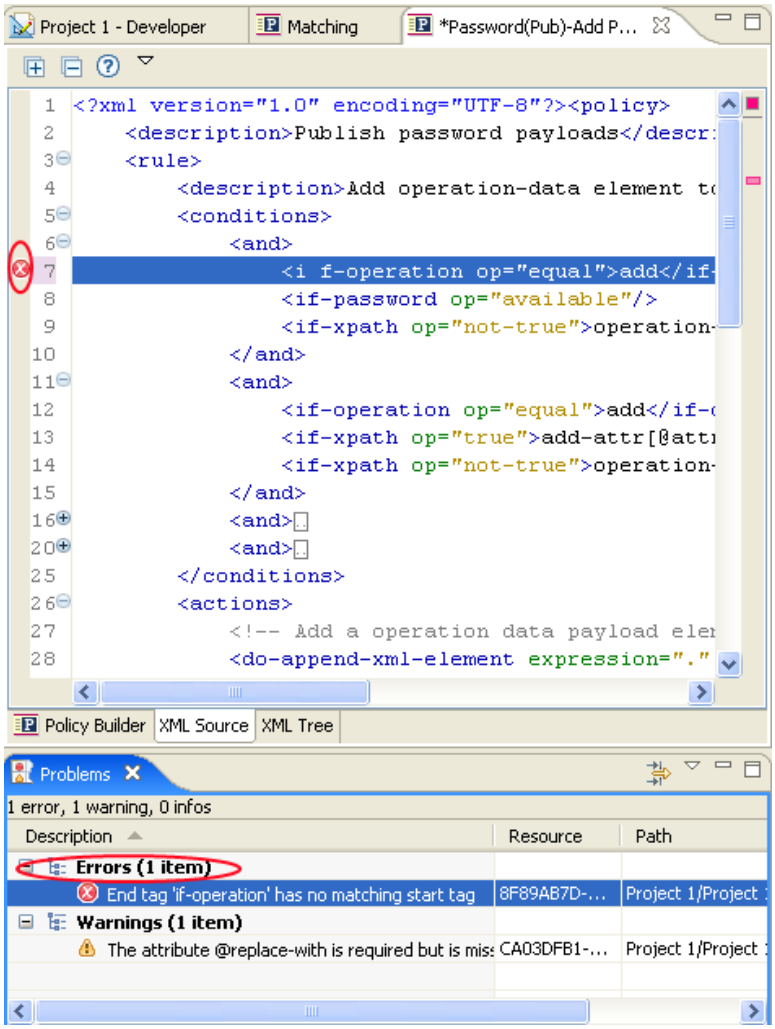
5 ［OK］をクリックします。

6 ポリシービルダをいったん閉じて再度開きます。

XML ソースの検証

XML エディタは、XML コードを検証します。右クリックし、[検証] を選択します。エラーがある場合は、その行に赤の「x」が表示されます。ウィンドウの下部の説明に、問題についての詳しい情報が示されます。

図 *2-13* *XML ソースの検証*



この例では、if-operation の終了タグに対応する開始タグがありません。

2.3 正規表現

正規表現とは、あるパターンに従ったテキスト文字列を照合するための式です。正規表現は、標準文字とメタ文字から構成されます。標準文字には、大文字と小文字、数字があります。メタ文字には特別な意味があります。次の表に、一般的なメタ文字とその意味を示します。

表 *2-6* 一般的な正規表現

メタ文字	説明
.	任意の 1 文字と一致します。
$	行の終わりを意味します。
^	行の先頭を意味します。
*	直前の文字が 0 個以上含まれることを意味します。
\	リテラルのエスケープ文字です。この文字を使用することで、検索対象にすべてのメタ文字を指定できます。たとえば、\$ と指定した場合、行の終わりではなく、$1000 が検索結果になります。
[ ]	角括弧で囲まれた文字のいずれかを意味します。
[0-9]	ハイフンの前後の文字範囲が対象になります。この例では、すべての数字を意味します。
[A-Za-z]	複数の範囲も同様に表します。この例では、すべての大文字と小文字が対象になります。

引数ビルダは、Java で定義されている正規表現を使用するように設計されています。Java Web サイト (http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html) で詳細情報を参照できます。

2.4 XPath 1.0 の式

条件、アクション、およびトークンの中には、引数で XPath 1.0 の式を使用するものがあります。XPath は、XSLT および XPointer とで共有される機能に対し、共通の構文とセマンティックを提供するために開発された言語です。主に XML ドキュメントのアドレス指定で使用されますが、文字列、数値およびブールなどのデータ操作を行う基本的な機能も備わっています。

XPath の仕様では、埋め込みアプリケーションが、複数のアプリケーションで定義された情報をコンテキストに提供する必要があります。DirXML スクリプト (11 ページのセクション 1.1.2「DirXML スクリプト」を参照) では、XPath は次のコンテキストで評価されます。

- コンテキストのノードが現在の操作。
- コンテキストの位置とサイズが 1。
- 使用可能な変数が次のように複数ある。
  - Identity Manager 内のスタイルシートに対するパラメータとして使用可能なもの (現在のところ、fromNDS、srcQueryProcessor、destQueryProcessor、srcCommandProcessor、destCommandProcessor および dnConverter)。
  - グローバル構成変数。
  - ローカルポリシーの変数。
  - 異なる変数ソース間で名前が衝突している場合は、優先順位はローカル変数、スタイルシートのパラメータ、グローバル変数の順になります。
- ポリシー要素上で宣言されたネームスペース。

- 使用可能な機能が次のように複数ある。
  - 組み込まれている XPath 1.0 のすべての機能。
  - NXSL で提供されている Java 拡張機能。

プリフィックスを Java クラスに関連付けるためのネームスペース宣言は、ポリシー要素で実行される必要があります。

W3 Web サイト (http://www.w3.org/TR/1999/REC-xpath-19991116) で詳細情報を参照できます。

2.5 条件

この節では、ポリシービルダインタフェースで使用できるすべての条件について、詳しく説明します。

2.5.1 「関連付け」条件

現在の操作または、現在のオブジェクトにある関連付けの値をテストします。

フィールド

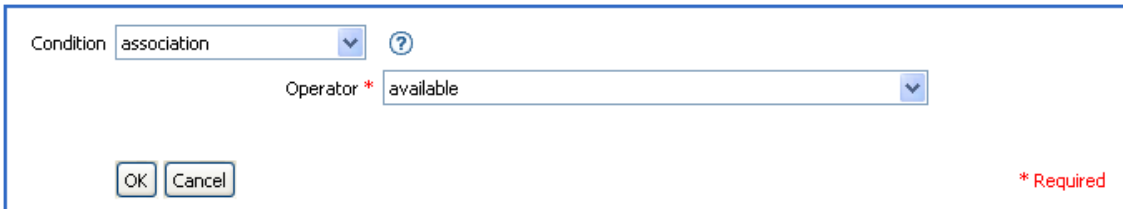
次の場合に演算子の条件に一致

演算子	次の場合に条件に一致
関連付けあり	現在のオブジェクトに確立された関連付けがある。
使用可能	現在の操作で指定された、空ではない関連付けの値がある。

演算子	次の場合に条件に一致
等しい	現在の操作で指定された関連付けの値が、「関連付け」条件の内容と完全に同じになる。
関連付けなし	現在のオブジェクトには確立された関連付けがない。
使用不可	現在のオブジェクトでは関連付けを使用できない。
等しくない	現在の操作で指定された関連付けの値が、「関連付け」条件の内容と異なる。

例

この例では、関連付けが使用可能かどうかを確認しています。この条件に一致する場合、定義されたアクションが実行されます。



2.5.2 「属性」条件

現在の操作または、ソースデータストアにある現在のオブジェクトの属性値をテストします。ソースデータストアまたは操作で条件が一致した場合にテストに適合するので、論理的には「操作属性」条件または「ソース属性」条件と考えることができます。

フィールド

名前

テストする属性の名前を指定します。

演算子

条件のテストタイプを選択します。

比較モード

比較モードを選択します。212 ページの 「比較モード」を参照してください。

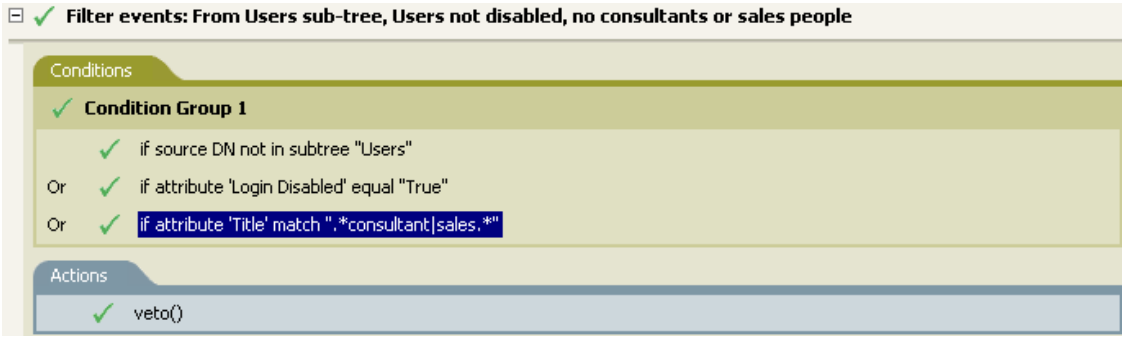
次の場合に演算子の条件に一致

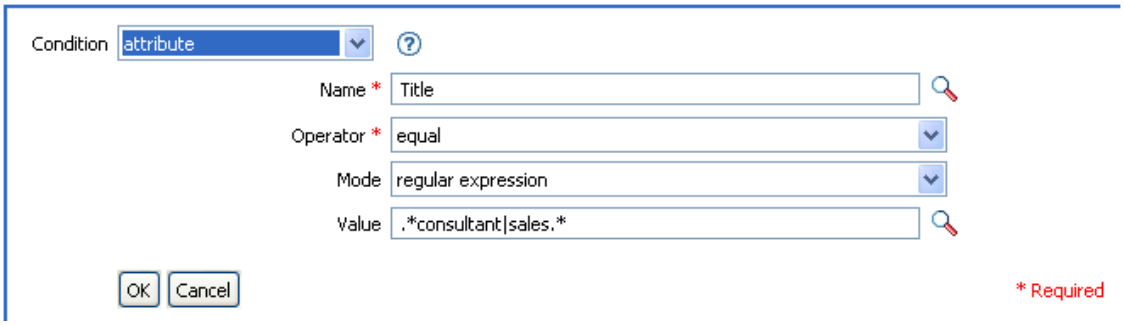
演算子	次の場合に条件に一致
使用可能	現在の操作またはソースデータストアに、指定した属性で使用できる値がある。
等しい	現在の操作またはソースデータストアに指定した属性に使用可能な値があり、指定された比較モードを使用して比較すると、指定した値と同じになる。
使用不可	「使用可能」の場合 False が返る。

演算子	次の場合に条件に一致
等しくない	「等しい」の場合 False が返る。

例

この例では、無効化されているかまたは一定の役職を持つユーザオブジェクトを抽出するために、条件として「属性」条件を使用しています。このポリシーは「Policy to Filter Events (イベントをフィルタ処理するためのポリシー)」であり、Novell のサポート Web サイトからダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。





この条件では、役職属性がコンサルタントまたは販売担当であるユーザが検索されます。

2.5.3 「クラス名」条件

現在の操作にあるオブジェクトクラス名をテストします。

フィールド

演算子

条件のテストタイプを選択します。

比較モード

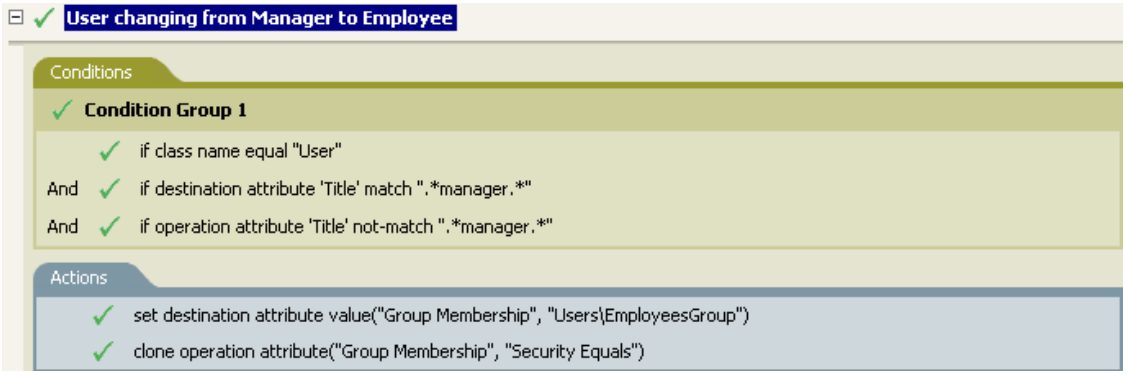
比較モードを選択します。212 ページの 「比較モード」を参照してください。

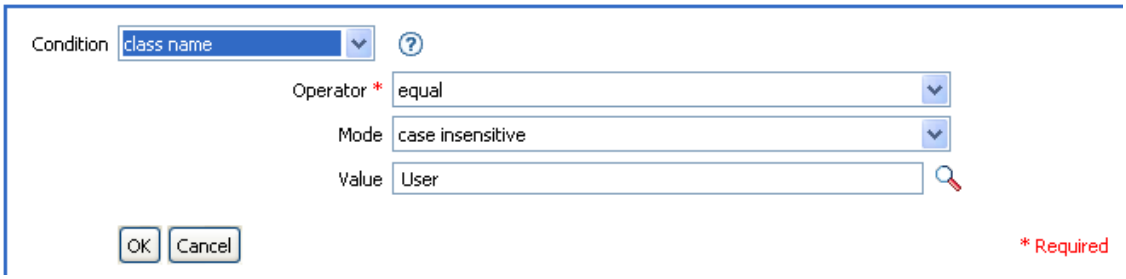
次の場合に演算子の条件に一致

演算子	次の場合に条件に一致
使用可能	現在の操作に使用可能なオブジェクトクラス名がある。
等しい	現在の操作に使用可能なオブジェクトクラス名があり、指定した比較モードを使用して比較すると、指定された値と同じになる。
使用不可	「使用可能」の場合 False が返る。
等しくない	「等しい」の場合 False が返る。

例

この例では、役職に基づいてユーザオブジェクトのグループメンバーシップを管理するため、条件として「クラス名条件」を使用しています。このポリシーは「Govern Groups for User Based on Title Attribute (役職属性に基づくユーザグループの管理)」で、Novell のサポート Web サイトからダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。





現在のオブジェクトのクラス名が「User」であるかどうかを確認します。

2.5.4 「ターゲット属性」条件

ターゲットデータストアにある現在のオブジェクトの属性値をテストします。

フィールド

名前

テストする属性の名前を指定します。

演算子

条件のテストタイプを選択します。

比較モード

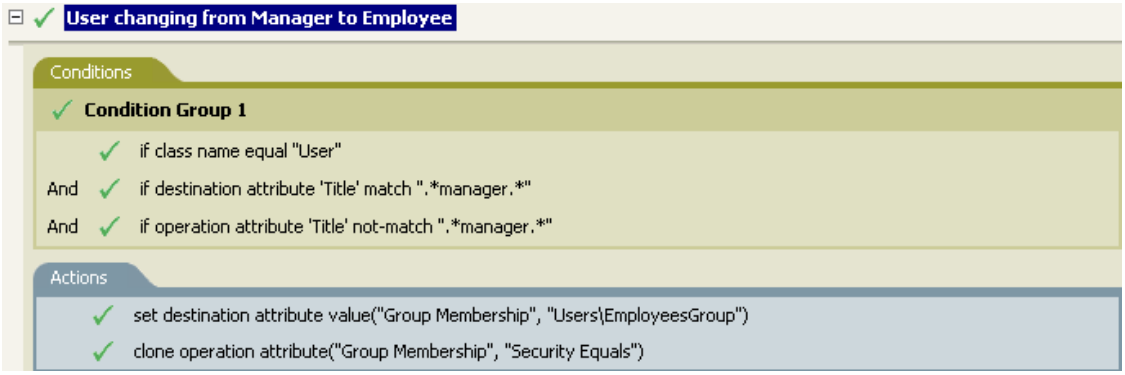
比較モードを選択します。212 ページの 「比較モード」を参照してください。

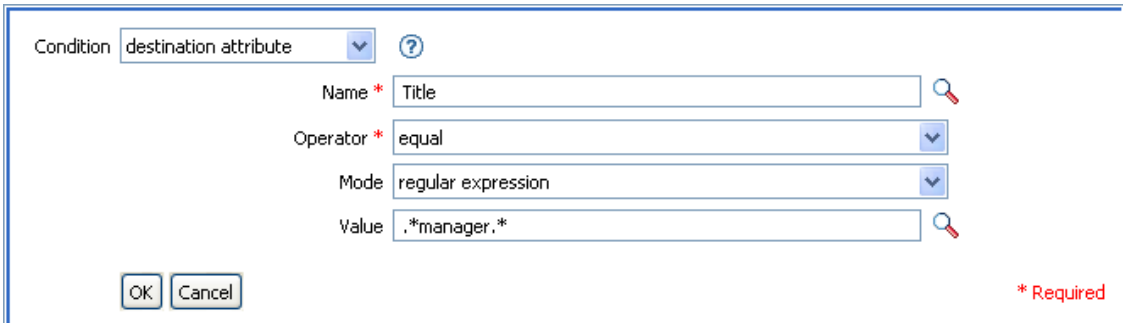
次の場合に演算子の条件に一致

演算子	次の場合に条件に一致
使用可能	ターゲットデータストアに、指定した属性で使用可能な値がある。
等しい	ターゲットデータストアに指定した属性で使用可能な値があり、指定された比較モードを使用して比較すると、指定された値と同じになる。
使用不可	「使用可能」の場合 False が返る。
等しくない	「等しい」の場合 False が返る。

例

この例では、役職に基づいてユーザオブジェクトのグループメンバーシップを管理するため、条件として「属性」条件を使用しています。これは「Govern Groups for User Based on Title Attribute (役職属性に基づくユーザグループの管理)」というポリシーで、Novell のサポート Web サイトからダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。





このポリシーでは、役職属性に「manager」が含まれているかどうかを確認します。

2.5.5 「ターゲット DN」条件

現在の操作のターゲット DN をテストします。

フィールド

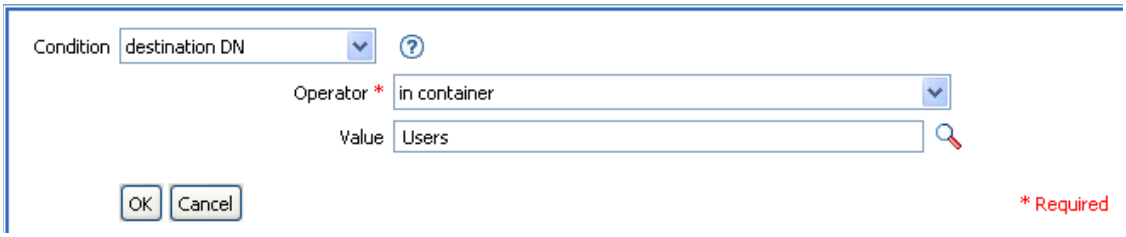
演算子

条件のテストタイプを選択します。

次の場合に演算子の条件に一致

演算子	次の場合に条件に一致
使用可能	使用可能なターゲット DN がある。
等しい	使用可能なターゲット DN があり、ターゲットデータストアの DN 形式に適したセマンティックを使用して比較すると、指定された値と同じになる。
コンテナ内にあり	使用可能なターゲット DN があり、ターゲットデータストアの DN 形式に適したセマンティックを使用して比較すると、値で指定されたコンテナ内のオブジェクトを示す。
サブツリー内にあり	使用可能なターゲット DN があり、ターゲットデータストアの DN 形式に適したセマンティックを使用して比較すると、値で指定されたサブツリー内のオブジェクトを示す。
使用不可	「使用可能」の場合 False が返されます。
等しくない	「等しい」の場合 False が返る。
コンテナ内になし	「コンテナ内にあり」の場合 False が返る。
サブツリー内になし	「サブツリー内にあり」の場合 False が返る。

例



2.5.6 「エンタイトルメント」条件

現在の操作またはアイデンティティボールトにある現在のオブジェクトのエンタイトルメントをテストします。

フィールド

名前

選択した条件をテストするエンタイトルメントの名前を指定します。

演算子

条件のテストタイプを選択します。

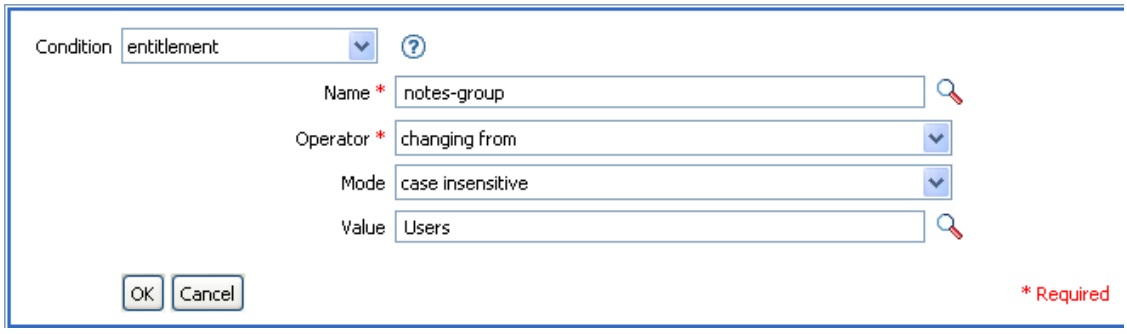
比較モード

比較モードを選択します。212 ページの 「比較モード」を参照してください。

次の場合に演算子の条件に一致

演算子	次の場合に条件に一致
使用可能	現在の操作またはアイデンティティボールトで、指定したエンタイトルメントを使用できる。
変更あり	現在の操作に、指定したエンタイトルメントの変更 (属性の変更または属性の追加) が含まれる。
削除指定の変更あり	現在の操作に、指定したエンタイトルメントの値を削除する変更 (値の削除) が含まれ、指定された比較モードを使用して比較すると、指定された値と同じ値がある。
追加指定の変更あり	現在の操作に、指定したエンタイトルメントに値を追加する変更 (値の追加または属性の追加) が含まれる。指定された比較モードを使用して比較すると、指定された値と同じ値があります。
等しい	ターゲットデータストアに指定した属性で使用可能な値があり、指定された比較モードを使用して比較すると、指定された値と同じになる。
使用不可	「使用可能」の場合 False が返る。
変更なし	「変更あり」の場合 False が返る。
削除指定の変更なし	「削除指定の変更あり」の場合 False が返る。
追加指定の変更なし	「追加指定の変更あり」の場合 False が返る。
等しくない	「等しい」の場合 False が返る。

例



2.5.7 「グローバル構成値」条件

グローバル構成変数をテストします。

フィールド

名前

選択した条件をテストするグローバル変数の名前を指定します。

演算子

条件のテストタイプを選択します。

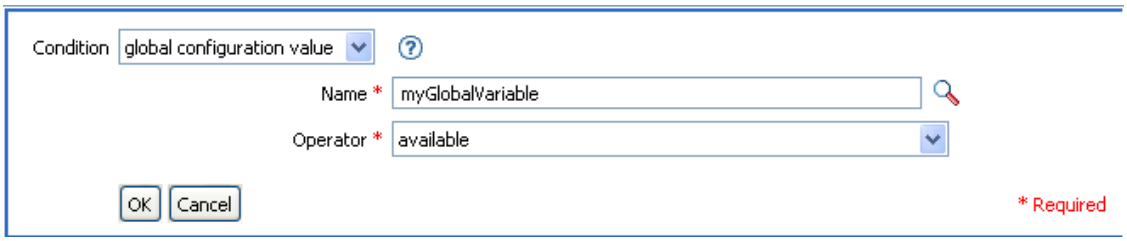
比較モード

比較モードを選択します。212 ページの 「比較モード」を参照してください。

次の場合に演算子の条件に一致

演算子	次の場合に条件に一致
使用可能	指定した名前のグローバル設定変数がある。
等しい	指定した名前のグローバル設定変数があり、その値が、指定された比較モードを使用して比較すると、指定された値と同じになる。
使用不可	「使用可能」の場合 False が返る。
等しくない	「等しい」の場合 False が返る。

例



2.5.8 「ローカル変数」条件

ローカル変数をテストします。

フィールド

名前

選択した条件をテストするローカル変数の名前を指定します。

演算子

条件のテストタイプを選択します。

比較モード

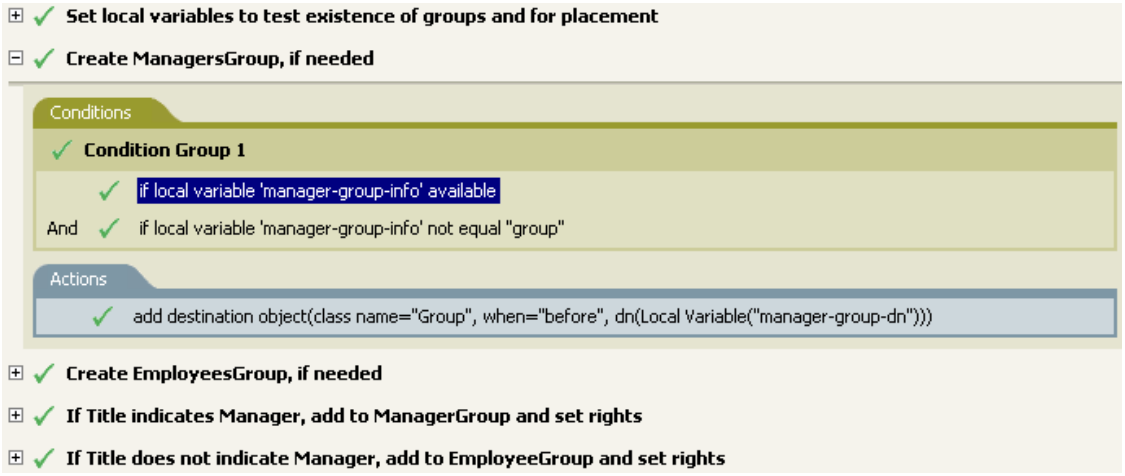
比較モードを選択します。212 ページの 「比較モード」を参照してください。

次の場合に演算子の条件に一致

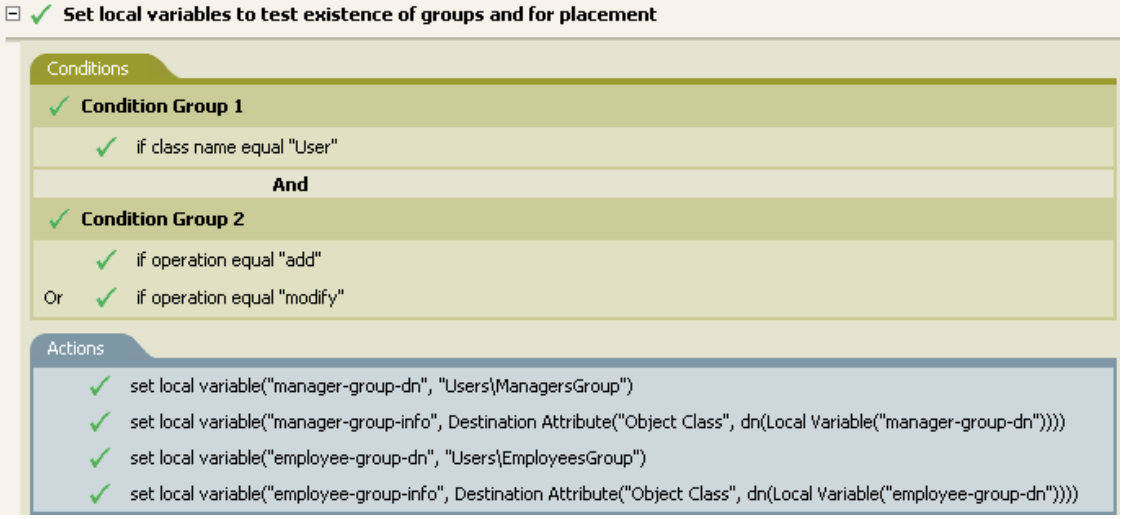
演算子	次の場合に条件に一致
使用可能	ポリシー内にある以前のルールのアクションですでに定義されている、指定した名前のローカル変数がある。
等しい	指定した名前のローカル変数があり、その値が、指定された比較モードを使用して比較すると、指定された値と同じになる。
使用不可	「使用可能」の場合 False が返る。
等しくない	「等しい」の場合 False が返る。

例

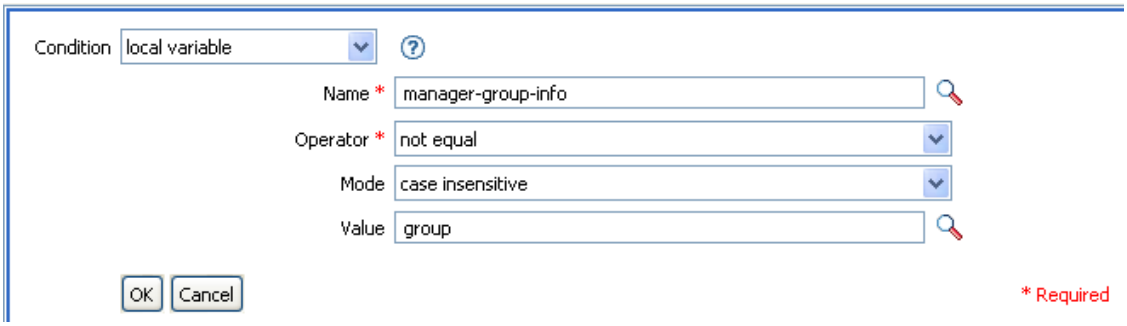
この例では、役職に基づいて、ユーザオブジェクトを適切なグループ (従業員またはマネージャ) に追加します。必要に応じてグループの作成も実行し、そのグループに同等セキュリティを設定します。このポリシーは「Govern Groups for User Based on Title Attribute (役職属性に基づくユーザグループの管理)」で、Novell のサポート Web サイトからダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。



このポリシーには、互いに依存する 5 つのルールが含まれています。



「ローカル変数」条件の条件を動作させるため、最初のルールで 4 つのローカル変数が設定され、グループとそのグループの配置場所がテストされます。



ルールが検索する条件では、ローカル変数 manager-group-info が使用可能かどうか、およびこの変数がグループと等しくないかどうかが確認されます。これらの条件が一致すると、グループのターゲットオブジェクトが追加されます。

2.5.9 「名前付きパスワード」条件

現在の操作にあるパスワードを、指定された名前でテストします。

フィールド

名前

選択した条件をテストする名前付きパスワードの名前を指定します。

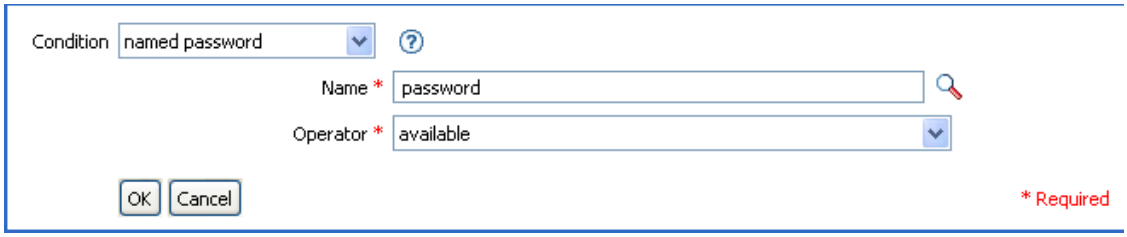
演算子

条件のテストタイプを選択します。

次の場合に演算子の条件に一致

演算子	次の場合に条件に一致
使用可能	指定した名前でパスワードを使用できる。
使用不可	「使用可能」の場合 False が返る。

例



2.5.10 「操作」条件

現在の操作の名前をテストします。

フィールド

演算子

条件のテストタイプを選択します。

次の場合に演算子の条件に一致

演算子	次の場合に条件に一致
等しい	現在の操作の名前が、「操作」条件の内容と完全に同じである。
等しくない	「等しい」の場合 False が返る。

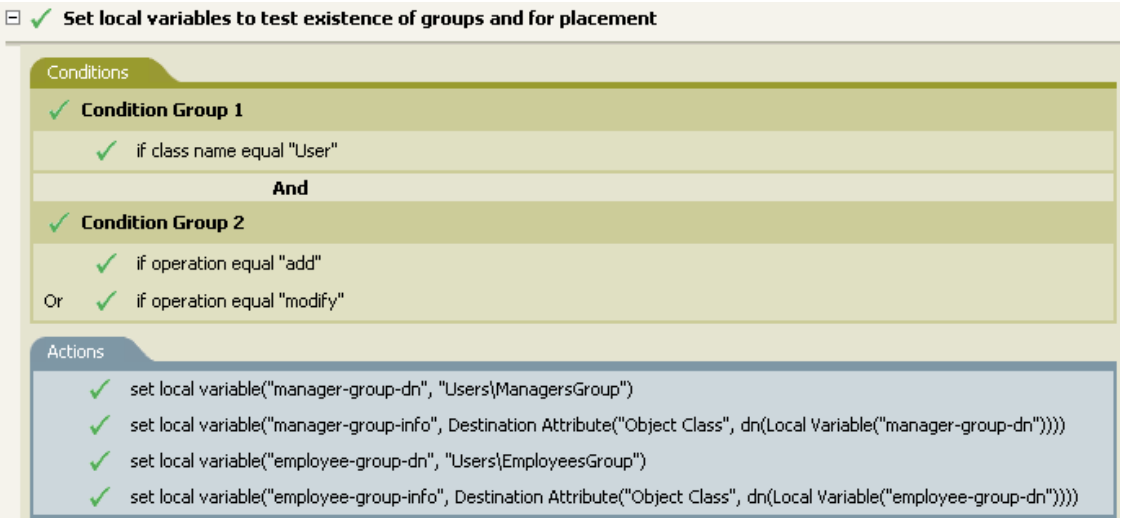
値

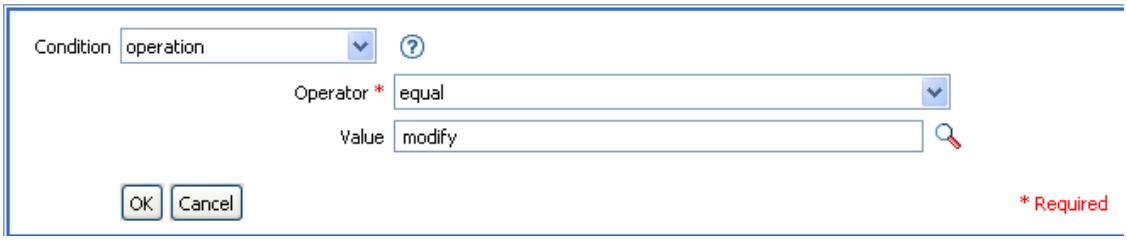
値は、メタディレクトリエンジンがこの条件で検索する操作です。

- 追加
- 関連付けの追加
- オブジェクトパスワードの確認
- 削除
- 名前付きパスワードの取得
- 変更
- 関連付けの変更
- パスワード変更
- 移動
- パラメータの開始
- インスタンス

例

この例では、役職に基づいて、ユーザオブジェクトを適切なグループ (従業員またはマネージャ) に追加します。必要に応じてグループも作成し、そのグループに同等セキュリティを設定します。これは「Govern Groups for User Based on Title Attribute (役職属性に基づくユーザグループの管理)」という名前のポリシーで、Novell のサポート Web サイトからダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。





この条件では、追加または変更の操作が発生したかどうかを確認しています。これらのいずれかが発生した場合、ローカル変数が設定されます。

2.5.11 「操作属性」条件

現在の操作の属性値をテストします。

フィールド

名前

テストする属性の名前を指定します。

演算子

条件のテストタイプを選択します。

比較モード

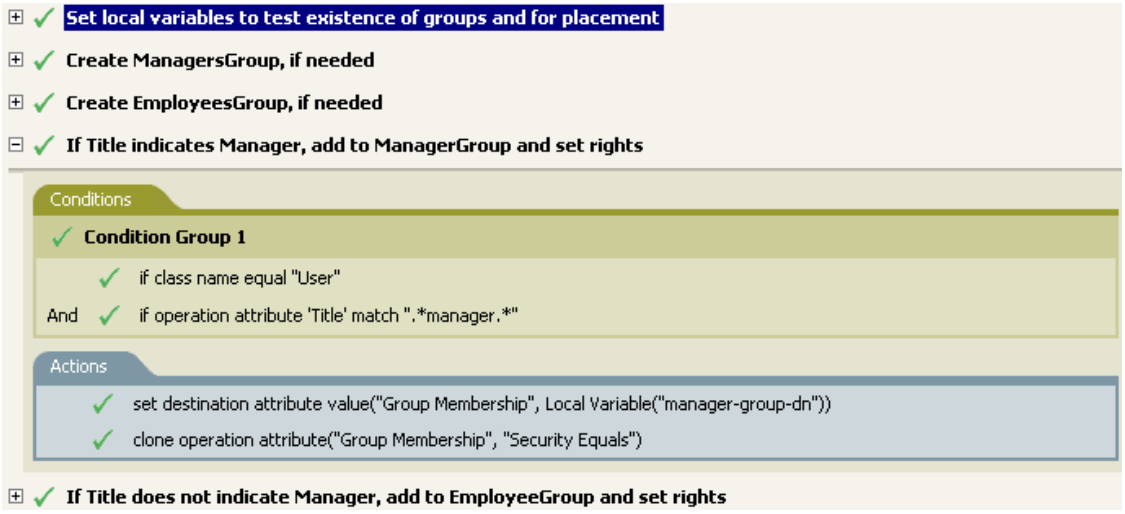
比較モードを選択します。212 ページの 「比較モード」を参照してください。

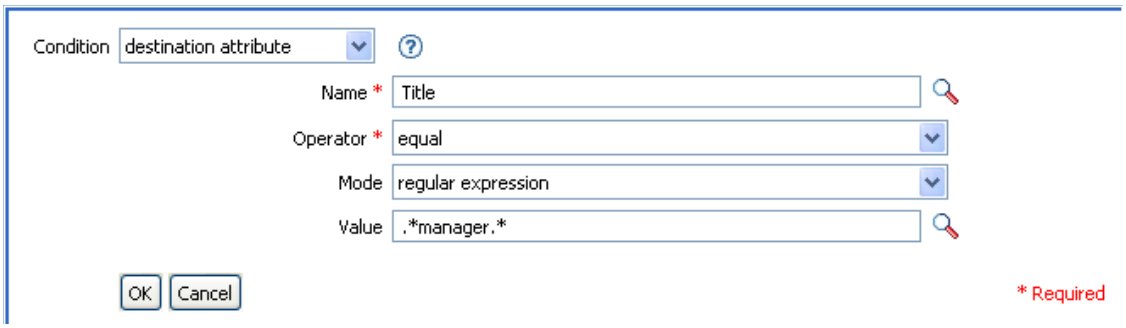
次の場合に演算子の条件に一致

演算子	次の場合に条件に一致
使用可能	現在の操作 (属性の追加、値の追加、属性) に、指定した属性で使用できる値がある。
変更あり	現在の操作に、指定した属性の変更 (属性の変更または属性の追加) がある。
削除指定の変更あり	現在の操作に、指定した属性の値を削除する変更 (値の削除) がある。指定された比較モードを使用して比較すると、指定された値と同じになります。
追加指定の変更あり	現在の操作に、指定した属性に値を追加する変更 (値の追加または属性の追加) が含まれる。指定された比較モードを使用して比較すると、指定された値と同じになります。
等しい	現在の操作 (値の削除以外) に、指定した属性で使用できる値がある。指定された比較モードを使用して比較すると、指定された値と同じになります。
使用不可	「使用可能」の場合 False が返る。
変更なし	「変更あり」の場合 False が返る。
削除指定の変更なし	「削除指定の変更あり」の場合 False が返る。
追加指定の変更なし	「追加指定の変更あり」の場合 False が返る。
等しくない	「等しい」の場合 False が返る。

例

この例では、役職に基づいて、ユーザオブジェクトを適切なグループ (従業員またはマネージャ) に追加します。必要に応じてグループも作成し、そのグループに同等セキュリティを設定します。これは「Govern Groups for User Based on Title Attribute (役職属性に基づくユーザグループの管理)」という名前のポリシーで、Novell のサポート Web サイトからダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。





この条件では、役職属性が正規表現「.*manager.*」に等しいかどうかを確認しています。つまり、manager の前に 0 個以上の文字を持ち、manager の後に 1 文字を持つ役職を検索しています。ユーザオブジェクトの役職が sales managers であった場合、一致として検出されます。

2.5.12 「操作プロパティ」条件

現在の操作の操作プロパティをテストします。

フィールド

名前

選択した条件をテストする操作プロパティの名前を指定します。

演算子

条件のテストタイプを選択します。

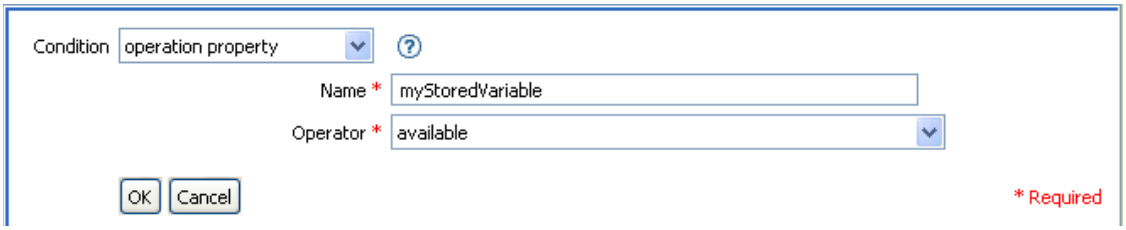
比較モード

比較モードを選択します。212 ページの 「比較モード」を参照してください。

次の場合に演算子の条件に一致

演算子	次の場合に条件に一致
使用可能	現在の操作に、指定した名前の操作プロパティがある。
等しい	指定した名前の操作プロパティが現在の操作にあり、その値が、指定された比較モードを使用して比較すると、指定された内容と同じになる。
使用不可	「使用可能」の場合 False が返る。
等しくない	「等しい」の場合 False が返る。

例



2.5.13 「パスワード」条件

現在の操作のパスワードをテストします。

フィールド

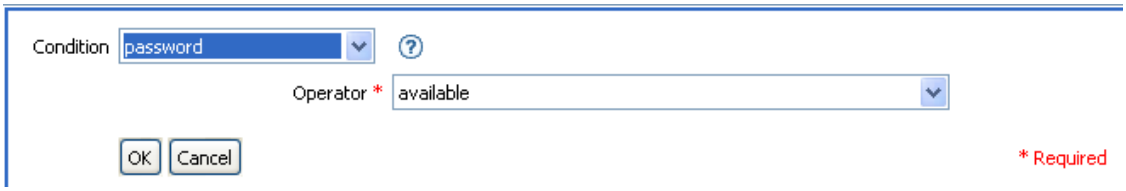
演算子

条件のテストタイプを選択します。

次の場合に演算子の条件に一致

演算子	次の場合に条件に一致
使用可能	現在の操作に、使用可能なパスワードがある。
使用不可	「使用可能」の場合 False が返る。

例



2.5.14 「ソース属性」条件

ソースデータストアにある現在のオブジェクトの属性値をテストします。

フィールド

名前

選択した条件をテストするソース属性の名前を指定します。

演算子

条件のテストタイプを選択します。

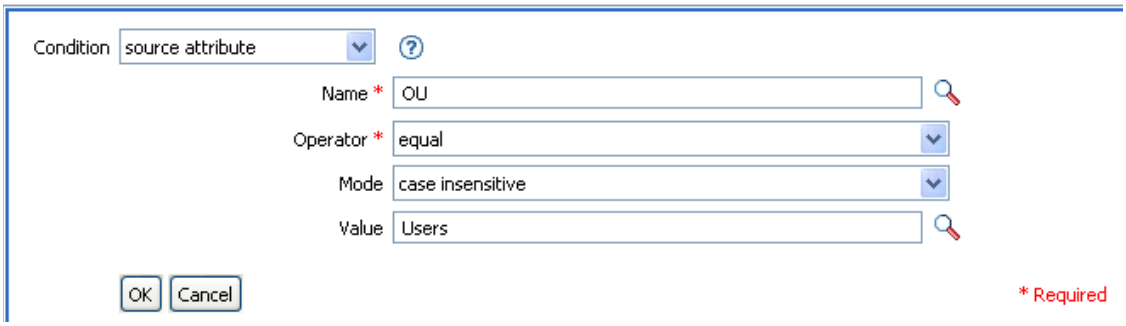
比較モード

比較モードを選択します。212 ページの 「比較モード」を参照してください。

次の場合に演算子の条件に一致

演算子	次の場合に条件に一致
使用可能	ソースデータストアに、指定した属性で使用可能な値がある。
等しい	ソースデータストアに、指定した属性で使用可能な値がある。指定された比較モードを使用して比較すると、指定された値と同じになります。
使用不可	「使用可能」の場合 False が返る。
等しくない	「等しい」の場合 False が返る。

例



2.5.15 「ソース DN」条件

現在の操作のソース DN をテストします。

フィールド

演算子

条件のテストタイプを選択します。

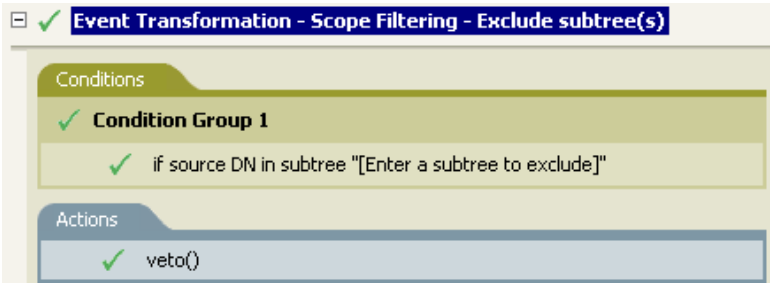
次の場合に演算子の条件に一致

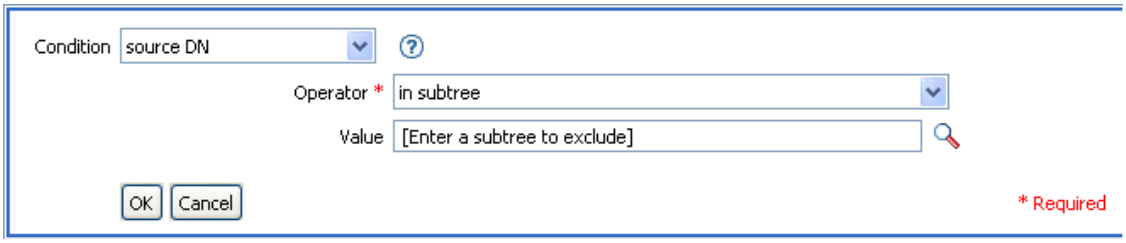
演算子	次の場合に条件に一致
使用可能	
	DN が使用可能である。
等しい	使用可能なソース DN があり、コンテナ内の指定された値と一致する。 使用可能なソース DN があり、指定された値で識別されるコンテナ内のオブジェクトを示す。
サブツリー内にあり	使用可能なソース DN があり、指定された値で識別されるサブツリー内のオブジェクトを示す。
使用不可	「使用可能」の場合 False が返る。
等しくない	「等しい」の場合 False が返されます。
コンテナ内になし	「コンテナ内にあり」の場合 False が返る。
サブツリー内になし	「サブツリー内にあり」の場合 False が返る。

例

この例では、ユーザオブジェクトがソース DN にあるかどうかを確認する条件として、「ソース DN」条件を使用しています。このルールは、Identity Manager 3.0 に付属している

事前定義されたルールです。詳細については、86 ページの 「イベント変換 - スコープフィルタリング - サブツリーの除外」を参照してください。





この例では、ソース DN がユーザコンテナにあるかどうかを確認しています。オブジェクトがこのコンテナ内にある場合は、拒否されます。

2.5.16 「XPath 式」条件

XPath 1.0 の式の評価結果をテストします。

フィールド

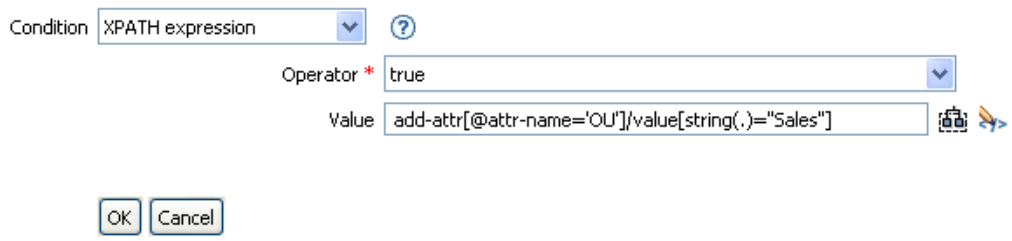
演算子

条件のテストタイプを選択します。

次の場合に演算子の条件に一致

演算子	次の場合に条件に一致
true	XPath 式が True に評価される。
false	「True」の場合 False が返る。

例



2.6 アクション

この節では、ポリシービルダインタフェースで使用できるすべてのアクションについて、詳しく説明します。

2.6.1 関連付けの追加

指定した関連付けと共に、関連付けの追加コマンドをアイデンティティボールトに送信します。

フィールド

モード

このアクションを現在の操作に追加するか、またはアイデンティティボールトへ直接書き込むかを選択します。

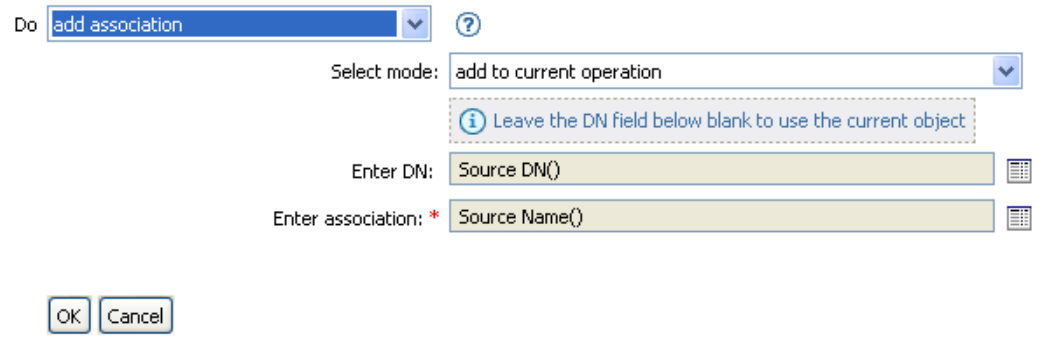
DN

ターゲットオブジェクトの DN を指定するか、または空白のままにして現在のオブジェクトを使用します。

関連付け

追加する関連付けの値を指定します。

例



2.6.2 ターゲット属性値の追加

ターゲットデータストア内のオブジェクトの属性に値を追加します。

フィールド

属性名

属性の名前を指定します。

クラス名

(オプション) ターゲットオブジェクトのクラス名を指定します。現在のオブジェクトのクラス名を使用するには、空白のままにします。

モード

このアクションを現在の操作の前と後のどちらに追加するか、または目的のデータストアへ直接書き込むかを選択します。

オブジェクト

ターゲットオブジェクトを選択します。このオブジェクトは現在のオブジェクトにすることも、DN または関連付けによって指定することもできます。

値のタイプ

追加する属性値の構文を選択します。

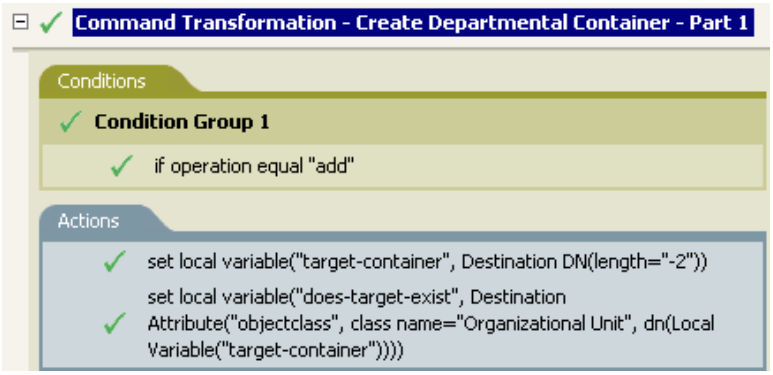
値

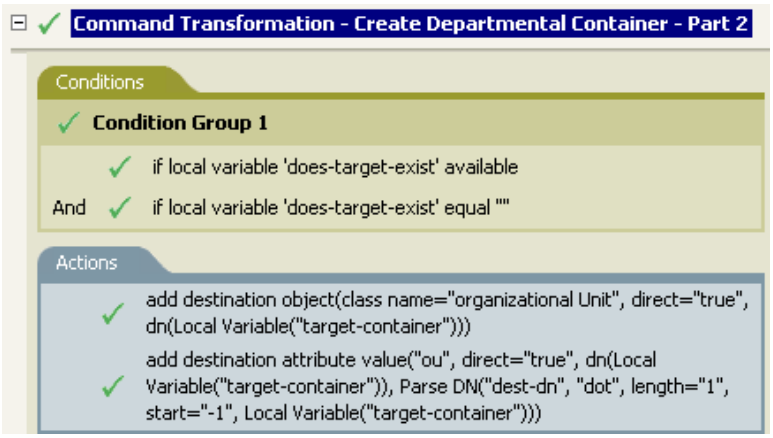
追加する属性値を指定します。

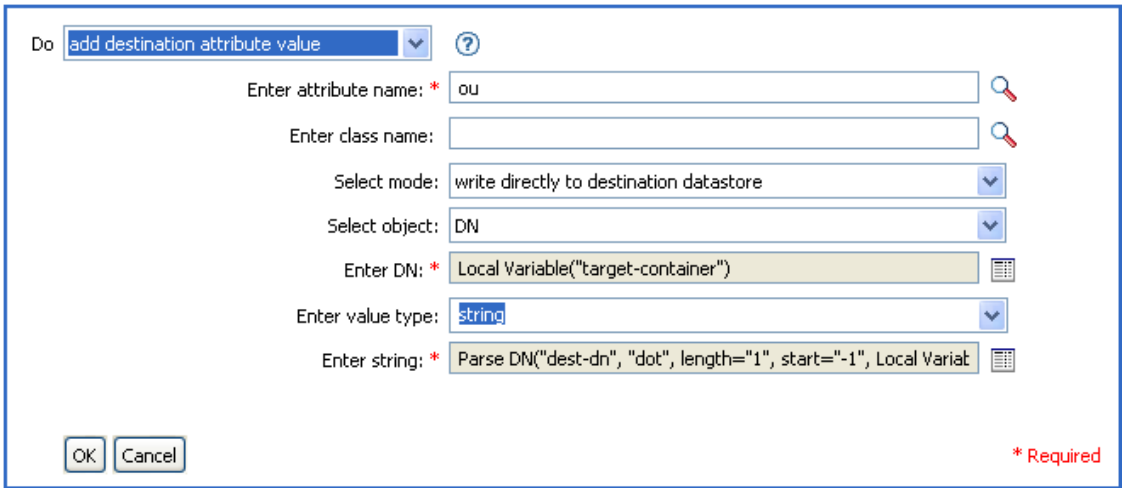
例

この例では、ターゲット属性値を OU 属性に追加します。作成されたローカル変数から値を生成します。このルールは、Identity Manager 3.0 に付属している事前定義されたルール

です。詳細については、74 ページの 「コマンド変換 - 部署別コンテナの作成 - パート 1 とパート 2」を参照してください。







2.6.3 ターゲットオブジェクトの追加

ターゲットデータストア内に新しいオブジェクトを作成します。

フィールド

クラス名

作成するオブジェクトのクラス名を指定します。

モード

このアクションを現在の操作の前と後のどちらに追加するか、または目的のデータストアへ直接書き込むかを選択します。

DN

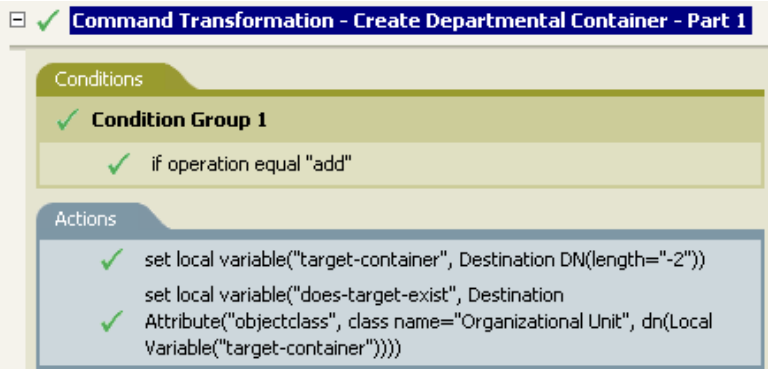
作成するオブジェクトの DN を指定します。

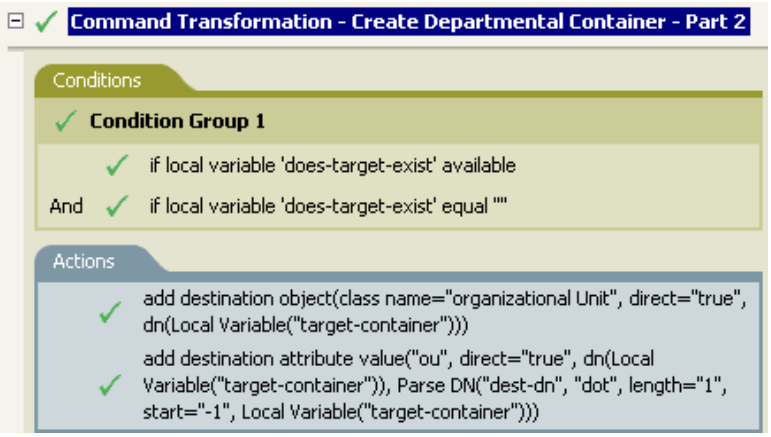
備考

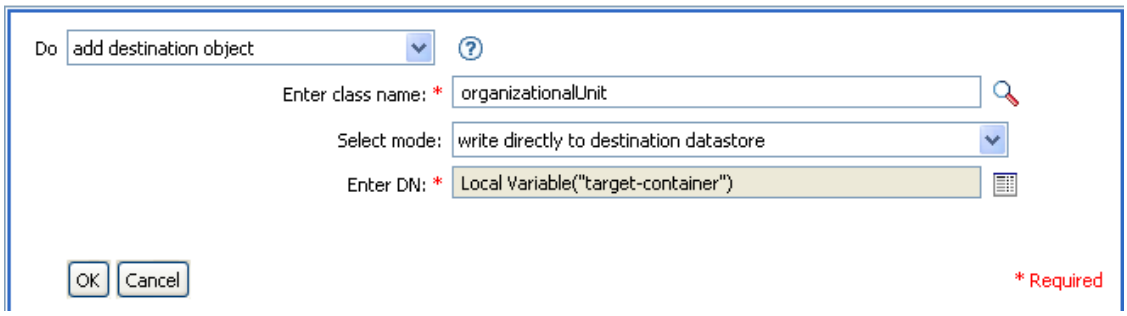
オブジェクト作成の一部として追加される任意の属性値は、次の 145 ページのセクション 2.6.2「ターゲット属性値の追加」アクションで同じ DN を使って追加する必要があります。

例

この例では、必要な部署別コンテナを作成します。このルールは、Identity Manager 3.0 に付属している事前定義されたルールです。詳細については、74 ページの 「コマンド変換 - 部署別コンテナの作成 - パート 1 とパート 2」を参照してください。







部門オブジェクトが作成されます。OU 属性の値は、このアクションの後に発生するターゲット属性値のアクションから作成されます。

2.6.4 ソース属性値の追加

ソースデータストア内のオブジェクトの属性に値を追加します。

フィールド

属性名

属性の名前を指定します。

クラス名

(オプション) ターゲットオブジェクトのクラス名を指定します。現在のオブジェクトのクラス名を使用するには、空白のままにします。

オブジェクト

ターゲットオブジェクトを選択します。このオブジェクトは現在のオブジェクトにすることも、DN または関連付けによって指定することもできます。

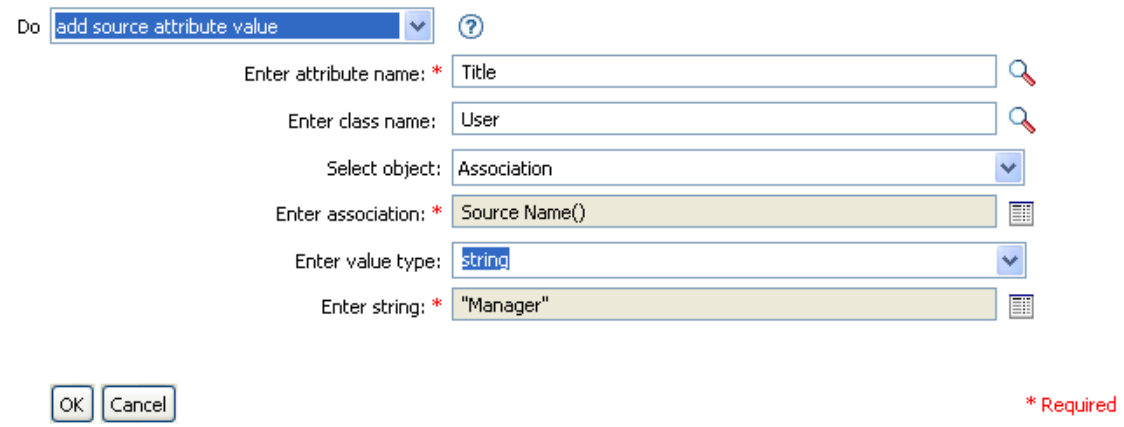
値のタイプ

追加する属性値の構文を選択します。

値

追加する属性値を指定します。

例



2.6.5 ソースオブジェクトの追加

ソースデータストア内に作成される、指定されたタイプのオブジェクトを作成します。オブジェクト作成の一部として追加される任意の属性値は、次の (148 ページ) ソース属性値の追加アクションで同じ DN を使って追加する必要があります。

フィールド

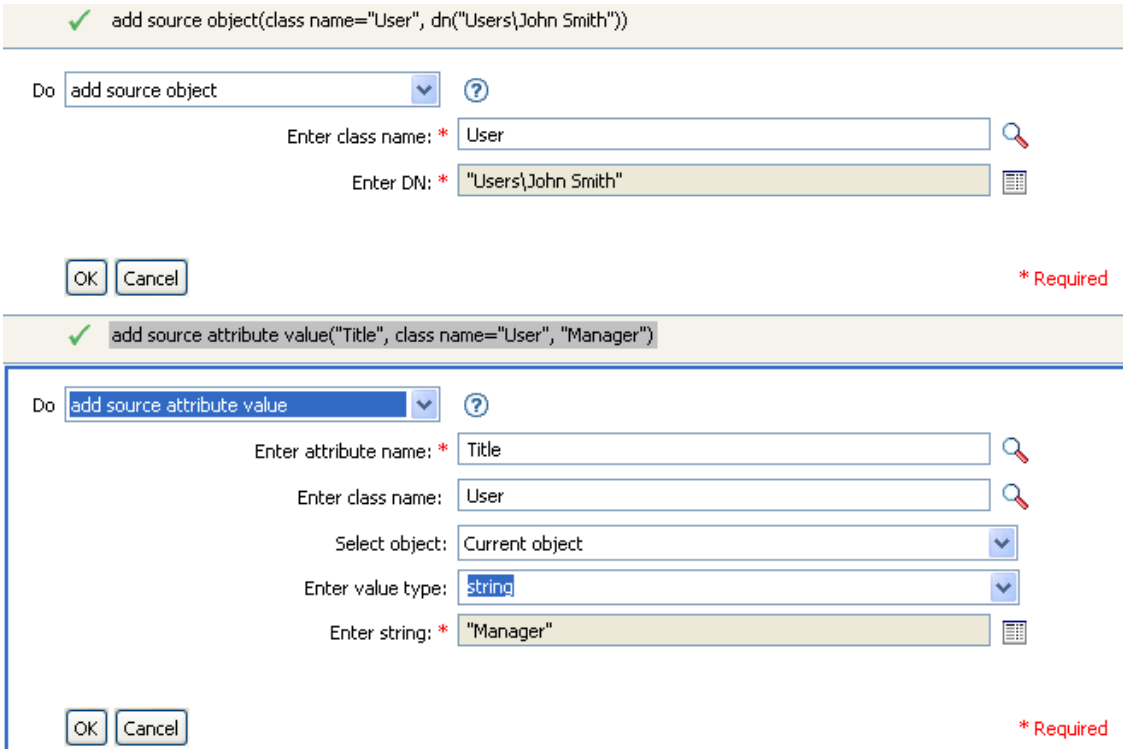
クラス名

追加するオブジェクトのクラス名を指定します。

DN

追加するオブジェクトの DN を指定します。

例



2.6.6 XML 要素の追加

XPath 式で選択された要素のセットに要素を追加します。

フィールド

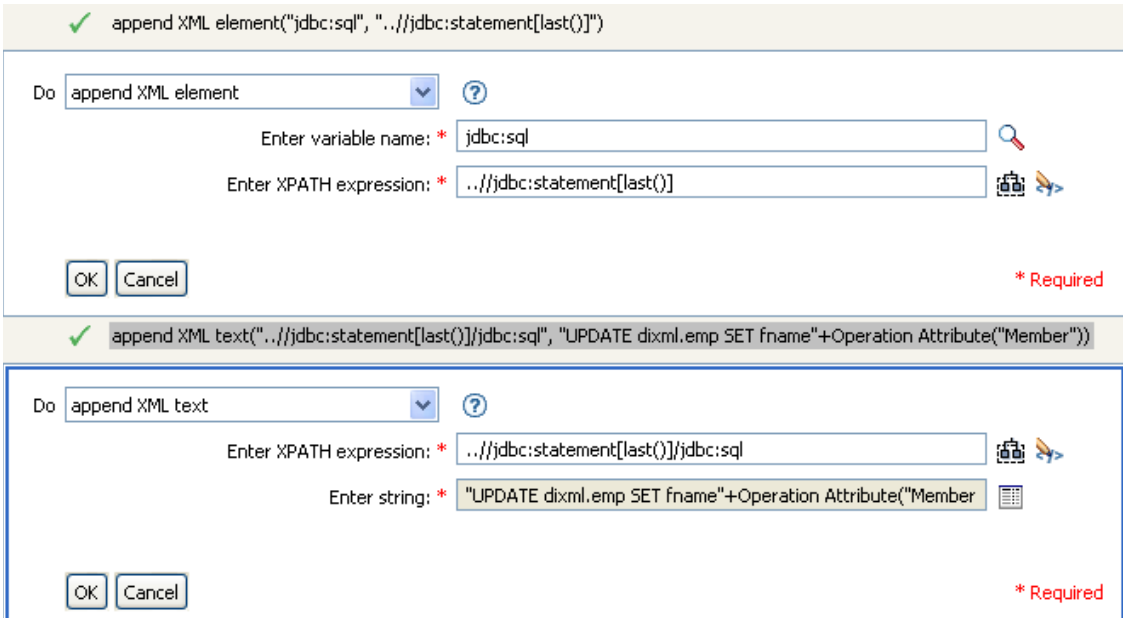
変数名

XML 要素のタグ名を指定します。この名前には、前にこのポリシーで定義されているネームスペースプリフィックスを含めることができます。

XPath 式

新しい要素の追加先になる要素を含むノードセットを返す XPath 1.0 の式を指定します。

例



2.6.7 XML テキストの追加

XPath 式で選択された要素のセットにテキストを追加します。

フィールド

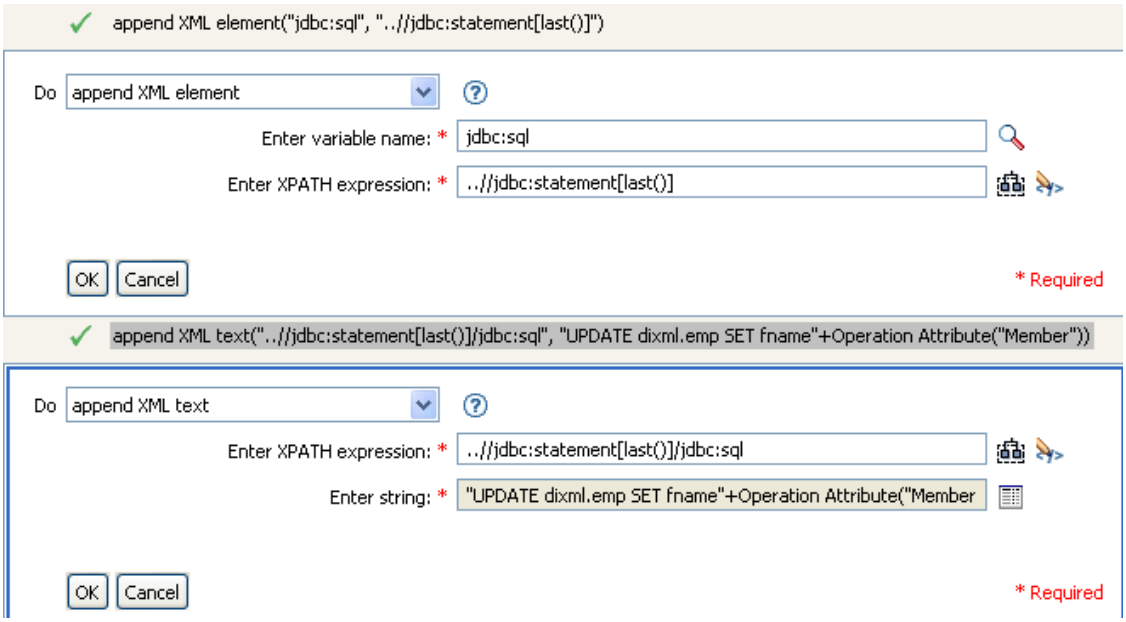
XPath 式

テキストの追加先になる要素が含まれるノードセットを返す XPATH 1.0 の式。

文字列

追加するテキストを指定します。

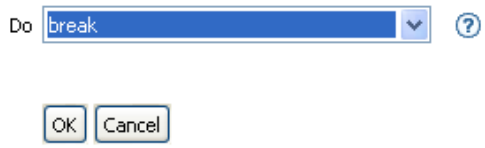
例



2.6.8 中断

現在のポリシーによる現在の操作の処理を終了します。

例



2.6.9 ターゲット属性値のクリア

ターゲットデータストア内の 1 つのオブジェクトから、すべての属性値を削除します。

フィールド

属性名

属性の名前を指定します。

クラス名

(オプション) ターゲットオブジェクトのクラス名を指定します。現在のオブジェクトのクラス名を使用するには、空白のままにします。

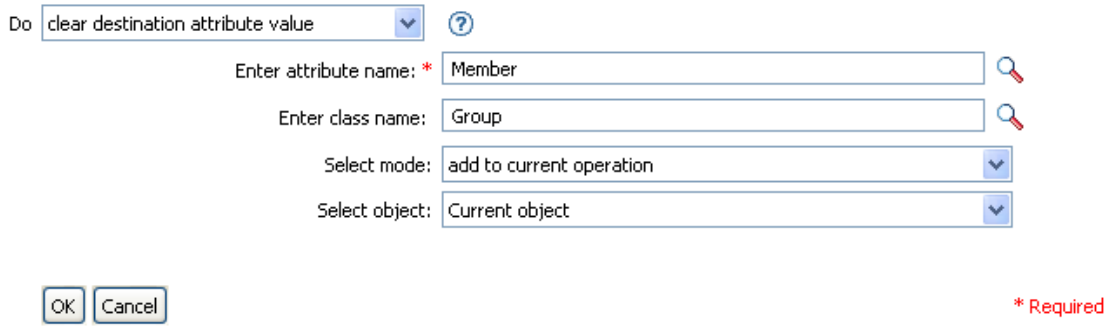
モード

このアクションを現在の操作の前と後のどちらに追加するか、または目的のデータストアへ直接書き込むかを選択します。

オブジェクト

ターゲットオブジェクトを選択します。このオブジェクトは現在のオブジェクトにすることも、DN または関連付けによって指定することもできます。

例



2.6.10 操作プロパティのクリア

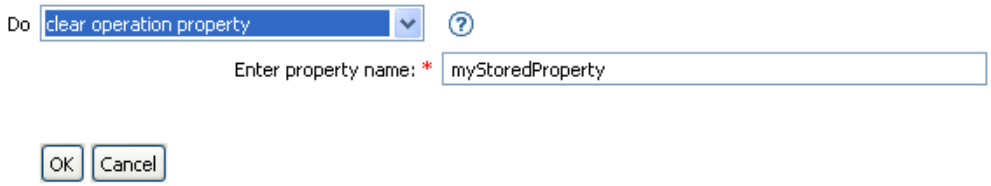
操作プロパティの現在の操作をクリアします。

フィールド

プロパティ名

クリアする操作プロパティの名前を指定します。

例



2.6.11 ソース属性値のクリア

ソースデータストア内の 1 つのオブジェクトから、すべての属性値を削除します。

フィールド

属性名

属性の名前を指定します。

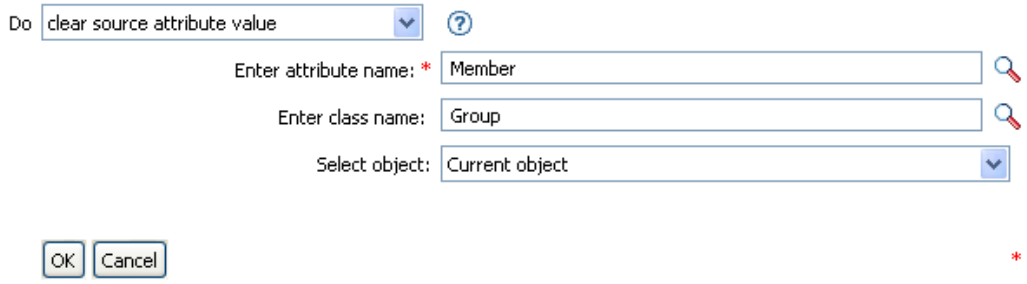
クラス名

(オプション) ターゲットオブジェクトのクラス名を指定します。現在のオブジェクトのクラス名を使用するには、空白のままにします。

オブジェクト

ターゲットオブジェクトを選択します。このオブジェクトは現在のオブジェクトにすることも、DN または関連付けによって指定することもできます。

例



2.6.12 SSO 資格情報のクリア

シングルサインオンの資格情報をクリアし、オブジェクトのプロビジョニングを解除できるようにします。このアクションは、資格情報のプロビジョニングポリシーの一部です。詳細については、333 ページの第 4 章「Novell 資格情報プロビジョニングポリシー」を参照してください。

フィールド

資格情報ストアオブジェクトの **DN**

リポジトリオブジェクトの DN を指定します。

ターゲットユーザの **DN**

ターゲットユーザの DN を指定します。

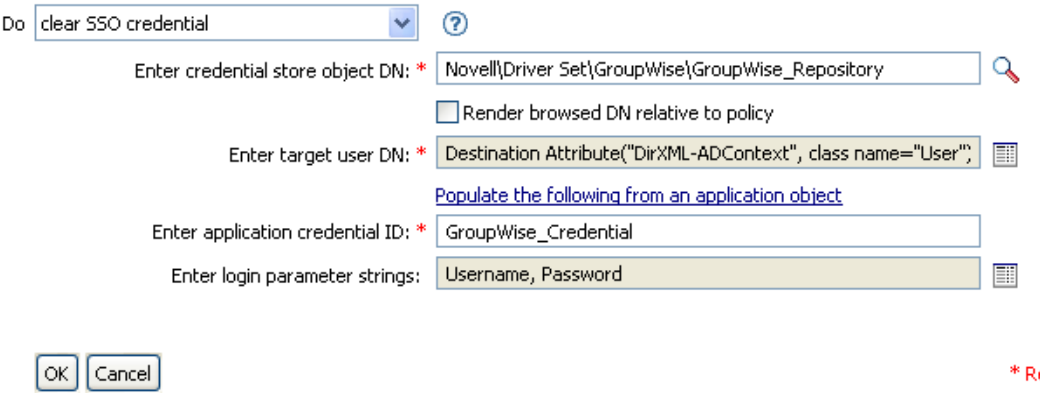
アプリケーションのアクティベーションキー **ID**

アプリケーションオブジェクト内に格納されるアプリケーションの資格情報を指定します。

ログインパラメータの文字列

アプリケーションのログインパラメータを指定します。ログインパラメータとは、アプリケーションオブジェクト内に格納されている認証キーです。

例



2.6.13 XPath 式によるクローン

XPath 式で選択された XML ノードのセットの詳細コピーを、他の XPath 式で選択された要素のセットに追加します。

フィールド

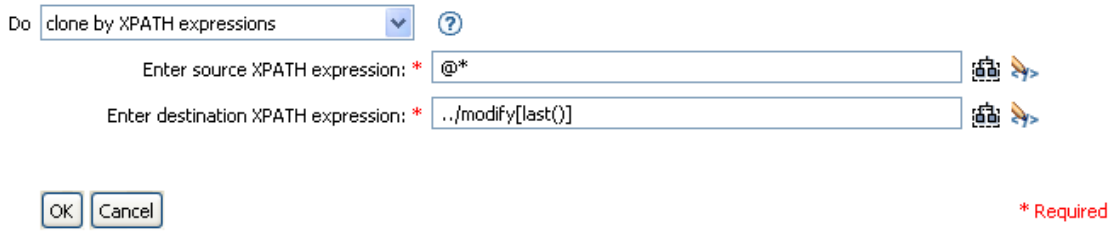
ソース **XPath** 式

コピーされるノードを含むノードセットを返す XPath 1.0 の式を指定します。

ターゲット **XPath** 式

コピーされたノードの追加先になる要素を含むノードセットを返す XPath 1.0 の式を指定します。

例



2.6.14 操作属性のクローン

現在の操作で属性に行った内容を、現在の操作内の別の属性にコピーします。

フィールド

ソース名

コピー元の属性の名前を指定します。

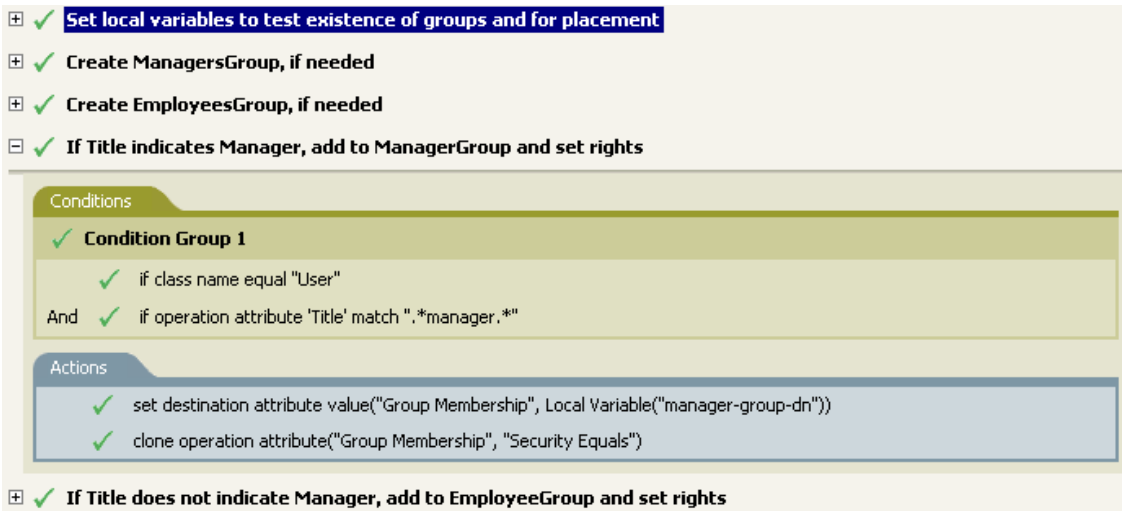
ターゲット名

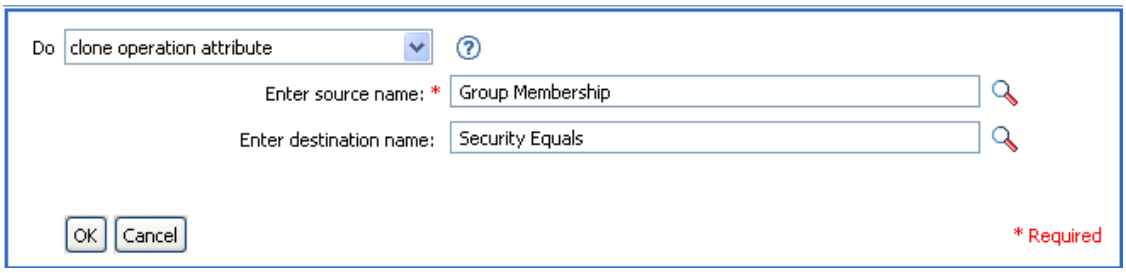
コピー先の属性の名前を指定します。

例

この例では、役職に基づいて、ユーザオブジェクトを適切なグループ (従業員またはマネージャ) に追加します。必要に応じてグループも作成し、そのグループに同等セキュリティを設定します。これは「Govern Groups for User Based on Title Attribute (役職属性に基づくユーザグループの管理)」というポリシーで、Novell のサポート Web サイトからダウ

ンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。





この「操作属性のクローン」の例では、グループメンバーシップ属性から情報を取得し、これに同等セキュリティを追加して同じ値になるようにしています。

2.6.15 ターゲットオブジェクトの削除

ターゲットデータストア内のオブジェクトを削除します。

フィールド

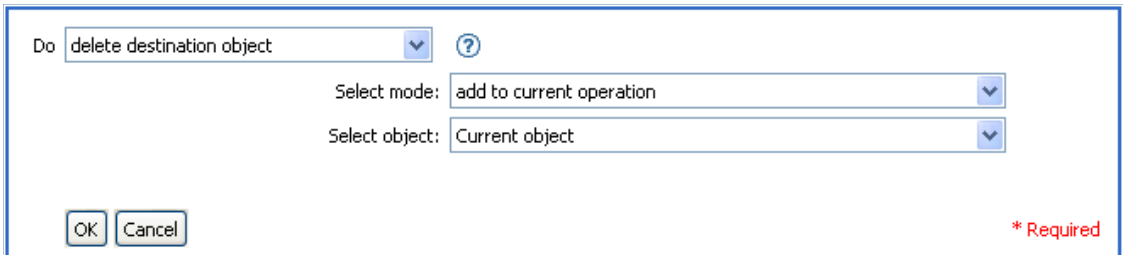
モード

このアクションを現在の操作の前と後のどちらに追加するか、または目的のデータストアへ直接書き込むかを選択します。

オブジェクト

ターゲットオブジェクトを選択します。このオブジェクトは現在のオブジェクトにすることも、DN または関連付けによって指定することもできます。

例



2.6.16 ソースオブジェクトの削除

ソースデータストア内のオブジェクトを削除します。

フィールド

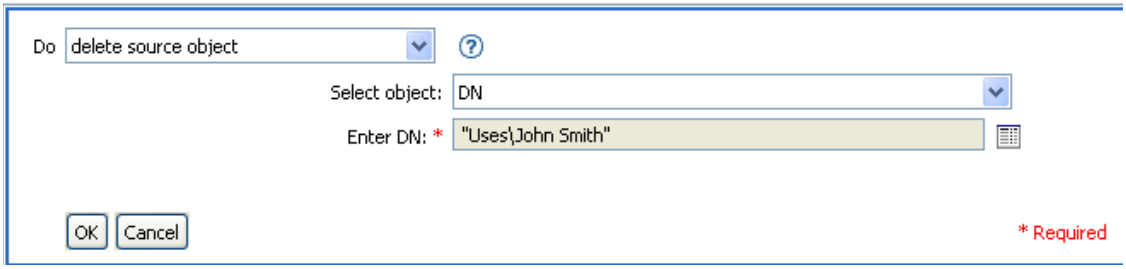
モード

このアクションを現在の操作の前と後のどちらに追加するか、または目的のデータストアへ直接書き込むかを選択します。

オブジェクト

ソースデータストア内の削除するターゲットオブジェクトを選択します。このオブジェクトは現在のオブジェクトにすることも、DN または関連付けによって指定することもできます。

例



2.6.17 一致オブジェクトの検索

ターゲットデータストアにある現在のオブジェクトに一致するものを検索します。

フィールド

スコープ

検索範囲を選択します。範囲は［エントリ］、［サブオーディネート］、または［サブツリー］になります。

DN

検索のベースとなる DN を指定します。

一致する属性

検索する属性値を指定します。

備考

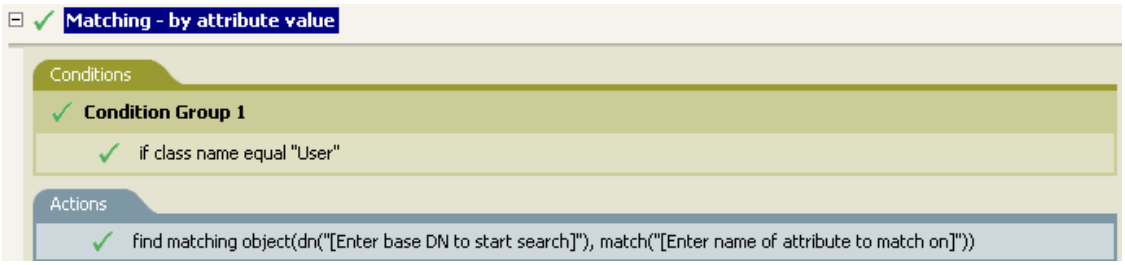
一致オブジェクトの検索は、現在の操作が追加される場合にのみ有効です。

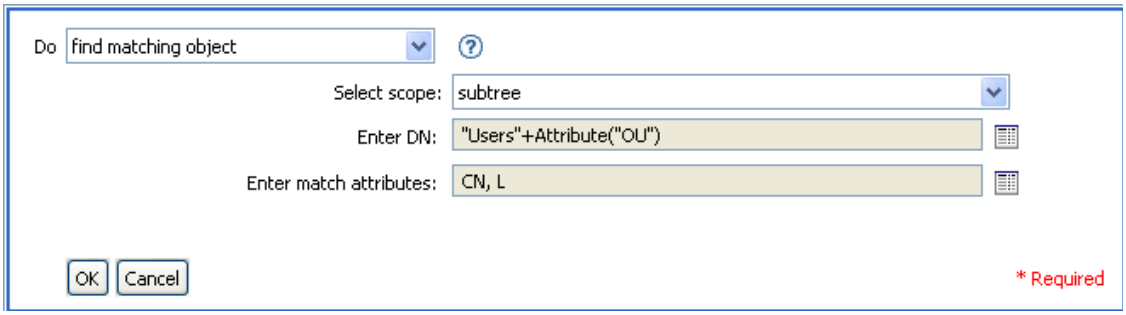
DN 引数は、スコープが［エントリ］の場合のみ必須で、それ以外の場合はオプションです。スコープが［サブツリー］または［サブオーディネート］の場合には、少なくとも 1 つの一致属性が必要です。スコープが［エントリ］の場合には結果は定義されず、一致属性が指定されます。ターゲットデータストアが接続されたアプリケーションである場合は、一致結果が返されるごとに関連付けが現在の操作に追加されます。現在の操作に空でない関連付けがすでにある場合はクエリが実行されないので、同じルール内に一致オブジェクトの検索アクションを複数指定しても問題ありません。

ターゲットデータストアがアイデンティティボールトである場合は、現在の操作のターゲット DN 属性が設定されます。現在の操作にすでに空でないターゲット DN 属性がある場合はクエリが実行されないので、同じルール内に一致オブジェクトの検索アクションを複数指定しても問題ありません。結果が 1 つだけ返され、それがまだ関連付けられていない場合は、現在の操作のターゲット DN が一致オブジェクトのソース DN に設定されます。結果が 1 つだけ返され、それがすでに関連付けられている場合は、現在の操作のターゲット DN が 1 文字の ￼ に設定されます。複数の結果が返される場合は、現在の操作のターゲット DN が 1 文字の &#xFFFD; に設定されます。

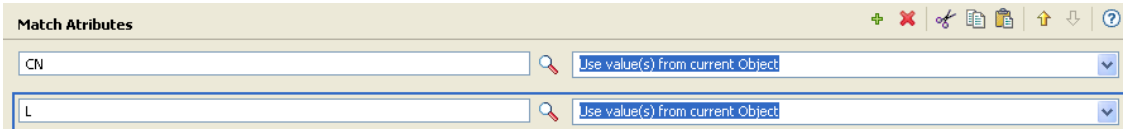
例

この例では、ユーザオブジェクトで属性 CN と L を使用して照合します。ルールが検索している場所のユーザコンテナで開始され、OU 属性内に格納された情報を DN に追加します。このルールは、Identity Manager 3.0 に付属している事前定義されたルールです。詳細については、94 ページの 「一致 - 属性値別」を参照してください。





引数ビルダのアイコンをクリックすると、一致属性ビルダが開きます。ビルダで照合する属性を指定します。この例では、CN および L の属性を使用しています。



2.6.18 繰り返し (For Each)

ノードセット内の各ノードに対し、アクションのセットを繰り返します。

フィールド

ノードセット

ノードセットを指定します。

アクション

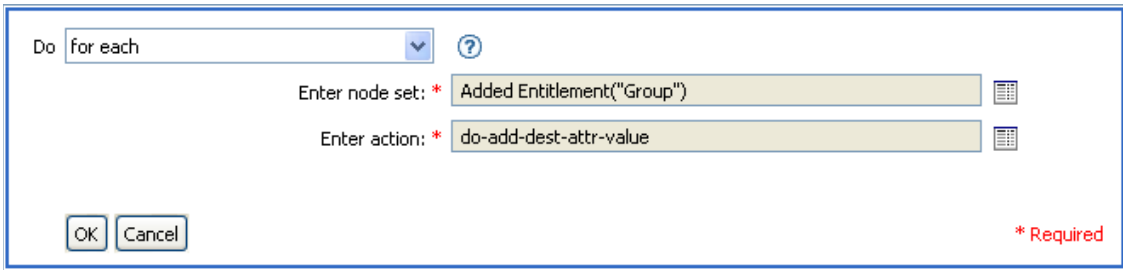
ノードセットの各ノードに対して実行するアクションを指定します。

備考

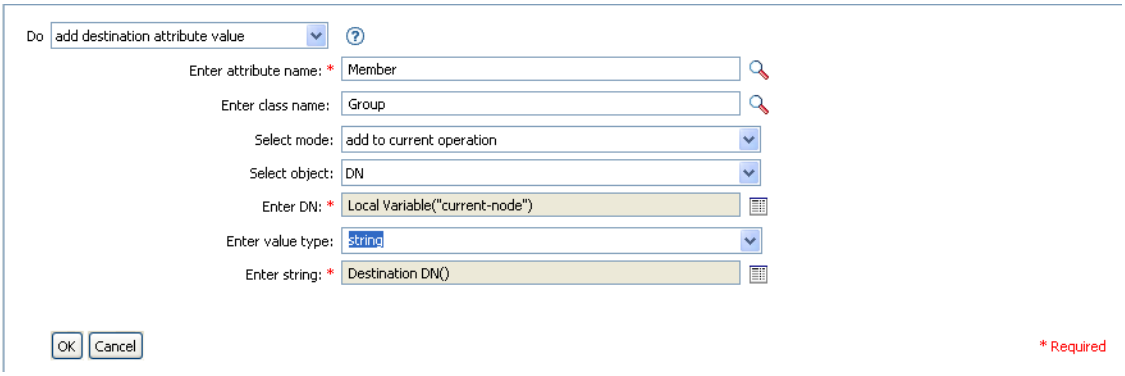
ローカル変数が使用される場合、アクションのセットが 1 回実行されるたびに、現在のノードは異なる値になります。

For each では、ノードセット内のあるノードがエンタイトルメントである場合、それに対しては黙示的に 162 ページの 「エンタイトルメントの実装」アクションが実行されます。

例



次に示すのは、引数アクションビルダの例で、アクションの引数を指定するために使用されます。



2.6.19 イベントの生成

ユーザ定義イベントを Novell Audit に送信します。

フィールド

ID

イベントの ID を指定します。この ID は、1000 ～ 1999 の範囲の整数である必要があります。

レベル

イベントのレベルを選択します。

レベル	説明
log-emergency	メタディレクトリエンジンまたはドライバがシャットダウンされるイベント。
log-alert	早急に注意が必要なイベント。
log-critical	メタディレクトリエンジンまたはドライバの一部が正常に動作しなくなるイベント。
log-error	メタディレクトリエンジンまたはドライバによって処理できるエラーを示すイベント。
log-warning	大きな問題としては取り上げられないネガティブなイベント。
log-notice	管理者が使い方や操作を理解または向上するのに使用できるイベント (ポジティブまたはネガティブ)。
log-info	何らかの重要性を持つポジティブなイベント。
log-debug	サポート担当者またはエンジニアがメタディレクトリエンジンまたはドライバの操作をデバッグするためのイベント。

文字列

イベントに含めるユーザ定義の文字列値、整数値、およびバイナリ値を指定します。これらの値は、名前付き文字列ビルダを使用して指定します。

文字列名	説明
target	イベントの対象になるオブジェクト。
target-type	ターゲットの定義済みの形式を示す整数。現在定義済みの target-type の値を示します。 ◆ 0 = なし ◆ 1 = スラッシュ表記 ◆ 2 = ドット表記 ◆ 3 = LDAP 表記
subTarget	イベントの対象になるターゲットのサブコンポーネント。
text1	ここに入力されるテキストは、text1 イベントフィールドに格納されます。
text2	ここに入力されるテキストは、text2 イベントフィールドに格納されます。
text3	ここに入力されるテキストは、text3 イベントフィールドに格納されます。
value	ここに入力される任意の数字は、value イベントフィールドに格納されます。
value3	ここに入力される任意の数字は、value3 イベントフィールドに格納されます。
data	ここに入力されるデータは、Blob イベントフィールドに格納されます。

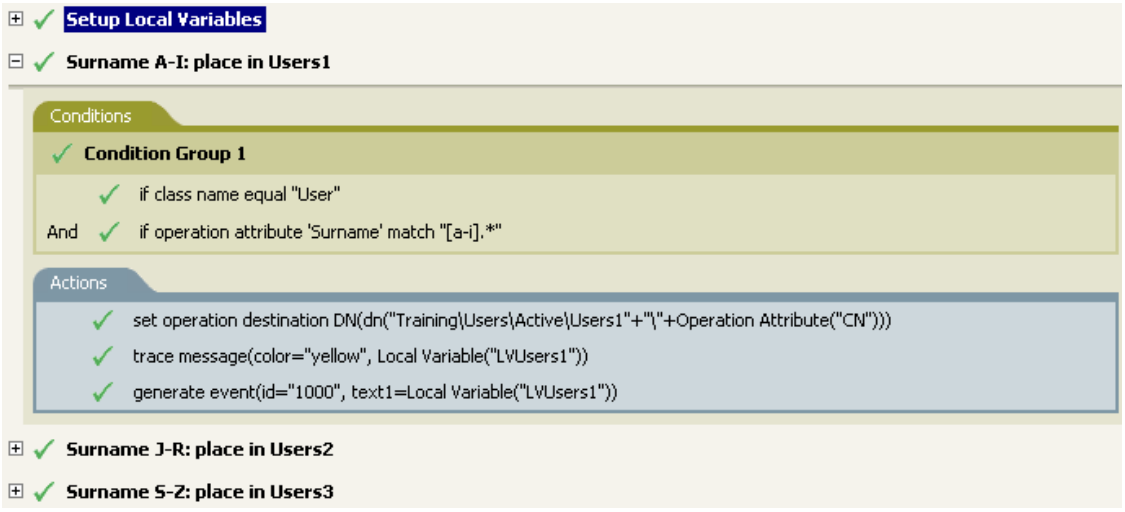
備考

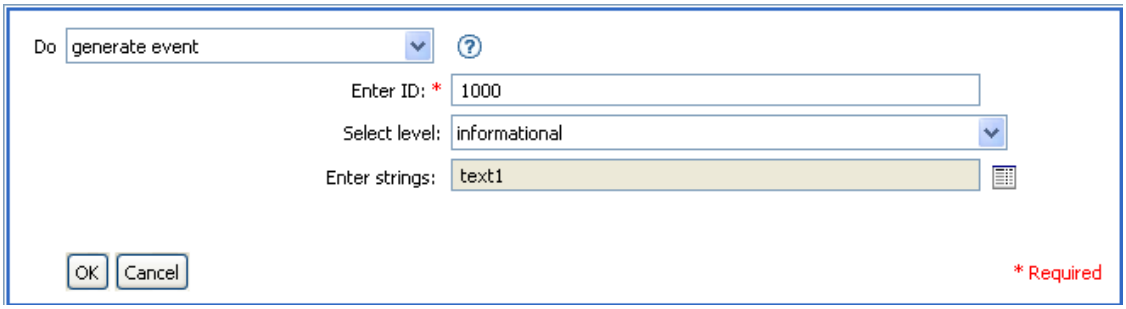
Novell Audit イベント構造には、1 つのターゲット、1 つのサブターゲット、3 つの文字列 (text1、text2、text3)、2 つの整数 (value、value3)、および 1 つの一般的なフィールド (data) が含まれます。テキストフィールドは 256 バイトに制限されています。データフィールドには 3KB までの情報を含めることができます。ただし、環境によってはこれより大きいデータフィールドを使用できる場合もあります。

例

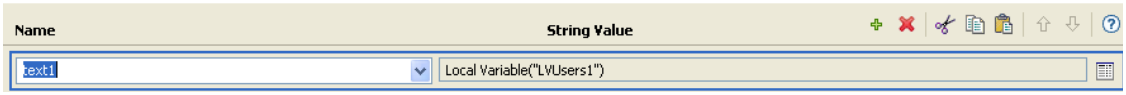
この例には 4 つのルールがあり、これらのルールでは名字属性の最初の文字に基づいてユーザオブジェクトに配置ポリシーを実装し、トレースメッセージおよびカスタムの Novell Audit イベントの両方を生成します。イベントの生成アクションは、Novell Audit にイベントを送信する場合に使用されます。これは、「Policy to Place by Surname (名字で配置するためのポリシー)」という名前のポリシーで、Novell のサポート Web サイトからダ

ウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。





次に示すのは、名前付き文字列ビルダの例で、文字列の引数を指定しているところです。



イベントの生成により、ID が 1000 のイベントを作成中で、LVUser1 のローカル変数で生成されたテキストを示しています。ローカル変数 LVUser1 は、+”“+”Training\Users\Active\Users1”+” コンテナ ” に追加されたユーザ : 操作属性 “cn” の文字列です。このイベントは、Trainging\Users\Active\Users1 コンテナに追加されたユーザ :jsmith を読み込みます。

2.6.20 エンタイトルメントの実装

エンタイトルメントを実装するアクションを指定することで、これらのエンタイトルメントのステータスが、そのエンタイトルメントを付与または取り消したエージェントにレポートされるようにします。

フィールド

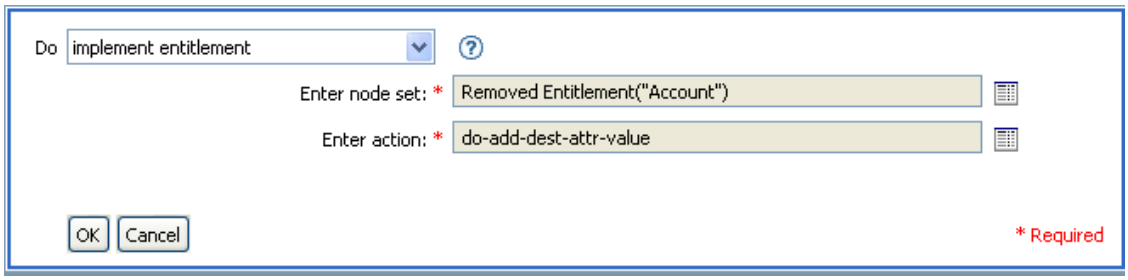
ノードセット

指定されたアクションによって実装中のエンタイトルメントが含まれるノードセット。

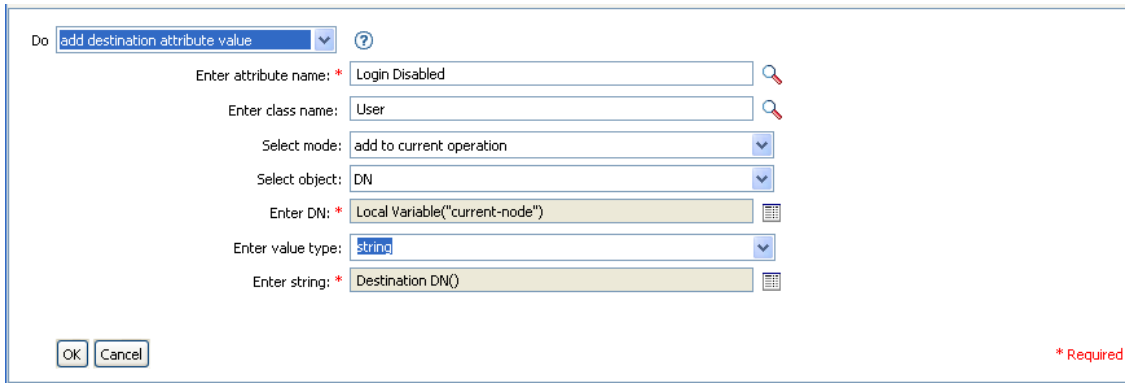
アクション

指定されたエンタイトルメントを実装するアクション。

例



次に示すのは、引数アクションビルダの例で、アクションの引数を指定する場合に使用されます。



2.6.21 ターゲットオブジェクトの移動

ターゲットデータストア内のオブジェクトを移動します。

フィールド

モード

このアクションを現在の操作の前と後のどちらに追加するか、または目的のデータストアへ直接書き込むかを選択します。

クラス名

(オプション) 移動するオブジェクトのクラス名を指定します。現在のオブジェクトのクラス名を使用するには、空白のままにします。

移動するオブジェクト

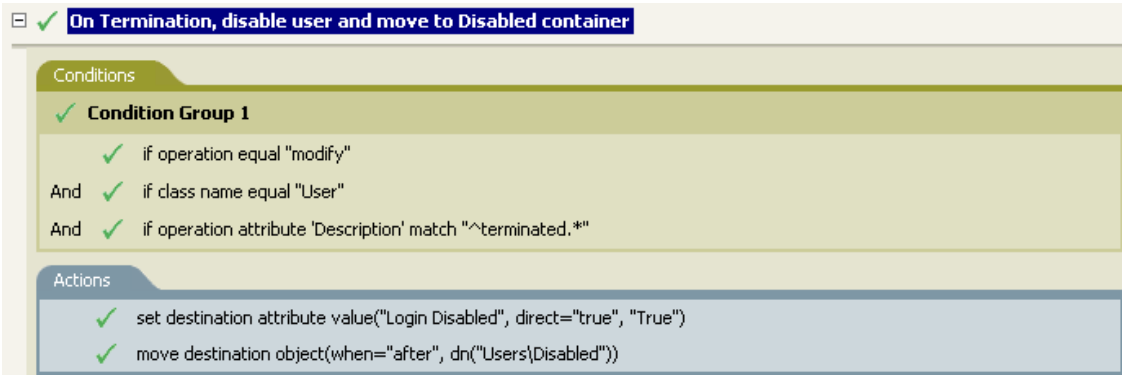
移動するオブジェクトを選択します。このオブジェクトは現在のオブジェクトにすることも、DN または関連付けによって指定することもできます。

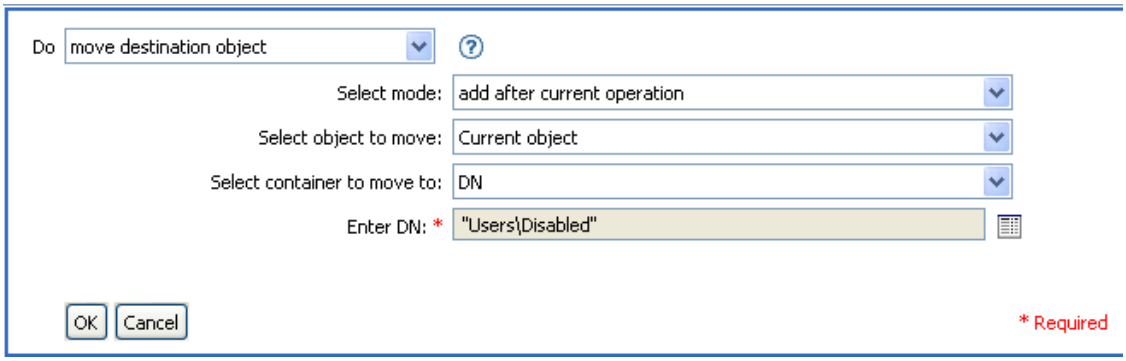
移動するコンテナ

ターゲットコンテナを選択します。このコンテナは、DN または関連付けによって指定します。

例

この例にはルールが 1 つ含まれています。このルールは、説明属性がユーザの終了を示している場合にユーザのアカウントを無効にし、アカウントを無効なコンテナに移動します。これは、「Disable User Account and Move When Terminated (終了時のユーザアカウントの無効化と移動)」という名前のポリシーで、Novel のサポート Web サイトからダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。





このポリシーでは、それがユーザオブジェクトの変更イベントであるかどうか、および説明属性に終了の値が含まれているかどうかを確認します。該当する場合、「ログインの無効化」の属性を True に設定し、そのオブジェクトを User\Disabled コンテナに移動します。

2.6.22 ソースオブジェクトの移動

ソースデータストア内のオブジェクトを移動します。

フィールド

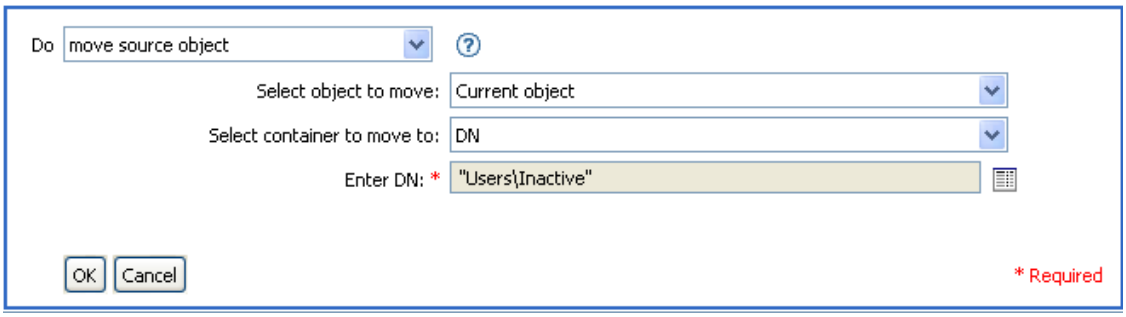
移動するオブジェクト

移動するオブジェクトを選択します。このオブジェクトは現在のオブジェクトにすることも、DN または関連付けによって指定することもできます。

移動するコンテナ

ターゲットコンテナを選択します。このコンテナは、DN または関連付けによって指定します。

例



2.6.23 操作属性の再フォーマット

パターンを使用して、現在の操作内にある属性のすべての値を再フォーマットします。

フィールド

名前

属性の名前を指定します。

値のタイプ

新しい属性値の構文を指定します。

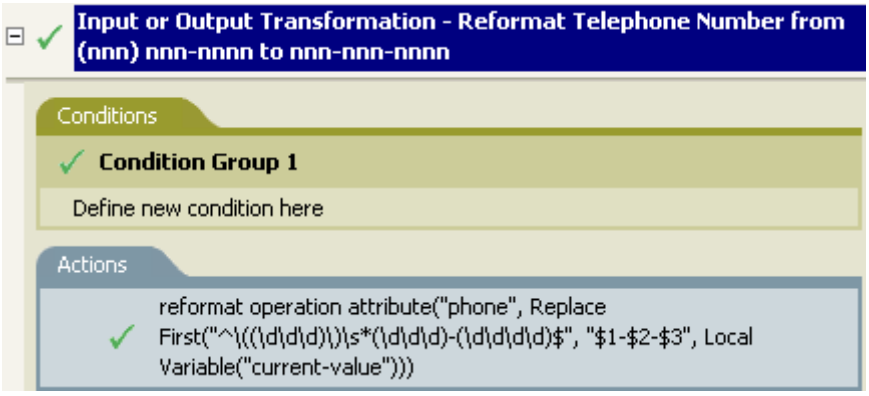
値

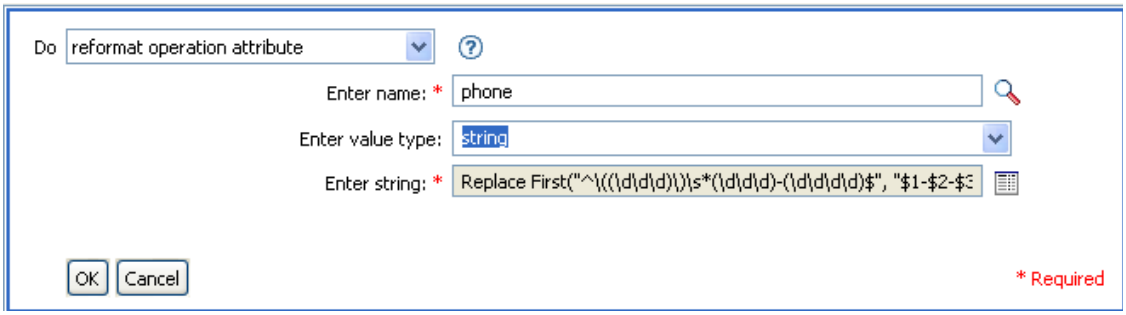
属性値の新しいフォーマットのパターンとして使用する値を指定します。新しい値を作成するのに元の値が必要な場合は、ローカル変数 current-value を参照することで取得する必要があります。

例

この例では、電話番号を再フォーマットします。(nnn)-nnn-nnnn から nnn-nnn-nnnn に変更します。このルールは、Identity Manager 3.0 に付属している事前定義されたルールです。

詳細については、88 ページの 「入出力変換 - 電話番号の形式を (nnn) nnn-nnnn から nnn-nnn-nnnn に変更」を参照してください。





このアクションの再フォーマット操作属性により、電話番号の形式を変更します。このルールでは、引数ビルダと正規表現を使用して、情報の表示方法を変更します。

2.6.24 関連付けを削除

関連付けを削除するコマンドをアイデンティティボールトに送信します。

フィールド

モード

このアクションを現在の操作の前と後のどちらに追加するか、または目的のデータストアへ直接書き込むかを選択します。

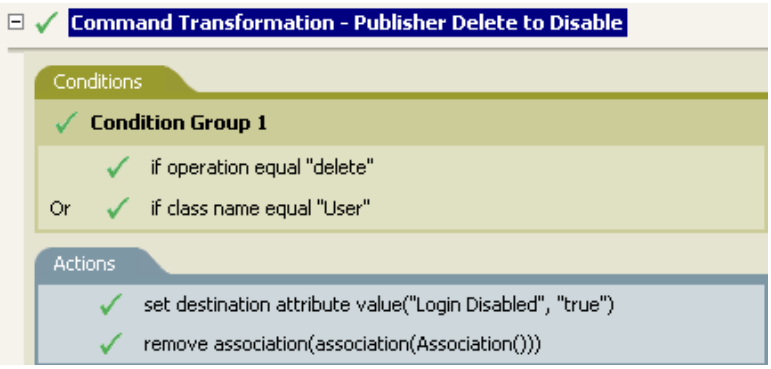
関連付け

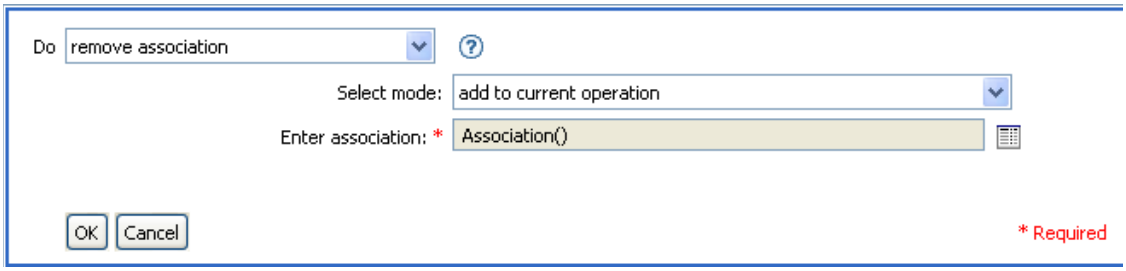
削除する関連付けの値を指定します。

例

この例では、削除操作を使用して、ユーザオブジェクトを無効にします。イベントは変換されます。このルールは、Identity Manager 3.0 に付属している事前定義されたルールで

す。詳細については、76 ページの 「コマンド変換 - 無効にする発行者の削除」を参照してください。





ユーザオブジェクトに対して削除操作が行われるときは、「ログインの無効化」の値が True に設定され、ユーザオブジェクトから関連付けが削除されます。関連付けが削除されるのは、接続アプリケーション内に関連付けられたオブジェクトが存在しなくなったためです。

2.6.25 ターゲット属性値の削除

ターゲットデータストア内のオブジェクトから、属性値を 1 つ削除します。

フィールド

属性名

属性の名前を指定します。

クラス名

(オプション) ターゲットオブジェクトのクラス名を指定します。現在のオブジェクトのクラス名を使用するには、空白のままにします。

モード

このアクションを現在の操作の前と後のどちらに追加するか、または目的のデータストアへ直接書き込むかを選択します。

オブジェクト

ターゲットオブジェクトを選択します。このオブジェクトは現在のオブジェクトにすることも、DN または関連付けによって指定することもできます。

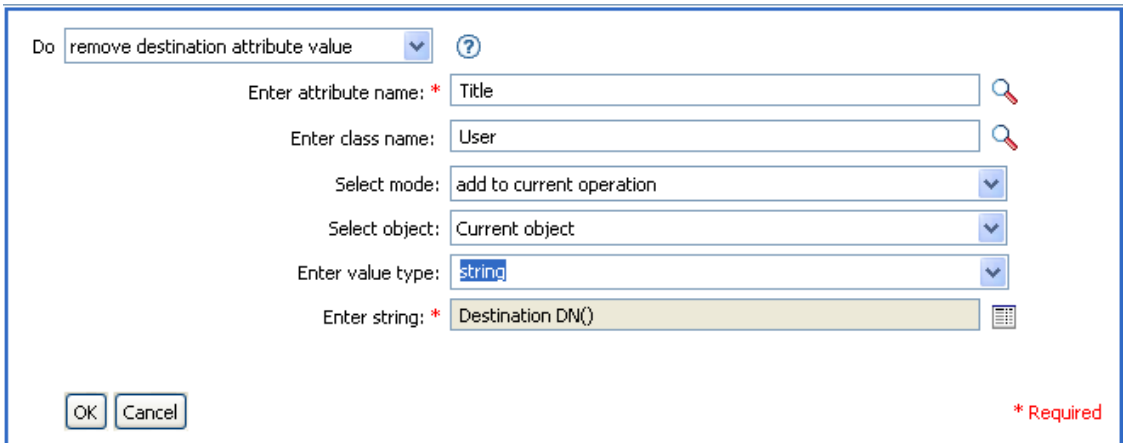
値のタイプ

削除する属性値の構文を指定します。

値

新しい属性値の値を指定します。

例



2.6.26 ソース属性値の削除

ソースデータストア内のオブジェクトにある名前付き属性から、指定した値を削除します。

フィールド

属性名

属性の名前を指定します。

クラス名

(オプション) ターゲットオブジェクトのクラス名を指定します。現在のオブジェクトのクラス名を使用するには、空白のままにします。

オブジェクト

ターゲットオブジェクトを選択します。このオブジェクトは現在のオブジェクトにすることも、DN または関連付けによって指定することもできます。

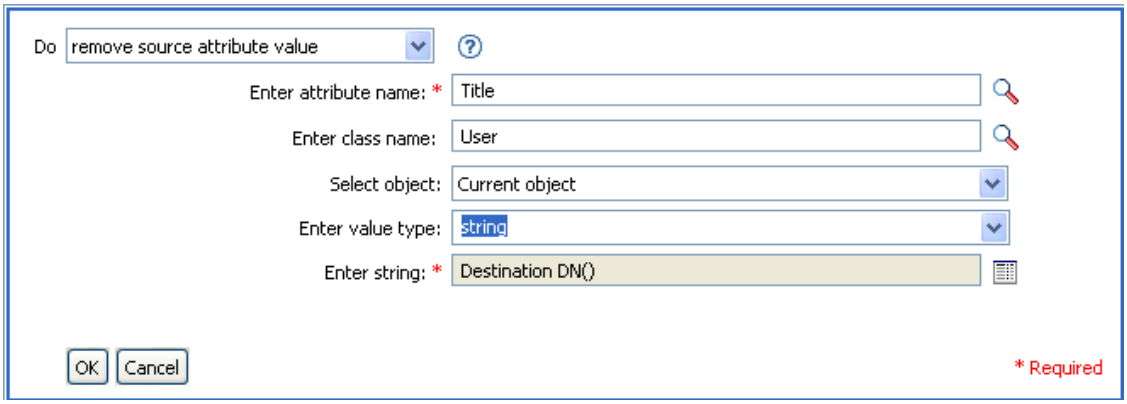
値のタイプ

削除する属性値の構文を指定します。

値

削除する属性値を指定します。

例



2.6.27 ターゲットオブジェクトの名前変更

ターゲットデータストア内のオブジェクトの名前を変更します。

フィールド

モード

このアクションを現在の操作の前と後のどちらに追加するか、または目的のデータストアへ直接書き込むかを選択します。

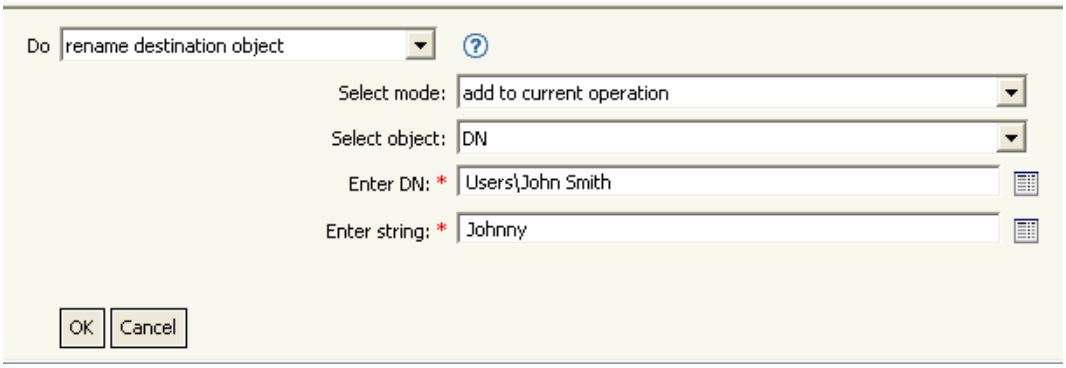
オブジェクト

ターゲットオブジェクトを選択します。このオブジェクトは現在のオブジェクトにすることも、DN または関連付けによって指定することもできます。

文字列

オブジェクトの新しい名前を指定します。

例



2.6.28 操作属性の名前変更

現在の操作で属性に行ったすべての内容に対する名前を変更します。

フィールド

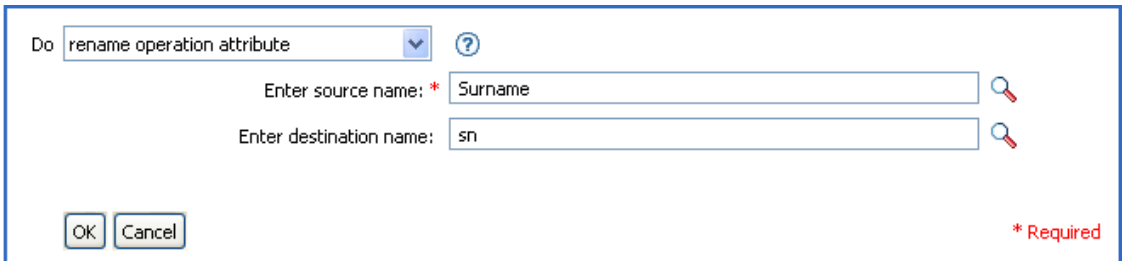
ソース名

変更前の属性名を指定します。

ターゲット名

新しい属性名を指定します。

例



2.6.29 ソースオブジェクトの名前変更

ソースデータストア内のオブジェクトの名前を変更します。

フィールド

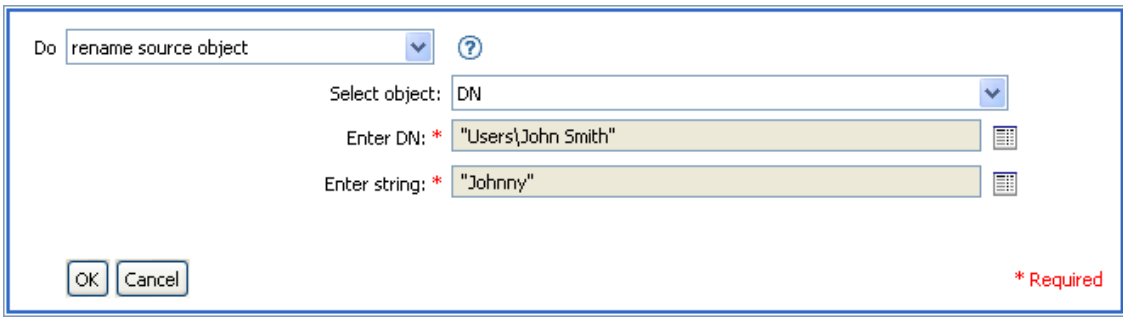
オブジェクト

ターゲットオブジェクトを選択します。このオブジェクトは現在のオブジェクト、DN または関連付けにすることができます。

文字列

オブジェクトの新しい名前を指定します。

例



2.6.30 電子メールの送信

電子メール通知を送信します。

フィールド

ID

(オプション) メッセージを送信する SMTP システムでのユーザ ID を指定します。

サーバ

SMTP サーバ名を指定します。

パスワード

(オプション)SMTP サーバのアカウントのパスワードを指定します。

重要 **:** パスワード属性の値はクリアテキストで保存されます。

タイプ

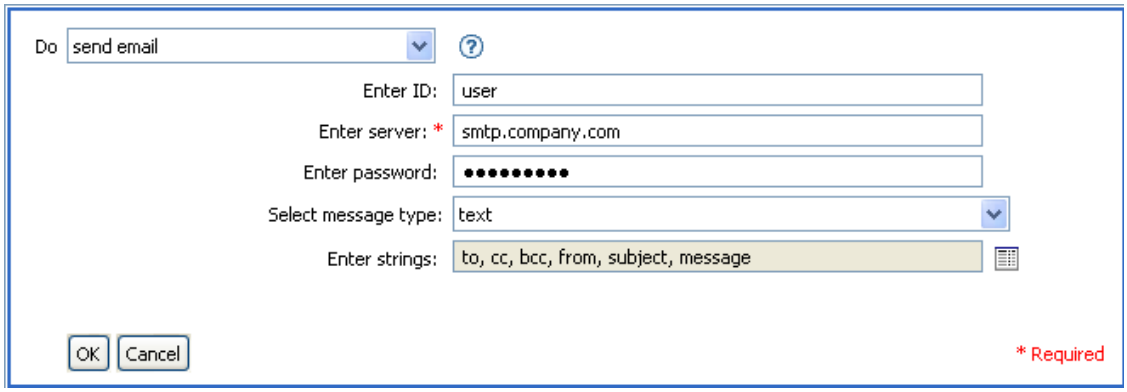
電子メールメッセージのタイプを選択します。

文字列

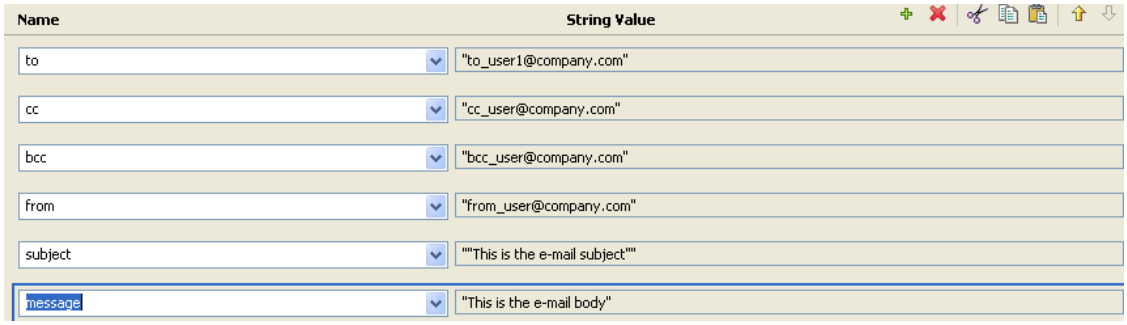
さまざまな電子メールアドレス、件名、およびメッセージなどの値を指定します。次の表に、有効な名前付き文字列の引数を示します。

文字列名	説明
宛先	電子メールの受信者リストにアドレスを追加します。複数のインスタンスを指定できます。
cc	電子メールの CC の受信者リストにアドレスを追加します。複数のインスタンスを指定できます。
bcc	電子メールの BCC の受信者リストにアドレスを追加します。複数のインスタンスを指定できます。
送信者	電子メールの発信アドレスとして使用されるアドレスを指定します。
返信先	電子メールメッセージの返信アドレスとして使用されるアドレスを指定します。
件名	電子メールの件名を指定します。
メッセージ	電子メールメッセージの内容を指定します。
エンコード	電子メールメッセージで使用する文字エンコードを指定します。

例



次に示すのは、名前付き文字列ビルダの例で、文字列の引数を指定しているところです。



2.6.31 テンプレートから電子メールを送信

テンプレートを使用して、電子メール通知を生成します。

フィールド

通知 **DN**

SMTP 通知設定オブジェクトのスラッシュ形式の DN を指定します。

テンプレート **DN**

電子メールテンプレートオブジェクトのスラッシュ形式の DN を指定します。

パスワード

(オプション)SMTP サーバのアカウントのパスワードを指定します。

重要 **:** パスワード属性の値はクリアテキストで保存されます。

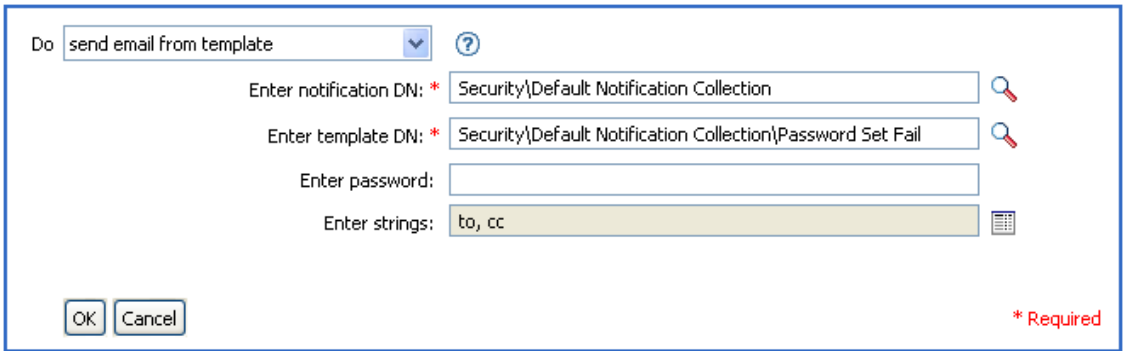
文字列

電子メールメッセージの他のフィールドを指定します。次の表に、さまざまな電子メールアドレスを指定する、予約済みのフィールド名を示します。

文字列名	説明
to	電子メールの受信者リストにアドレスを追加します。複数のインスタンスを指定できます。
cc	電子メールの CC の受信者リストにアドレスを追加します。複数のインスタンスを指定できます。
bcc	電子メールの BCC の受信者リストにアドレスを追加します。複数のインスタンスを指定できます。
reply-to	電子メールメッセージの返信アドレスとして使用されるアドレスを指定します。
encoding	電子メールメッセージで使用する文字エンコードを指定します。

各テンプレートでは、電子メールメッセージの件名および本文で置き換えられるフィールドも定義できます。

例



次に示すのは、名前付き文字列ビルダの例で、文字列の引数を指定しているところです。

2.6.32 デフォルト属性値の設定

属性に値が指定されていない場合に、現在の操作にデフォルト値を追加します (オプションで、ソースデータストア内の現在のオブジェクトにも追加します)。これは、現在の操作が「追加」の場合のみ有効です。

フィールド

属性名

デフォルト属性の名前を指定します。

ライトバック

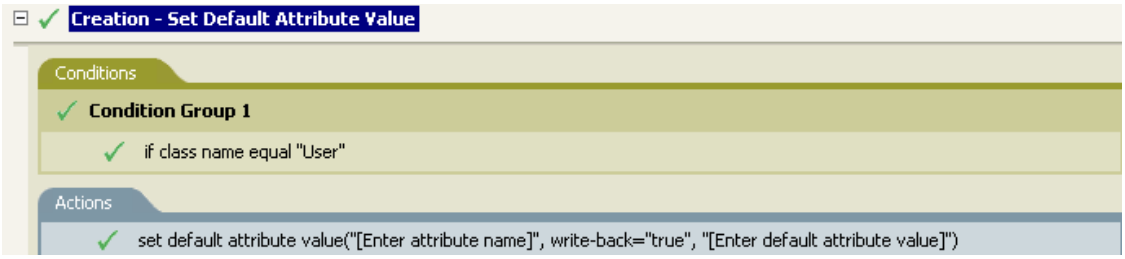
デフォルト値をソースデータストアにもライトバックするかどうかを選択します。

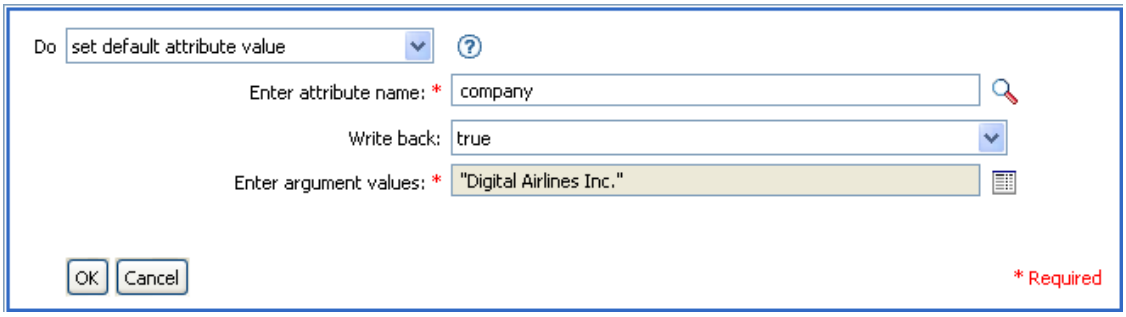
値

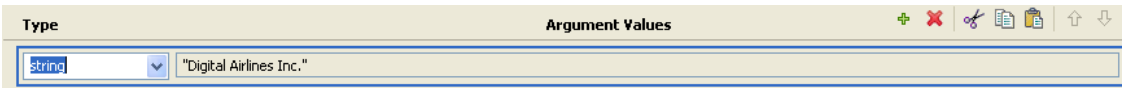
属性のデフォルト値を指定します。

例

この例では、属性「company」のデフォルト値を設定します。必要な属性に値を設定できます。このルールは、Identity Manager 3.0 に付属している事前定義されたルールです。詳細については、81 ページの 「作成 - デフォルト属性値の設定」を参照してください。







値を作成するには、引数値リストビルダを起動します。このビルダの詳細については、65 ページの 「引数値リストビルダ」を参照してください。必要な値を設定できます。この場合、引数ビルダを使用して、「company」という名前のテキストを入力しました。

2.6.33 ターゲット属性値の設定

ターゲットデータストアにあるオブジェクトの属性に値を追加し、その属性に設定されている他の値をすべて削除します。

フィールド

属性名

属性の名前を指定します。

クラス名

(オプション) ターゲットデータストア内のターゲットオブジェクトのクラス名を指定します。現在のオブジェクトのクラス名を使用するには、空白のままにします。

モード

このアクションを現在の操作の前と後のどちらに追加するか、または目的のデータストアへ直接書き込むかを選択します。

オブジェクト

ターゲットオブジェクトを選択します。このオブジェクトは現在のオブジェクトにすることも、DN または関連付けによって指定することもできます。

値のタイプ

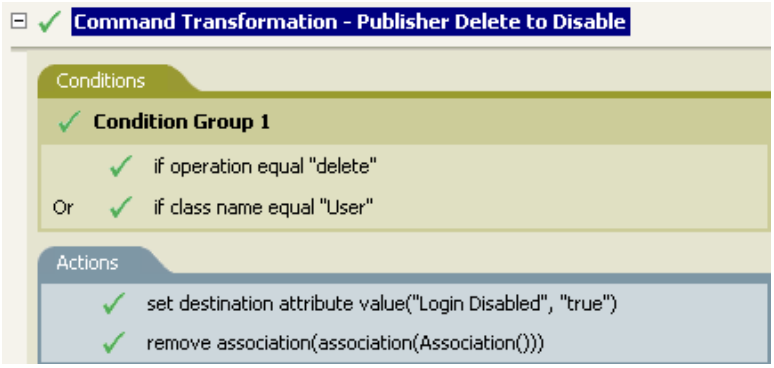
設定する属性値の構文を選択します。

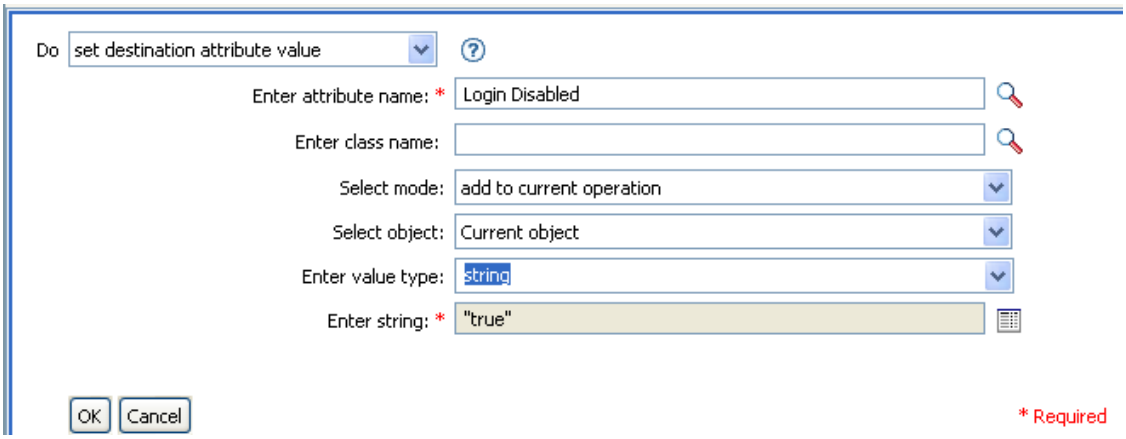
値

設定する属性値を指定します。

例

この例では、削除操作を使用して、ユーザオブジェクトを無効にします。このルールは、Identity Manager 3.0 に付属している事前定義されたルールです。詳細については、76 ページの 「コマンド変換 - 無効にする発行者の削除」を参照してください。





このルールでは、「ログインの無効化」の属性値を True に設定します。このルールでは、引数ビルダを使用して、この属性値としてテキスト「True」を追加します。このビルダの詳細については、61 ページの 「引数ビルダ」を参照してください。

2.6.34 ターゲットパスワードの設定

ターゲットデータストアにある現在のオブジェクトのパスワードを設定します。

フィールド

モード

このアクションを現在の操作の前と後のどちらに追加するか、または目的のデータストアへ直接書き込むかを選択します。

オブジェクト

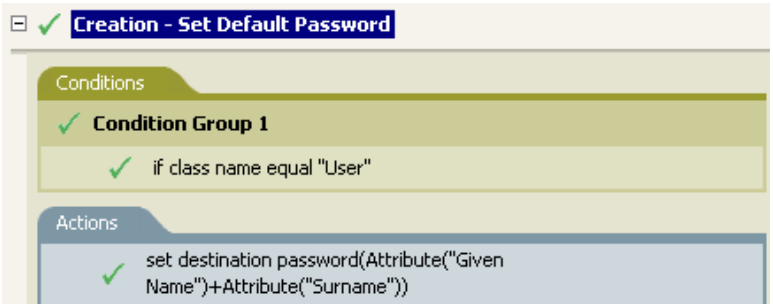
ターゲットオブジェクトを選択します。このオブジェクトは現在のオブジェクトにすることも、DN または関連付けによって指定することもできます。

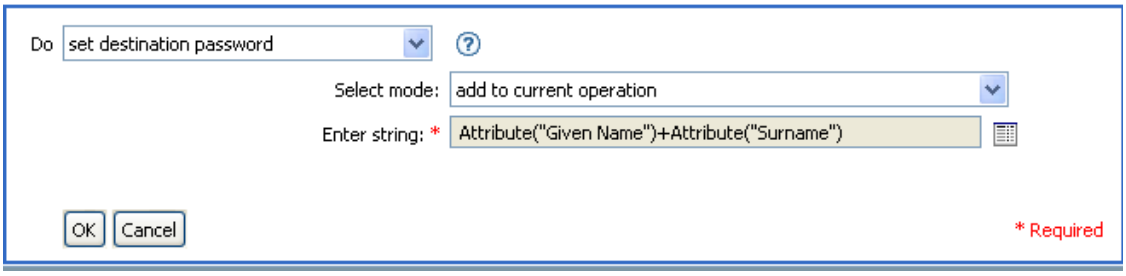
文字列

設定するパスワードを指定します。

例

この例では、作成されるユーザオブジェクトのデフォルトのパスワードを設定します。このルールは、Identity Manager 3.0 に付属している事前定義されたルールです。詳細については、83 ページの 「作成 - デフォルトパスワードの設定」を参照してください。





ユーザオブジェクトが作成され場合、パスワードは、名前属性に名字属性を加えたもの設定されます。

2.6.35 ローカル変数の設定

ローカル変数を設定します。

フィールド

変数名

ローカル変数の名前を指定します。

変数タイプ

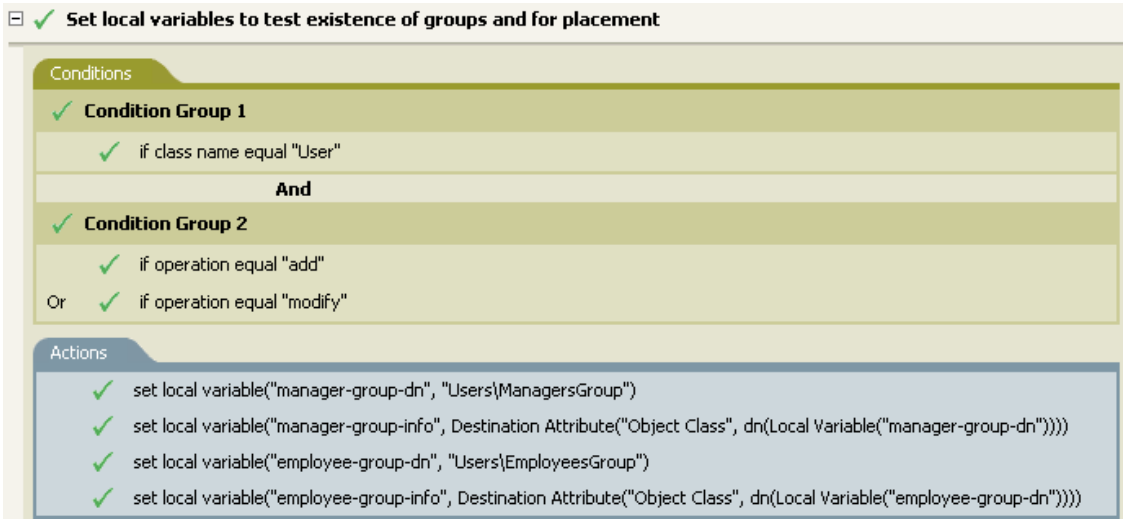
ローカル変数のタイプを選択します。文字列、XPath 1.0 ノードセット、または Java オブジェクトにできます。

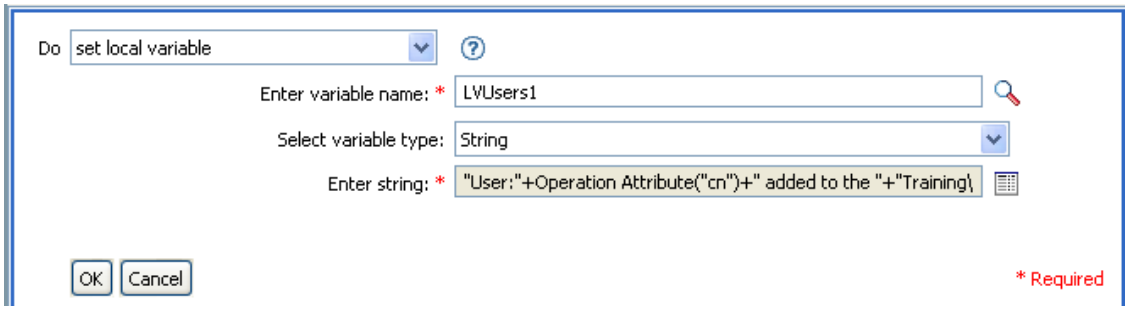
値

新しいローカル変数の値を指定します。

例

この例では、役職に基づいて、ユーザオブジェクトを適切なグループ (従業員またはマネージャ) に追加します。必要に応じてグループも作成し、そのグループに同等セキュリティを設定します。これは「Govern Groups for User Based on Title (役職に基づくユーザグループの管理)」という名前のポリシーで、Novell のサポート Web サイトからダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。





ローカル変数は、ユーザオブジェクトのターゲット属性 (オブジェクトクラスとローカル変数 manager-group-info) の値に設定されます。引数ビルダは、ローカル変数の作成に使用されます。詳細については、61 ページの 「引数ビルダ」を参照してください。

2.6.36 操作関連付けの設定

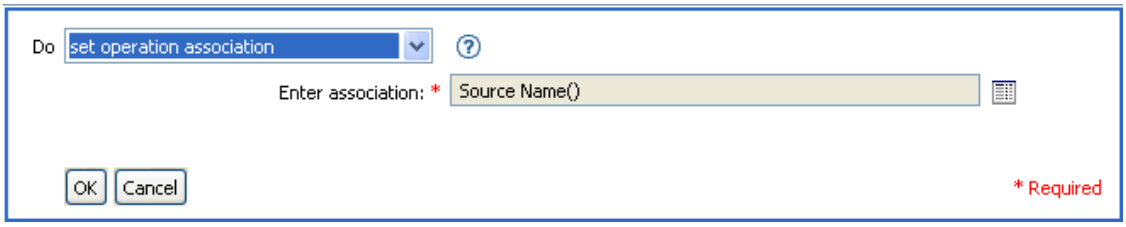
現在の操作に関連付けの値を設定します。

フィールド

関連付け

新しい関連付けの値を指定します。

例



2.6.37 操作クラス名の設定

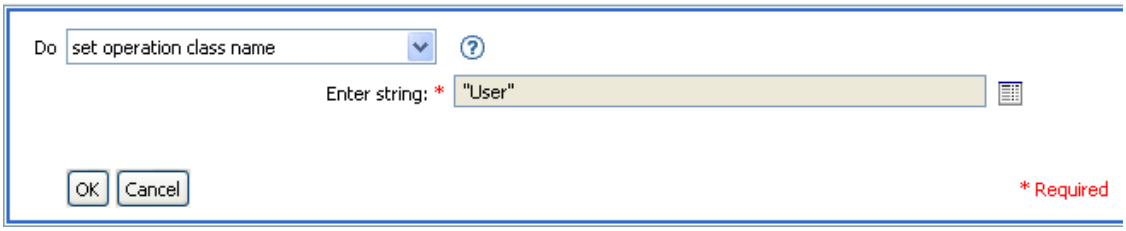
現在の操作のオブジェクトクラス名を設定します。

フィールド

文字列

新しいクラス名を指定します。

例



2.6.38 操作ターゲット DN の設定

現在の操作のターゲット DN を設定します。

フィールド

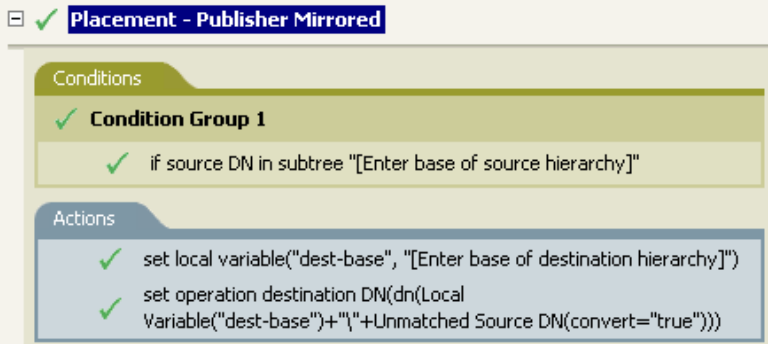
DN

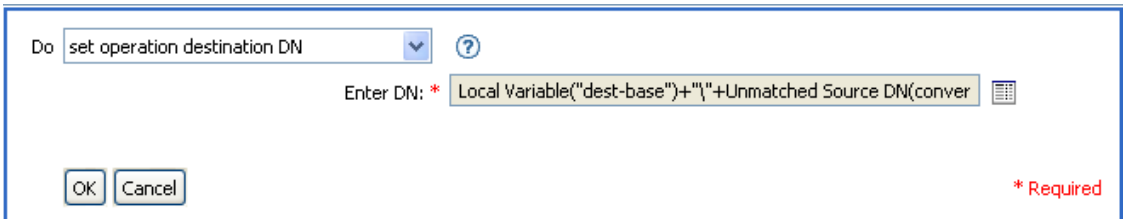
新しいターゲット DN を指定します。

例

この例では、接続システムからミラー化された構造を使用して、アイデンティティボールト内にオブジェクトを配置します。ソースおよびターゲットのデータストアで、ミラー化

を開始するポイントを定義する必要があります。このルールは、Identity Manager 3.0 に付属している事前定義されたルールです。詳細については、81 ページの 「作成 - デフォルト属性値の設定」を参照してください。





このルールでは、操作ターゲット DN をターゲットのベースロケーションとソース DN のローカル変数として設定します。

2.6.39 操作プロパティの設定

操作プロパティを設定します。操作プロパティは、操作内に保存される名前付きの値です。一般に、操作の結果を処理するポリシーで必要になる可能性がある追加のコンテキストを提供するために使用されます。

フィールド

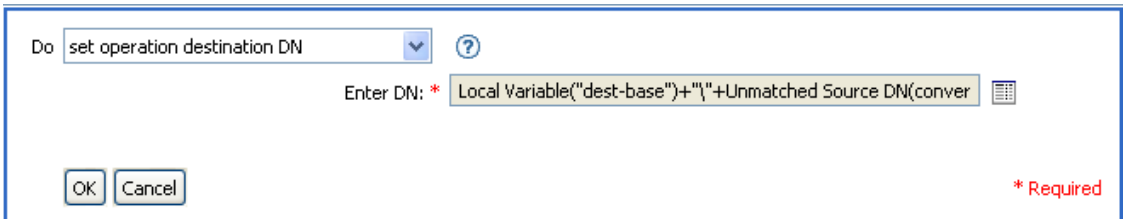
プロパティ名

操作プロパティの名前を指定します。

文字列

操作プロパティの名前を指定します。

例



2.6.40 操作ソース DN の設定

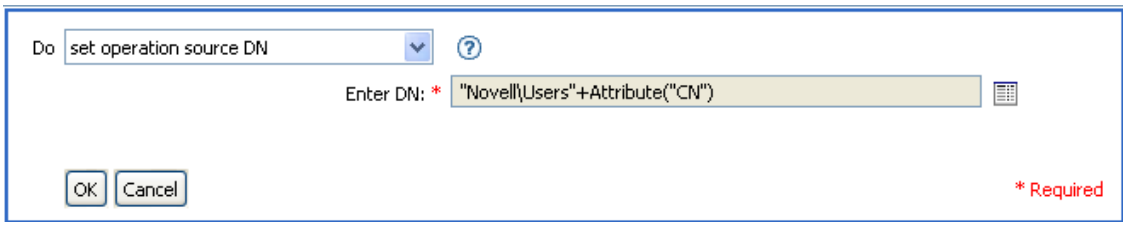
現在の操作のソース DN を設定します。

フィールド

DN

新しいソース DN を指定します。

例



2.6.41 操作テンプレート DN の設定

現在の操作のテンプレート DN を、指定した値に設定します。このアクションは、現在の操作が「追加」の場合のみ有効です。

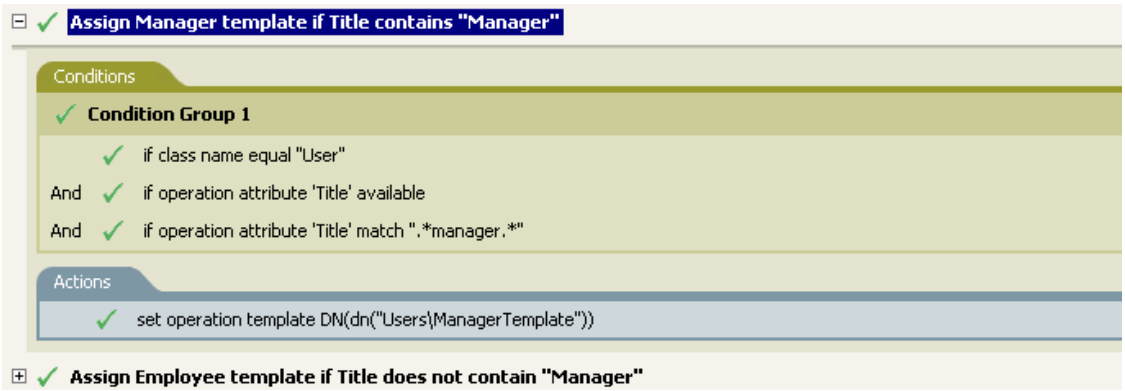
フィールド

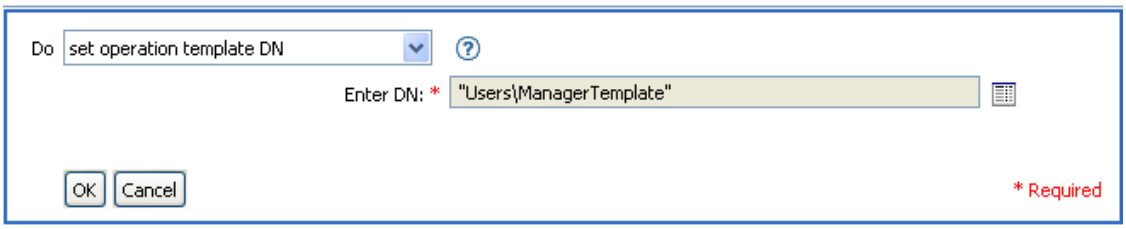
DN

新しいテンプレート DN を指定します。

例

この例では、役職属性に「Manager」という語句が含まれている場合に、Manager テンプレートを適用します。これは「Policy: Assign Template to User Based on Title (ポリシー : 役職名に基づくユーザへのテンプレート割り当て)」という名前のポリシーで、Novell のサポート Web サイトからダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。





テンプレート「Manager Template」は、使用可能な役職属性を持っていて、役職名のどこかに「manager」という語句が含まれているユーザオブジェクトに適用されます。このポリシーでは、一致するすべてのものを検索する正規表現を使用しています。

2.6.42 ソース属性値の設定

ソースデータストアにあるオブジェクトの属性に値を追加し、その属性に設定されている他の値をすべて削除します。

フィールド

属性名

属性の名前を指定します。

クラス名

(オプション) ソースデータストア内のターゲットオブジェクトのクラス名を指定します。現在のオブジェクトのクラス名を使用するには、空白のままにします。

オブジェクト

ターゲットオブジェクトを選択します。このオブジェクトは現在のオブジェクトにすることも、DN または関連付けによって指定することもできます。

値のタイプ

属性値の構文を選択します。

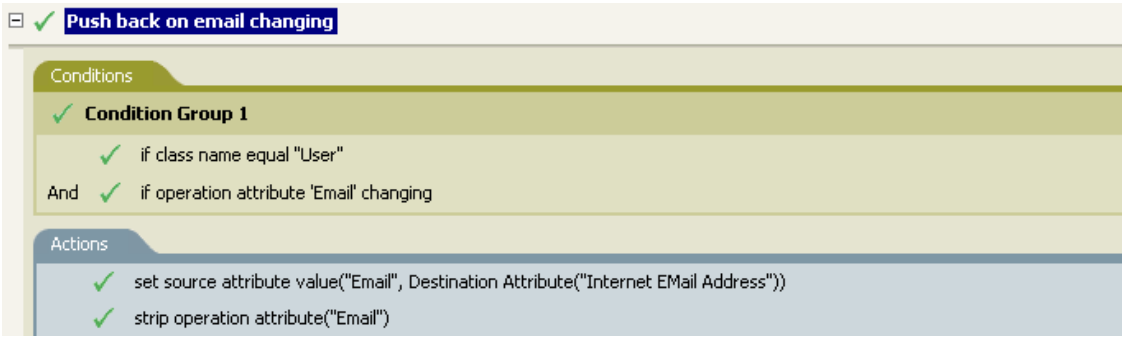
値

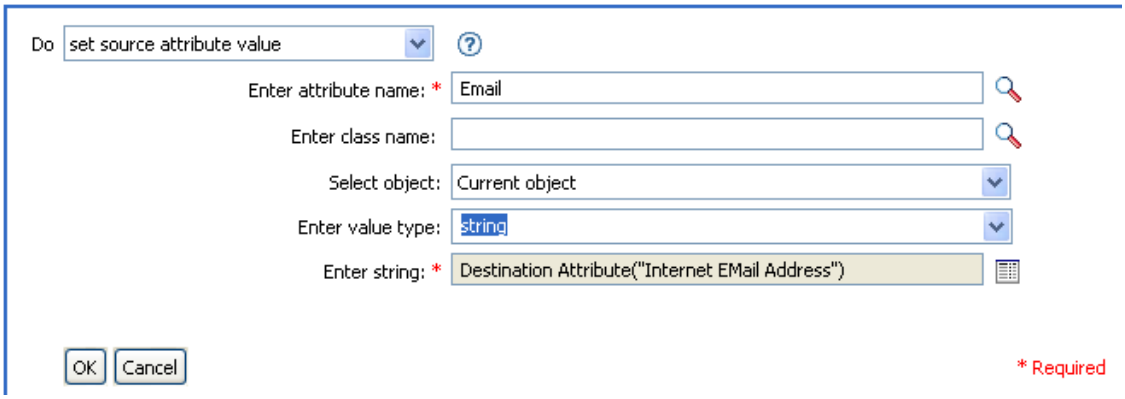
設定する属性値を指定します。

例

この例では、電子メールアドレスが変更された場合に、この変更を元の状態に戻します。これは「Policy: Reset Value of the E-mail Attribute (ポリシー : 電子メール属性値のリセット)」という名前のポリシーで、Novell のサポート Web サイトからダウンロードできます。

詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。





このアクションでは、ターゲット属性「Internet EMail Address」の値を取得し、電子メールのソース属性をこの値と同じに設定します。

2.6.43 ソースパスワードの設定

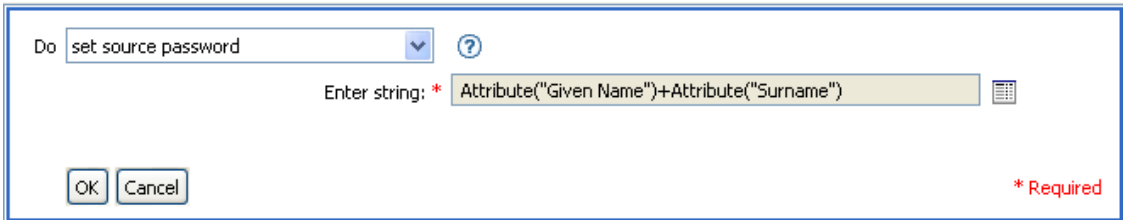
ソースデータストアにある現在のオブジェクトのパスワードを設定します。

フィールド

文字列

設定するパスワードを指定します。

例



2.6.44 SSO 資格情報の設定

ユーザオブジェクトの作成またはパスワードの変更が実施されるときの、SSO 資格情報を設定します。このアクションは、資格情報のプロビジョニングポリシーの一部です。詳細については、333 ページの第 4 章「Novell 資格情報プロビジョニングポリシー」を参照してください。

フィールド

資格情報ストアオブジェクトの **DN**

リポジトリオブジェクトの DN を指定します。

ターゲットユーザの **DN**

ターゲットユーザの DN を指定します。

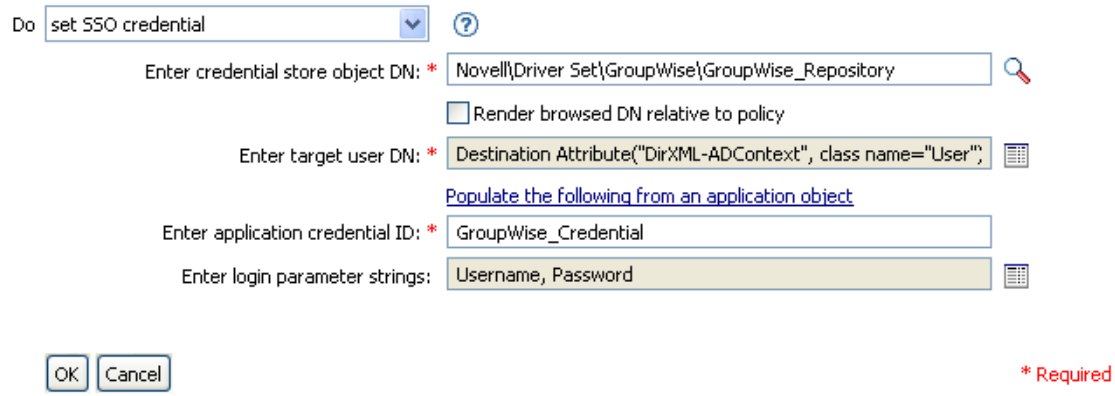
アプリケーションのアクティベーションキー **ID**

アプリケーションオブジェクト内に格納されるアプリケーションの資格情報を指定します。

ログインパラメータの文字列

アプリケーションのログインパラメータを指定します。ログインパラメータとは、アプリケーションオブジェクト内に格納されている認証キーです。

例



2.6.45 SSO パスフレーズの設定

ユーザオブジェクトがプロビジョニングされるときの Novell SecureLogin® のパスフレーズおよび回答を設定します。このアクションは、資格情報のプロビジョニングポリシーの一部です。詳細については、333 ページの第 4 章「Novell 資格情報プロビジョニングポリシー」を参照してください。

フィールド

資格情報ストアオブジェクトの **DN**

リポジトリオブジェクトの DN を指定します。

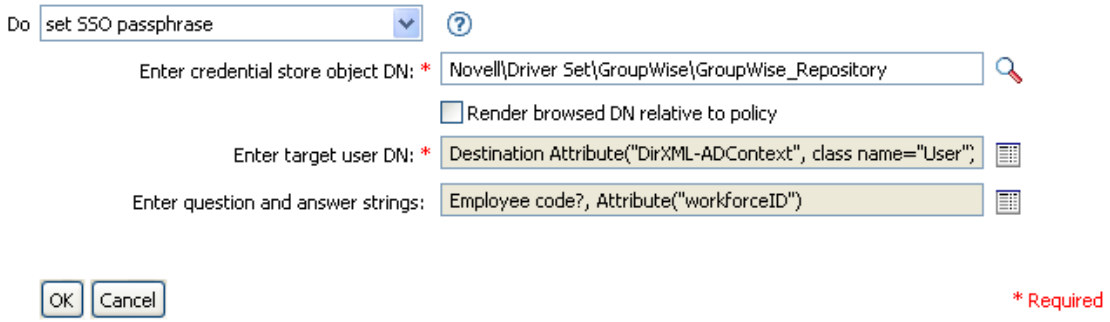
ターゲットユーザの **DN**

ターゲットユーザの DN を指定します。

Question and Answer Strings (質問と回答の文字列 **)**

SecureLogin パスフレーズの質問と回答を指定します。

例



SecureLogin パスフレーズの質問と回答は、ポリシー内に文字列として保存されます。[これらの文字列を編集します] アイコンをクリックして、文字列ビルダを起動します。パスフレーズの質問と回答を指定します。

2.6.46 XML 属性の設定

XPath 式で選択された要素のセットに XML 属性を設定します。

フィールド

名前

XML 属性の名前を指定します。この名前には、前にこのポリシーで定義されているネームスペースプリフィックスを含めることができます。

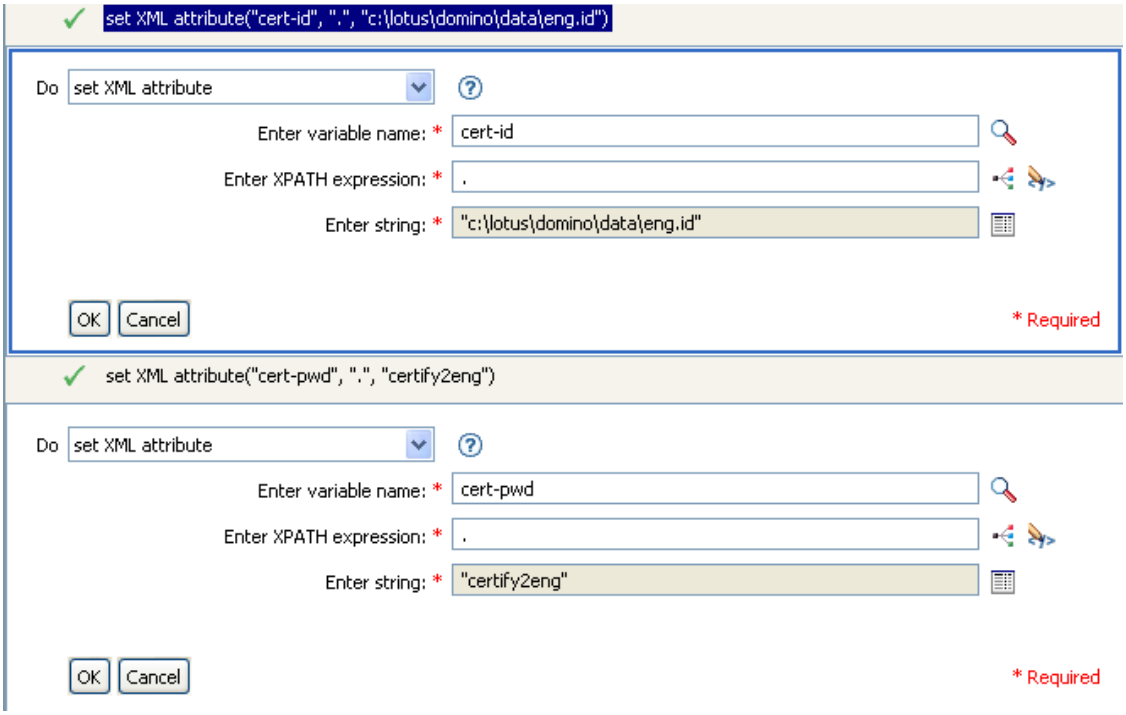
XPath 式

XML 属性の設定先になる要素を含むノードセットを返す XPath 1.0 式。

文字列

XML 属性の値を指定します。

例



2.6.47 ステータス

ステータス通知を生成します。

フィールド

レベル

通知のステータスレベルを指定します。

メッセージ

引数ビルダを使用してステータスメッセージを指定できます。

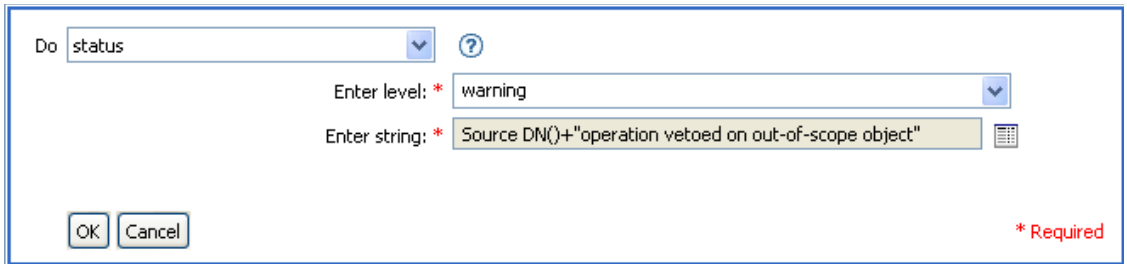
備考

レベルが「再試行」である場合、ポリシーは入力ドキュメントの処理をただちに中止して、現在処理中のイベントの再試行をスケジュールします。

レベルが「致命的エラー」である場合、ポリシーは入力ドキュメントの処理をただちに中止して、ドライバのシャットダウンを開始します。

現在の操作にイベント ID が割り当てられている場合、そのイベント ID がステータス通知に使用されます。割り当てられていない場合は、イベント ID はレポートされません。

例



2.6.48 操作属性のストリップ

現在の操作から属性に行ったすべての内容を除去します。

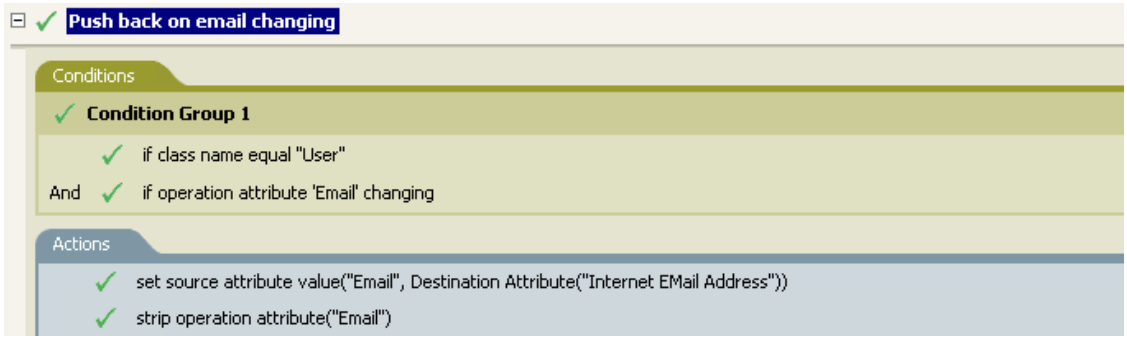
フィールド

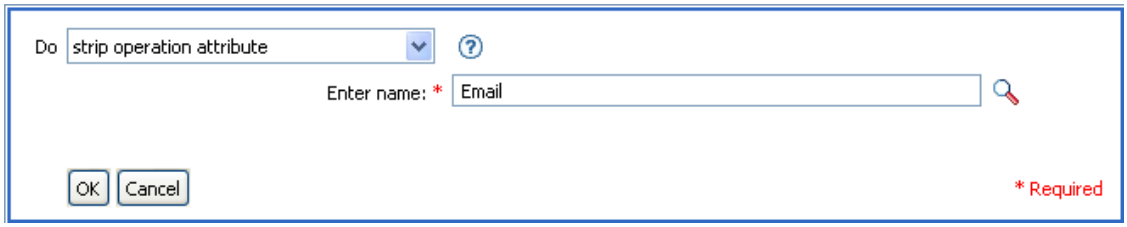
名前

除去する属性の名前を指定します。

例

この例では、電子メールアドレスが変更された場合に、この変更を元の状態に戻します。これは「Policy: Reset Value of the E-mail Attribute (ポリシー : 電子メール属性値のリセット)」という名前のポリシーで、Novell のサポート Web サイトからダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」 を参照してください。





このアクションでは、電子メールの属性を除去します。保持されている値は、ターゲットの電子メール属性内にあったものです。

2.6.49 XPath のストリップ

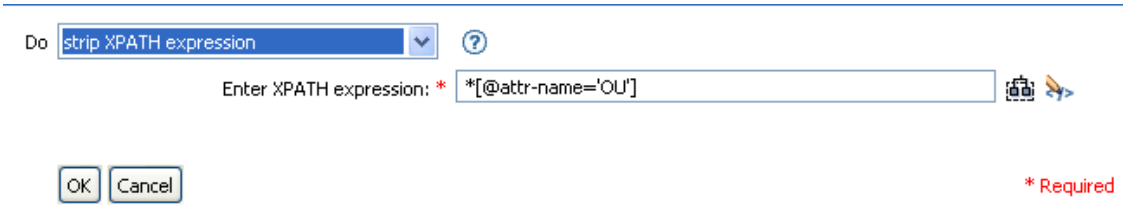
XPath 式で選択されたノードを除去します。

フィールド

XPath 式

除去されるノードを含むノードセットを返す XPath 1.0 の式を指定します。

例



2.6.50 メッセージのトレース

DSTRAC へメッセージを送信します。

フィールド

レベル

メッセージのトレースレベルを指定します。デフォルトのレベルは 0 です。メッセージは、指定したトレースレベルがドライバで設定されているトレースレベル以下である場合にのみ表示されます。

ドライバのトレースレベルの設定方法についての詳細は、『*Novell Identity Manager 3.0 管理ガイド*』の「Identity Manager のプロセスの表示」を参照してください。

色

トレースメッセージの色を選択します。

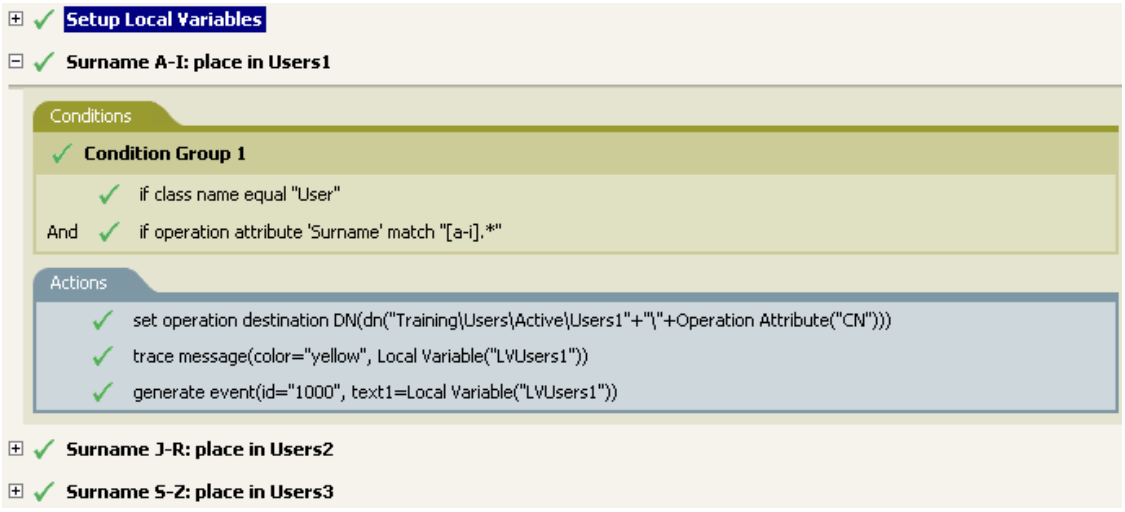
文字列

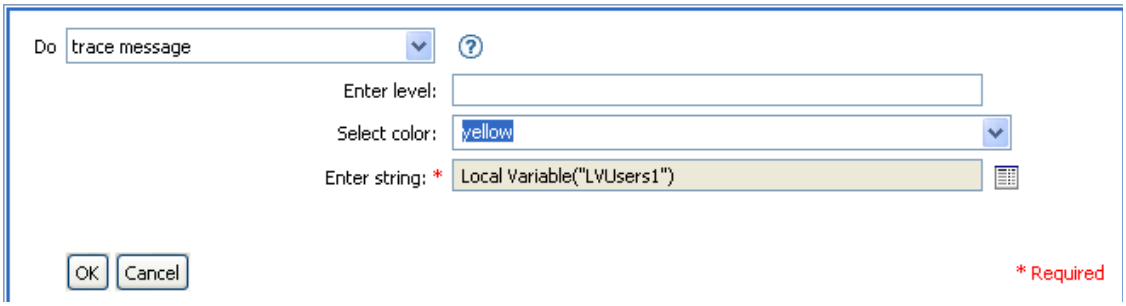
トレースメッセージの値を指定します。

例

この例には 4 つのルールがあり、これらのルールでは名字属性の最初の文字に基づいてユーザオブジェクトに配置ポリシーを実装し、トレースメッセージおよびカスタムの Novell Audit イベントの両方を生成します。メッセージのトレースアクションは、DSTRACE へのトレースメッセージを送信する場合に使用されます。これは、「Policy to Place by Surname (名字で配置するためのポリシー)」という名前のポリシーで、Novell の

サポート Web サイトからダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。





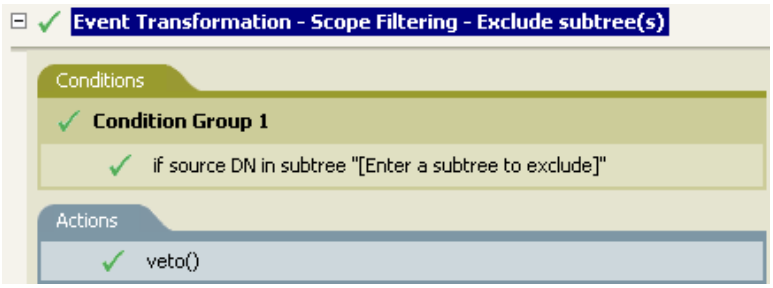
DSTRAC へトレースメッセージを送信します。ローカル変数の内容は LVUsers1 で、DSTRACE では黄色で表示されます。

2.6.51 拒否

現在の操作を拒否します。

例

この例では、指定されたサブツリーからのイベントをすべて除外します。このルールは、Identity Manager 3.0 に付属している事前定義されたルールです。詳細については、86 ページの 「イベント変換 - スコープフィルタリング - サブツリーの除外」を参照してください。



このアクションでは、指定されたサブツリーからのイベントをすべて拒否します。

2.6.52 操作属性値がない場合は拒否

現在の操作内の属性の使用状況に基づき、条件付きで現在の操作をキャンセルして現在のポリシーの処理を終了します。

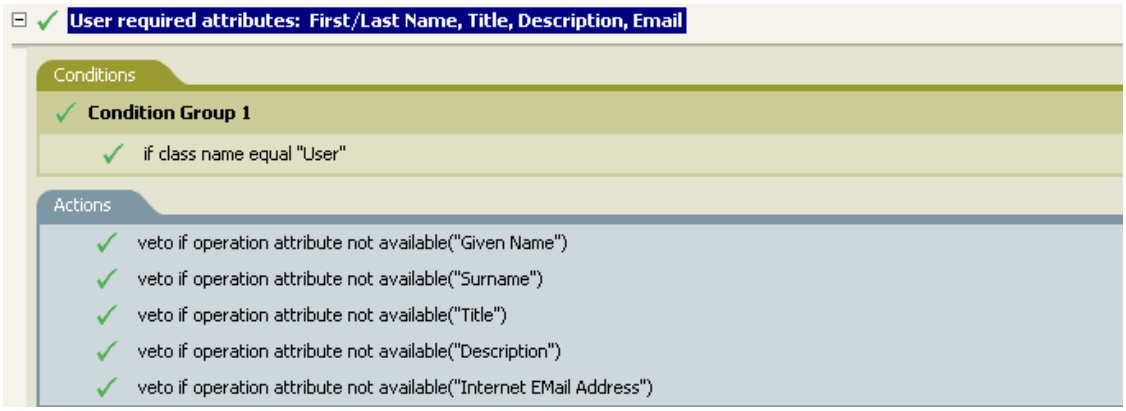
フィールド

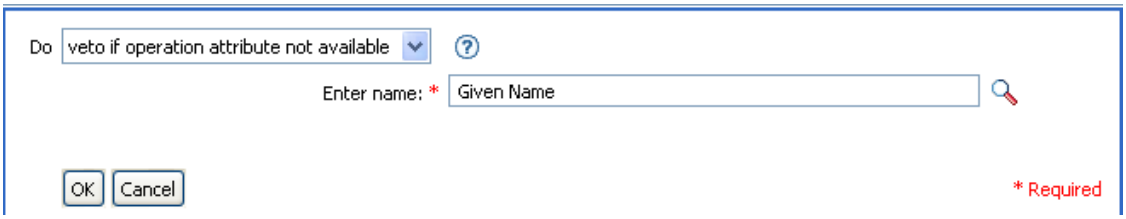
名前

属性の名前を指定します。

例

この例では、属性「名前」、「名字」、「役職」、「説明」、および「インターネット電子メールアドレス」が使用できない場合、ユーザオブジェクトは作成されません。これは「Policy to Enforce the Presences of Attributes (属性の存在を強制するポリシー)」という名前のポリシーで、Novell のサポート Web サイトからダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。





このアクションでは、属性「名前」、「名字」、「役職」、「説明」、「インターネット電子メールアドレス」が使用できない場合、操作を拒否します。

2.7 名詞トークン

この節では、引数ビルダインタフェースで使用できるすべての名詞トークンについて、詳しく説明します。

2.7.1 追加されたエンタイトルメント

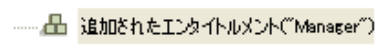
現在の操作で付与されたエンタイトルメントの値に展開します。

フィールド

名前

エンタイトルメントの名前。

例



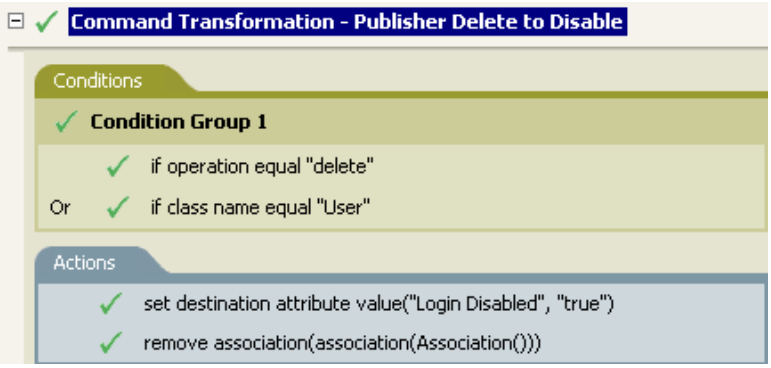
2.7.2 関連付け

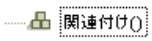
現在の操作から関連付けの値に展開します。

例

この例は、Identity Manager 3.0 に付属している事前定義されたルールからのものです。事前定義されたルールの詳細については、76 ページの 「コマンド変換 - 無効にする発行者の削除」を参照してください。

関連付けを削除するアクションでは、関連付けトークンを使用して、現在の操作から値を取得します。このルールでは、ユーザオブジェクトから関連付けを削除することで、新しいイベントが発生してもユーザオブジェクトに影響を与えないようにします。





2.7.3 属性

現在の操作およびソースデータストア内の現在のオブジェクトからの属性値に展開します。これは、論理的には、操作属性のトークンとソース属性のトークンの結合と考えることができます。変更操作で削除された値は含まれません。

フィールド

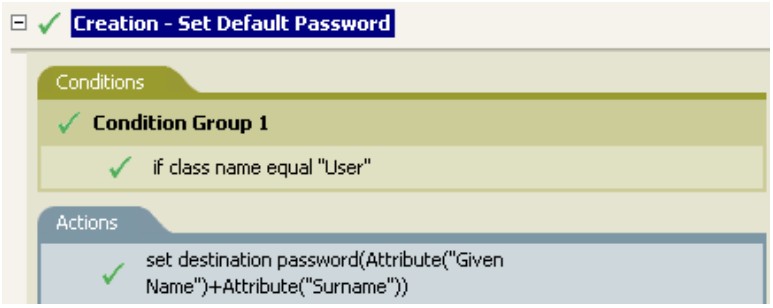
名前

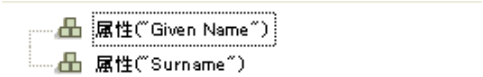
属性の名前を指定します。

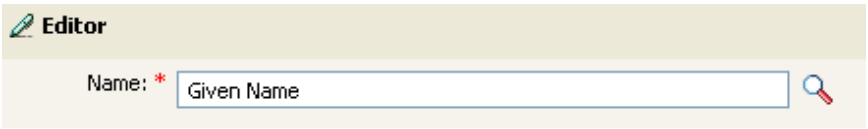
例

この例は、Identity Manager 3.0 に付属している事前定義されたルールからのものです。詳細については、83 ページの 「作成 - デフォルトパスワードの設定」を参照してください。

ターゲットパスワードの設定のアクションでは、属性トークンを使用してパスワードを作成します。パスワードは、名前属性と名字属性から作成されます。引数ビルダのエディタから、使用する属性を参照して選択します。



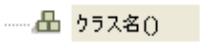




2.7.4 クラス名

現在の操作からオブジェクトクラス名に展開します。

例



2.7.5 ターゲット属性

ターゲットデータストアの現在のオブジェクト、DN、または関連付けの指定した属性値に展開します。

フィールド

クラス名

(オプション) ターゲットオブジェクトのクラス名を指定します。現在のオブジェクトのクラス名を使用するには、空白のままにします。

名前

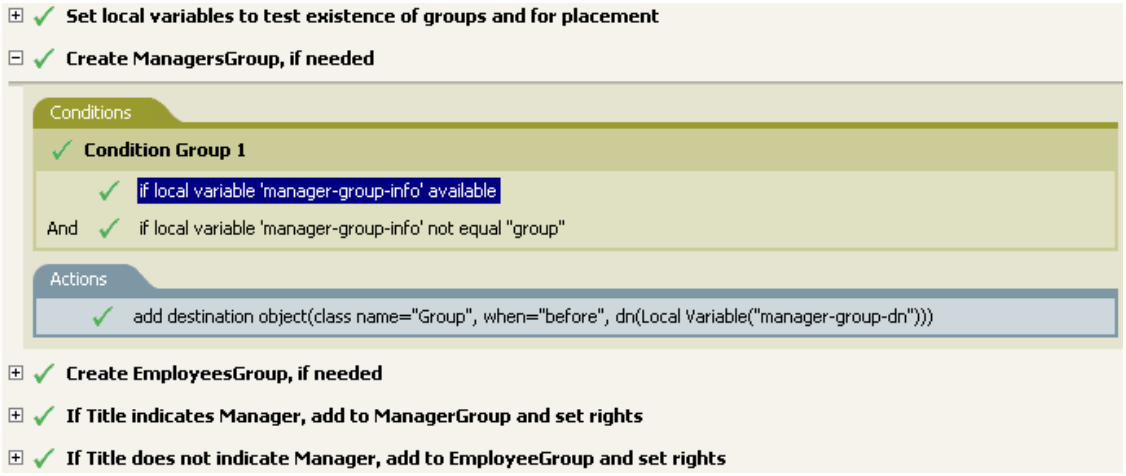
属性の名前。

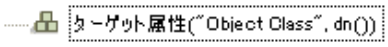
例

この例は「Govern Groups for User Based on Title (役職名に基づくユーザグループの管理)」ポリシーからのもので、Novell のサポート Web サイトからダウンロードできます。詳細

については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。

このポリシーでは、引数ビルダを使用してターゲット属性を作成します。ローカル変数の設定のアクションには、ターゲット属性のトークンが含まれています。





ターゲット属性はエディタを使用して作成します。この例では、オブジェクトクラスの属性が設定されます。DN は、ターゲットオブジェクトの選択に使用されます。DN の値は、ローカル変数 manager-group-dn です。

2.7.6 ターゲット DN

現在の操作からターゲット DN に展開します。

フィールド

変換

DN をソースデータストアで使用される形式に変換するかどうかを選択します。

開始

開始の RDN インデックスを指定します。

- インデックス 0 はルートに最も近い RDN
- 正のインデックスはルートに最も近い RDN からのオフセット

- インデックス -1 はリーフに最も近いセグメント
- 負のインデックスは、リーフに最も近い RDN からルートに最も近い RDN 方向へのオフセット

長さ

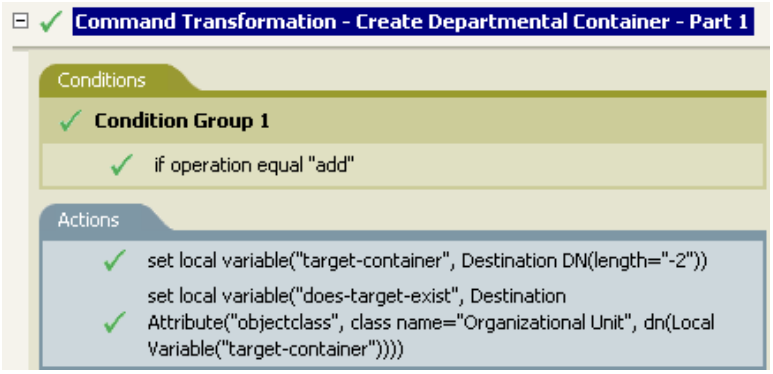
含める RDN の数を指定します。負の数は (セグメント総数 + 長さ) +1 のように解釈されます (たとえば、セグメント数が 5 の DN では、長さが -1 の場合は -1 = (5 + (-1)) + 1 = 5、長さが -2 の場合は -2 = (5 + (-2)) + 1 = 4)。

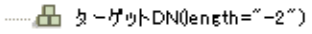
備考

「開始」または「長さ」がデフォルト値 {0、-1} に設定されている場合は、DN 全体が使用されます。それ以外の場合は、「開始」および「長さ」で指定された DN の部分が使用されます。

例

この例では、ターゲット DN のトークンを使用して、ローカル変数 target-container の値を設定します。このポリシーでは、ユーザオブジェクトの部署別コンテナがない場合に、そのコンテナを作成します。このポリシーは、Identity Manager 3.0 に付属している事前定義されたルールからのものです。詳細については、74 ページの 「コマンド変換 - 部署別コンテナの作成 - パート 1 とパート 2」を参照してください。

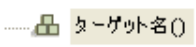




2.7.7 ターゲット名

現在の操作から指定されたターゲット DN の非修飾の相対識別名 (RDN) に展開します。

例



2.7.8 エンタイトルメント

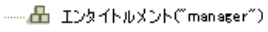
現在のオブジェクトから付与されたエンタイトルメントの値に展開します。

フィールド

名前

エンタイトルメントの名前を指定します。

例



2.7.9 グローバル構成値

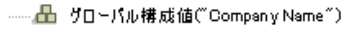
グローバル構成値の値に展開します。

フィールド

名前

グローバル構成値の名前。

例



2.7.10 ローカル変数

ローカル変数の値に展開します。

フィールド

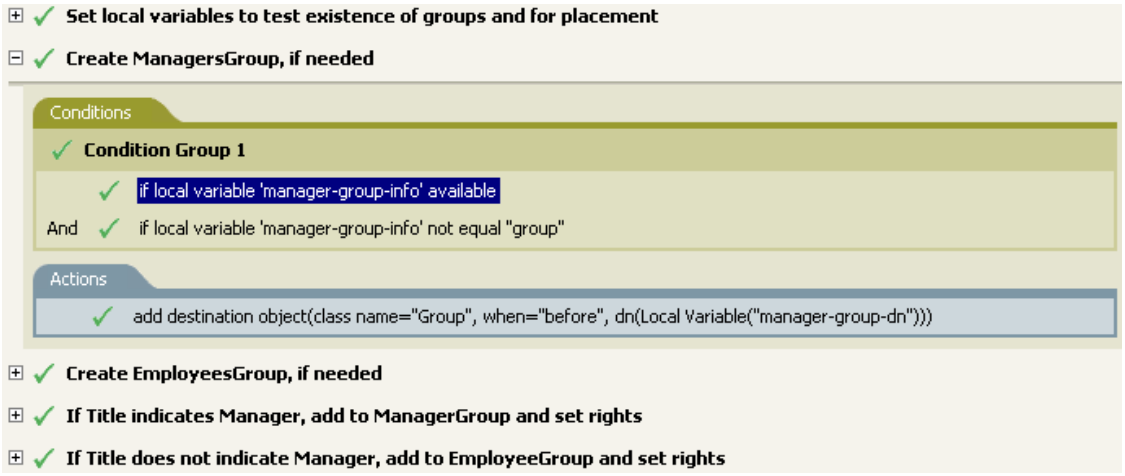
名前

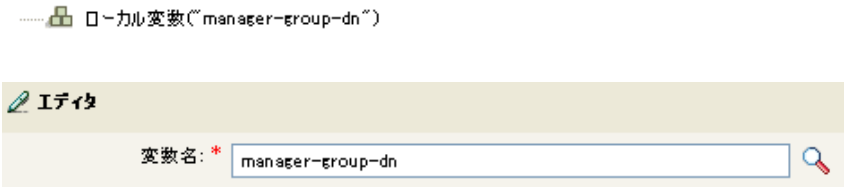
ローカル変数の名前を指定します。

例

この例は「Govern Groups for User Based on Title (役職名に基づくユーザグループの管理)」ポリシーからのもので、Novell のサポート Web サイトからダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。

ターゲットオブジェクトの追加アクションでは、ローカル変数のトークンを使用します。





ローカル変数は、ローカル変数の設定アクションがポリシーで以前使用されていた場合にのみ使用されます。ローカル変数に保存される値を設定します。エディタで、参照アイコンをクリックすると、定義済みのすべてのローカル変数がリストされます。正しいローカル変数を選択します。

ローカル変数の値は、group-manager-dn です。これは 1 つ前のルール、マネージャのグループ Users\ManagersGroup の DN として group-manager-dn が定義されたローカル変数の設定アクションです。

2.7.11 名前付きパスワード

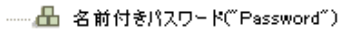
ドライバの名前付きパスワードに展開します。

フィールド

名前

パスワードの名前を指定します。

例



2.7.12 操作

現在の操作の名前に展開します。

例

2.7.13 操作属性

現在の操作から属性の値に展開します。変更操作で削除された値は含まれません。

フィールド

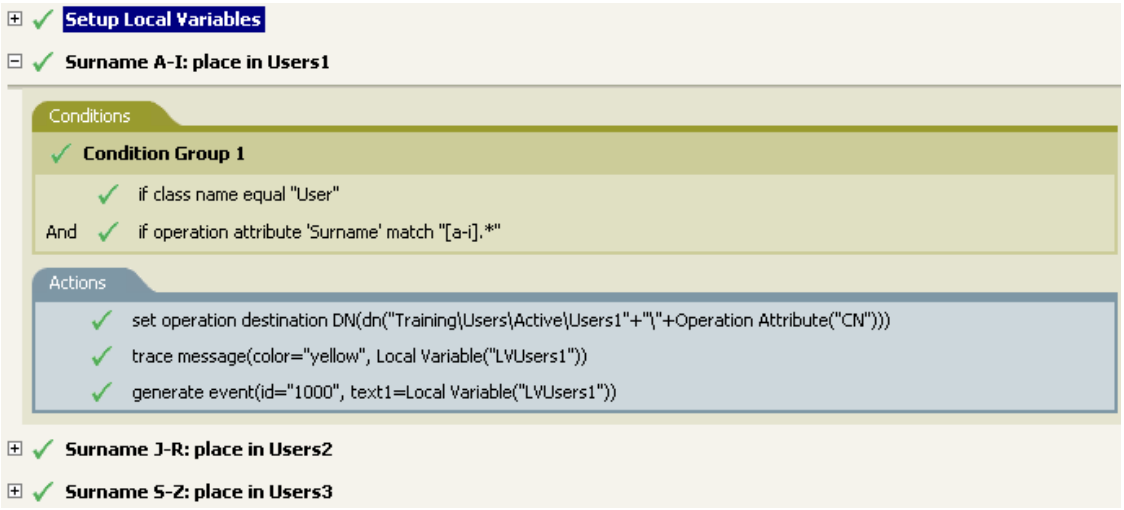
名前

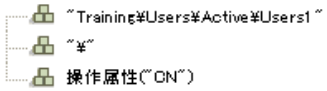
属性の名前を指定します。

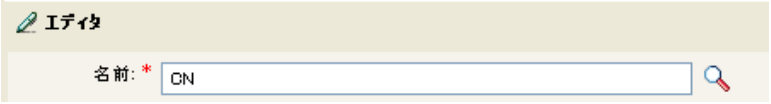
例

この例には 4 つのルールがあり、これらのルールでは名字属性の最初の文字に基づいてユーザオブジェクトに配置ポリシーを実装し、トレースメッセージおよびカスタムの Novell Audit イベントの両方を生成します。これは、「Policy to Place by Surname (名字で配置するためのポリシー)」という名前のポリシーで、Novell のサポート Web サイトからダ

ウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。







操作ターゲット DN の設定アクションには、操作属性のトークンが含まれています。操作属性のトークンは、ターゲット DN を CN 属性に設定します。このルールでは、Training\Users\Active\Users のコンテキストを取得して、\ および CN 属性の値を追加します。

2.7.14 操作プロパティ

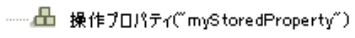
現在の操作から操作プロパティの値に展開します。

フィールド

名前

操作プロパティの名前を指定します。

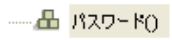
例



2.7.15 パスワード

現在の操作からパスワードに展開します。

例



2.7.16 削除された属性

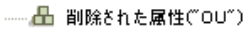
現在の操作で削除されている属性の値に展開します。変更操作の場合にのみ適用されます。

フィールド

名前

属性の名前を指定します。

例



2.7.17 削除されたエンタイトルメント

現在の操作で取り消されたエンタイトルメントの値に展開します。

フィールド

名前

エンタイトルメントの名前を指定します。

例

2.7.18 ソース属性

ソースデータストア内の 1 つのオブジェクトからの属性値に展開します。

フィールド

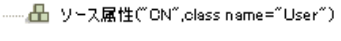
クラス名

(オプション) ターゲットオブジェクトのクラス名を指定します。現在のオブジェクトのクラス名を使用するには、空白のままにします。

名前

属性の名前。

例



2.7.19 ソース DN

現在の操作からソース DN に展開します。

フィールド

変換

DN をターゲットデータストアで使用される形式に変換するかどうかを選択します。

開始

開始の RDN インデックスを指定します。

- インデックス 0 はルートに最も近い RDN
- 正のインデックスはルートに最も近い RDN からのオフセット
- インデックス -1 はリーフに最も近いセグメント
- 負のインデックスは、リーフに最も近い RDN からルートに最も近い RDN 方向へのオフセット

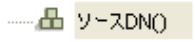
長さ

含める RDN のセグメントの数です。負の数は (セグメント総数 + 長さ) +1 のように解釈されます (たとえば、セグメント数が 5 の DN では、長さが -1 の場合は -1 = (5 + (-1)) + 1 = 5、長さが -2 の場合は -2 = (5 + (-2)) + 1 = 4)。

備考

「開始」または「長さ」がデフォルト値 {0、-1} に設定されている場合は、DN 全体が使用されます。それ以外の場合は、「開始」または「長さ」で指定された DN の一部分が使用されます。

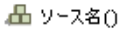
例



2.7.20 ソース名

現在の操作からソース DN の非修飾の相対識別名 (RDN) に展開します。

例



2.7.21 テキスト

テキストに展開します。

フィールド

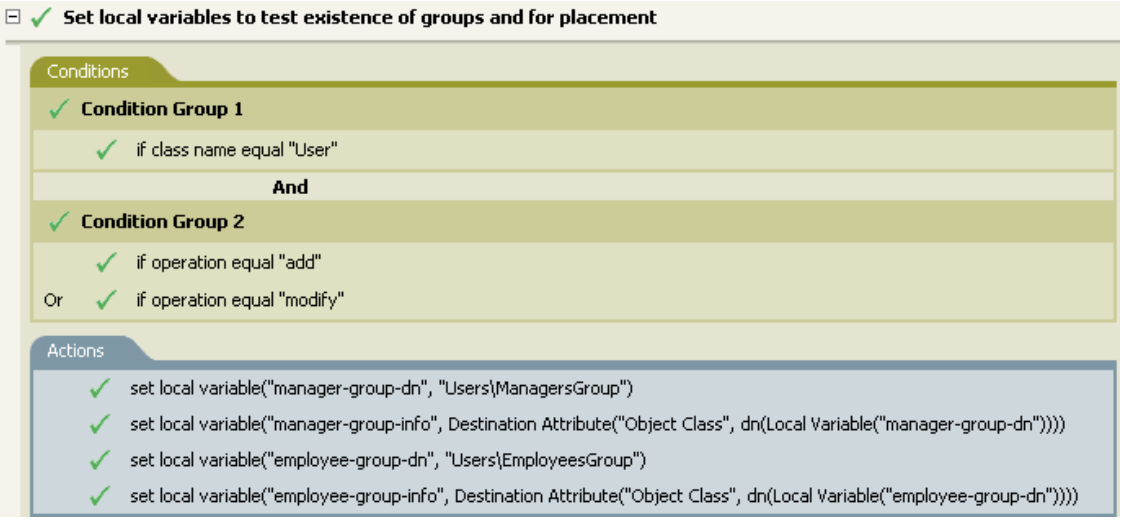
テキスト

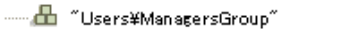
テキストを指定します。

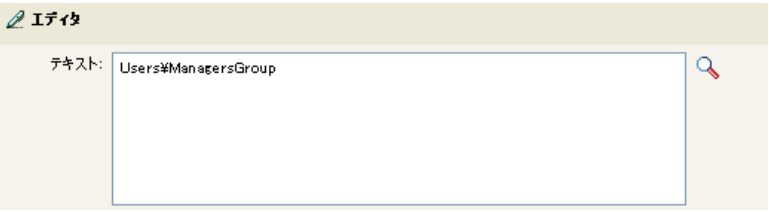
例

この例は「Govern Groups for User Based on Title (役職名に基づくユーザグループの管理)」ポリシーからのもので、Novell のサポート Web サイトからダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。

テキストトークンは、マネージャのグループの DN を定義するため、ローカル変数の設定アクションで使用されます。テキストトークンには、オブジェクトまたはプレーンテキストを含められます。







テキスト名詞には、マネージャのグループの DN が含まれます。使用するオブジェクトを参照するか、またはエディタに情報を入力します。

2.7.22 一意の名前

指定された条件に従って、ターゲットデータストアで一意の、パターンに基づいた名前に展開します。

フィールド

名前

一意性をチェックする属性の名前を指定します。

スコープ

一意性をチェックするスコープを指定します。

検索の開始

検索を開始するポイントを選択します。開始ポイントは、データストアのルートにするか、DN で指定するか、または関連付けにすることができます。

パターン

引数ビルダを使用して一意の値を生成する場合に使用するパターンを指定します。

カウンタの開始

一意の名前を検索する必要がある場合に使用する、カウンタを開始する数値を指定します。

桁

カウンタの桁数を指定します。デフォルトは 1 です。桁数に満たない値の場合、桁数が一致するように値の前に「0」のカウンタが付加されます。たとえば、3 桁を指定すると、1 桁の値には 001、002 などのように 0 が付加されます。

備考

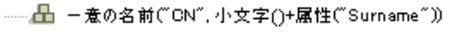
指定されたパターンごとに、ターゲットデータストアに対してのクエリが実行されます。このとき、指定された属性名、スコープおよび検索の開始値が使用されます。指定された各パターンは、見つかったオブジェクトを返さない値が検出されるまで、順に試行されます。

指定されたパターンがすべてなくなった場合は、最後のパターンにカウンタが追加され、クエリがインスタンスを返さなくなるまで、そのパターンが繰り返し試行されます (カウンタが毎回増えます)。

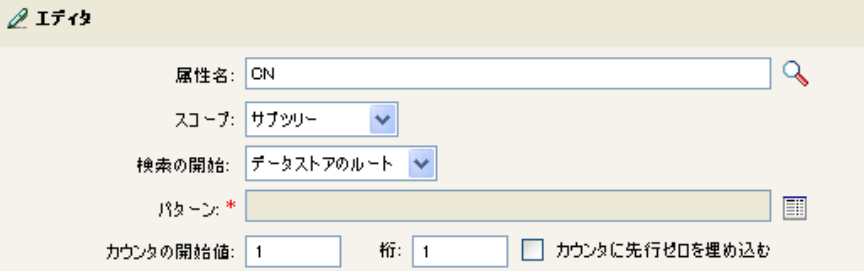
開始番号として別の番号を設定するには、［カウンタの開始］フィールドを使用します。カウンタは、［桁］フィールドで指定された桁数を使用します。桁数が指定された桁数より少ない場合、カウンタは右に詰められ、0 でパディングされます。桁数が指定された桁数より多い場合、一意の名前は生成されず、トークンで指定しているルールがエラーステータスを返します。

ターゲットデータストアがアイデンティティボールトであり、［名前］フィールドが空白のままである場合は、擬似属性「[Entry].rdn」に対して検索が実行されます。これは、命名属性が何であるかにかかわらず、オブジェクトの RDN を示します。ターゲットデータストアが接続アプリケーションの場合、［名前］フィールドは必須です。

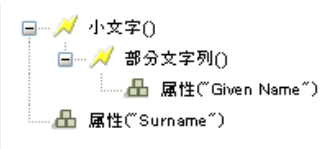
例



次に示すのは、一意の名前引数を作成するときの［エディタ］ペインの例です。



次のパターンは、一意の名前を提供するために作成されました。



このパターンで一意の名前を生成しない場合は、数値が 1 つ追加され、指定された桁数になるまで増分されます。この例では、エラーが発生するまで、数字を追加することで一意の名前が 9 つ生成されます (パターン 1 からパターン 9)。

2.7.23 一致しないソース DN

「ソース DN」条件の条件との最後の検索で一致しなかった DN の一部に対応する、現在の操作に含まれるソース DN の一部分に展開します。

フィールド

変換

DN をターゲットデータストアで使用される形式に変換するかどうかを選択します。

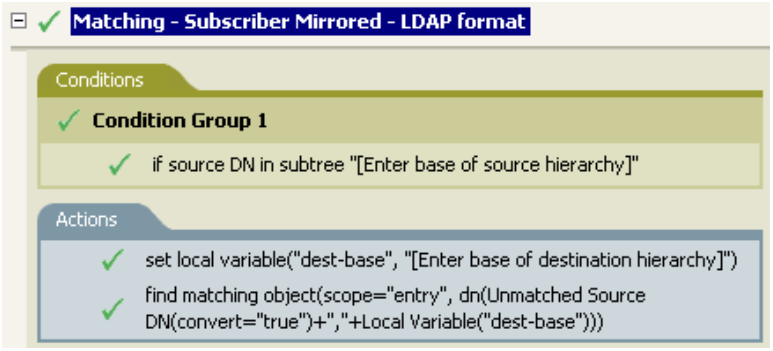
備考

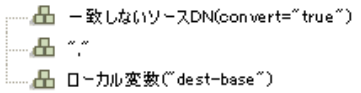
一致するものがなかった場合は、DN 全体が使用されます。

例

この例は、Identity Manager 3.0 に付属している事前定義されたルールからのものです。詳細については、92 ページの 「一致 - 購読者 (ミラーリング) - LDAP 形式」を参照してください。

一致オブジェクトの検索アクションでは、一致しないソース DN トークンを使用して、一致情報を LDAP 形式で作成します。ソース DN の一致しなかった部分を使用して、一致作業を行います。





2.7.24 XPath

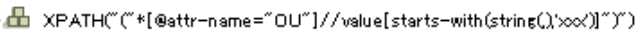
XPath 1.0 の式の評価結果に展開します。

フィールド

式

評価する XPath 1.0 の式を指定します。

例



2.8 動詞トークン

この節では、引数ビルダインタフェースで使用できるすべての動詞トークンについて、詳しく説明します。

- 205 ページのセクション 2.8.1「ターゲット DN のエスケープ」
- 205 ページのセクション 2.8.2「ソース DN のエスケープ」
- 206 ページのセクション 2.8.3「小文字」
- 206 ページのセクション 2.8.4「DN の解析」
- 208 ページのセクション 2.8.5「すべて置換」
- 209 ページのセクション 2.8.6「最初を置換」

- 210 ページのセクション 2.8.7「部分文字列」
- 211 ページのセクション 2.8.8「大文字」

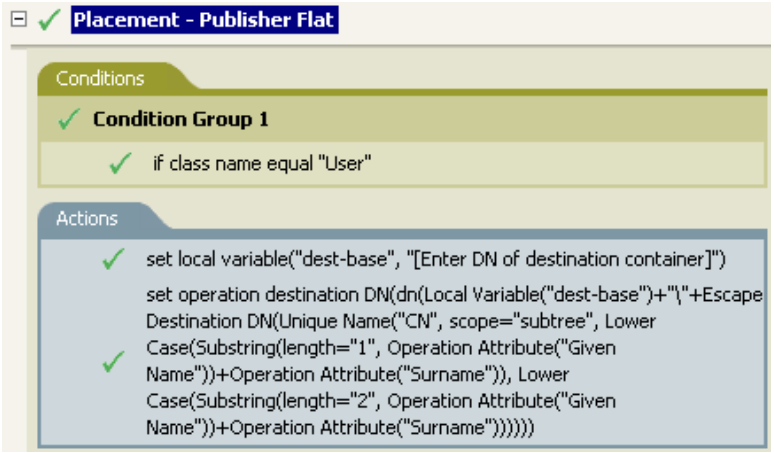
2.8.1 ターゲット DN のエスケープ

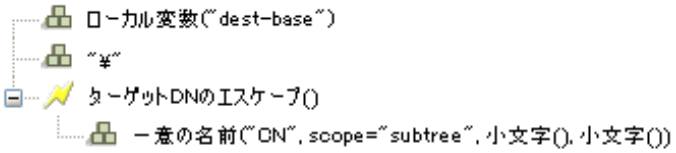
ターゲットデータストアの DN フォーマットのルールに従って文字列をエスケープします。

例

この例は、Identity Manager 3.0 に付属している事前定義されたルールからのものです。詳細については、99 ページの 「配置 - 発行者 (フラット)」を参照してください。

操作ターゲット DN の設定アクションでは、ターゲット DN のエスケープトークンを使用して、ユーザオブジェクトのターゲット DN を作成します。



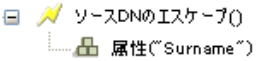


ターゲット DN のエスケープでは、一意の名前の値を取得して、これをターゲット DN の形式に設定します。

2.8.2 ソース DN のエスケープ

ソースデータストアの DN フォーマットのルールに従って文字列をエスケープします。

例

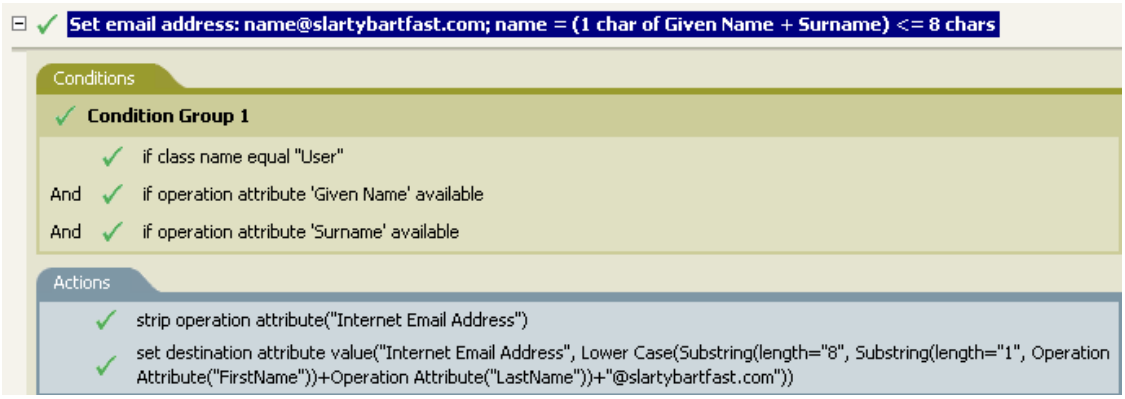


2.8.3 小文字

文字列内の文字を小文字に変換します。

例

この例では、電子メールアドレスを「name@slartybartfast.com」に設定します。name の部分は、名前と名字の最初の文字になります。ポリシーが表示されます。Create E-mail from Given Name and Surname, and it is available for download at Novell's support Web site. 詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。



小文字トークンは、ターゲット属性値の設定アクションの情報を、すべて小文字に設定します。

2.8.4 DN の解析

DN を別の形式に変換します。

フィールド

開始

開始の RDN インデックスを指定します。

- インデックス 0 はルートに最も近い RDN
- 正のインデックスはルートに最も近い RDN からのオフセット
- インデックス -1 はリーフに最も近いセグメント
- 負のインデックスは、リーフに最も近い RDN からルートに最も近い RDN 方向へのオフセット

長さ

含める RDN の数です。負の数は (セグメント総数 + 長さ) +1 のように解釈されます (たとえば、セグメント数が 5 の DN では、長さが -1 の場合は -1 = (5 + (-1)) + 1 = 5、長さが -2 の場合は -2 = (5 + (-2)) + 1 = 4)。

ソース **DN** のフォーマット

ソース DN の解析に使用されるフォーマットを指定します。

ターゲット **DN** のフォーマット

解析された DN の出力に使用されるフォーマットを指定します。

ソース **DN** 区切り文字

ソース DN のフォーマットが［カスタム］に設定されている場合に、カスタムのソース DN 区切り文字を指定します。

ターゲット **DN** 区切り文字

ターゲット DN のフォーマットが［カスタム］に設定されている場合に、カスタムのターゲット DN 区切り文字を指定します。

備考

「開始」または「長さ」がデフォルト値 {0、-1} に設定されている場合は、DN 全体が使用されます。それ以外の場合は、「開始」または「長さ」で指定された DN の一部分が使用されます。

カスタムの DN フォーマットを指定する場合、区切り文字を構成する 8 文字は次のように定義されます。

1. タイプ付きの名前のブールフラグ : 0 は名前がタイプなし、1 はタイプ付きであることを示します。

2. Unicode No-Map 文字のブールフラグ :0 は、マップ不可能な Unicode 文字を、出力しない、または ¥FEFF などのエスケープ処理された 16 進数字の文字列として変換しないことを意味します。eDirectory では、Unicode 文字の 0xfeff、0xfffe、0xfffd、および 0xffff は使用できません。

3. 相対 RDN 区切り文字

4. RDN 区切り文字

5. 名前ディバイダ

6. 名前の値の区切り文字

7. ワイルドカード文字

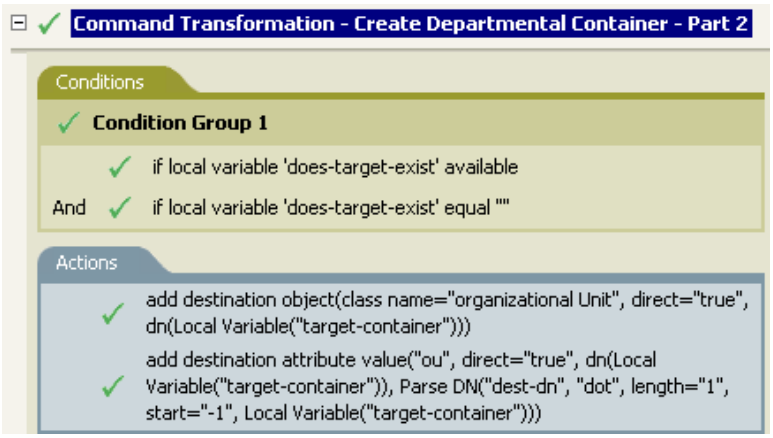
8. エスケープ文字

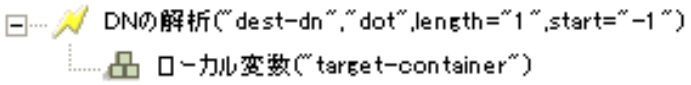
RDN 区切り文字と相対 RDN 区切り文字が同じ文字である場合、名前の向きは右から左、それ以外の場合は左から右になります。

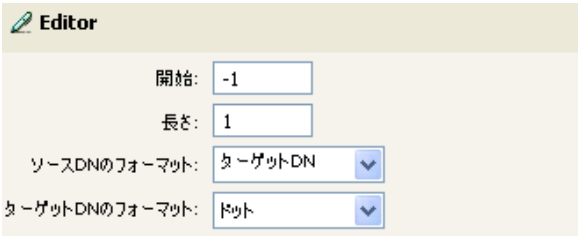
区切り文字セットが 8 文字を超える場合、超過した文字はエスケープ処理が必要な文字と見なされるだけで、それ以外の特別な意味は考慮されません。

例

この例では、DN の解析トークンを使用して、ターゲット属性値の追加アクションの値を作成します。この例は、Identity Manager 3.0 に付属している事前定義されたルールからのものです。詳細については、74 ページの 「コマンド変換 - 部署別コンテナの作成 - パート 1 とパート 2」を参照してください。







DN の解析トークンは、ソース DN から情報を取得し、これをドット表記に変更します。DN の解析からの情報は、OU の属性値に保存されます。

2.8.5 すべて置換

文字列内の正規表現と一致したものをすべて置換します。

フィールド

正規表現

置換される部分文字列と一致させる正規表現を指定します。

置換文字列

置換する文字列を指定します。

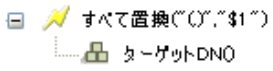
備考

正規表現の作成についての詳細は、次を参照してください。

- Sun の Java Web サイト (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html)
- Sun の Java Web サイト (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Matcher.html#replaceAll (java.lang.String))

［パターン］のオプションには CASE_INSENSITIVE、DOTALL、および UNICODE_CASE が使用されますが、適切な埋め込みエスケープを使用して逆の意味を指定することができます。

例



2.8.6 最初を置換

文字列内の正規表現と最初に一致したものを置換します。

フィールド

正規表現

置換される部分文字列と一致させる正規表現を指定します。

置換文字列

置換する文字列を指定します。

備考

一致したインスタンスは、［置換文字列］フィールドで指定された値で指定された文字列に置き換えられます。

正規表現の作成についての詳細は、次を参照してください。

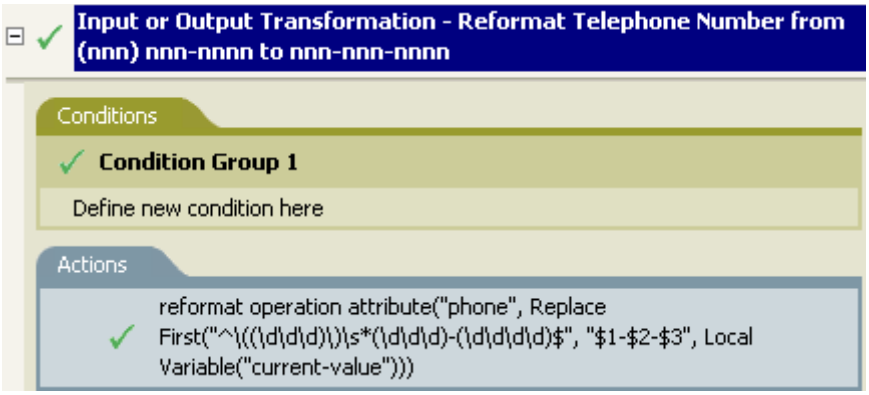
- http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html)
- http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Matcher.html#replaceAll (java.lang.String) (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Matcher.html#replaceAll (java.lang.String))

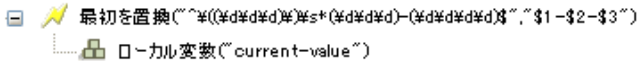
［パターン］のオプションには CASE_INSENSITIVE、DOTALL、および UNICODE_CASE が使用されますが、適切な埋め込みエスケープを使用して逆の意味を指定することができます。

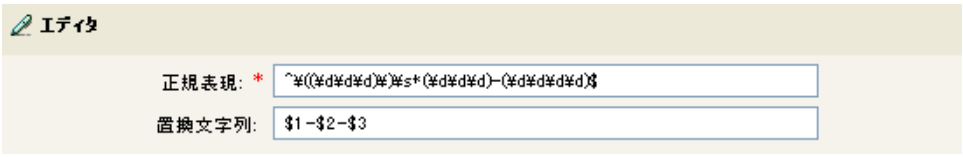
例

この例では、電話番号 (nnn)-nnn-nnnn を nnn-nnn-nnnn を再フォーマットします。このルールは、Identity Manager 3.0 に付属している事前定義されたルールです。詳細については、88 ページの 「入出力変換 - 電話番号の形式を (nnn) nnn-nnnn から nnn-nnn-nnnn に変更」を参照してください。

［最初を置換］トークンは、［操作属性の再フォーマット］アクションで使用されます。







正規表現 ^\(((\d\d\d)\)\s*(\d\d\d)-(\d\d\d\d)$ は、(nnn) nnn-nnnn を、正規表現 $1-$2-$3 は nnn を示しています。このルールでは、電話番号の形式を (nnn) nnn-nnnn から nnn-nnn-nnnn に変更します。

2.8.7 部分文字列

文字列の一部を抽出します。

フィールド

開始

開始文字のインデックスを指定します。

- インデックス 0 は 1 文字目です。
- 正のインデックスは文字列の先頭からのオフセットです。
- インデックス -1 は最後の文字です。
- 負のインデックスは、最後の文字から文字列の先頭方向へのオフセットです。

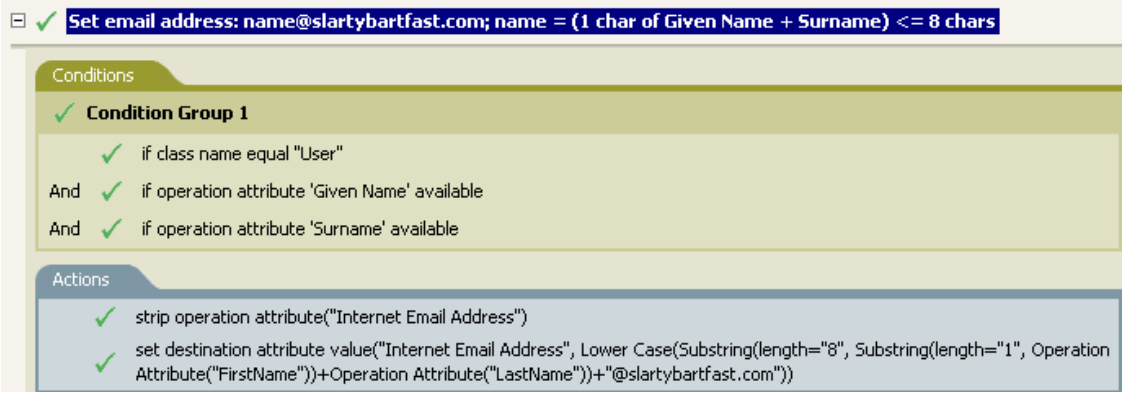
たとえば、開始が -2 に設定されると、最後の文字から読み込みが開始されます。-3 が指定されると、最後から 2 文字目で開始されます。

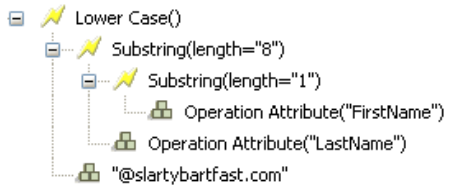
長さ

部分文字列に含める、開始位置からの文字数。負の数は (文字総数 + 長さ) + 1 のように解釈されます。たとえば、-1 の場合は全長または元の文字列を表します。-2 が指定されると、「全体の長さ -1」になります。5 文字の文字列の場合、長さが -1 の場合は -1= (5 + (-1)) + 1 = 5、長さが -2 の場合は -2 = (5 + (-2)) + 1 = 4 になります。

例

この例では、電子メールアドレスを「name@slartybartfast.com」に設定します。name の部分は、名前と名字の最初の文字になります。これは「Policy: Create E-mail from Given Name and Surname (ポリシー : 名前と名字から電子メールを作成)」という名前のポリシーで、Novell のサポート Web サイトでダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。





部分文字列トークンは、ターゲット属性値の設定アクションで 2 度使用されます。名前属性の最初の文字列を取得し、名字属性の 8 文字を追加して、1 つの部分文字列を作成します。

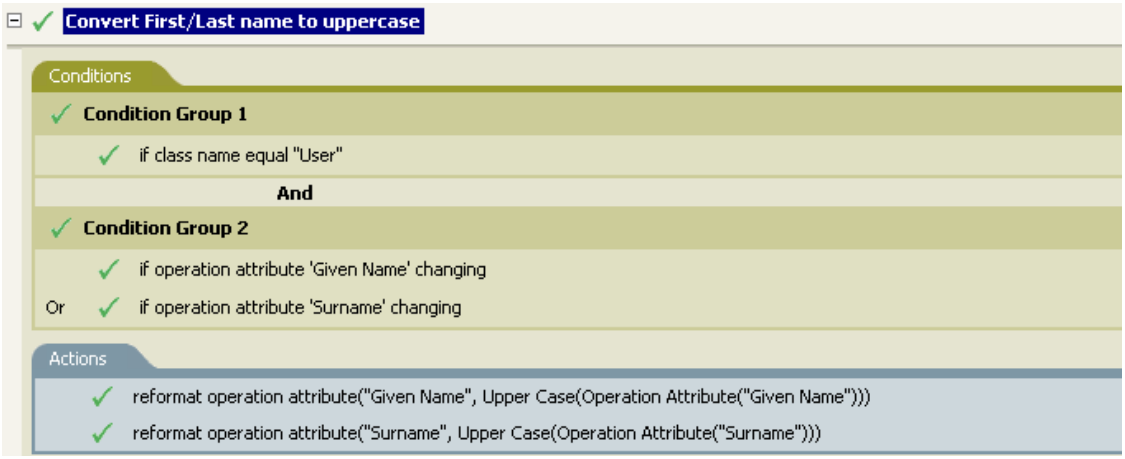
2.8.8 大文字

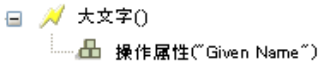
文字列内の文字を大文字に変換します。

例

この例では、ユーザオブジェクトの名前と名字の属性を大文字に変換します。これは「Policy: Convert First/Last Name to Upper Case (ポリシー : 名前と名字を大文字に変換)」と

いうポリシーで、Novell のサポート Web サイトでダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。





2.9 値

この節では、ポリシービルダに共通の値を一覧表示しています。

2.9.1 比較モード

表 2-7 比較モード

モード	説明
case (大文字と小文字の区別あり)	1 文字ずつ比較する (大文字と小文字の区別あり)。
nocase (大文字と小文字の区別なし)	1 文字ずつ比較する (大文字と小文字の区別なし)。

モード	説明
正規表現	文字列全体を正規表現で比較する。デフォルトでは大文字と小文字は区別されませんが、式でエスケープして変更できます。 Sun の Java Web サイト (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html) および Sun の Java Web サイト (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Matcher.html#matches()) を参照してください。 ［パターン］のオプションには CASE_INSENSITIVE、DOTALL、および UNICODE_CASE が使用されますが、適切な埋め込みエスケープを使用して逆の意味を指定することができます。
src-dn (ソース DN)	ソースデータストアの DN のフォーマットに対する適切なセマンティックを使用して比較します。
dest-dn (ターゲット DN)	ターゲットデータストアの DN のフォーマットに対する適切なセマンティックを使用して比較します。
数字	数字を比較します。
octet (オクテット)	オクテット値 (Base64 でエンコード) で比較します。
構造	属性の構造構文の比較ルールに従って、構造属性を比較します。

iManagerのポリシービルダを使用したポリシーの定義

3

ポリシービルダは、接続システム間でのデータのやりとりを定義するポリシーを作成、および管理するための機能を完備したグラフィカルインタフェースです。

この節では、次のようなポリシーおよびポリシービルダの使用方法について説明します。

- 35 ページのセクション 2.1「ポリシー」
- 216 ページのセクション 3.2「iManager におけるポリシービルダーのタスク」

次の内容についても詳しく説明します。

- 252 ページのセクション 3.5「条件」
- 270 ページのセクション 3.6「アクション」
- 310 ページのセクション 3.7「名詞トークン」
- 324 ページのセクション 3.8「動詞トークン」

3.1 ポリシー

ポリシーの動作を理解するには、まず、ポリシーのコンポーネントを理解する必要があります。

- ポリシーは複数のルールで構成されています。
- ルールとは、定義したアクション (270 ページの「アクション」を参照) が実行されるために満たされていなければならない条件 (252 ページの「条件」を参照) のセットです。
- アクションは実行時に展開されるトークンから派生する動的な引数を持つことができます。
- トークンは、名詞 (310 ページの「名詞トークン」を参照) と動詞 (324 ページの「動詞トークン」を参照) の 2 つに分類できます。
  - 名詞トークンは現在の操作、ソースやターゲットのデータストア、または外部ソースなどから派生する値を展開します。
  - 動詞トークンは、そのトークンのサブオーディネイトにある他のトークンの連結された結果を変更します。
- 正規表現 (250 ページの「正規表現」を参照) および XPath 1.0 の式 (251 ページの「XPath 1.0 の式」を参照) は、一般的には、ポリシーに対し適した結果を作成するためにルールで使用されます。
- ポリシーとは XDS ドキュメント上で操作を実行するもので、その主な目的はドキュメントを調べて変更を加えることです。
- 操作とは XDS ドキュメント内の要素のことで、入力要素と出力要素の子になります。これらの要素は Novell' の nds.dtd の一部です。詳細については、「NDS DTD (http://developer.novell.com/ndk/doc/dirxml/dirxmlbk/ref/ndsdtd/index.html)」を参照してください。

- 通常、1 つの操作は 1 つのイベント、コマンドまたはステータスを表します。
- ポリシーは、操作ごとに個別に適用されます。ポリシーが各操作に順番に適用されるので、その操作が現在の操作になります。各ルールは現在の操作に順次適用されます。直前のルールによって実行されたアクションが原因で、ルールがそれ以降適用されなくなる場合を除き、すべてのルールが現在の操作に適用されます。
- ポリシーはドキュメント外のコンテキストを取得して、結果のドキュメントに反映されない副次的動作を発生させることもできます。

3.2 iManager におけるポリシービルダーのタスク

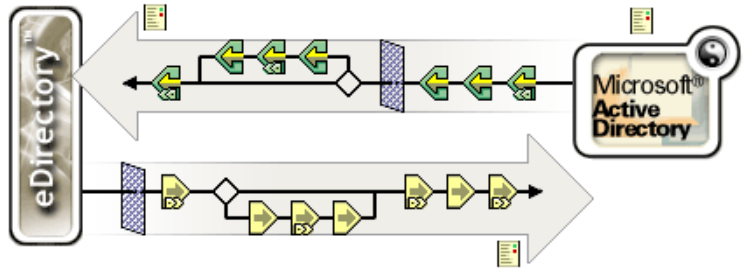
この節では、ポリシービルダの一般的なタスクを実行する手順を説明します。

- 216 ページのセクション 3.2.1「ポリシービルダの起動」
- 216 ページのセクション 3.2.2「ポリシーの作成」
- 226 ページのセクション 3.2.5「ポリシーの変更」
- 217 ページのセクション 3.2.3「ポリシー内での各ルールの定義」
- 219 ページのセクション 3.2.4「ルール内での各引数の定義」
- 228 ページのセクション 3.2.12「事前定義されたルールの使用」

3.2.1 ポリシービルダの起動

1. iManager で、［Identity Manager］役割を展開し、［Identity Manager の概要］をクリックします。
2. ドライバセットを指定します。
3. ポリシーを管理するドライバをクリックします。［Identity Manager ドライバの概要］が開きます。

図 3-1 *Identity Manager* ドライバの概要



ポリシーは、［Identity Manager ドライバの概要］から管理します。

3.2.2 ポリシーの作成

1. 管理するドライバの［Identity Manager ドライバの概要］を開きます。
2. 定義するポリシーを示しているアイコンをクリックします。
   は未定義のポリシーを示しています。
   は定義されたポリシーを示しています。

3 ［挿入］をクリックします。

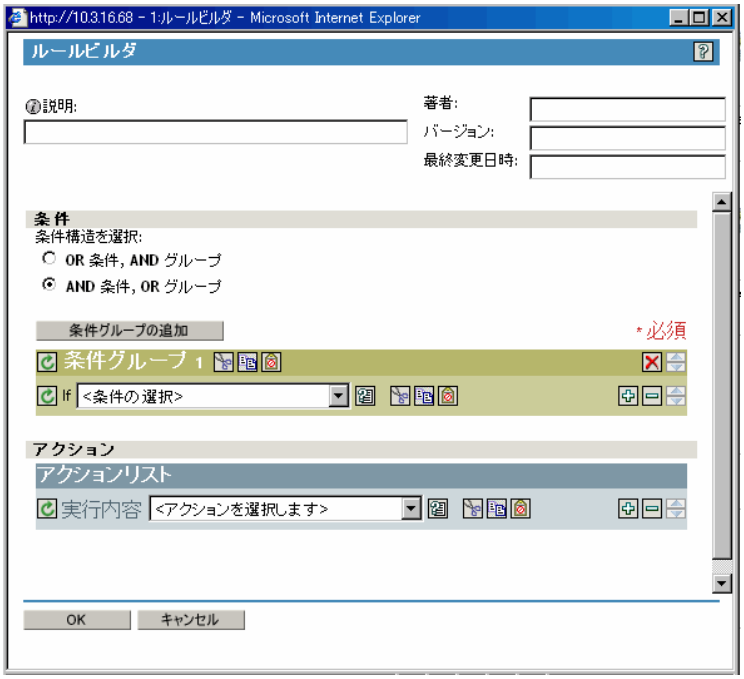
4 新しいポリシーの名前を入力し、［ポリシービルダ］を選択します。

5 ポリシーが表示されます。このポリシーに 1 つ以上のポリシーを定義するには、［新しいルールを追加する］をクリックし、217 ページのセクション 3.2.3「ポリシー内での各ルールの定義」の手順を実行します。

3.2.3 ポリシー内での各ルールの定義

ルールは、ポリシービルダの［ルールビルダ］ウィンドウで定義します。

図 3-2 ポリシービルダの［ルールビルダ］ウィンドウ



ルールビルダのインタフェースでは、インテリジェントなドロップダウンメニューを使用してルールをすばやく作成および変更できます。

ルールビルダでは、定義したアクションが実行されるために満たされていなければならない条件のセットを定義します。

たとえば、環境内に新しいオブジェクトを追加できないようにするルールを作成する場合は、次のように定義します。追加操作が発生すると、その操作を拒否します。

この論理をルールビルダで実装するには、次の条件を選択します。

図 3-3 ルールビルダインタフェースにおけるユーザの移動条件

次のアクションも選択します。

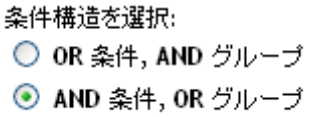
図 3-4 ルールビルダインタフェースにおける拒否アクション

252 ページのセクション 3.5「条件」および 270 ページのセクション 3.6「アクション」、またはルールビルダで参照できる条件とアクションの詳細を参照してください。

ヒント

より複雑な条件を作成するには、条件および条件グループと、AND/OR ステートメントを組み合わせることができます。これらの方法は、条件構造を選択し、組み合わせることで変更できます。

図 3-5 条件構造のラジオボタン



- フィールド値のリストを表示するには、アイコンをクリックします。前の例では、このアイコンをクリックすると有効なクラス名のリストが表示されます。
- 引数ビルダインタフェースを使用して引数を作成するには、アイコンをクリックします。
- ポリシー、ルール、条件、またはアクションを無効にするには、アイコンをクリックします。これらを再度有効にするには、アイコンをクリックします。
- ポリシーまたはルールにコメントを追加するには、アイコンをクリックします。コメントは、ポリシーまたはルールに直接保存され、必要な限り保持できます。
- ポリシービルダのクリップボードを使用するには、[切り取り]、[コピー]、または[貼り付け] アイコン を使用します。クリップボード内の現在の内容がその場所では使用できない場合、[貼り付け] アイコンは無効になります。
- 条件を追加、削除、位置付けるには、アイコンを使用します。
- 条件グループを追加するには、条件グループの追加 アイコンを使用します。
- 条件グループを削除し、位置付けるには、アイコンを使用します。

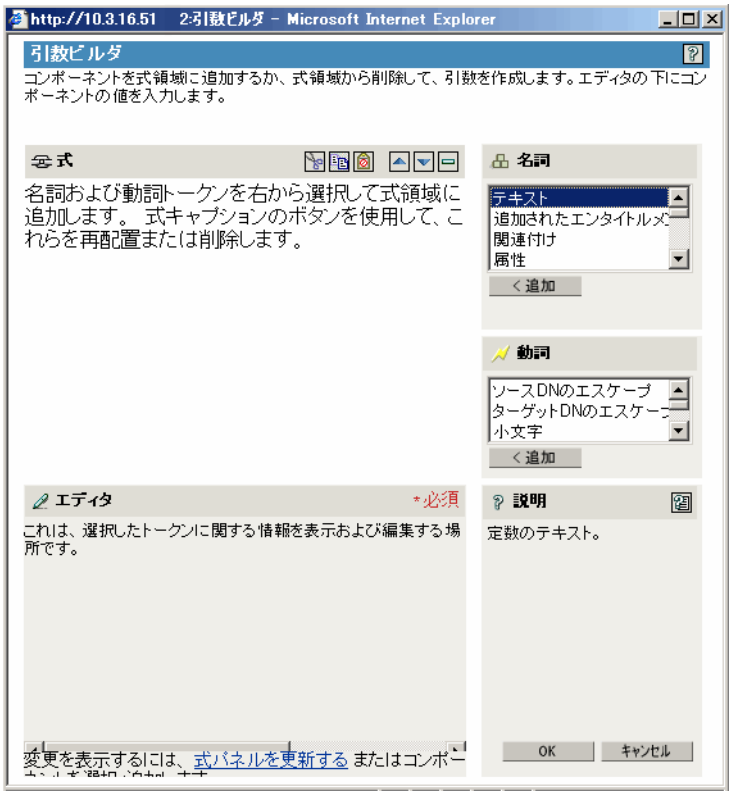
3.2.4 ルール内での各引数の定義

引数ビルダでは、動的なグラフィカルインタフェースによって、ルールビルダで使用する複雑な引数の式を作成できます。引数ビルダにアクセスするには、221 ページの 「引数ビルダ」を参照してください。

引数はアクションによって動的に使用されるもので、実行時に展開されるトークンから派生します。

トークンは、名詞と動詞の 2 つに分類できます。名詞トークンは、現在の操作、ソースやターゲットのデータストア、または外部ソースなどから派生する値を展開します。動詞トークンは、そのトークンのサブオーディネイトにある他のトークンの連結された結果を変更します。

図 *3-6* 引数ビルダのデフォルトインタフェース



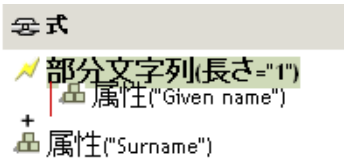
式を定義するには、値、オブジェクト、変数などの名詞トークンを 1 つ以上選択し、これらを「部分文字列」、「エスケープ」、「大文字」および「小文字」などの動詞トークンと組み合わせ、引数を作成します。複数のトークンを組み合わせることで、複雑な引数を作成できます。

たとえば、属性値に引数セットを指定する場合は、属性トークンを選択してから、属性名を選択します。

図 *3-7* テキスト引数として *ds.novell* を表示しているエディタ

この属性の一部だけを使用する場合は、属性トークンを「部分文字列」トークンと組み合わせます。

図 3-8 名前属性の長さが *1* の部分文字列を表示している式 *(* 名字属性との組み合わせ *)*



トークンの追加後は、エディタでそのトークンのフィールドを編集できます。

引数ビルダで使用できる名詞と動詞の詳細については、310 ページのセクション 3.7「名詞トークン」および 324 ページのセクション 3.8「動詞トークン」を参照してください。

ヒント

- より複雑な条件を作成するには、条件または条件グループと、AND/OR ステートメントを組み合わせることができます。
- 名詞トークンおよび動詞トークンを移動、または削除するには、  を使用します。
- フィールド値のリストを表示するには、  アイコンをクリックします。
- 名詞トークンまたは動詞トークンを追加した後は、エディタを使用して値を入力してから、次の名詞トークンまたは動詞トークンを追加します。変更を適用するために［式］ペインを更新する必要はなく、次の操作を実行すれば変更は反映されます。

ほとんどの引数は引数ビルダで定義できますが、ポリシービルダ内の条件エディタおよびアクションエディタで使用されるビルダが他にもいくつかあります。各ビルダは、次に示すどのビルダでも再帰的に呼び出すことができます。

- 220 ページの 「引数アクションビルダ」
- 221 ページの 「引数ビルダ」
- 222 ページの 「一致属性ビルダ」
- 223 ページの 「アクションの引数コンポーネントビルダ」
- 224 ページの 「引数値リストビルダ」
- 224 ページの 「名前付き文字列ビルダ」
- 225 ページの 「条件の引数コンポーネントビルダ」

引数アクションビルダ

引数アクションビルダにより、(283 ページ) For Each アクションおよび (287 ページ) エンタイトルメントの実装アクションで必要なアクションを設定できます。

次の例では、ターゲット属性値の追加アクションが Group エンタイトルメントごとに実行され、現在の操作に追加されます。

図 *3-9* 引数アクションビルダ

ターゲット属性値の追加アクションを定義するには、引数アクションビルダを起動するアイコンをクリックします。引数アクションビルダで、目的のアクションを定義します。次の例では、メンバー属性が、追加された各 Group エンタイトルメントのターゲットオブジェクトに追加されます。

図 *3-10* 引数アクションビルダ

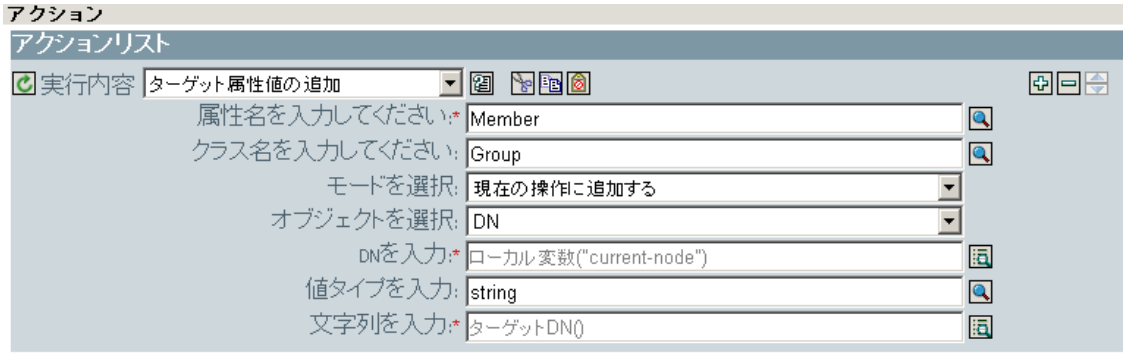
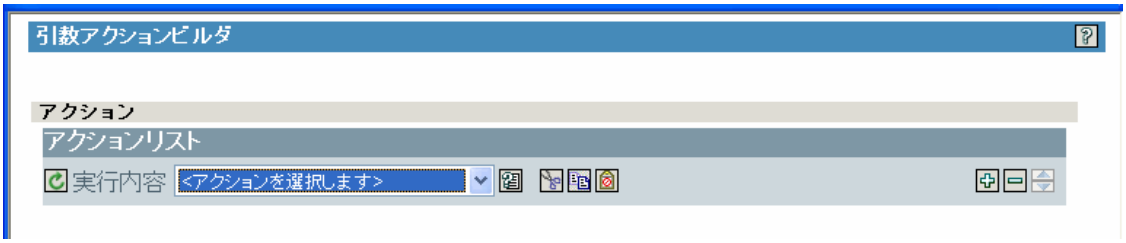


図 *3-11* 引数アクションビルダ



引数ビルダ

引数ビルダは、次のアクションから［引数の編集］アイコンをクリックすることで起動します。

- (271 ページ) 関連付けの追加
- (272 ページ) ターゲット属性値の追加
- (273 ページ) ターゲットオブジェクトの追加
- (274 ページ) ソース属性値の追加
- (277 ページ) XML テキストの追加
- (278 ページ) ターゲット属性値のクリア選択されたオブジェクトが［DN］または［関連付け］である場合。

- (279 ページ) ソース属性値のクリア選択されたオブジェクトが［DN］または［関連付け］である場合。
- (281 ページ) ターゲットオブジェクトの削除選択されたオブジェクトが［DN］または［関連付け］である場合。
- (282 ページ) ソースオブジェクトの削除選択されたオブジェクトが［DN］または［関連付け］である場合。
- (282 ページ) 一致オブジェクトの検索
- (283 ページ) For Each
- (287 ページ) ターゲットオブジェクトの移動
- (288 ページ) ソースオブジェクトの移動
- (289 ページ) 操作属性の再フォーマット
- (290 ページ) 関連付けを削除
- (290 ページ) ターゲット属性値の削除
- (291 ページ) ソース属性値の削除
- (292 ページ) ターゲットオブジェクトの名前変更選択されたオブジェクトが［DN］または［関連付け］および［文字列を入力］である場合。
- (293 ページ) ソースオブジェクトの名前変更選択されたオブジェクトが［DN］または［関連付け］および［文字列を入力］である場合。
- (297 ページ) ターゲット属性値の設定選択されたオブジェクトが［DN］または［関連付け］であり、［値タイプを入力］が指定されていない場合。
- (298 ページ) ターゲットパスワードの設定
- (299 ページ) ローカル変数の設定
- (299 ページ) 操作関連付けの設定
- (300 ページ) 操作クラス名の設定
- (300 ページ) 操作ターゲット DN の設定
- (301 ページ) 操作プロパティの設定
- (301 ページ) 操作ソース DN の設定
- (301 ページ) 操作テンプレート DN の設定
- (302 ページ) ソース属性値の設定
- (303 ページ) ソースパスワードの設定
- (305 ページ) XML 属性の設定
- (306 ページ) ステータス
- (308 ページ) メッセージのトレース

一致属性ビルダ

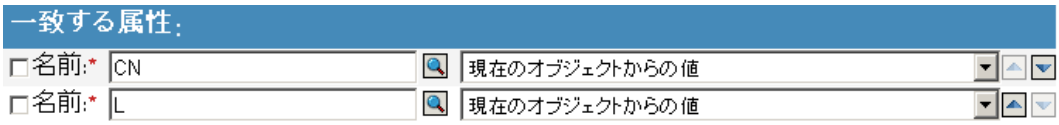
一致属性ビルダでは、データストアに一致するオブジェクトが存在するかどうかを判断するために、282 ページのセクション 3.6.17「一致オブジェクトの検索」によって使用される属性および値を選択できます。

たとえば、共通名と場所に基づいてユーザを一致させる場合は、次の条件を選択します。

図 3-12 一致オブジェクトの検索

［一致属性の入力］フィールドの横にある［引数の編集］アイコンをクリックして、一致属性ビルダのインタフェースを起動します。

図 3-13 一致属性ビルダ



［属性の参照］アイコンを選択して、一致させる属性を参照し、選択します。この例では、L および CN です。

2 番目の列では、［現在のオブジェクトからの値］を選択することで、属性内に保存された現在の値に一致させることができます。［その他の値］を選択すると、他の値と一致させることができます。一致させる値として任意の値を指定できます。値のタイプを選択すると、適切なビルダが［状態を入力］フィールドで使用可能になります。

アクションの引数コンポーネントビルダ

アクションの引数コンポーネントビルダは、［値のタイプを入力］で［構造］が選択されている場合に、次のアクションを選択すると起動します。

- (272 ページ) ターゲット属性値の追加
- (274 ページ) ソース属性値の追加
- (289 ページ) 操作属性の再フォーマット
- (290 ページ) ターゲット属性値の削除
- (291 ページ) ソース属性値の削除
- (296 ページ) デフォルト属性値の設定
- (302 ページ) ソース属性値の設定

図 3-14 アクションの引数コンポーネントビルダ

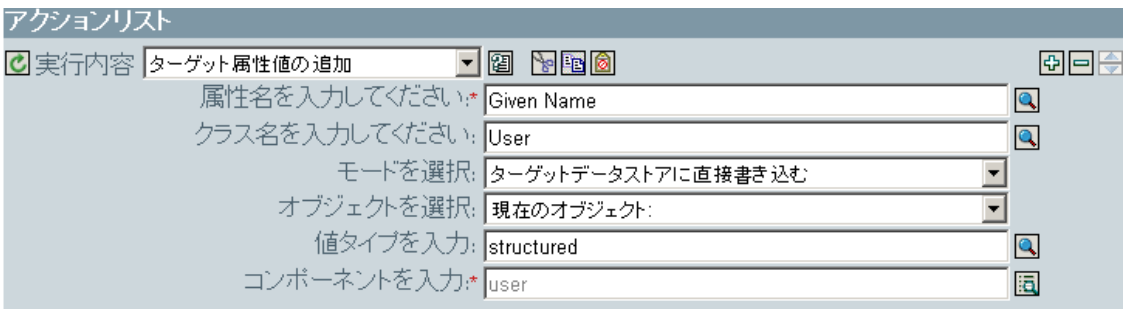
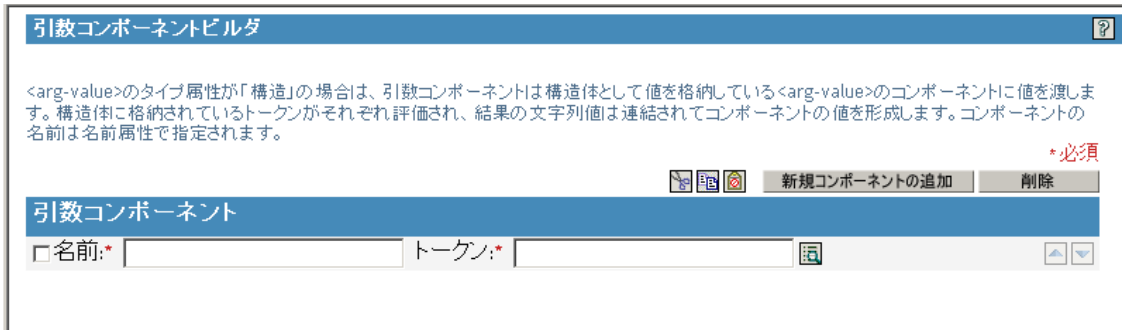


図 *3-15* アクションの引数コンポーネントビルダ



引数値リストビルダ

引数値リストビルダでは、(296 ページ) デフォルト属性値の設定アクションに使用するデフォルトの引数値を作成できます。

たとえば、デフォルトの場所として「不明」を設定する場合は、次のアクションを選択します。

図 *3-16* 引数値リストビルダ

[値を入力] フィールドの横にあるアイコンをクリックして、引数値リストビルダのインタフェースを起動し、次のような引数を作成します。

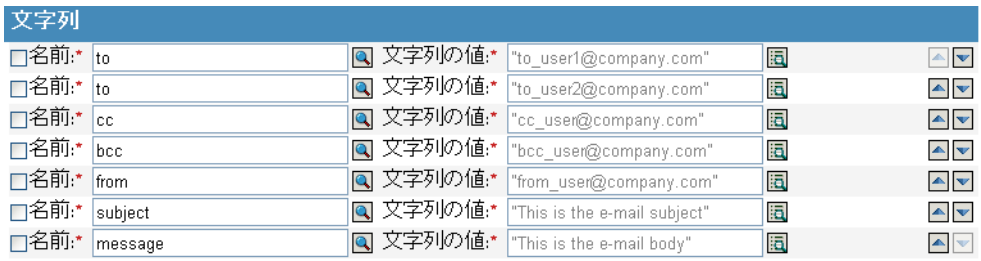
図 *3-17* 引数値リストビルダ

名前付き文字列ビルダ

名前付き文字列ビルダでは、(284 ページ) イベントの生成、(293 ページ) 電子メールの送信および (294 ページ) テンプレートから電子メールを送信など、特定のアクションで使用される名前と値のペアを作成できます。

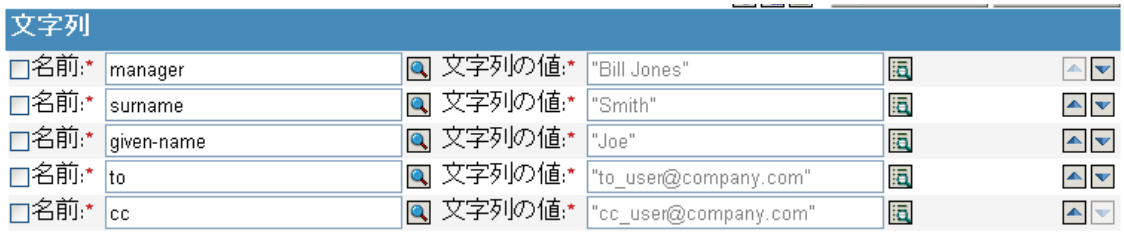
イベントの生成アクションの場合、名前付き文字列は、次のイベントとともに指定できるカスタム値フィールドに対応しています。

図 *3-18* 名前付き文字列ビルダ



メール送信アクションの場合、名前付き文字列は電子メールの要素に対応します。

図 3-19 メール送信アクション



指定できる値の完全なリストは、名前付き文字列ビルダを起動するアクションに対応するヘルプファイルに含まれています。

条件の引数コンポーネントビルダ

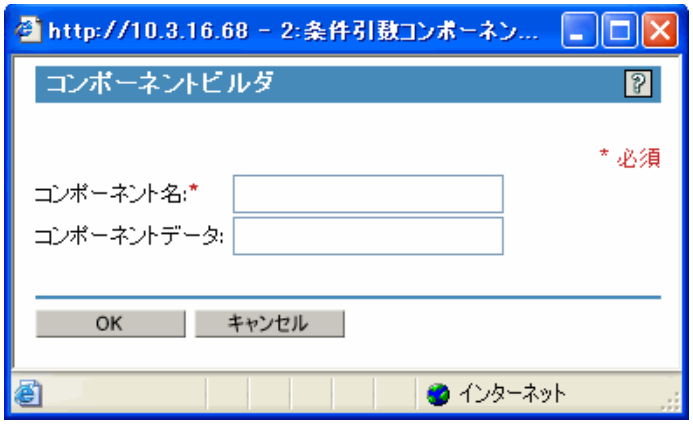
条件の引数コンポーネントビルダは、[引数の編集] アイコンをクリックすることで起動します。

このアイコンを表示するには、次の条件のモードとして [構造] を選択する必要があります。

- (253 ページ) If 属性
- (255 ページ) If ターゲット属性
- (266 ページ) If ソース属性

図 3-20 [構造] オプション

図 *3-21* 条件の引数コンポーネントビルダ



3.2.5 ポリシーの変更

1 管理するドライバの［Identity Manager ドライバの概要］を開きます。
2 変更するポリシーを示しているアイコンをクリックします。
3 変更するポリシーを選択し、［編集］をクリックします。

3.2.6 ポリシーの削除

選択したポリシーセットからポリシーを削除しても、そのポリシーは削除されません。

1 管理するドライバの［Identity Manager ドライバの概要］を開きます。
2 削除するポリシーを示しているアイコンをクリックします。

ポリシーセットに関連付けられているポリシーを表示するには :

1 管理するドライバの［Identity Manager ドライバの概要］を開きます。
2 ［すべてのポリシーを表示］アイコンをクリックします。

削除したポリシーをポリシーセットに追加しなおすには :

1 管理するドライバの［Identity Manager ドライバの概要］を開きます。
2 ポリシーを追加するポリシーセットをクリックします。
3 ［挿入］をクリックします。
4 ［既存のポリシーを使用する］を選択し、［参照］ボタンをクリックします。
5 追加するポリシーを参照します。

ヒント : そのポリシーを表示する適切なコンテナが表示されていることを確認します。

6 ［OK］をクリックします。
7 ［閉じる］をクリックします。

3.2.7 ポリシーの名前変更

1 管理するドライバの［Identity Manager ドライバの概要］を開きます。
2 名前変更するポリシーを示しているアイコンをクリックします。
3 ［名前変更］をクリックし、ポリシーの名前を変更します。
4 ［OK］をクリックします。
5 ［閉じる］をクリックします。

3.2.8 ポリシーの削除

1 管理するドライバの［Identity Manager ドライバの概要］を開きます。
2 削除するポリシーを示しているアイコンをクリックします。
3 削除するポリシーを選択し、［削除］をクリックします。

3.2.9 XML ファイルからのポリシーのインポート

1 管理するドライバの［Identity Manager ドライバの概要］を開きます。
2 インポートするポリシーを示しているアイコンをクリックします。
3 ポリシーを選択してから、［編集］をクリックします。
4 ［挿入］ボタンをクリックしてから、［DirXML スクリプトを含む XML ファイルをインポートする］を選択します。
5 インポートするポリシーファイルを参照して選択し、［OK］をクリックします。

3.2.10 XML ファイルへのポリシーのエクスポート

1 管理するドライバの［Identity Manager ドライバの概要］を開きます。
2 エクスポートするポリシーを示しているアイコンをクリックします。
3 ポリシーを選択してから、［編集］をクリックします。
4 ［名前を付けて保存］ボタンをクリックしてから、DirXML スクリプトを含む XML ファイルを保存する場所を選択します。
5 ［保存］をクリックします。

3.2.11 ポリシーの参照の作成

ポリシーの参照では、ポリシーを 1 つ作成し、それを複数の場所で参照できます。複数のドライバまたはポリシーによって使用されるポリシーがある場合は、参照を 1 つ作成することで、このポリシーの管理を簡素化できます。

1 管理するドライバの［Identity Manager ドライバの概要］を開きます。
2 参照を追加するポリシーを示しているアイコンをクリックします。
3 ポリシーを選択してから、［編集］をクリックします。
4 ［挿入］ボタンをクリックしてから、［DirXML スクリプトを含むポリシーへの参照を追加します。］を選択します。
5 参照するポリシーオブジェクトを指定して選択し、［OK］をクリックします。

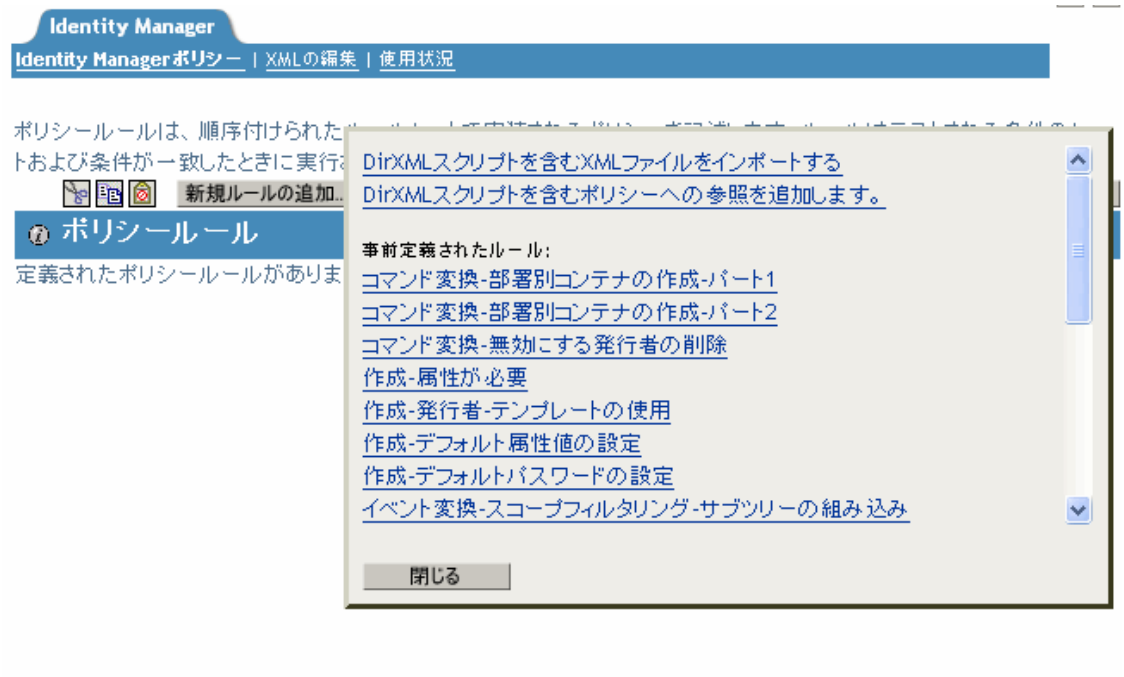
3.2.12 事前定義されたルールの使用

iManager には、20 の事前定義されたルールが備わっています。これらのルールをインポートすることで、ルールを自分で作成する場合と同様に使用できます。これらのルールには、管理者が使用する一般的なタスクが含まれています。ルールをカスタマイズするには、各自の環境に合わせた情報を指定する必要があります。

- 229 ページの 「コマンド変換 - 部署別コンテナの作成 - パート 1 とパート 2」
- 231 ページの 「コマンド変換 - 無効にする発行者の削除」
- 232 ページの 「作成 - 属性が必要」
- 233 ページの 「作成 - 発行者 - テンプレートの使用」
- 234 ページの 「作成 - デフォルト属性値の設定」
- 235 ページの 「作成 - デフォルトパスワードの設定」
- 236 ページの 「イベント変換 - スコープフィルタリング - サブツリーの組み込み」
- 237 ページの 「イベント変換 - スコープフィルタリング - サブツリーの除外」
- 238 ページの「入出力変換 - 電話番号の形式を (nnn) nnn-nnnn から nnn-nnn-nnnn に変更」
- 239 ページの「入出力変換 - 電話番号の形式を nnn-nnn-nnnn から (nnn) nnn-nnnn に変更」
- 240 ページの 「一致 - 発行者 (ミラーリング)」
- 241 ページの 「一致 - 購読者 (ミラーリング) - LDAP 形式」
- 242 ページの 「一致 - 属性値別」
- 243 ページの 「配置 - 発行者 (ミラーリング)」
- 244 ページの 「配置 - 購読者 (ミラーリング)-LDAP 形式」
- 245 ページの 「配置 - 発行者 (フラット)」
- 247 ページの 「配置 - 購読者 (フラット)-LDAP 形式」
- 248 ページの 「配置 - 部署別発行者」
- 249 ページの 「配置 - 部署別購読者 -LDAP 形式」

事前定義されたルールにアクセスするには :

1. 管理するドライバの［Identity Manager ドライバの概要］を開きます。
2. 事前定義されたルールを追加するポリシーを示すアイコンをクリックします。
3. ポリシーを選択し、［編集］をクリックします。
4. ［挿入］をクリックし、使用する事前定義されたルールを選択します。



コマンド変換 - 部署別コンテナの作成 - パート 1 とパート 2

ターゲットデータストア内に部署別コンテナがない場合に、そのコンテナを作成します。このルールは、ドライバ内の購読者コマンド変換ポリシーまたは発行者コマンド変換ポリシーに実装します。

この事前定義されたルールを使用するには、コマンド変換ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加するコマンド変換ポリシーがすでにある場合は、229 ページの 「事前定義されたルールのインポート」へ進みます。

ポリシーの作成

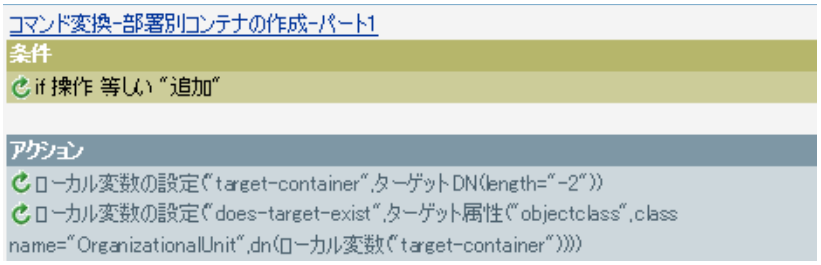
1. 管理するドライバの［Identity Manager ドライバの概要］を開きます。
2. 発行者チャネルまたは購読者チャネルのコマンド変換ポリシーオブジェクトをクリックします。
3. ［挿入］をクリックします。
4. ポリシーの名前を指定し、このポリシーをポリシービルダで実装することを確認したら、［OK］をクリックします。

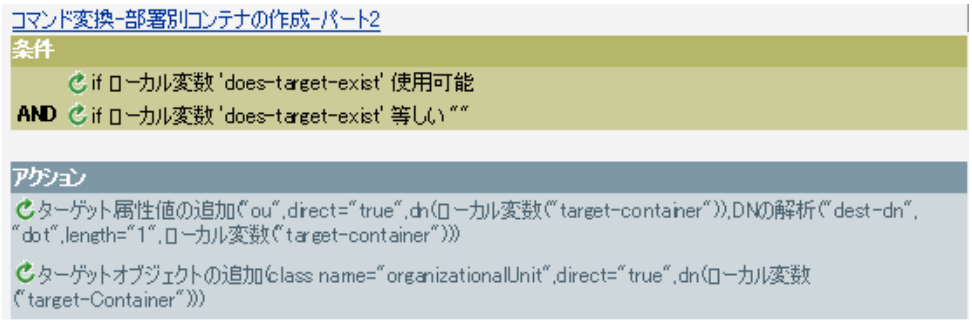
ルールビルダが起動します。

事前定義されたルールのインポート

1. ルールビルダで、［挿入］をクリックします。
2. ［コマンド変換 - 部署別のコンテナの作成 - パート 1］を選択します。
3. ［挿入］をクリックします。
4. ［コマンド変換 - 部署別のコンテナの作成 - パート 2］を選択します。
5. ［OK］をクリックします。

このルールには、環境に応じて変更すべき情報はありません。





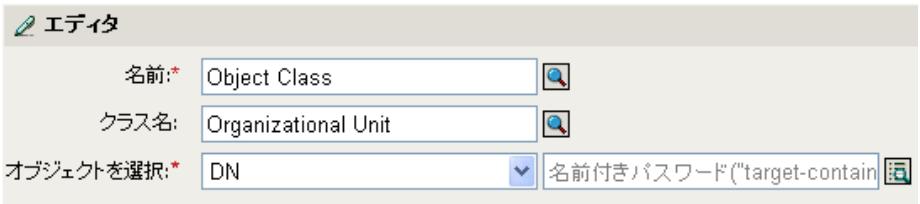
重要 : ルールが順序どおりに表示されていることを確認します。パート 1 は 2 よりも先に実行する必要があります。

ルールでのロジックの動作

このルールは、オブジェクトのターゲットの場所が存在しない場合に使用されます。このルールでは、オブジェクトが配置できない場合、作成を拒否する代わりにコンテナが作成され、その中にオブジェクトが配置されます。

パート 1 では追加操作が想定されます。追加操作が発生すると、2 つのローカル変数が設定されます。最初のローカル変数は、target-container という名前になります。target-container の値が、ターゲット DN に設定されます。2 つ目のローカル変数は、does-target-exist という名前になります。does-target-exist の値は、objectclass のターゲット属性値に設定されます。クラスは OrganizationalUnit に設定されます。OrganizationalUnit の DN は、ローカル変数 target-container に設定されます。

図 *3-22* コンテナの作成



パート 2 では、ローカル変数 does-target-exist が使用可能かどうかが確認されます。また、ローカル変数 does-target-exist の値が空白に設定されているかどうかも確認されます。値が空白である場合、部門オブジェクトが作成されます。部門の DN は、ローカル変数 target-container の値に設定されます。また、OU 属性の値も追加されます。OU 属性の値

は、ローカル変数 target-container の値を解析することで取得される新しい部門の名前に設定されます。

コマンド変換 - 無効にする発行者の削除

ユーザオブジェクトの削除操作を、eDirectory™ 内のターゲットユーザオブジェクトを無効にする変更操作に変換します。このルールは、ドライバ内の発行者コマンド変換ポリシーに実装します。

この事前定義されたルールを使用するには、コマンド変換ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加するコマンド変換ポリシーがすでにある場合は、(231 ページ) 事前定義されたルールのインポートへ進みます。

ポリシーの作成

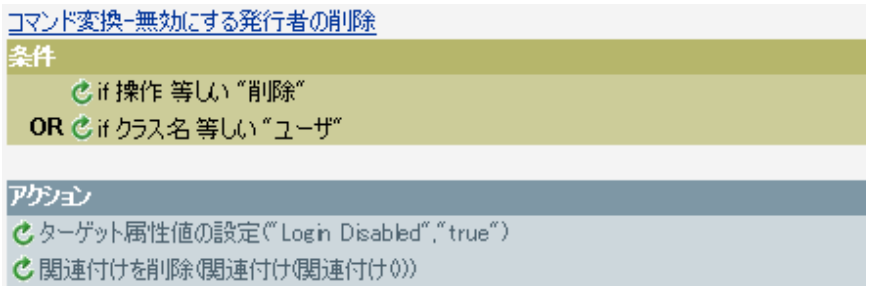
1. 管理するドライバの［Identity Manager ドライバの概要］を開きます。
2. 発行者チャネルのコマンド変換ポリシーオブジェクトをクリックします。
3. ［挿入］をクリックします。
4. ポリシーの名前を指定し、このポリシーをポリシービルダで実装することを確認したら、［OK］をクリックします。

ルールビルダが起動します。

事前定義されたルールのインポート

1. ルールビルダで、［挿入］をクリックします。
2. ［コマンド変換 - 無効にする発行者の削除］を選択します。
3. ［OK］をクリックします。

このルールには、環境に応じて変更すべき情報はありません。



ルールでのロジックの動作

このルールは、通常は接続システムで発生した削除イベントに対して、削除コマンドが識別ボールトに送信されるときに使用されます。ユーザオブジェクトは識別ボールトで削除される代わりに、無効になります。削除コマンドがユーザオブジェクトに対して処理されるときは、「ログインの無効化」のターゲット属性値が True に設定され、ユーザオブジェクトから関連付けが削除され、削除コマンドが拒否されます。ユーザオブジェクトは、Novell eDirectory ツリーへはログインできなくなりますが、ユーザオブジェクトは削除されません。

作成 - 属性が必要

必要な属性が入力されない場合にユーザオブジェクトを作成できないようにします。このルールは、ドライバ内の購読者作成ポリシーまたは発行者作成ポリシーに実装します。

この事前定義されたルールを使用するには、作成ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する作成ポリシーがすでにある場合は、232 ページの 「事前定義されたルールのインポート」へ進みます。

ポリシーの作成

1. 管理するドライバの［Identity Manager ドライバの概要］を開きます。
2. 発行者チャネルまたは購読者チャネルの作成ポリシーオブジェクトをクリックします。
3. ［挿入］をクリックします。
4. ポリシーの名前を指定し、このポリシーをポリシービルダで実装することを確認したら、［OK］をクリックします。

ルールビルダが起動します。

事前定義されたルールのインポート

1. ルールビルダで、［挿入］をクリックします。
2. ［作成 - 属性が必要］を選択します。
3. ルールを編集するため、ルールビルダで［作成 - 属性が必要］をクリックします。
4. ［名前を入力］フィールドから、［必要な属性の名前を入力してください］を削除します。
5. (オプション) 必須属性を複数指定する場合は、プラスアイコンをクリックして新しいアクションを追加します。
6. ［操作属性値がない場合は拒否］を選択し、別の必須属性を参照します。
7. ［OK］をクリックします。

ルールでのロジックの動作

このルールは、ビジネスプロセスにおいて、ターゲットユーザオブジェクトが作成される前に、ソースユーザオブジェクトに特別な属性が必要な場合に使用されます。ソースデータストア内にユーザオブジェクトを作成する場合、このルールでは、ターゲットユーザオブジェクトの作成時に必須属性が入力されないと、ターゲットデータストア内へのオブジェクトの作成が拒否されます。必須属性は複数指定できます。

作成 - 発行者 - テンプレートの使用

ユーザオブジェクトの作成時に、Novell eDirectory のテンプレートオブジェクトを使用できるようにします。このルールは、ドライバ内の発行者作成ポリシーに実装します。

この事前定義されたルールを使用するには、作成ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する作成ポリシーがすでにある場合は、233 ページの 「事前定義されたルールのインポート」へ進みます。

ポリシーの作成

1. 管理するドライバの［Identity Manager ドライバの概要］を開きます。
2. 発行者チャネルまたは購読者チャネルの作成ポリシーオブジェクトをクリックします。
3. ［挿入］をクリックします。
4. ポリシーの名前を指定し、このポリシーをポリシービルダで実装することを確認したら、［OK］をクリックします。

ルールビルダが起動します。

事前定義されたルールのインポート

1. ルールビルダで、［挿入］をクリックします。
2. ［作成 - 発行者 - テンプレートの使用］を選択します。
3. ルールを編集するため、ルールビルダで［作成 - 発行者 - テンプレートの使用］をクリックします。
4. ［DN を入力］フィールドから、[テンプレートオブジェクトの DN を入力してください] を削除します。
5. ［引数の編集］アイコンをクリックして、引数ビルダを起動します。
6. 名詞リストから［テキスト］を選択し、［追加］をクリックします。
7. エディタで、参照アイコンをクリックし、テンプレートオブジェクトを参照して選択し、［OK］をクリックします。
8. ［OK］をクリックします。

ルールでのロジックの動作

このルールは、テンプレートオブジェクトに基づいて識別ボールト内にユーザを作成する場合に使用されます。ユーザに共通の属性がある場合、テンプレートを使用することで時間を節約できます。テンプレートオブジェクトに情報を入力してユーザオブジェクトを作成すると、識別ボールトはテンプレートの属性値を使用してユーザオブジェクトを作成します。

ユーザオブジェクトを作成するとき、このルールは操作テンプレート DN の設定アクションを実行します。このアクションは Identity Manager に対し、オブジェクトの作成時に参照先テンプレートを使用するよう指定します。

作成 - デフォルト属性値の設定

ユーザオブジェクトの作成時に割り当てられる属性のデフォルト値を設定できます。このルールは、ドライバ内の購読者作成ポリシーまたは発行者作成ポリシーに実装します。

この事前定義されたルールを使用するには、作成ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する作成ポリシーがすでにある場合は、234 ページの 「事前定義されたルールのインポート」へ進みます。

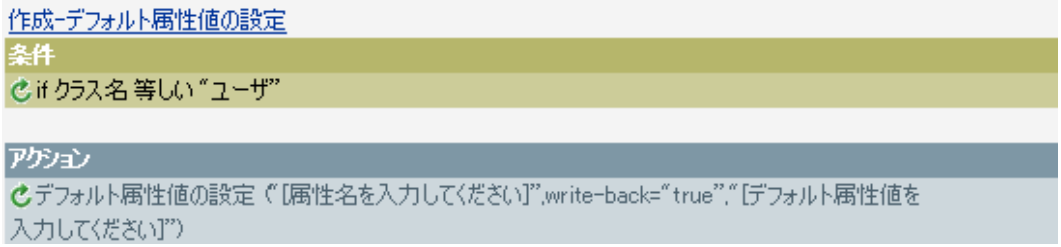
ポリシーの作成

1. 管理するドライバの［Identity Manager ドライバの概要］を開きます。
2. 発行者チャネルまたは購読者チャネルの作成ポリシーオブジェクトをクリックします。
3. ［挿入］をクリックします。
4. ポリシーの名前を指定し、このポリシーをポリシービルダで実装することを確認したら、［OK］をクリックします。

ルールビルダが起動します。

事前定義されたルールのインポート

1. ルールビルダで、［挿入］をクリックします。
2. ［作成 - デフォルト属性値の設定］を選択します。
3. ルールを編集するため、ルールビルダで［デフォルト属性値の設定］をクリックします。
4. ［属性名を入力してください］フィールドから、［属性名を入力してください］を削除します。
5. 参照アイコンをクリックして、作成する属性を参照して選択します。
6. ［引数値を入力］フィールドから、［デフォルト属性値を入力してください］を削除します。
7. ［引数の編集］アイコンをクリックして、引数値リストビルダを起動します。
8. デフォルト値にするデータのタイプを選択します。
9. ［引数の編集］アイコンをクリックして、引数ビルダを起動します。
10. 引数ビルダで、引数にする値を作成し、［OK］をクリックします。
11. ［OK］をクリックします。



ルールでのロジックの動作

このルールは、ユーザオブジェクトを作成するときにデフォルト属性値を指定する場合に使用されます。ユーザオブジェクトが作成されるときに、ソースオブジェクトによって値が指定されていない場合に限り、指定済みの属性がこのルールによって追加されます。

複数の属性を定義する場合は、アクションを右クリックして［New (新規)］>［アクション］の順にクリックします。アクションを選択し、デフォルト属性値を設定した後、上記の手順を実行して属性値を割り当てます。

作成 - デフォルトパスワードの設定

ユーザオブジェクトの作成時に、ユーザオブジェクトのデフォルトパスワードを設定します。このルールは、ドライバ内の購読者作成ポリシーまたは発行者作成ポリシーに実装します。

この事前定義されたルールを使用するには、作成ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する作成ポリシーがすでにある場合は、235 ページの 「事前定義されたルールのインポート」へ進みます。

ポリシーの作成

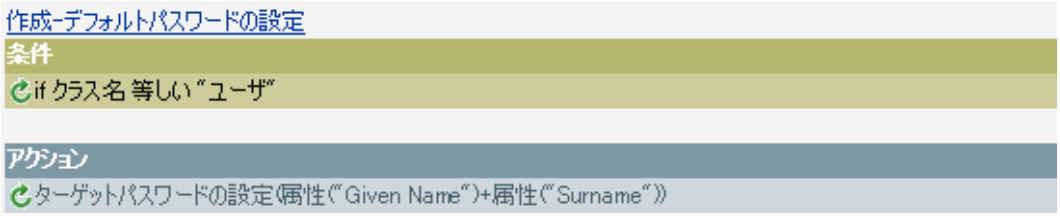
1. 管理するドライバの［Identity Manager ドライバの概要］を開きます。
2. 発行者チャネルまたは購読者チャネルの作成ポリシーオブジェクトをクリックします。
3. ［挿入］をクリックします。
4. ポリシーの名前を指定し、このポリシーをポリシービルダで実装することを確認したら、［OK］をクリックします。

ルールビルダが起動します。

事前定義されたルールのインポート

1. ルールビルダで、［挿入］をクリックします。
2. ［作成 - デフォルトパスワードの設定］を選択します。
3. ［OK］をクリックします。

このルールには、環境に応じて変更すべき情報はありません。



ルールでのロジックの動作

このルールは、デフォルトパスワードを使用してユーザオブジェクトを作成する場合に使用されます。ユーザオブジェクトの作成時に、ユーザオブジェクトに設定されるパスワードは、そのユーザオブジェクトの名前属性に名字属性を加えたものになります。

デフォルトパスワードの値は、引数を編集することで変更できます。パスワードは、引数ビルダを使用して任意の値に設定できます。

イベント変換 - スコープフィルタリング - サブツリーの組み込み

特定のサブツリーの外で発生するすべてのイベントを除外します。このルールは、ドライバ内の購読者イベント変換ポリシーまたは発行者イベント変換ポリシーに実装します。

この事前定義されたルールを使用するには、イベント変換ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加するイベント変換ポリシーがすでにある場合は、236 ページの 「事前定義されたルールのインポート」へ進みます。

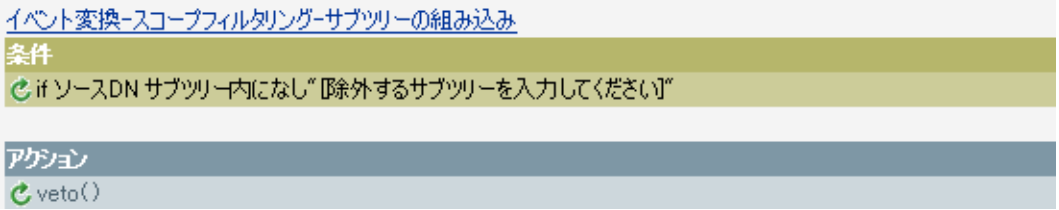
ポリシーの作成

1. 管理するドライバの［Identity Manager ドライバの概要］を開きます。
2. 発行者チャネルまたは購読者チャネルのイベント変換ポリシーオブジェクトをクリックします。
3. ［挿入］をクリックします。
4. ポリシーの名前を指定し、このポリシーをポリシービルダで実装することを確認したら、［OK］をクリックします。

ルールビルダが起動します。

事前定義されたルールのインポート

1. ルールビルダで、［挿入］をクリックします。
2. ［イベント変換 - スコープフィルタリング - サブツリーの組み込み］を選択します。
3. ルールを編集するため、ルールビルダで［イベント変換 - スコープフィルタリング - サブツリーの組み込み］をクリックします。
4. ［値］フィールドの［組み込むサブツリーを入力してください］を削除します。
5. ［参照］ボタンをクリックして識別ボールトを参照し、イベントを同期させるツリーの部分を選択し、［OK］をクリックします。
6. ［OK］をクリックします。



ルールでのロジックの動作

このルールは、識別ボールトと接続システムとの間で、特定のサブツリーのみを同期させる場合に使用されます。イベントが識別ボールトの指定された以外の場所で発生した場合、そのイベントは拒否されます。同期対象のサブツリーは、268 ページのセクション 3.5.15「If ソース DN」条件をコピーして貼り付けることで追加できます。

イベント変換 - スコープフィルタリング - サブツリーの除外

特定のサブツリー内で発生するすべてのイベントを除外します。このルールは、ドライバ内の購読者イベント変換ポリシーまたは発行者イベント変換ポリシーに実装します。

この事前定義されたルールを使用するには、イベント変換ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加するイベント変換ポリシーがすでにある場合は、237 ページの 「事前定義されたルールのインポート」へ進みます。

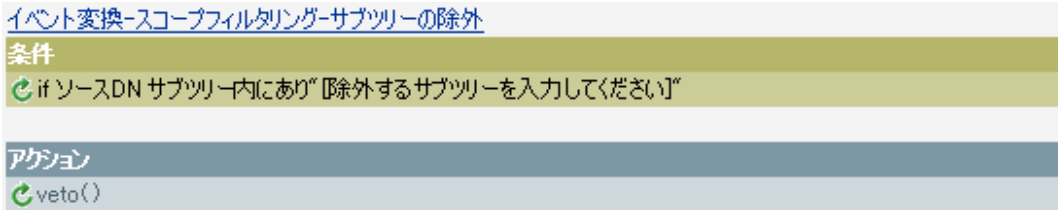
ポリシーの作成

1. 管理するドライバの［Identity Manager ドライバの概要］を開きます。
2. 発行者チャネルまたは購読者チャネルのイベント変換ポリシーオブジェクトをクリックします。
3. ［挿入］をクリックします。
4. ポリシーの名前を指定し、このポリシーをポリシービルダで実装することを確認したら、［OK］をクリックします。

ルールビルダが起動します。

事前定義されたルールのインポート

1. ルールビルダで、［挿入］をクリックします。
2. ［イベント変換 - スコープフィルタリング - サブツリーの除外］を選択します。
3. ルールを編集するため、ルールビルダで［イベント変換 - スコープフィルタリング - サブツリーの除外］をクリックします。
4. ［値］フィールドの［除外するサブツリーを入力してください］を削除します。
5. ［参照］ボタンをクリックして識別ボールトを参照し、同期から除外するイベントのツリーの部分を選択し、［OK］をクリックします。
6. ［OK］をクリックします。



ルールでのロジックの動作

このルールは、識別ボールトまたは接続システムの一部を同期から除外する場合に使用されます。識別ボールトの特定部分でイベントが発生した場合、そのイベントは拒否されます。除外対象のサブツリーは、if ソース DN 条件をコピーして貼り付けることで追加できます。

入出力変換 - 電話番号の形式を (nnn) nnn-nnnn から nnn-nnn-nnnn に変更

電話番号の形式を変換します。このルールは、ドライバ内の入出力変換ポリシーに実装します。通常、このルールを入力変換上で使用する場合は、このルールを使用して、出力変換上で電話番号の形式を nnn-nnn-nnnn から (nnn) nnn-nnnn に変更したり、(nnn) nnn-nnnn から nnn-nnn-nnnn に変更しなおしたりします。

この事前定義されたルールを使用するには、入出力変換ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する入出力変換ポリシーがすでにある場合は、238 ページの 「事前定義されたルールのインポート」へ進みます。

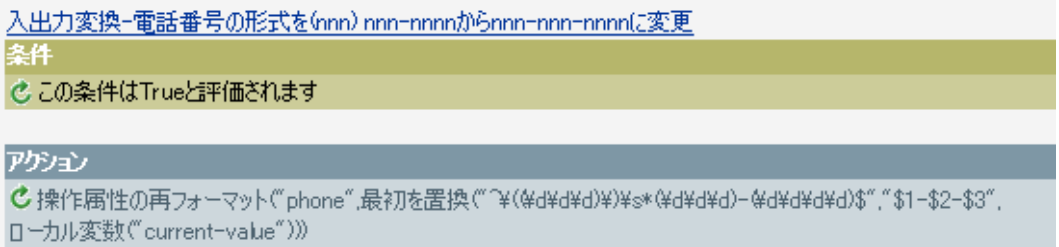
ポリシーの作成

1. 管理するドライバの［Identity Manager ドライバの概要］を開きます。
2. 発行者チャネルまたは購読者チャネルの入出力変換ポリシーオブジェクトをクリックします。
3. ［挿入］をクリックします。
4. ポリシーの名前を指定し、このポリシーをポリシービルダで実装することを確認したら、［OK］をクリックします。

ルールビルダが起動します。

事前定義されたルールのインポート

1. ルールビルダで、［挿入］をクリックします。
2. ［入出力変換 - 電話番号の形式を (nnn) nnn-nnnn から nnn-nnn-nnnn に変更］を選択します。
3. ルールを編集するため、ルールビルダで［入出力変換 - 電話番号の形式を (nnn) nnn-nnnn から nnn-nnn-nnnn に変更］を選択します。
4. 電話番号の形式変更が実行されるときの条件を定義します。
5. ［OK］をクリックします。



ルールでのロジックの動作

このルールは、電話番号の形式を変換する場合に使用されます。現在の操作における電話番号の属性値で、(nnn) nnn-nnnn に一致するパターンをすべて検索し、これを nnn-nnn-nnnn に置き換えます。

入出力変換 - 電話番号の形式を nnn-nnn-nnnn から (nnn) nnn-nnnn に変更

電話番号の形式を変換します。このルールは、入出力変換ポリシーに実装します。通常、このルールを出力変換上で使用する場合は、このルールを使用して、入力変換上で電話番号の形式を (nnn) nnn-nnnn から nnn-nnn-nnnn に変更したり、nnn-nnn-nnnn から (nnn) nnn-nnnn に変更しなおしたりします。

この事前定義されたルールを使用するには、入出力変換ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する入出力変換ポリシーがすでにある場合は、239 ページの 「事前定義されたルールのインポート」へ進みます。

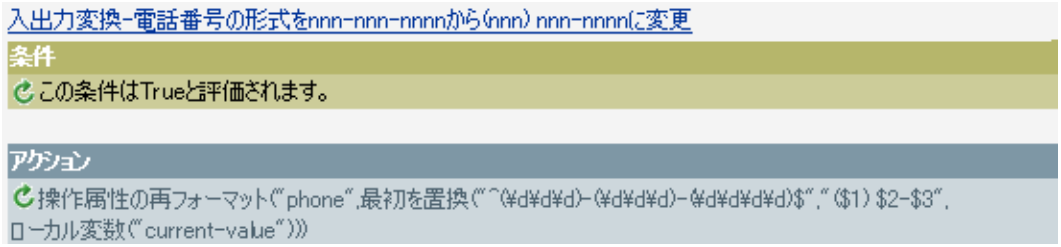
ポリシーの作成

1. 管理するドライバの［Identity Manager ドライバの概要］を開きます。
2. 発行者チャネルまたは購読者チャネルの入出力変換ポリシーオブジェクトをクリックします。
3. ［挿入］をクリックします。
4. ポリシーの名前を指定し、このポリシーをポリシービルダで実装することを確認したら、［OK］をクリックします。

ルールビルダが起動します。

事前定義されたルールのインポート

1. ルールビルダで、［挿入］をクリックします。
2. ［入出力変換 - 電話番号の形式を nnn-nnn-nnnn から (nnn) nnn-nnnn に変更］を選択します。
3. ルールを編集するため、ルールビルダで［入出力変換 - 電話番号の形式を nnn-nnn-nnnn から (nnn) nnn-nnnn に変更］を選択します。
4. 電話番号の形式変更が実行されるときの条件を定義します。
5. ［OK］をクリックします。



ルールでのロジックの動作

このルールは、電話番号の形式を変更する場合に使用されます。現在の操作における電話番号の属性値で、(nnn) nnn-nnnn に一致するパターンをすべて検索し、これを nnn-nnn-nnnn に置き換えます。

一致 - 発行者 (ミラーリング)

識別ボールト内で接続システム内のオブジェクトに一致するものを、名前と場所に基づいて検索します。このルールは、ドライバ内の発行者一致ポリシーに実装します。

この事前定義されたルールを使用するには、一致ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する一致ポリシーがすでにある場合は、240 ページの 「事前定義されたルールのインポート」へ進みます。

ポリシーの作成

1. 管理するドライバの［Identity Manager ドライバの概要］を開きます。
2. 発行者チャネルの一致ポリシーオブジェクトをクリックします。
3. ［挿入］をクリックします。
4. ポリシーの名前を指定し、このポリシーをポリシービルダで実装することを確認したら、［OK］をクリックします。

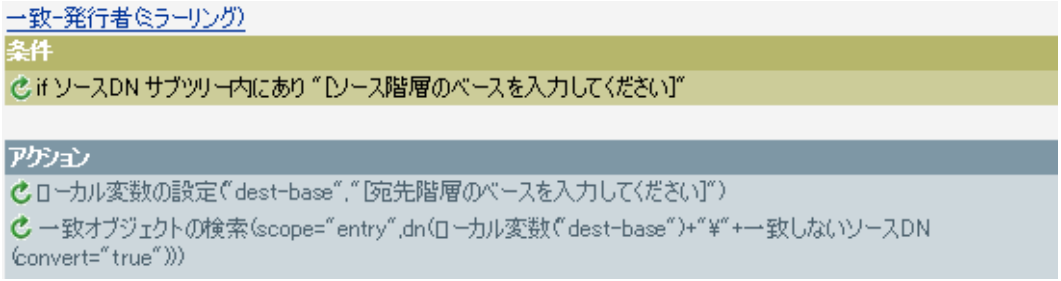
ルールビルダが起動します。

事前定義されたルールのインポート

1. ルールビルダで、［挿入］をクリックします。
2. ［一致 - 発行者 (ミラーリング)］を選択します。
3. ルールを編集するため、ルールビルダで［一致 - 発行者 (ミラーリング)］をクリックします。
4. ［値］フィールドから、［ソース階層のベースを入力してください］を削除します。
5. 照合を開始するソース階層内のコンテナを参照し、［OK］をクリックします。
6. ［OK］をクリックします。
7. ［文字列を入力］フィールドから、［宛先階層のベースを入力してください］を削除します。
8. ［引数の編集］アイコンをクリックして、引数ビルダを起動します。
9. 名詞リストから［テキスト］を選択し、［追加］をクリックします。

10 エディタで参照アイコンをクリックして、ソース構造を一致させる宛先階層内のコンテナを参照し、選択したら［OK］をクリックします。

11 ［OK］をクリックします。



ルールでのロジックの動作

特定のソースサブツリー内にある接続システム内のオブジェクトでイベントが発生した場合、このルールでは、指定されたターゲットサブツリーに対して同じオブジェクト名と場所を表す DN が識別ボールト内に作成されます。ターゲットオブジェクトが存在し、そのオブジェクトが目的のオブジェクトクラスである場合、一致したとみなされます。ソース (接続システム) の DN およびターゲット (識別ボールト) のサブツリーを指定する必要があります。

一致 - 購読者 (ミラーリング) - LDAP 形式

LDAP 形式の DN を使用している接続システム内で、識別ボールト内のオブジェクトに一致するものを、名前と場所に基づいて検索します。このルールは、ドライバ内の購読者一致ポリシーに実装します。

この事前定義されたルールを使用するには、一致ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する一致ポリシーがすでにある場合は、241 ページの 「事前定義されたルールのインポート」へ進みます。

ポリシーの作成

1 管理するドライバの［Identity Manager ドライバの概要］を開きます。

2 購読者チャネルの一致ポリシーオブジェクトをクリックします。

3 ［挿入］をクリックします。

4 ポリシーの名前を指定し、このポリシーをポリシービルダで実装することを確認したら、［OK］をクリックします。

ルールビルダが起動します。

事前定義されたルールのインポート

1 ルールビルダで、［挿入］をクリックします。

2 ［一致 - 購読者 (ミラーリング)-LDAP 形式］を選択します。

3 ルールを編集するため、ルールビルダで［一致 - 購読者 (ミラーリング)-LDAP 形式］をクリックします。

4 ［値］フィールドから、［ソース階層のベースを入力してください］を削除します。

5 照合を開始するソース階層内のコンテナを参照し、［OK］をクリックします。

6 ［OK］をクリックします。

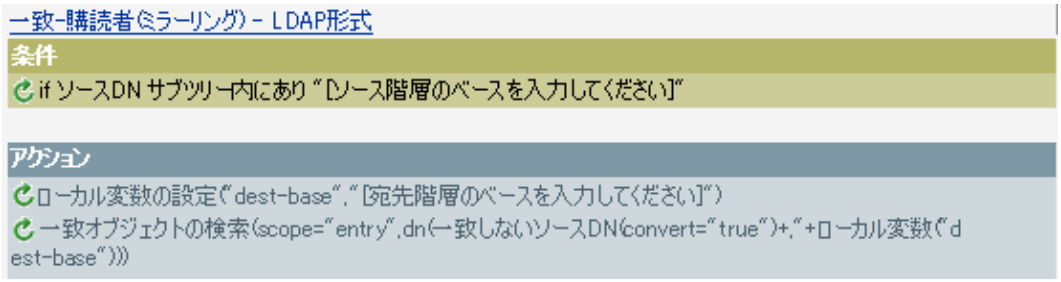
7 ［文字列を入力］フィールドから、［宛先階層のベースを入力してください］を削除します。

8 ［引数の編集］アイコンをクリックして、引数ビルダを起動します。

9 名詞リストから［テキスト］を選択し、［追加］をクリックします。

10 エディタで、参照アイコンをクリックして、ソース構造を照合する宛先階層内のコンテナを参照し、選択したら［OK］をクリックします。

11 ［OK］をクリックします。



ルールでのロジックの動作

特定のソースサブツリー内にある識別ボールト内のオブジェクトで追加イベントが発生した場合、このルールでは、指定されたターゲットサブツリーに関連する接続システム内の同じオブジェクト名と場所を表す DN が作成されます。ターゲットオブジェクトが存在し、そのオブジェクトが目的のオブジェクトクラスである場合、一致したとみなされます。ソース (識別ボールト) の DN およびターゲット (接続システム) のサブツリーを指定する必要があります。また、接続システムでは LDAP 形式の DN を使用しなければなりません。

一致 - 属性値別

一致するオブジェクトを特定の属性値で検索します。このルールは、ドライバ内の購読者一致ポリシーまたは発行者一致ポリシーに実装します。

この事前定義されたルールを使用するには、一致ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する一致ポリシーがすでにある場合は、243 ページの 「事前定義されたルールのインポート」へ進みます。

ポリシーの作成

1 管理するドライバの［Identity Manager ドライバの概要］を開きます。

2 発行者チャネルの一致ポリシーオブジェクトをクリックします。

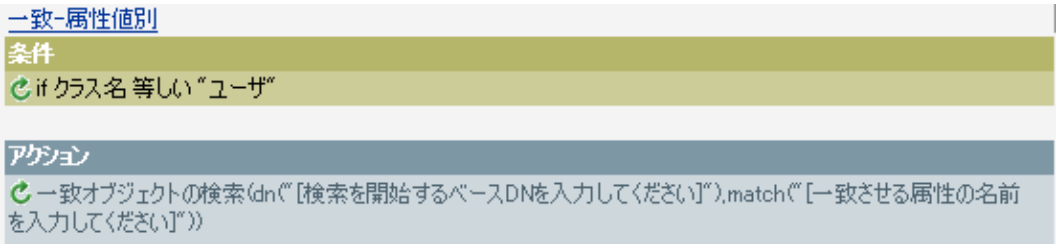
3 ［挿入］をクリックします。

4 ポリシーの名前を指定し、このポリシーをポリシービルダで実装することを確認したら、［OK］をクリックします。

ルールビルダが起動します。

事前定義されたルールのインポート

1 ルールビルダで、［挿入］をクリックします。
2 ［一致 - 属性値別］を選択します。
3 ルールを編集するため、ルールビルダで［一致 - 属性値別］をクリックします。
4 ［DN を入力］フィールドから、［検索を開始するベース DN を入力してください］を削除します。
5 ［引数の編集］アイコンをクリックして、引数ビルダを起動します。
6 名詞リストから［テキスト］を選択し、［追加］をクリックします。
7 エディタで、参照アイコンをクリックして、検索を開始するコンテナを参照し、選択したら［OK］をクリックします。
8 ［一致属性を入力］フィールドから、［Enter name of attribute to match on (一致させる属性名を入力してください)］を削除します。
9 ［引数の編集］アイコンをクリックして、一致属性ビルダを起動します。
10 参照アイコンをクリックして、一致させる属性を選択します。一致させる属性を 1 つ以上選択したら、［OK］をクリックします。
11 ［OK］をクリックします。



ルールでのロジックの動作

ソースデータストア内のオブジェクトで追加イベントが発生した場合、このルールでは、指定された属性と同じ値を持つオブジェクトをターゲットデータストア内で検索します。接続システム内で検索するサブツリーのベース DN、および一致させる属性名を指定する必要があります。

配置 - 発行者 (ミラーリング)

接続システムの名前と場所に基づき、識別ボールト内にオブジェクトを配置します。このルールは、ドライバ内の発行者配置ポリシーに実装します。

この事前定義されたルールを使用するには、配置ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する配置ポリシーがすでにある場合は、244 ページの 「事前定義されたルールのインポート」へ進みます。

ポリシーの作成

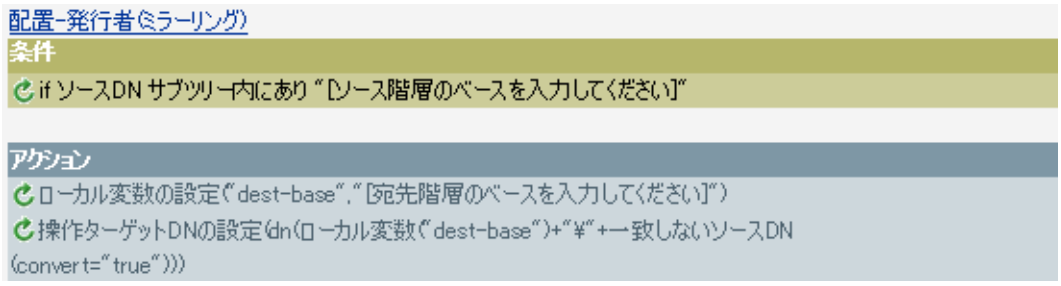
1 管理するドライバの［Identity Manager ドライバの概要］を開きます。
2 発行者チャネルの配置ポリシーオブジェクトをクリックします。
3 ［挿入］をクリックします。

4 ポリシーの名前を指定し、このポリシーをポリシービルダで実装することを確認したら、［OK］をクリックします。

ルールビルダが起動します。

事前定義されたルールのインポート

1 ルールビルダで、［挿入］をクリックします。
2 ［配置 - 発行者 (ミラーリング)］を選択します。
3 ルールを編集するため、ルールビルダで［配置 - 発行者 (ミラーリング)］をクリックします。
4 ［値］フィールドから、［ソース階層のベースを入力してください］を削除します。
5 オブジェクトをイベントの対象にするソース階層でコンテナを参照し、選択したら［OK］をクリックします。
6 ［文字列を入力］フィールドから、［宛先階層のベースを入力してください］を削除します。
7 ［引数の編集］アイコンをクリックして、引数ビルダを起動します。
8 名詞リストから［テキスト］を選択し、［追加］をクリックします。
9 エディタで、参照アイコンをクリックして、オブジェクトを配置する宛先階層内のコンテナを参照し、選択したら［OK］をクリックします。
10 ［OK］をクリックします。



ルールでのロジックの動作

ユーザオブジェクトが接続システム内の指定されたソースサブツリー内にある場合、そのオブジェクトは、同じ相対名と同じ場所で識別ボールト内に配置されます。ソース (接続システム) の DN およびターゲット (識別ボールト) のサブツリーを指定する必要があります。

配置 - 購読者 (ミラーリング)-LDAP 形式

指定したポイントから識別ボールト内のミラー化された構造を使用して、オブジェクトをデータストア内に配置します。このルールは、ドライバ内の配置ポリシーに実装します。このルールは、購読者チャネルにのみ実装できます。

この事前定義されたルールを使用するには、配置ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する配置ポリシーがすでにある場合は、245 ページの 「事前定義されたルールのインポート」へ進みます。

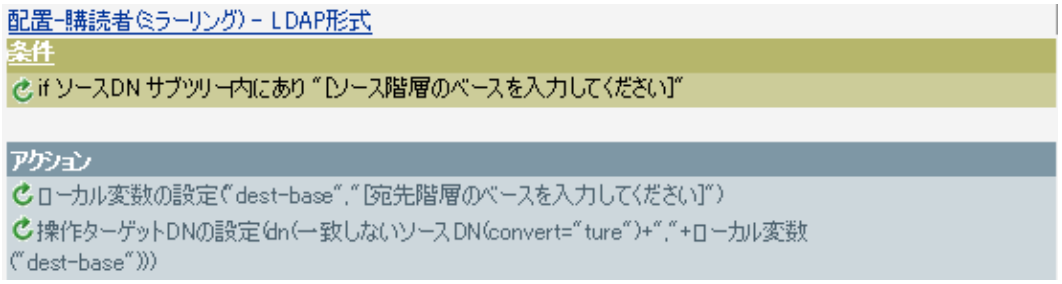
ポリシーの作成

1 管理するドライバの［Identity Manager ドライバの概要］を開きます。
2 購読者チャネルの配置ポリシーオブジェクトをクリックします。
3 ［挿入］をクリックします。
4 ポリシーの名前を指定し、このポリシーをポリシービルダで実装することを確認したら、［OK］をクリックします。

ルールビルダが起動します。

事前定義されたルールのインポート

1 ルールビルダで、［挿入］をクリックします。
2 ［配置 - 購読者 (ミラーリング)-LDAP 形式］を選択します。
3 ルールを編集するため、ルールビルダで［配置 - 購読者 (ミラーリング)-LDAP 形式］をクリックします。
4 ［値］フィールドから、［ソース階層のベースを入力してください］を削除します。
5 オブジェクトをイベントの対象にするソース階層でコンテナを参照し、選択したら［OK］をクリックします。
6 ［文字列を入力］フィールドから、［宛先階層のベースを入力してください］を削除します。
7 ［引数の編集］アイコンをクリックして、引数ビルダを起動します。
8 名詞リストから［テキスト］を選択し、［追加］をクリックします。
9 エディタで、参照アイコンをクリックして、オブジェクトを配置する宛先階層内のコンテナを参照し、選択したら［OK］をクリックします。
10 ［OK］をクリックします。



ルールでのロジックの動作

指定されたソースサブツリー内にユーザオブジェクトがある場合、そのオブジェクトは、同じ相対名と同じ場所で識別ボールト内に配置されます。ソース (識別ボールト) の DN およびターゲット (接続システム) のサブツリーを指定する必要があります。また、接続システムでは LDAP 形式の DN を使用しなければなりません。

配置 - 発行者 (フラット)

データストアのオブジェクトを識別ボールトの 1 つのコンテナ内に配置します。このルールは、ドライバ内の発行者配置ポリシーに実装します。

この事前定義されたルールを使用するには、配置ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する配置ポリシーがすでにある場合は、246 ページの 「事前定義されたルールのインポート」へ進みます。

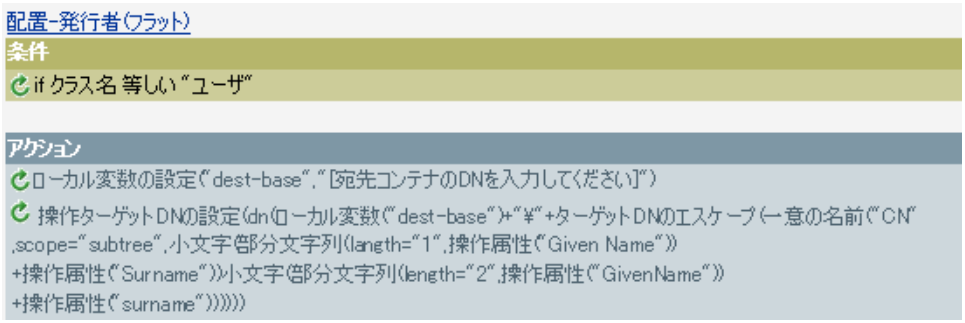
ポリシーの作成

1. 管理するドライバの［Identity Manager ドライバの概要］を開きます。
2. 発行者チャネルの配置ポリシーオブジェクトをクリックします。
3. ［挿入］をクリックします。
4. ポリシーの名前を指定し、このポリシーをポリシービルダで実装することを確認したら、［OK］をクリックします。

ルールビルダが起動します。

事前定義されたルールのインポート

1. ルールビルダで、［挿入］をクリックします。
2. ［配置 - 発行者 (フラット)］を選択します。
3. ルールを編集するため、ルールビルダで［配置 - 発行者 (フラット)］をクリックします。
4. ［文字列を入力］フィールドから、［宛先コンテナの DN を入力してください］を削除します。
5. ［引数の編集］アイコンをクリックして、引数ビルダを起動します。
6. 名詞リストから［テキスト］を選択し、［追加］をクリックします。
7. エディタで、参照アイコンをクリックして、すべてのユーザオブジェクトを配置するターゲットコンテナを参照し、選択したら［OK］をクリックします。
8. ［OK］をクリックします。



ルールでのロジックの動作

このルールは、すべてのユーザオブジェクトをターゲット DN に配置します。このルールでは、ターゲットコンテナの DN をローカル変数 dest-base として設定します。その後で、ターゲット DN を dest-base\CN 属性に設定します。ユーザオブジェクトの CN 属性は、名前属性および名字属性の最初の 2 文字 (小文字) になります。このルールではスラッシュ形式を使用します。

配置 - 購読者 (フラット)-LDAP 形式

識別ボールトのオブジェクトをデータストア内の 1 つのコンテナに配置します。このルールは、ドライバ内の購読者配置ポリシーに実装します。

この事前定義されたルールを使用するには、配置ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する配置ポリシーがすでにある場合は、247 ページの 「事前定義されたルールのインポート」へ進みます。

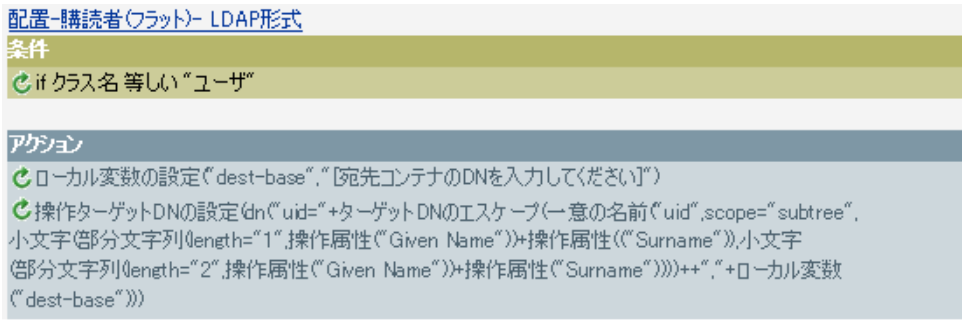
ポリシーの作成

1. 管理するドライバの［Identity Manager ドライバの概要］を開きます。
2. 購読者チャネルの配置ポリシーオブジェクトをクリックします。
3. ［挿入］をクリックします。
4. ポリシーの名前を指定し、このポリシーをポリシービルダで実装することを確認したら、［OK］をクリックします。

ルールビルダが起動します。

事前定義されたルールのインポート

1. ルールビルダで、［挿入］をクリックします。
2. ［配置 - 購読者 (フラット)-LDAP 形式］を選択します。
3. ルールを編集するため、ルールビルダで［配置 - 購読者 (フラット)-LDAP 形式］をクリックします。
4. ［文字列を入力］フィールドから、［宛先コンテナの DN を入力してください］を削除します。
5. ［引数の編集］アイコンをクリックして、引数ビルダを起動します。
6. 名詞リストから［テキスト］を選択し、［追加］をクリックします。
7. エディタで、すべてのユーザオブジェクトを配置するターゲットコンテナを追加します。コンテナが LDAP 形式で指定されていることを確認し、［OK］をクリックします。
8. ［OK］をクリックします。



ルールでのロジックの動作

このルールは、すべてのユーザオブジェクトをターゲット DN に配置します。このルールでは、ターゲットコンテナの DN をローカル変数 dest-base として設定します。その後で、

ターゲット DN を uid= 一意の名前、dest-base に設定します。ユーザオブジェクトの uid 属性は、名前属性および名字属性の最初の 2 文字 (小文字) になります。このルールでは LDAP 形式を使用します。

配置 - 部署別発行者

OU 属性の値に基づいて、オブジェクトを、データストア内の 1 つのコンテナから識別ボールト内の複数のコンテナ内に配置します。このルールは、ドライバ内の発行者配置ポリシーに実装します。

この事前定義されたルールを使用するには、配置ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する配置ポリシーがすでにある場合は、248 ページの 「事前定義されたルールのインポート」へ進みます。

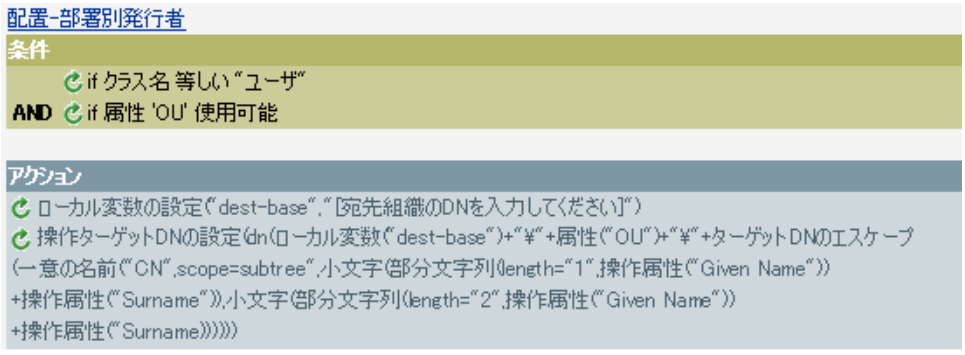
ポリシーの作成

1. 管理するドライバの［Identity Manager ドライバの概要］を開きます。
2. 発行者チャネルの配置ポリシーオブジェクトをクリックします。
3. ［挿入］をクリックします。
4. ポリシーの名前を指定し、このポリシーをポリシービルダで実装することを確認したら、［OK］をクリックします。

ルールビルダが起動します。

事前定義されたルールのインポート

1. ルールビルダで、［挿入］をクリックします。
2. ［配置 - 部署別発行者］を選択します。
3. ルールを編集するため、［配置 - 部署別発行者］をクリックします。
4. ［文字列を入力］フィールドから、［宛先組織の DN を入力してください］を削除します。
5. ［引数の編集］アイコンをクリックして、引数ビルダを起動します。
6. 名詞リストから［テキスト］を選択し、［追加］をクリックします。
7. エディタで、参照アイコンをクリックして、識別ボールト内の親コンテナを参照し、選択します。すべての部署別コンテナがこの DN の子コンテナであることを確認し、［OK］をクリックします。
8. ［OK］をクリックします。



ルールでのロジックの動作

このルールでは、OU 属性に格納された値に基づいて、ユーザオブジェクトを適切な部署に配置します。配置する必要があり、使用可能な OU 属性を持っているユーザオブジェクトの場合は、「dest-base\OU 属性 \CN 属性の値」に配置されます。

dest-base はローカル変数です。DN は、部署別コンテナのルートの相対パスである必要があります。このパスは組織または部門になります。OU 属性に格納された値は、ローカル変数 dest-base の子コンテナ名である必要があります。

OU 属性の値は、子コンテナ名である必要があります。OU 属性が存在しない場合、このルールは実行されません。

ユーザオブジェクトの CN 属性は、名前属性および名字属性の最初の 2 文字 (小文字) になります。このルールではスラッシュ形式を使用します。

配置 - 部署別購読者 -LDAP 形式

OU 属性に基づいて、オブジェクトを、識別ボールト内の 1 つのコンテナからデータストア内の複数のコンテナ内に配置します。このルールは、ドライバ内の配置ポリシーに実装します。このルールは、購読者チャネルにのみ実装できます。

この事前定義されたルールを使用するには、配置ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する配置ポリシーがすでにある場合は、249 ページの 「事前定義されたルールのインポート」へ進みます。

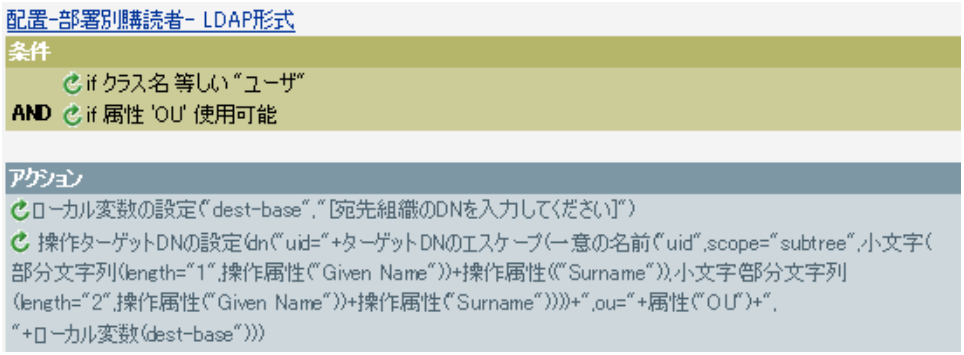
ポリシーの作成

1. 管理するドライバの［Identity Manager ドライバの概要］を開きます。
2. 購読者チャネルの配置ポリシーオブジェクトをクリックします。
3. ［挿入］をクリックします。
4. ポリシーの名前を指定し、このポリシーをポリシービルダで実装することを確認したら、［OK］をクリックします。

ルールビルダが起動します。

事前定義されたルールのインポート

1. ルールビルダで、［挿入］をクリックします。
2. ［配置 - 部署別購読者 -LDAP 形式］を選択します。
3. ルールを編集するため、ルールビルダで［配置 - 部署別購読者 -LDAP 形式］をクリックします。
4. ［文字列を入力］フィールドから、［宛先組織の DN を入力してください］を削除します。
5. ［引数の編集］アイコンをクリックして、引数ビルダを起動します。
6. 名詞リストから［テキスト］を選択し、［追加］をクリックします。
7. エディタで、データストアに親コンテナを追加します。この親コンテナは、LDAP 形式で指定する必要があります。すべての部署別コンテナがこの DN の子コンテナであることを確認し、［OK］をクリックします。
8. ［OK］をクリックします。



ルールでのロジックの動作

このルールでは、OU 属性に格納された値に基づいて、ユーザオブジェクトを適切な部署に配置します。配置する必要があり、使用可能な OU 属性を持っているユーザオブジェクトの場合は、「uid= 一意の名前 ,ou=OU 属性の値 ,dest-base」に配置されます。

dest-base はローカル変数です。DN は、部署別コンテナのルートの相対パスである必要があります。このパスは組織または部門になります。OU 属性に格納された値は、ローカル変数 dest-base の子コンテナ名である必要があります。

OU 属性の値は、子コンテナ名である必要があります。OU 属性が存在しない場合、このルールは実行されません。

ユーザオブジェクトの uid 属性は、名前属性および名字属性の最初の 2 文字 (小文字) になります。このルールでは LDAP 形式を使用します。

3.3 正規表現

正規表現とは、あるパターンに従ったテキスト文字列を照合するための式です。正規表現は、標準文字とメタ文字から構成されます。標準文字には、大文字と小文字、数字があります。メタ文字には特別な意味があります。次の表に、一般的なメタ文字とその意味を示します。

表 *3-1* 一般的な正規表現

メタ文字	説明
.	任意の 1 文字を意味します。
$	行の終わりを意味します。
^	行の先頭を意味します。
*	直前の文字が 0 個以上含まれることを意味します。
\	リテラルのエスケープ文字です。この文字を使用することで、検索対象にすべてのメタ文字を指定できます。たとえば、\$ と指定した場合、行の終わりではなく、$1000 が検索結果になります。
[ ]	角括弧で囲まれた文字のいずれかを意味します。

メタ文字	説明
[0-9]	ハイフンの前後の文字範囲が対象になります。この例では、すべての数字を意味します。
[A-Za-z]	複数の範囲も同様に表します。この例では、すべての大文字と小文字が対象になります。

引数ビルダは、Java* で定義されている正規表現を使用するように設計されています。Java Web サイト (http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html) で詳細情報を参照できます。

3.4 XPath 1.0 の式

条件、アクション、およびトークンの中には、引数で XPath 1.0 の式を使用するものがあります。XPath は、XSLT および XPointer とで共有される機能に対し、共通の構文とセマンティックを提供するために開発された言語です。主に XML ドキュメントのアドレス指定で使用されますが、文字列、数値およびブールなどのデータ操作を行う基本的な機能も備わっています。

XPath の仕様では、埋め込みアプリケーションが、情報を定義された複数のアプリケーションにコンテキストを提供する必要があります。DirXML スクリプト (11 ページのセクション 1.1.2「DirXML スクリプト」を参照) では、XPath は次のコンテキストで評価されます。

- コンテキストのノードが現在の操作。
- コンテキストの位置とサイズが 1。
- 使用可能な変数
  - Identity Manager 内のスタイルシートに対するパラメータとして使用可能なもの (現在のところ、fromNDS、srcQueryProcessor、destQueryProcessor、srcCommandProcessor、destCommandProcessor および dnConverter)。
  - グローバル設定変数。
  - ローカルポリシーの変数。
  - 異なる変数ソース間で名前が衝突している場合は、優先順位はローカル変数、スタイルシートのパラメータ、グローバル変数の順になります。
- ポリシー要素上で宣言されたネームスペース。
- 使用可能な機能
  - 組み込まれている XPath 1.0 のすべての機能
  - NXSL で提供されている Java 拡張機能
    - プリフィックスを Java クラスに関連付けるためのネームスペース宣言は、ポリシー要素で実行される必要があります。

W3 Web サイト (http://www.w3.org/TR/1999/REC-xpath-19991116) で詳細情報を参照できます。

3.5 条件

この節では、ポリシービルダインタフェースで使用できるすべての条件について、詳しく説明します。

3.5.1 If 関連付け

現在の操作または、現在のオブジェクトにある関連付けの値をテストします。

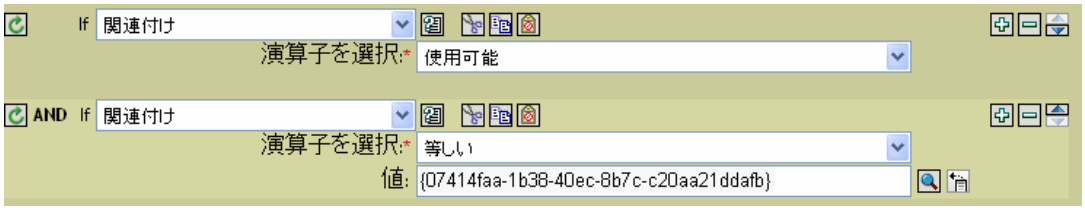
フィールド

次の場合に演算子の条件に一致

演算子	次の場合に条件に一致
関連付けあり	現在のオブジェクトに確立された関連付けがあります。
使用可能	現在の操作で指定された、空ではない関連付けの値があります。
等しい	現在の操作で指定された関連付けの値が、If 関連付けの内容と完全に同じになります。
関連付けなし	現在のオブジェクトには確立された関連付けはありません。
使用不可	現在のオブジェクトでは関連付けを使用できません。
等しくない	現在の操作で指定された関連付けの値が、If 関連付けの内容と異なります。

例

この例では、関連付けが使用可能かどうかを確認しています。この条件が満たされると、定義されたアクションが実行されます。



3.5.2 If 属性

現在の操作または、ソースデータストアにある現在のオブジェクトの属性値をテストします。ソースデータストアまたは操作で条件が一致した場合にテストに適合するので、論理的には If 操作属性または If ソース属性と考えることができます。

フィールド

名前

テストする属性の名前を指定します。

演算子

条件のテストタイプを選択します。

比較モード

比較モードを選択します。332 ページの 「比較モード」を参照してください。

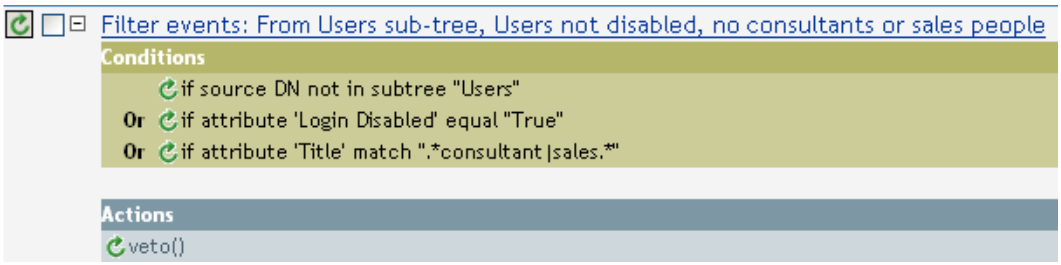
次の場合に演算子の条件に一致

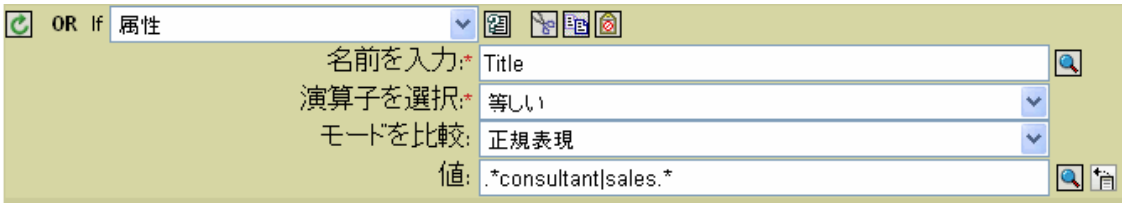
演算子	次の場合に条件に一致
使用可能	現在の操作またはソースデータストアに、指定した属性で使用できる値があります。
等しい	現在の操作またはソースデータストアに指定した属性で使用可能な値があり、指定された比較モードを使用して比較すると、指定した値と同じになります。
使用不可	「使用可能」の場合 False が返されます。
等しくない	「等しい」の場合 False が返されます。

例

この例では、無効または任意の役職名を持つユーザオブジェクトをフィルタ処理する場合に、条件として I f 属性を使用しています。これは「Policy to Filter Events (イベントをフィルタ処理するためのポリシー)」というポリシーで、Novell のサポート Web サイトか

らダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。





この条件では、役職属性がコンサルタントまたは販売担当であるユーザが検索されます。

3.5.3 If クラス名

現在の操作にあるオブジェクトクラス名をテストします。

フィールド

演算子

条件のテストタイプを選択します。

比較モード

比較モードを選択します。332 ページのセクション 3.9.1「比較モード」を参照してください。

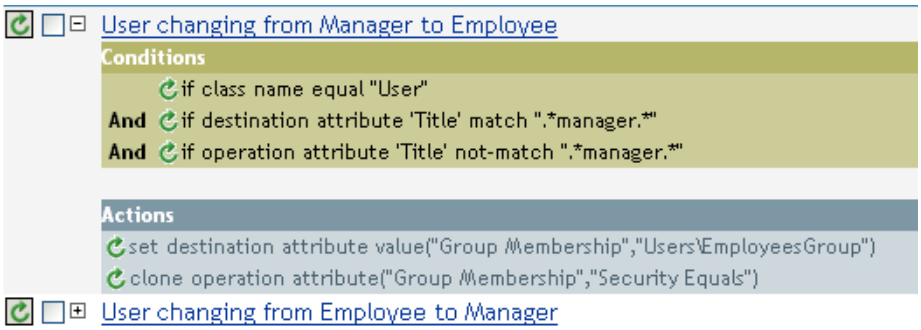
次の場合に演算子の条件に一致

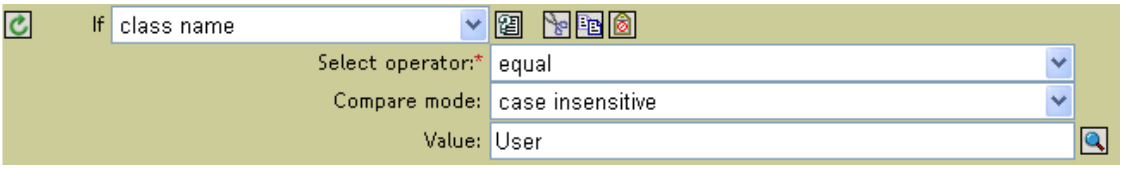
演算子	次の場合に条件に一致
使用可能	現在の操作に使用可能なオブジェクトクラス名があります。
等しい	現在の操作に使用可能なオブジェクトクラス名があり、指定した比較モードを使用して比較すると、指定された値と同じになります。
使用不可	「使用可能」の場合 False が返されます。
等しくない	「等しい」の場合 False が返されます。

例

この例では、役職に基づいてユーザオブジェクトのグループメンバーシップを管理するため、条件として If クラス名を使用しています。これは「Govern Groups for User Based on Title Attribute (役職属性に基づくユーザグループの管理)」というポリシーで、Novell の

サポート Web サイトからダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。





現在のオブジェクトのクラス名が「User」であるかどうかを確認します。

3.5.4 If ターゲット属性

ターゲットデータストアにある現在のオブジェクトの属性値をテストします。

フィールド

名前

テストする属性の名前を指定します。

演算子

条件のテストタイプを選択します。

比較モード

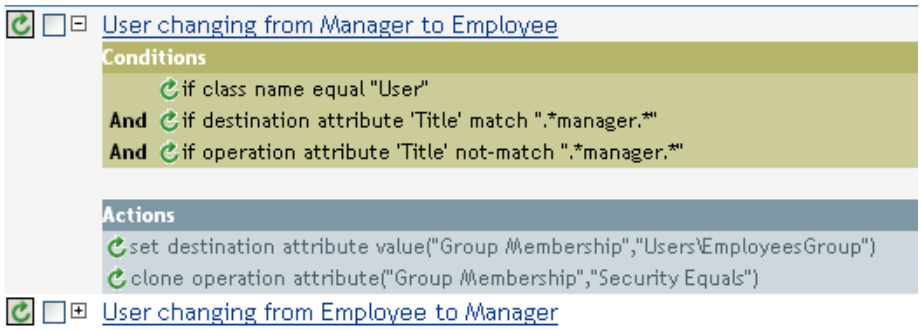
比較モードを選択します。332 ページの 「比較モード」を参照してください。

次の場合に演算子の条件に一致

演算子	次の場合に条件に一致
使用可能	ターゲットデータストアに、指定した属性で使用可能な値があります。
等しい	ターゲットデータストアに指定した属性で使用可能な値があり、指定された比較モードを使用して比較すると、指定された値と同じになります。
使用不可	「使用可能」の場合 False が返されます。
等しくない	「等しい」の場合 False が返されます。

例

この例では、役職に基づいてユーザオブジェクトのグループメンバーシップを管理するため、条件として If 属性を使用しています。これは「Govern Groups for User Based on Title Attribute (役職属性に基づくユーザグループの管理)」というポリシーで、Novell のサポート Web サイトからダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。



このポリシーでは、役職属性に「manager」が含まれているかどうかを確認します。

3.5.5 If ターゲット DN

現在の操作のターゲット DN をテストします。実行されるテストは、指定された演算子によって異なります。

フィールド

演算子

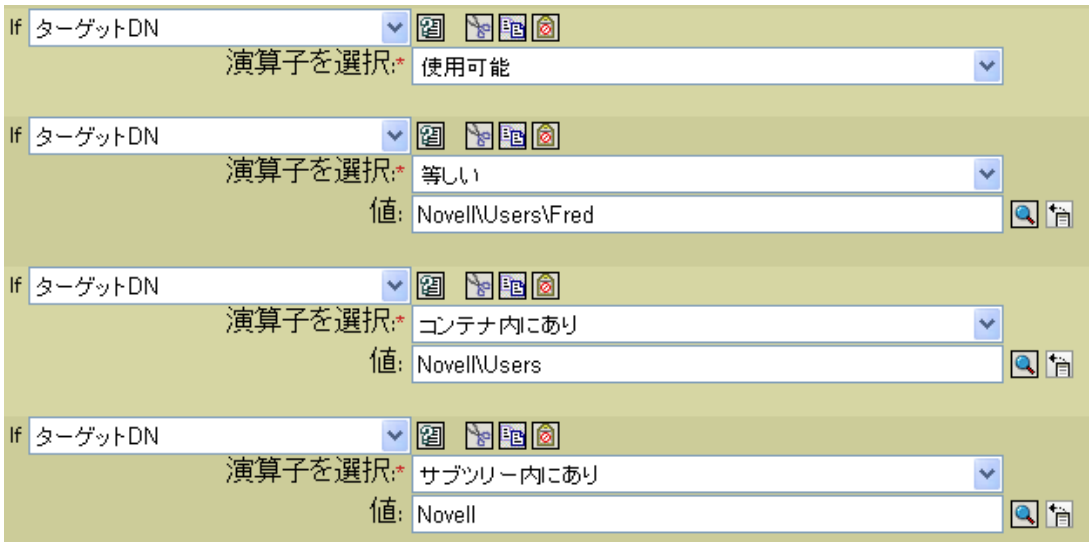
条件のテストタイプを選択します。

次の場合に演算子の条件に一致

演算子	次の場合に条件に一致
使用可能	使用可能なターゲット DN があります。
等しい	使用可能なターゲット DN があり、ターゲットデータストアの DN のフォーマットに適したセマンティックを使用して比較すると、指定された値と同じになります。
コンテナ内にあり	使用可能なターゲット DN があり、ターゲットデータストアの DN のフォーマットに適したセマンティックを使用して比較すると、値で指定されたコンテナ内のオブジェクトを示します。

演算子	次の場合に条件に一致
サブツリー内にあり	使用可能なターゲット DN があり、ターゲットデータストアの DN のフォーマットに適したセマンティックを使用して比較すると、値で指定されたサブツリー内のオブジェクトを示します。
使用不可	「使用可能」の場合 False が返されます。
等しくない	「等しい」の場合 False が返されます。
コンテナ内になし	「コンテナ内にあり」の場合 False が返されます。
サブツリー内になし	「サブツリー内にあり」の場合 False が返されます。

例



3.5.6 If エンタイトルメント

現在の操作または識別ボールトにある現在のオブジェクトのエンタイトルメントをテストします。

フィールド

名前

選択した条件をテストするエンタイトルメントの名前を指定します。

演算子

条件のテストタイプを選択します。

比較モード

比較モードを選択します。332 ページの 「比較モード」を参照してください。

次の場合に演算子の条件に一致

演算子	次の場合に条件に一致
使用可能	現在の操作または識別ボールトで、指定したエンタイトルメントを使用できます。
変更あり	現在の操作に、指定したエンタイトルメントの変更 (属性の変更または属性の追加) が含まれます。
削除指定の変更あり	現在の操作に、指定したエンタイトルメントの値を削除する変更 (値の削除) が含まれ、指定された比較モードを使用して比較すると、指定された値と同じ値があります。
追加指定の変更あり	現在の操作に、指定したエンタイトルメントに値を追加する変更 (値の追加または属性の追加) が含まれます。指定された比較モードを使用して比較すると、指定された値と同じ値があります。
等しい	ターゲットデータストアに指定した属性に使用可能な値があり、指定された比較モードを使用して比較すると、指定された値と同じになります。
使用不可	「使用可能」の場合 False が返されます。
変更なし	「変更あり」の場合 False が返されます。
削除指定の変更なし	「削除指定の変更あり」の場合 False が返されます。
追加指定の変更なし	「追加指定の変更あり」の場合 False が返されます。
等しくない	「等しい」の場合 False が返されます。

例

3.5.7 If グローバル構成値

グローバル設定変数をテストします。

フィールド

名前

選択した条件をテストするグローバル変数の名前を指定します。

演算子

条件のテストタイプを選択します。

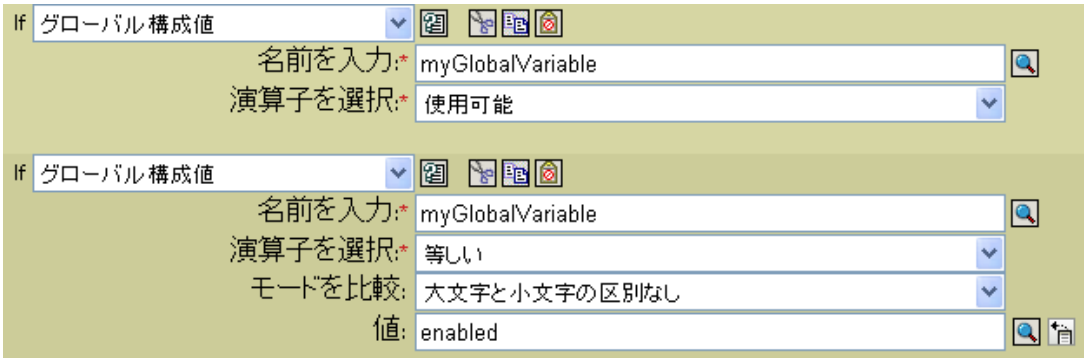
比較モード

比較モードを選択します。332 ページの 「比較モード」 を参照してください。

次の場合に演算子の条件に一致

演算子	次の場合に条件に一致
使用可能	指定した名前のグローバル設定変数があります。
等しい	指定した名前のグローバル設定変数があり、その値が、指定された比較モードを使用して比較すると、指定された値と同じになります。
使用不可	「使用可能」の場合 False が返されます。
等しくない	「等しい」の場合 False が返されます。

例



3.5.8 If ローカル変数

ローカル変数をテストします。

フィールド

名前

選択した条件をテストするローカル変数の名前を指定します。

演算子

条件のテストタイプを選択します。

比較モード

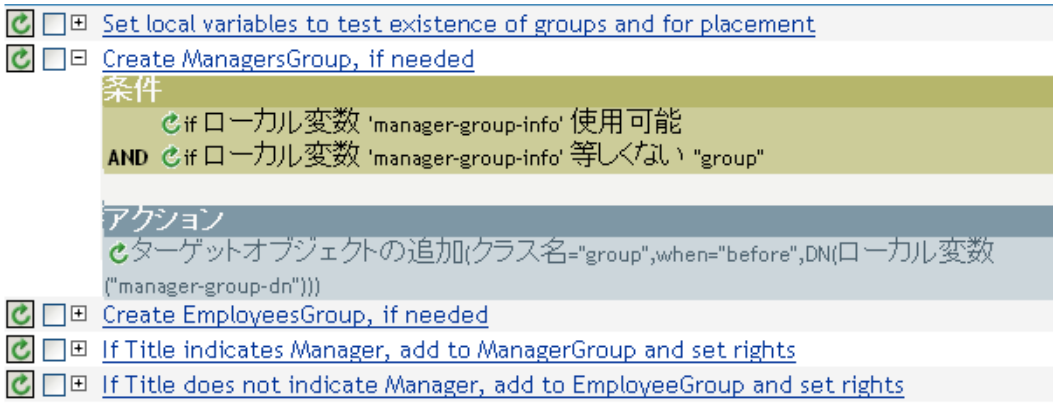
比較モードを選択します。332 ページの 「比較モード」を参照してください。

次の場合に演算子の条件に一致

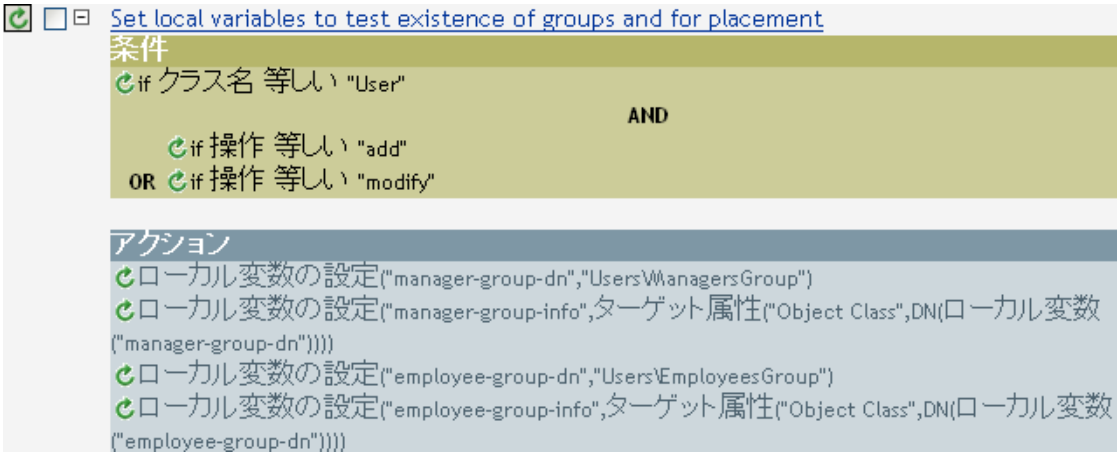
演算子	次の場合に条件に一致
使用可能	ポリシー内にある以前のルールのアクションですでに定義されている、指定した名前のローカル変数があります。
等しい	指定した名前のローカル変数があり、その値が、指定された比較モードを使用して比較すると、指定された値と同じになります。
使用不可	「使用可能」の場合 False が返されます。
等しくない	「等しい」の場合 False が返されます。

例

この例では、役職に基づいて、ユーザオブジェクトを適切なグループ (従業員またはマネージャ) に追加します。必要に応じてグループも作成し、そのグループに同等のセキュリティを設定します。これは「Govern Groups for User Based on Title Attribute (役職属性に基づくユーザグループの管理)」というポリシーで、Novell のサポート Web サイトからダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。



このポリシーには、互いに依存する 5 つのルールが含まれています。



If ローカル変数の条件を動作させるため、最初のルールで 4 つのローカル変数が設定され、グループとそのグループの配置場所がテストされます。

ルールが検索する条件では、ローカル変数 manager-group-info が使用可能かどうか、およびこの変数がグループと等しくないかどうかが確認されます。これらの条件が満たされると、グループのターゲットオブジェクトが追加されます。

3.5.9 If 名前付きパスワード

現在の操作にあるパスワードを、指定された名前でテストします。

フィールド

名前

選択した条件をテストする名前付きパスワードの名前を指定します。

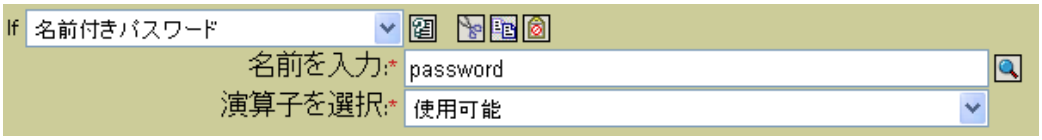
演算子

条件のテストタイプを選択します。

次の場合に演算子の条件に一致

演算子	次の場合に条件に一致
使用可能	指定した名前でパスワードを使用できます。
使用不可	「使用可能」の場合 False が返されます。

例



3.5.10 If 操作

現在の操作の名前をテストします。

フィールド

演算子

条件のテストタイプを選択します。

次の場合に演算子の条件に一致

演算子	次の場合に条件に一致
等しい	現在の操作の名前が、If 操作の内容と完全に同じです。
等しくない	「等しい」の場合 False が返されます。

値

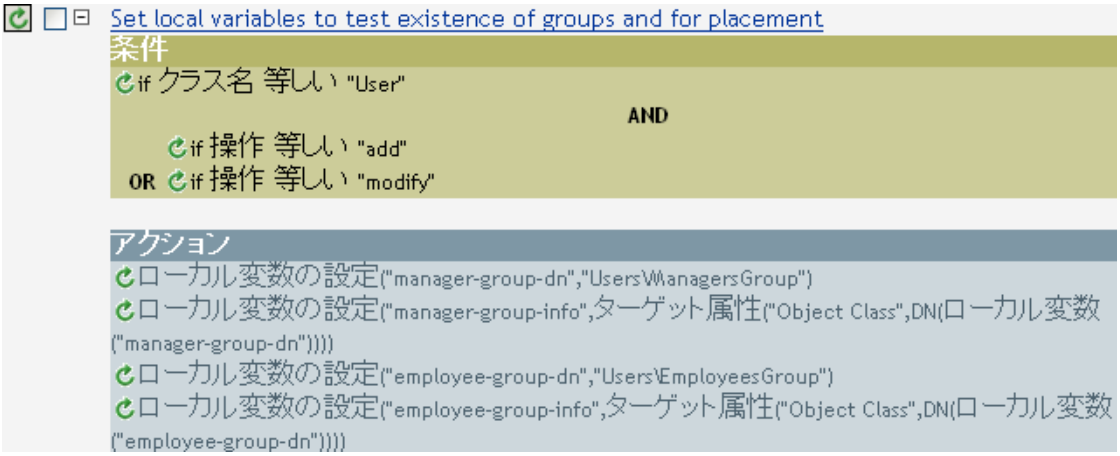
値は、メタディレクトリエンジンがこの条件で検索する操作です。

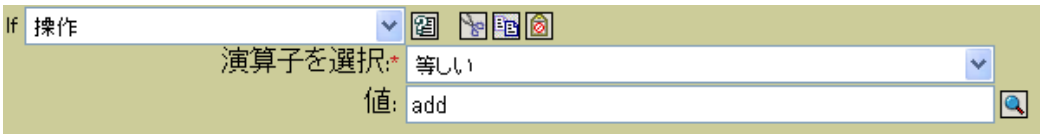
- 追加
- 関連付けの追加
- オブジェクトパスワードの確認
- 削除

- 名前付きパスワードの取得
- 変更
- 関連付けの変更
- パスワード変更
- 移動
- パラメータの開始
- インスタンス

例

この例では、役職に基づいて、ユーザオブジェクトを適切なグループ (従業員またはマネージャ) に追加します。必要に応じてグループも作成し、そのグループに同等のセキュリティを設定します。これは「Govern Groups for User Based on Title Attribute (役職属性に基づくユーザグループの管理)」という名前のポリシーで、Novell のサポート Web サイトからダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。





この条件では、追加または変更の操作が発生したかどうかを確認しています。これらのいずれかが発生した場合、ローカル変数が設定されます。

3.5.11 If 操作属性

現在の操作の属性値をテストします。実行されるテストは、指定された演算子によって異なります。

フィールド

名前

テストする属性の名前を指定します。

演算子

条件のテストタイプを選択します。

比較モード

比較モードを選択します。332 ページの 「比較モード」を参照してください。

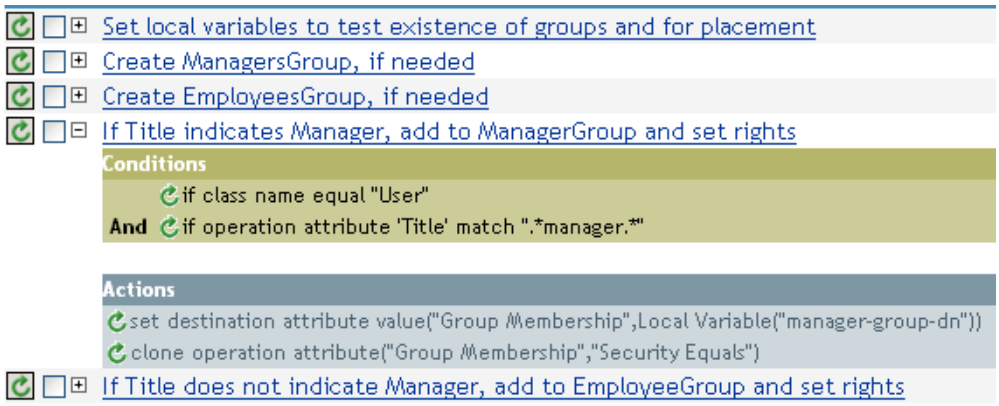
次の場合に演算子の条件に一致

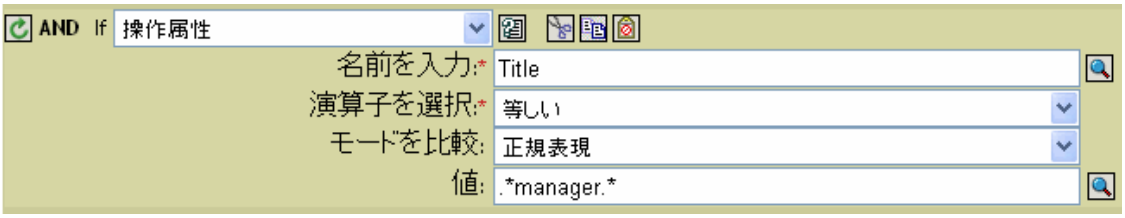
演算子	次の場合に条件に一致
使用可能	現在の操作 (属性の追加、値の追加、属性) に、指定した属性で使用できる値があります。
変更あり	現在の操作に、指定した属性の変更 (属性の変更または属性の追加) があります。
削除指定の変更あり	現在の操作に、指定した属性の値を削除する変更 (値の削除) があります。指定された比較モードを使用して比較すると、指定された値と同じになります。
追加指定の変更あり	現在の操作に、指定した属性に値を追加する変更 (値の追加または属性の追加) が含まれます。指定された比較モードを使用して比較すると、指定された値と同じになります。
等しい	現在の操作 (値の削除以外) に、指定した属性で使用できる値があります。指定された比較モードを使用して比較すると、指定された値と同じになります。
使用不可	「使用可能」の場合 False が返されます。
変更なし	「変更あり」の場合 False が返されます。
削除指定の変更なし	「削除指定の変更あり」の場合 False が返されます。
追加指定の変更なし	「追加指定の変更あり」の場合 False が返されます。
等しくない	「等しい」の場合 False が返されます。

例

この例では、役職に基づいて、ユーザオブジェクトを適切なグループ (従業員またはマネージャ) に追加します。必要に応じてグループも作成し、そのグループに同等のセキュリティを設定します。これは「Govern Groups for User Based on Title Attribute (役職属性に基づくユーザグループの管理)」という名前のポリシーで、Novell のサポート Web サイト

からダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。





この条件では、役職属性が正規表現「.*manager.*」に等しいかどうかを確認しています。つまり、manager の前に 0 個以上の文字を持ち、manager の後に 1 文字を持つ役職を検索しています。ユーザオブジェクトの役職が sales managers であった場合、一致として検出されます。

3.5.12 If 操作プロパティ

現在の操作の操作プロパティをテストします。

フィールド

名前

選択した条件をテストする操作プロパティの名前を指定します。

演算子

条件のテストタイプを選択します。

比較モード

比較モードを選択します。332 ページの 「比較モード」を参照してください。

次の場合に演算子の条件に一致

演算子	次の場合に条件に一致
使用可能	現在の操作に、指定した名前の操作プロパティがあります。

演算子	次の場合に条件に一致
等しい	指定した名前の操作プロパティが現在の操作にあり、その値が、指定された比較モードを使用して比較すると、指定された内容と同じになります。
使用不可	「使用可能」の場合 False が返されます。
等しくない	「等しい」の場合 False が返されます。

例

3.5.13 If パスワード

現在の操作のパスワードをテストします。

フィールド

演算子

条件のテストタイプを選択します。

次の場合に演算子の条件に一致

演算子	次の場合に条件に一致
使用可能	現在の操作に使用可能なパスワードがあります。
使用不可	「使用可能」の場合 False が返されます。

例

3.5.14 If ソース属性

ソースデータストアにある現在のオブジェクトの属性値をテストします。

フィールド

名前

選択した条件をテストするソース属性の名前を指定します。

演算子

条件のテストタイプを選択します。

比較モード

比較モードを選択します。332 ページのセクション 3.9.1「比較モード」を参照してください。

次の場合に演算子の条件に一致

演算子	次の場合に条件に一致
使用可能	ソースデータストアに、指定した属性で使用可能な値があります。
等しい	ソースデータストアに、指定した属性で使用可能な値があります。指定された比較モードを使用して比較すると、指定された値と同じになります。
使用不可	「使用可能」の場合 False が返されます。
等しくない	「等しい」の場合 False が返されます。

フィールド

名前

選択した条件をテストするソース属性の名前を指定します。

演算子

条件のテストタイプを選択します。

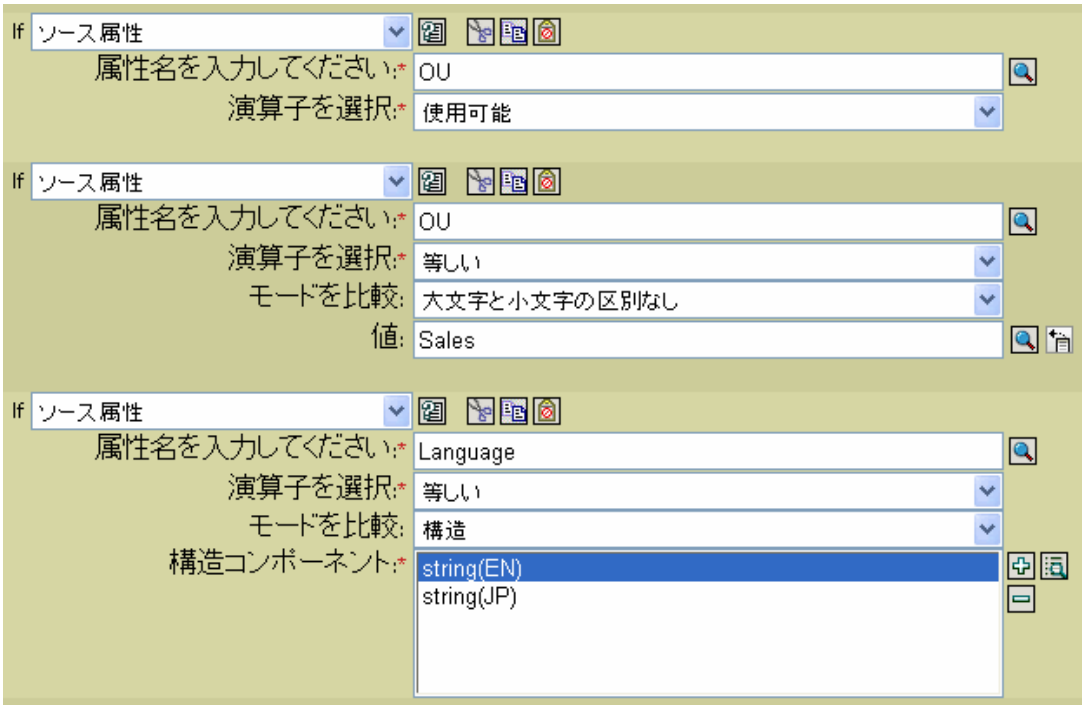
比較モード

比較モードを選択します。332 ページの 「比較モード」を参照してください。

次の場合に演算子の条件に一致

演算子	次の場合に条件に一致
使用可能	ソースデータストアに、指定した属性で使用可能な値があります。
等しい	ソースデータストアに、指定した属性で使用可能な値があります。指定された比較モードを使用して比較すると、指定された値と同じになります。
使用不可	「使用可能」の場合 False が返されます。
等しくない	「等しい」の場合 False が返されます。

例



3.5.15 If ソース DN

現在の操作のソース DN をテストします。

フィールド

演算子

条件のテストタイプを選択します。

次の場合に演算子の条件に一致

演算子	次の場合に条件に一致
使用可能	使用可能なソース DN があります。
等しい	使用可能なソース DN があり、コンテナ内の指定された値と一致します。使用可能なソース DN があり、指定された値で識別されるコンテナ内のオブジェクトを示しています。
サブツリー内にあり	使用可能なソース DN があり、指定された値で識別されるサブツリー内のオブジェクトを示しています。
使用不可	「使用可能」の場合 False が返されます。
等しくない	「等しい」の場合 False が返されます。
コンテナ内になし	「コンテナ内にあり」の場合 False が返されます。
サブツリー内になし	「サブツリー内にあり」の場合 False が返されます。

フィールド

演算子

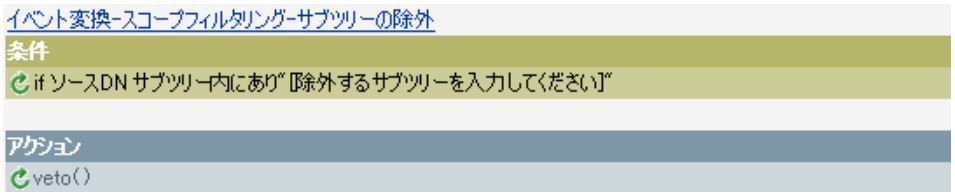
条件のテストタイプを選択します。

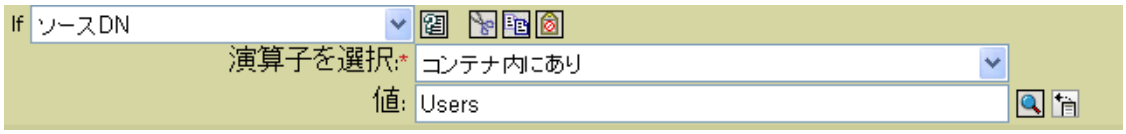
次の場合に演算子の条件に一致

演算子	次の場合に条件に一致
使用可能	使用可能なソース DN があります。
等しい	使用可能なソース DN があり、コンテナ内の指定された値と一致します。使用可能なソース DN があり、指定された値で識別されるコンテナ内のオブジェクトを示しています。
サブツリー内にあり	使用可能なソース DN があり、指定された値で識別されるサブツリー内のオブジェクトを示しています。
使用不可	「使用可能」の場合 False が返されます。
等しくない	「等しい」の場合 False が返されます。
コンテナ内になし	「コンテナ内にあり」の場合 False が返されます。
サブツリー内になし	「サブツリー内にあり」の場合 False が返されます。

例

この例では、ユーザオブジェクトがソース DN にあるかどうかを確認する条件として、If ソース DN を使用しています。このルールは、Identity Manager 3.0 に付属している事前定義されたルールです。詳細については、237 ページの 「イベント変換 - スコープフィルタリング - サブツリーの除外」を参照してください。





この例では、ソース DN がユーザコンテナにあるかどうかを確認しています。オブジェクトがこのコンテナ内にある場合は、拒否されます。

3.5.16 If XPath 式

XPath 1.0 の式の評価結果をテストします。

フィールド

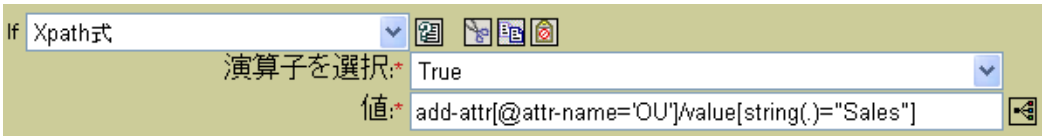
演算子

条件のテストタイプを選択します。

次の場合に演算子の条件に一致

演算子	次の場合に条件に一致
true	XPath 式が True に評価されます。
false	「True」の場合 False が返されます。

例



3.6 アクション

この節では、ポリシービルダインタフェースで使用できるすべてのアクションについて、詳しく説明します。

3.6.1 関連付けの追加

指定した関連付けと共に、関連付けの追加コマンドを識別ボールトに送信します。

フィールド

モード

このアクションを現在の操作に追加するか、または識別ボールトへ直接書き込むかを選択します。

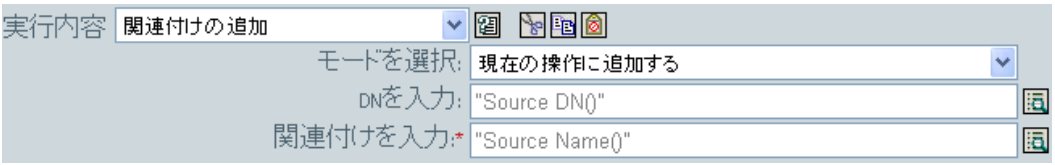
DN

ターゲットオブジェクトの DN を指定するか、または空白のままにして現在のオブジェクトを使用します。

関連付け

追加する関連付けの値を指定します。

例



3.6.2 ターゲット属性値の追加

ターゲットデータストア内のオブジェクトの属性に値を追加します。

フィールド

属性名

属性の名前を指定します。

クラス名

(オプション) ターゲットオブジェクトのクラス名を指定します。現在のオブジェクトのクラス名を使用するには、空白のままにします。

モード

このアクションを現在の操作の前と後のどちらに追加するか、または目的のデータストアへ直接書き込むかを選択します。

オブジェクト

ターゲットオブジェクトを選択します。このオブジェクトは現在のオブジェクトにすることも、DN または関連付けによって指定することもできます。

値のタイプ

追加する属性値の構文を選択します。

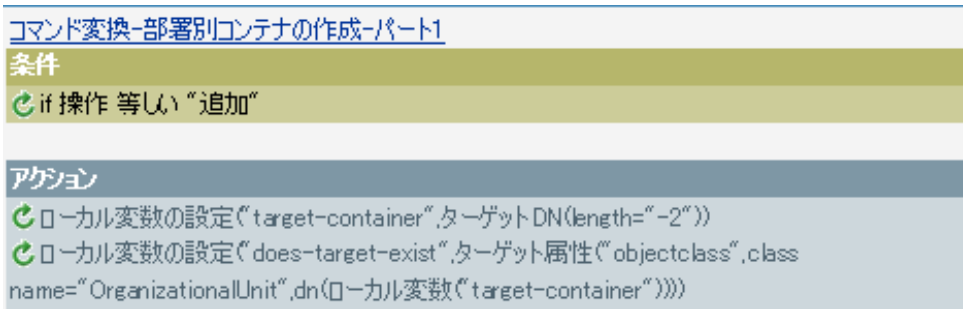
値

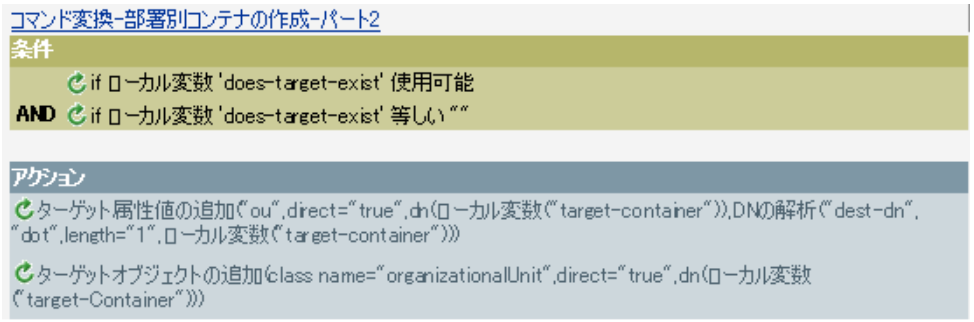
追加する属性値を指定します。

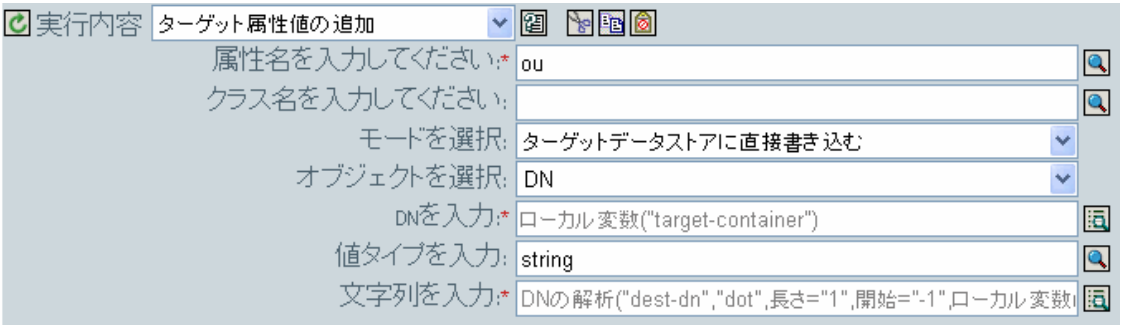
例

この例では、ターゲット属性値を OU 属性に追加します。作成されたローカル変数から値を生成します。このルールは、Identity Manager 3.0 に付属している事前定義されたルール

です。詳細については、229 ページの 「コマンド変換 - 部署別コンテナの作成 - パート 1 とパート 2」を参照してください。







3.6.3 ターゲットオブジェクトの追加

ターゲットデータストアに、指定されたタイプの新しいオブジェクトを作成します。

フィールド

クラス名

作成するオブジェクトのクラス名を指定します。

モード

このアクションを現在の操作の前と後のどちらに追加するか、または目的のデータストアへ直接書き込むかを選択します。

DN

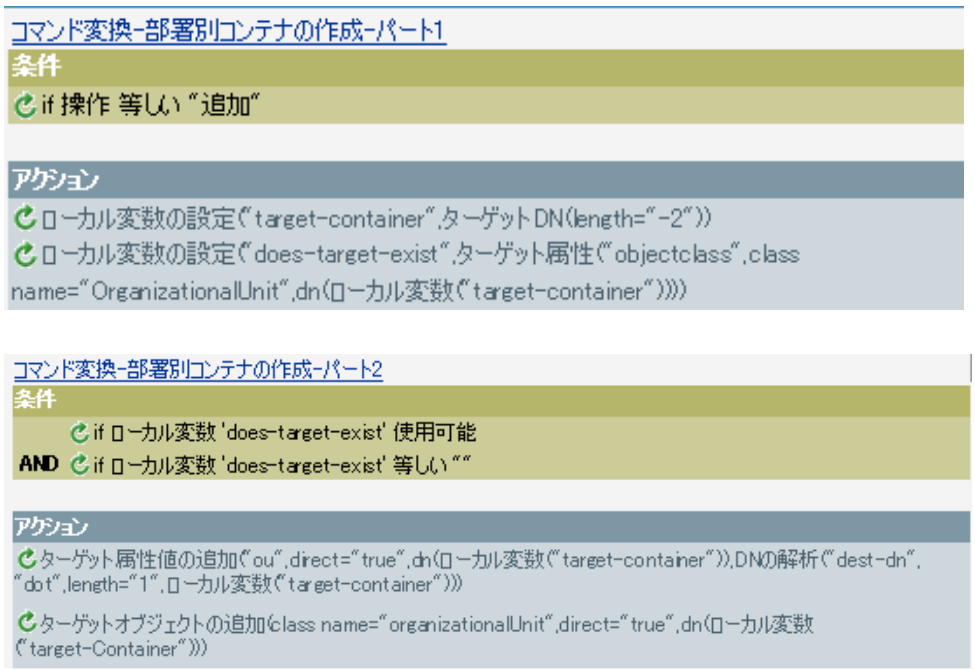
作成するオブジェクトの DN を指定します。

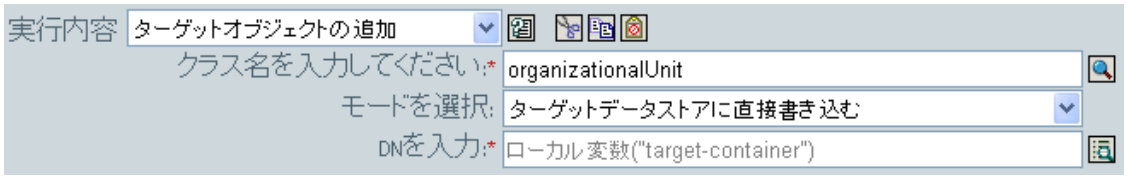
備考

オブジェクト作成の一部として追加される任意の属性値は、次の 272 ページの 「ターゲット属性値の追加」アクションで同じ DN を使って追加する必要があります。

例

この例では、必要な部署別コンテナを作成します。このルールは、Identity Manager 3.0 に付属している事前定義されたルールです。詳細については、229 ページの 「コマンド変換 - 部署別コンテナの作成 - パート 1 とパート 2」を参照してください。





OU オブジェクトが作成されます。OU 属性の値は、このアクションの後に発生するターゲット属性値のアクションから作成されます。

3.6.4 ソース属性値の追加

ソースデータストア内のオブジェクトに指定した属性に、指定した値を追加します。ターゲットオブジェクトは現在のオブジェクト、DN または関連付けです。

フィールド

属性名

属性の名前を指定します。

クラス名

(オプション) ターゲットオブジェクトのクラス名を指定します。現在のオブジェクトのクラス名を使用するには、空白のままにします。

オブジェクト

ターゲットオブジェクトを選択します。このオブジェクトは現在のオブジェクトにすることも、DN または関連付けによって指定することもできます。

値のタイプ

追加する属性値の構文を選択します。

値

追加する属性値を指定します。

例

3.6.5 ソースオブジェクトの追加

ソースデータストア内に作成される、指定されたタイプのオブジェクトを作成します。オブジェクト作成の一部として追加される任意の属性値は、次の (274 ページ) ソース属性値の追加アクションで同じ DN を使って追加する必要があります。

フィールド

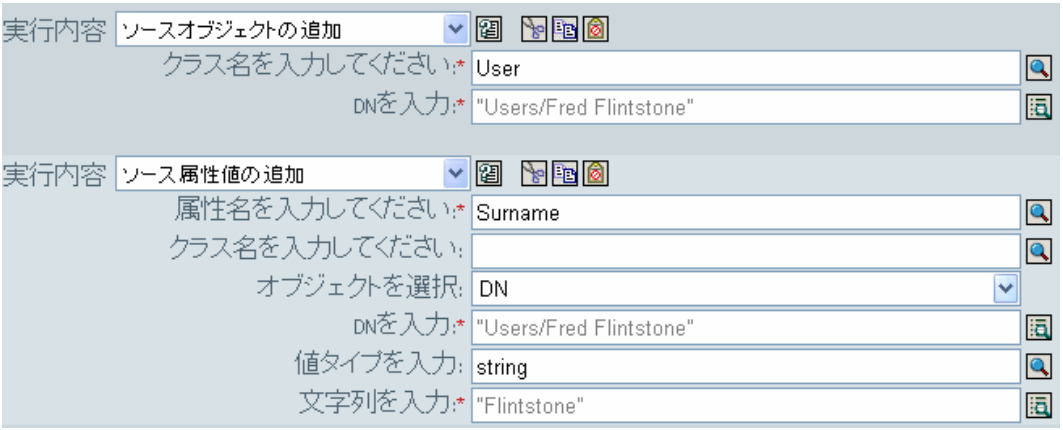
クラス名

追加するオブジェクトのクラス名を指定します。

DN

追加するオブジェクトの DN を指定します。

例



フィールド

クラス名

ソースデータストアに追加するオブジェクトのクラス名を指定します。

DN

ソースデータストアに追加する新しいオブジェクトの DN を指定します。

3.6.6 XML 要素の追加

XPath 式で選択された要素のセットに要素を追加します。

フィールド

名前

XML 要素のタグ名を指定します。この名前には、前にこのポリシーで定義されているネームスペースプリフィックスを含めることができます。

XPATH 式

新しい要素の追加先になる要素を含むノードセットを返す XPath 1.0 の式を指定します。

例

3.6.7 XML テキストの追加

XPath 式で選択された要素のセットにテキストを追加します。

フィールド

XPATH 式

新しい要素の追加先になる要素を含むノードセットを返す XPath 1.0 の式。

文字列

追加するテキストを指定します。

例

3.6.8 中断

現在のポリシーによる現在の操作の処理を終了します。

例

3.6.9 ターゲット属性値のクリア

ターゲットデータストア内のオブジェクトから、名前付き属性のすべての値を削除します。

フィールド

属性名

属性の名前を指定します。

クラス名

(オプション) ターゲットオブジェクトのクラス名を指定します。現在のオブジェクトのクラス名を使用するには、空白のままにします。

モード

このアクションを現在の操作の前と後のどちらに追加するか、または目的のデータストアへ直接書き込むかを選択します。

オブジェクト

ターゲットオブジェクトを選択します。このオブジェクトは現在のオブジェクトにすることも、DN または関連付けによって指定することもできます。

例

3.6.10 操作プロパティのクリア

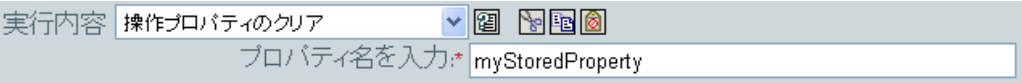
操作プロパティの現在の操作をクリアします。

フィールド

プロパティ名

クリアする操作プロパティの名前を指定します。

例



3.6.11 SSO 資格情報のクリア

シングルサインオンの資格情報をクリアし、オブジェクトのプロビジョニングを解除できるようにします。このアクションは、資格情報のプロビジョニングポリシーの一部です。詳細については、333 ページの第 4 章「Novell 資格情報プロビジョニングポリシー」を参照してください。

フィールド

資格情報ストアオブジェクトの **DN**

リポジトリオブジェクトの DN を指定します。

ターゲットユーザの **DN**

ターゲットユーザの DN を指定します。

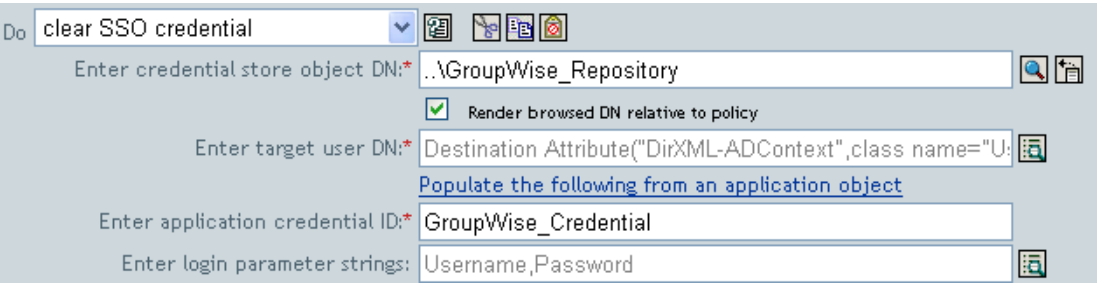
アプリケーションのアクティベーションキー **ID**

アプリケーションオブジェクト内に格納されたアプリケーションの資格情報を指定します。

ログインパラメータの文字列

アプリケーションのログインパラメータを指定します。ログインパラメータとは、アプリケーションオブジェクト内に格納されている認証キーです。

例



3.6.12 ソース属性値のクリア

ソースデータストア内の 1 つのオブジェクトから、すべての属性値を削除します。

フィールド

属性名

属性の名前を指定します。

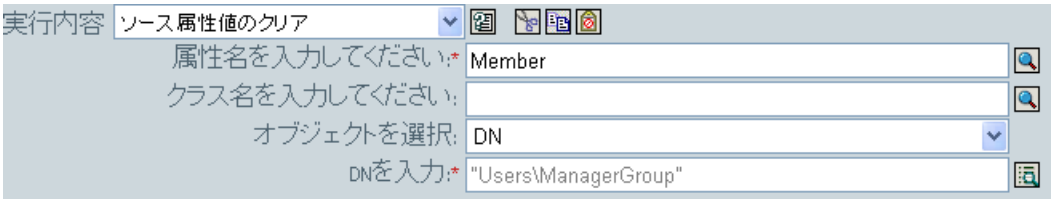
クラス名

(オプション) ターゲットオブジェクトのクラス名を指定します。現在のオブジェクトのクラス名を使用するには、空白のままにします。

オブジェクト

ターゲットオブジェクトを選択します。このオブジェクトは現在のオブジェクトにすることも、DN または関連付けによって指定することもできます。

例



3.6.13 XPath 式によるクローン

XPath 式で選択された XML ノードのセットの詳細コピーを、他の XPath 式で選択された要素のセットに追加します。

フィールド

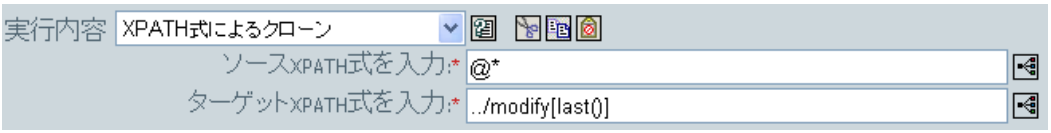
ソース **XPATH** 式

コピーされるノードを含むノードセットを返す XPath 1.0 の式を指定します。

ターゲット **XPATH** 式

コピーされたノードの追加先になる要素を含むノードセットを返す XPath 1.0 の式を指定します。

例



3.6.14 操作属性のクローン

現在の操作で属性に行った内容を、現在の操作内の別の属性にコピーします。

フィールド

ソース名

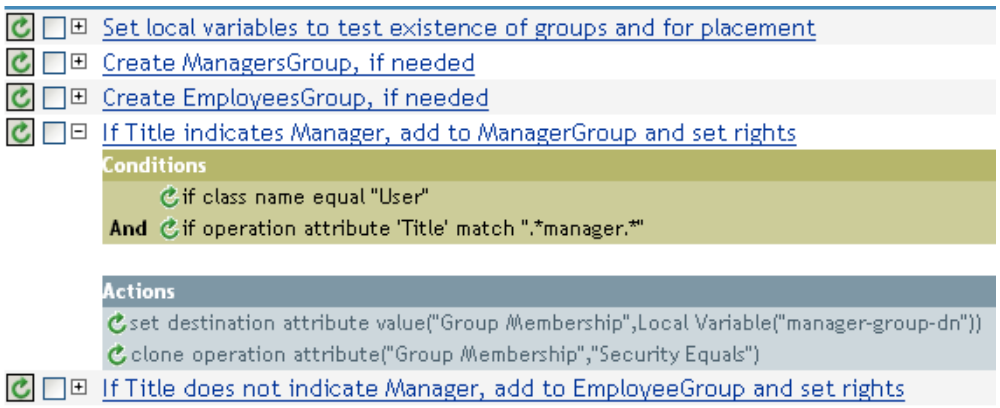
コピー元の属性の名前を指定します。

ターゲット名

コピー先の属性の名前を指定します。

例

この例では、役職に基づいて、ユーザオブジェクトを適切なグループ (従業員またはマネージャ) に追加します。必要に応じてグループも作成し、そのグループに同等のセキュリティを設定します。これは「Govern Groups for User Based on Title Attribute (役職属性に基づくユーザグループの管理)」というポリシーで、Novell のサポート Web サイトからダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。



操作属性のクローンでは、グループメンバーシップ属性から情報を取得し、これに同等セキュリティを追加して同じ値になるようにします。

3.6.15 ターゲットオブジェクトの削除

ターゲットデータストア内のオブジェクトを削除します。

フィールド

モード

このアクションを現在の操作の前と後のどちらに追加するか、または目的のデータストアへ直接書き込むかを選択します。

オブジェクト

ターゲットデータストア内の削除するターゲットオブジェクトを選択します。このオブジェクトは現在のオブジェクトにすることも、DN または関連付けによって指定することもできます。

例

3.6.16 ソースオブジェクトの削除

ソースデータストア内のオブジェクトを削除します。

フィールド

モード

このアクションを現在の操作の前と後のどちらに追加するか、または目的のデータストアへ直接書き込むかを選択します。

オブジェクト

ソースデータストア内の削除するターゲットオブジェクトを選択します。このオブジェクトは現在のオブジェクトにすることも、DN または関連付けによって指定することもできます。

例

3.6.17 一致オブジェクトの検索

ターゲットデータストアにある現在のオブジェクトに一致するものを検索します。

フィールド

スコープ

検索スコープを選択します。スコープは［エントリ］、［サブオーディネート］、または［サブツリー］になります。

DN

検索のベースとなる DN を指定します。

一致属性

検索する属性値を指定します。

備考

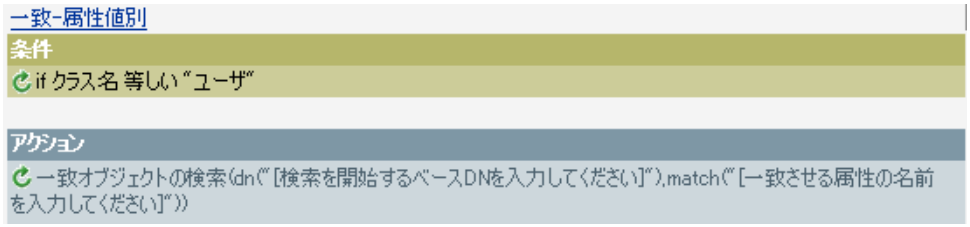
一致オブジェクトの検索は、現在の操作が追加の場合にのみ有効です。

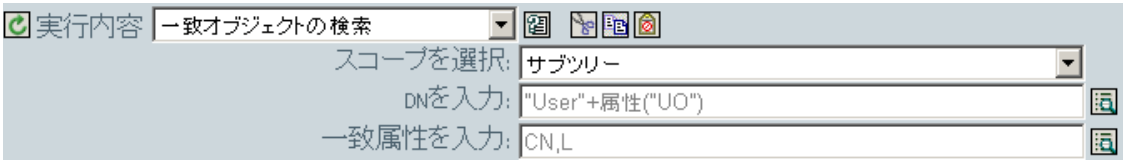
DN 引数は、スコープが［エントリ］の場合のみ必須で、それ以外の場合はオプションです。スコープが［サブツリー］または［サブオーディネート］の場合には、少なくとも 1 つの一致属性が必要です。スコープが［エントリ］の場合には結果は定義されず、一致属性が指定されます。ターゲットデータストアが接続アプリケーションである場合は、一致結果が返されるごとに関連付けが現在の操作に追加されます。現在の操作に空でない関連付けがすでにある場合はクエリが実行されないので、同じルール内に一致オブジェクトの検索アクションを複数指定しても問題ありません。

ターゲットデータストアが識別ボールトである場合は、現在の操作のターゲット DN 属性が設定されます。現在の操作にすでに空でないターゲット DN 属性がある場合はクエリが実行されないので、同じルール内に一致オブジェクトの検索アクションを複数指定しても問題ありません。結果が 1 つだけ返され、それがまだ関連付けられていない場合は、現在の操作のターゲット DN が一致オブジェクトのソース DN に設定されます。結果が 1 つだけ返され、それがすでに関連付けられている場合は、現在の操作のターゲット DN が 1 文字の ￼ に設定されます。複数の結果が返される場合は、現在の操作のターゲット DN が 1 文字の &#xFFFD; に設定されます。

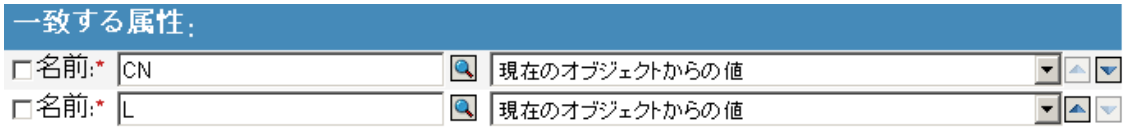
例

この例では、ユーザオブジェクトで属性 CN と L を使用して照合します。ルールが検索している場所のユーザコンテナから始まり、OU 属性内に格納された情報を DN に追加します。このルールは、Identity Manager 3.0 に付属している事前定義されたルールです。詳細については、94 ページの 「一致 - 属性値別」を参照してください。





引数ビルダのアイコンをクリックすると、一致属性ビルダが開きます。ビルダ上で照合する属性を指定します。この例では、CN および L の属性を使用しています。



3.6.18 For Each

ノードセット内の各ノードに対し、アクションのセットを繰り返します。

フィールド

ノードセット

ノードセットを指定します。

アクション

ノードセットの各ノードに対して実行するアクションを指定します。

備考

ローカル変数が使用される場合、アクションを繰り返すと、現在のノードは異なる値になります。

ノードセット内のノードがエンタイトルメントである場合、それぞれに対して黙示的に287 ページの 「エンタイトルメントの実装」アクションを実行します。

例

次に示すのは、引数アクションビルダの例で、アクションの引数を指定する場合に使用されます。

3.6.19 イベントの生成

ユーザ定義イベントを Novell Audit に送信します。

フィールド

ID

イベントの ID。java.lang.Integer の parseInt メソッドを使用して解析したときに、1000 ～ 1999 の整数になる値を指定する必要があります。

レベル

イベントのレベル。

レベル	説明
log-emergency	メタディレクトリエンジンまたはドライバがシャットダウンされるイベント。
log-alert	早急に注意が必要なイベント。
log-critical	メタディレクトリエンジンまたはドライバの一部が正常に動作しなくなるイベント。
log-error	メタディレクトリエンジンまたはドライバによって処理できるエラーを示すイベント。
log-warning	大きな問題としては取り上げられないネガティブなイベント。
log-notice	管理者が使い方や操作を理解または向上するのに使用できるイベント (ポジティブまたはネガティブ)。
log-info	何らかの重要性を持つポジティブなイベント。
log-debug	サポート担当者またはエンジニアがメタディレクトリエンジンまたはドライバの操作をデバッグするためのイベント。

文字列

イベントに含めるユーザ定義の文字列値、整数値、およびバイナリ値を指定します。これらの値は、名前付き文字列ビルダを使用して指定します。

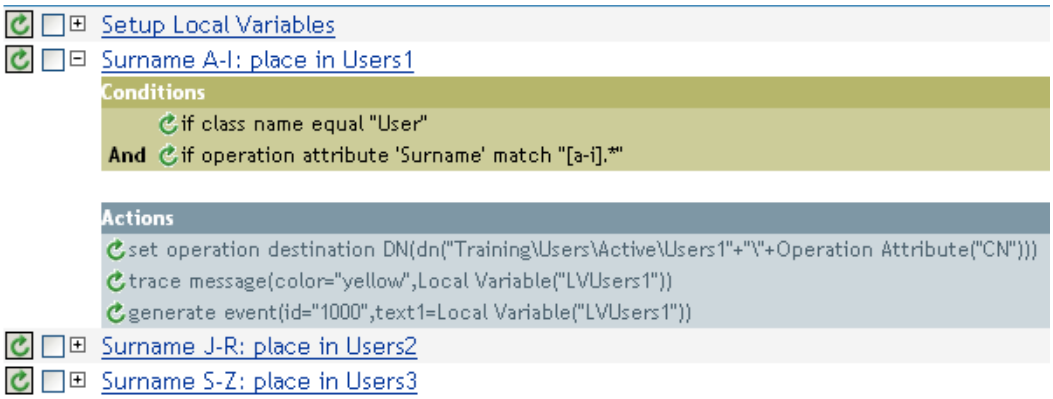
タグ	説明
target	イベントの対象になるオブジェクト。
target-type	ターゲットの定義済みの形式を示す整数。現在定義済みの target-type の値を示します。 ◆ 0 = なし ◆ 1 = スラッシュ表記 ◆ 2 = ドット表記 ◆ 3 = LDAP 表記
subTarget	イベントの対象になるターゲットのサブコンポーネント。
text1	ここに入力されるテキストは、text1 イベントフィールドに格納されます。
text2	ここに入力されるテキストは、text2 イベントフィールドに格納されます。
text3	ここに入力されるテキストは、text3 イベントフィールドに格納されます。
value	ここに入力される任意の数字は、value イベントフィールドに格納されます。
value3	ここに入力される任意の数字は、value3 イベントフィールドに格納されます。
data	ここに入力されるデータは、Blob イベントフィールドに格納されます。

備考

Novell Audit イベント構造には、1 つのターゲット、1 つのサブターゲット、3 つの文字列 (text1、text2、text3)、2 つの整数 (value、value3)、および 1 つの一般的なフィールド (data) が含まれます。テキストフィールドは 256 バイトに制限されています。データフィールドには 3KB までの情報を含めることができます。ただし、環境によってはこれより大きいデータフィールドを使用できる場合もあります。

例

この例には 4 つのルールがあり、これらのルールでは名字属性の最初の文字に基づいてユーザオブジェクトに配置ポリシーを実装し、トレースメッセージおよびカスタムの Novell Audit イベントの両方を生成します。イベントの生成アクションは、Novell Audit にイベントを送信する場合に使用されます。これは、「Policy to Place by Surname (名字で配置するためのポリシー)」という名前のポリシーで、Novell のサポート Web サイトからダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。



次に示すのは、文字列の引数を指定する場合に使用される、名前付き文字列ビルダの例です。

イベントの生成により、ID が 1000 のイベントを作成中で、LVUser1 のローカル変数で生成されたテキストを示しています。ローカル変数 LVUser1 は、+”“+”Training\Users\Active\Users1”+” コンテナ ” に追加されたユーザ : 操作属性 “cn” の文字列です。このイベントは、Trainging\Users\Active\Users1 コンテナに追加されたユーザ :jsmith を読み込みます。

3.6.20 エンタイトルメントの実装

エンタイトルメントを実装するアクションを指定することで、これらのエンタイトルメントのステータスが、そのエンタイトルメントを付与または取り消したエージェントにレポートされるようにします。

フィールド

ノードセット

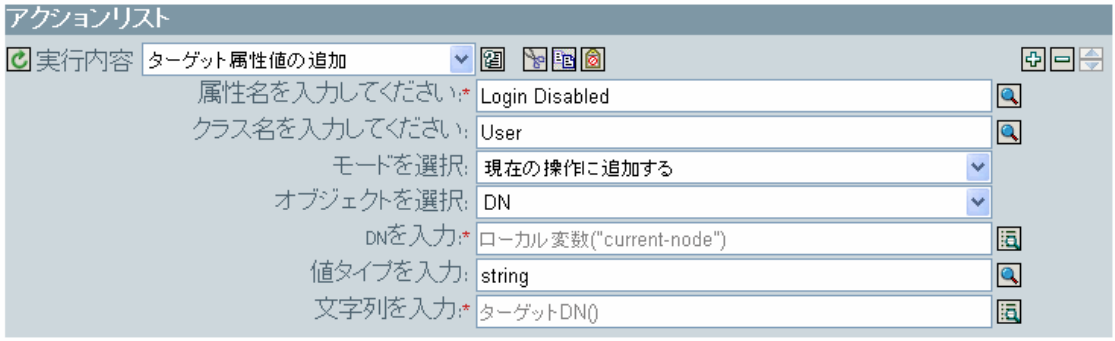
指定されたアクションによって実装中のエンタイトルメントが含まれるノードセット。

アクション

指定されたエンタイトルメントを実装するアクション。

例

次に示すのは、引数アクションビルダの例で、アクションの引数を指定する場合に使用されます。



3.6.21 ターゲットオブジェクトの移動

ターゲットデータストア内のオブジェクトを移動します。

フィールド

モード

このアクションを現在の操作の前と後のどちらに追加するか、または目的のデータストアへ直接書き込むかを選択します。

クラス名

(オプション) 移動するオブジェクトのクラス名を指定します。現在のオブジェクトのクラス名を使用するには、空白のままにします。

移動するオブジェクト

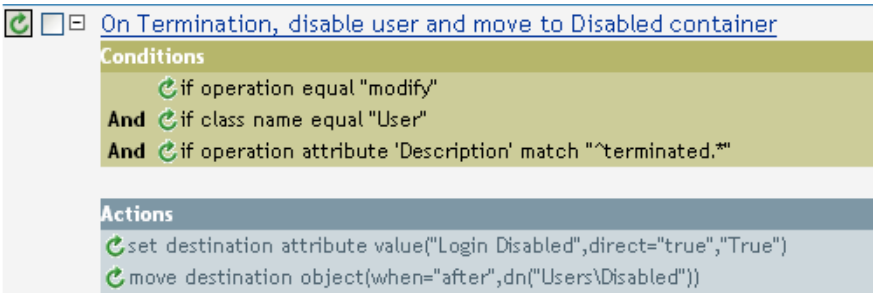
移動するオブジェクトを選択します。このオブジェクトは現在のオブジェクトにすることも、DN または関連付けによって指定することもできます。

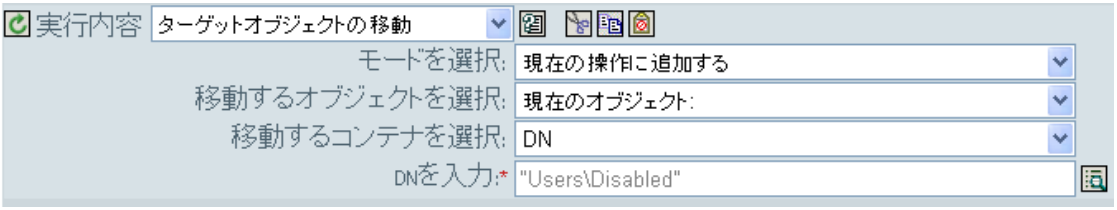
コンテナ

オブジェクトを受け取るコンテナを選択します。このコンテナは、DN または関連付けによって指定します。

例

この例にはルールが 1 つ含まれています。このルールは説明属性がルールの実行終了を示している場合にユーザのアカウントを無効にし、アカウントを無効なコンテナに移動します。これは、「Disable User Account and Move When Terminated (終了時のユーザアカウントの無効化と移動)」という名前のポリシーで、Novel のサポート Web サイトからダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。





このポリシーでは、それがユーザオブジェクトの変更イベントであるかどうか、および説明属性に終了の値が含まれているかどうかを確認します。該当する場合、「ログインの無効化」の属性を True に設定し、そのオブジェクトを User\Disabled コンテナに移動します。

3.6.22 ソースオブジェクトの移動

ソースデータストア内のオブジェクトを移動します。

フィールド

移動するオブジェクト

移動するオブジェクトを選択します。このオブジェクトは現在のオブジェクトにすることも、DN または関連付けによって指定することもできます。

コンテナの選択

オブジェクトを受け取るコンテナを選択します。このコンテナは、DN または関連付けによって指定します。

例

3.6.23 操作属性の再フォーマット

パターンを使用して、現在の操作内にある属性のすべての値を再フォーマットします。

フィールド

名前

属性の名前を指定します。

値のタイプ

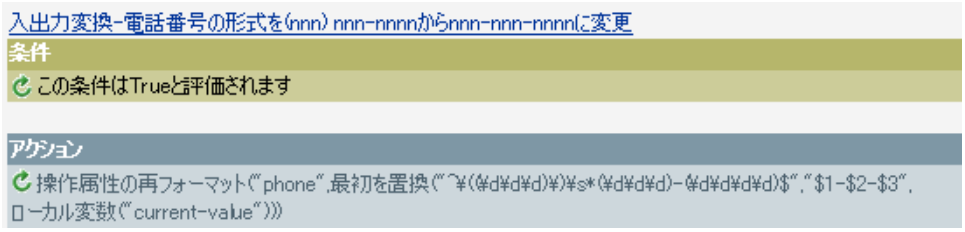
新しい属性値の構文を指定します。

値

属性値の新しいフォーマットのパターンとして使用する値を指定します。新しい値を作成するのに元の値が必要な場合は、ローカル変数 current-value を参照することで取得する必要があります。

例

この例では、電話番号を再フォーマットします。(nnn)-nnn-nnnn から nnn-nnn-nnnn に変更します。このルールは、Identity Manager 3.0 に付属している事前定義されたルールです。詳細については、238 ページの 「入出力変換 - 電話番号の形式を (nnn) nnn-nnnn から nnn-nnn-nnnn に変更」を参照してください。



再フォーマット操作アクションの属性により、電話番号の形式を変更します。このルールでは、引数ビルダと正規表現を使用して、情報の表示方法を変更します。

3.6.24 関連付けを削除

関連付けを削除するコマンドを識別ボールトに送信します。

フィールド

モード

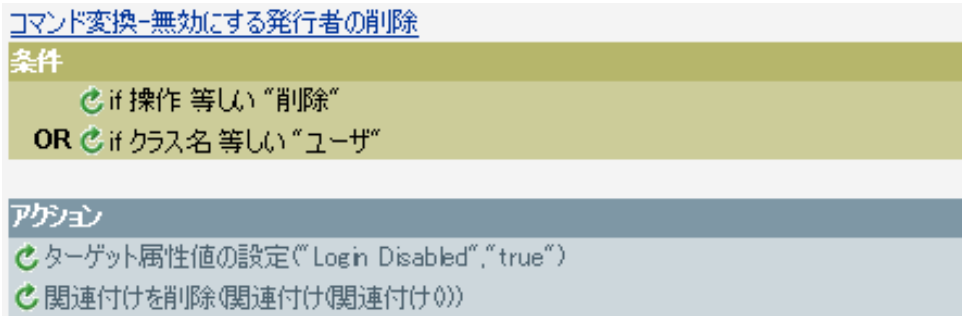
このアクションを現在の操作の前と後のどちらに追加するか、または目的のデータストアへ直接書き込むかを選択します。

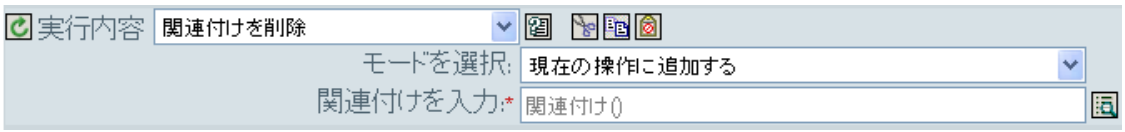
関連付け

削除する関連付けの値を指定します。

例

この例では、削除操作を代用してユーザオブジェクトを無効にします。イベントは変換されます。このルールは、Identity Manager 3.0 に付属している事前定義されたルールです。詳細については、231 ページの 「コマンド変換 - 無効にする発行者の削除」を参照してください。





ユーザオブジェクトに対して削除操作が行われるときは、「ログインの無効化」の値が True に設定され、ユーザオブジェクトから関連付けが削除されます。関連付けが削除されるのは、接続アプリケーション内に関連付けられたオブジェクトが存在しなくなったためです。

3.6.25 ターゲット属性値の削除

ターゲットデータストア内のオブジェクトから、属性値を 1 つ削除します。

フィールド

属性名

属性の名前を指定します。

クラス名

(オプション) ターゲットオブジェクトのクラス名を指定します。現在のオブジェクトのクラス名を使用するには、空白のままにします。

モード

このアクションを現在の操作の前と後のどちらに追加するか、または目的のデータストアへ直接書き込むかを選択します。

オブジェクトの選択

ターゲットオブジェクトを選択します。このオブジェクトは現在のオブジェクトにすることも、DN または関連付けによって指定することもできます。

値のタイプ

新しい属性値の構文を指定します。

値

新しい属性の値を指定します。

例

3.6.26 ソース属性値の削除

ソースデータストア内のオブジェクトにある名前付き属性から、指定した値を削除します。

フィールド

属性名

属性の名前を指定します。

クラス名

(オプション) ターゲットオブジェクトのクラス名を指定します。現在のオブジェクトのクラス名を使用するには、空白のままにします。

オブジェクト

ターゲットオブジェクトを選択します。このオブジェクトは現在のオブジェクトにすることも、DN または関連付けによって指定することもできます。

値のタイプ

削除する属性値の構文を指定します。

値

削除する属性値を指定します。

例

3.6.27 ターゲットオブジェクトの名前変更

ターゲットデータストア内のオブジェクトの名前を変更します。

フィールド

モード

このアクションを現在の操作の前と後のどちらに追加するか、または目的のデータストアへ直接書き込むかを選択します。

オブジェクト

ターゲットオブジェクトを選択します。このオブジェクトは現在のオブジェクトにすることも、DN または関連付けによって指定することもできます。

文字列

オブジェクトの新しい名前を指定します。

例

3.6.28 操作属性の名前変更

現在の操作内で出現したすべての属性の名前を変更します。

フィールド

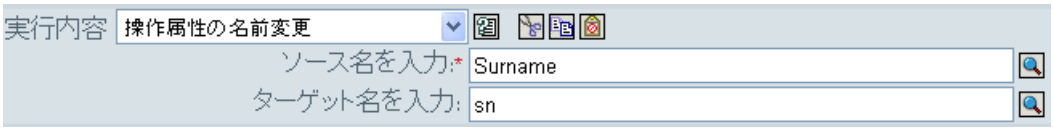
ソース名

変更前の属性名を指定します。

ターゲット名

新しい属性名を指定します。

例



3.6.29 ソースオブジェクトの名前変更

ソースデータストア内のオブジェクトの名前を変更します。

フィールド

オブジェクトの選択

ターゲットオブジェクトを選択します。このオブジェクトは現在のオブジェクト、DN または関連付けにすることができます。

文字列

オブジェクトの新しい名前を指定します。

例

3.6.30 電子メールの送信

電子メール通知を送信します。

フィールド

ID

(オプション) メッセージを送信する SMTP システムでのユーザ ID を指定します。

サーバ

SMTP サーバ名を指定します。

パスワード

(オプション)SMTP サーバのアカウントのパスワードを指定します。

重要 **:** パスワード属性の値はクリアテキストで保存されます。

タイプ

電子メールメッセージのタイプを選択します。

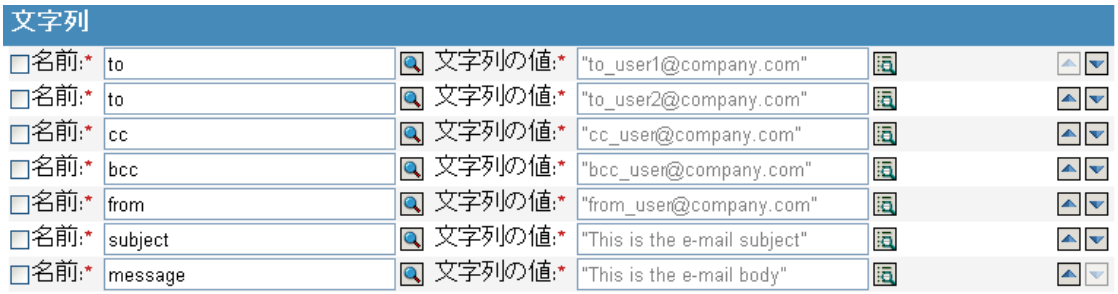
文字列

さまざまな電子メールアドレス、件名、およびメッセージなどの値を指定します。次の表に、有効な名前付き文字列の引数を示します。

文字列名	説明
to	電子メールの受信者リストにアドレスを追加します。複数のインスタンスを指定できます。
cc	電子メールの CC の受信者リストにアドレスを追加します。複数のインスタンスを指定できます。
bcc	電子メールの BCC の受信者リストにアドレスを追加します。複数のインスタンスを指定できます。
from	電子メールの発信アドレスとして使用されるアドレスを指定します。
reply-to	電子メールメッセージの返信アドレスとして使用されるアドレスを指定します。
subject	電子メールの件名を指定します。
message	電子メールメッセージの内容を指定します。
encoding	電子メールメッセージで使用する文字エンコードを指定します。

例

次に示すのは、名前付き文字列ビルダの例で、文字列の引数を指定しているところです。



3.6.31 テンプレートから電子メールを送信

テンプレートを使用して、電子メール通知を生成します。

フィールド

通知 **DN**

SMTP 通知設定オブジェクトのスラッシュ形式の DN を指定します。

テンプレート **DN**

電子メールテンプレートオブジェクトのスラッシュ形式の DN を指定します。

パスワード

(オプション)SMTP サーバのアカウントのパスワードを指定します。

重要 **:** パスワード属性の値はクリアテキストで保存されます。

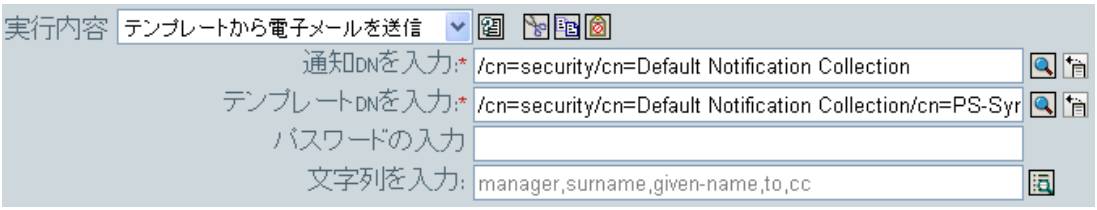
文字列

電子メールメッセージの他のフィールドを指定します。次の表に、さまざまな電子メールアドレスを指定する予約済みのフィールド名を示します。

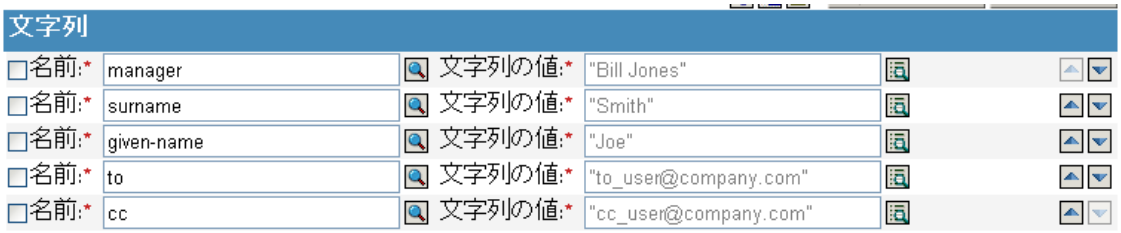
文字列名	説明
to	電子メールの受信者リストにアドレスを追加します。複数のインスタンスを指定できます。
cc	電子メールの CC の受信者リストにアドレスを追加します。複数のインスタンスを指定できます。
bcc	電子メールの BCC の受信者リストにアドレスを追加します。複数のインスタンスを指定できます。
reply-to	電子メールメッセージの返信アドレスとして使用されるアドレスを指定します。
encoding	電子メールメッセージで使用する文字エンコードを指定します。

各テンプレートでは、電子メールメッセージの件名および本文で置き換えられるフィールドも定義できます。

例



次に示すのは、名前付き文字列ビルダの例で、文字列の引数を指定する場合に使用されます。



3.6.32 デフォルト属性値の設定

属性に値が指定されていない場合に、現在の操作にデフォルト値を追加します (オプションで、ソースデータストア内の現在のオブジェクトにも追加します)。これは、現在の操作が「追加」の場合のみ有効です。

フィールド

属性名

デフォルト属性の名前を指定します。

ライトバック

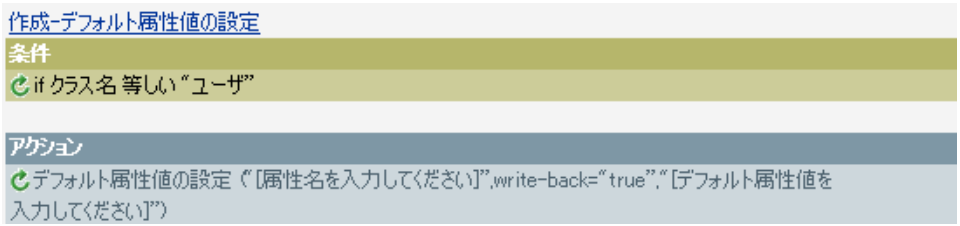
デフォルト値をソースデータストアにもライトバックするかどうかを選択します。

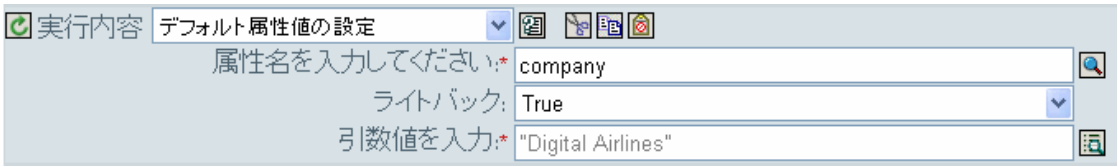
値

属性のデフォルト値を指定します。

例

この例では、属性「company」のデフォルト値を設定します。必要な属性に値を設定できます。このルールは、Identity Manager 3.0 に付属している事前定義されたルールです。詳細については、234 ページの 「作成 - デフォルト属性値の設定」を参照してください。





値を作成するには、引数値リストビルダを起動します。このビルダの詳細については、224 ページの 「引数値リストビルダ」を参照してください。必要な値を設定できます。この場合、引数ビルダを使用して「company」という名前のテキストを入力しました。

3.6.33 ターゲット属性値の設定

ターゲットデータストアにあるオブジェクトの属性に値を追加し、その属性に設定されている他の値をすべて削除します。

フィールド

属性名

属性の名前を指定します。

クラス名

(オプション) ターゲットデータストア内のターゲットオブジェクトのクラス名を指定します。現在のオブジェクトのクラス名を使用するには、空白のままにします。

モード

このアクションを現在の操作の前と後のどちらに追加するか、または目的のデータストアへ直接書き込むかを選択します。

オブジェクト

ターゲットオブジェクトを選択します。このオブジェクトは現在のオブジェクトにすることも、DN または関連付けによって指定することもできます。

値のタイプ

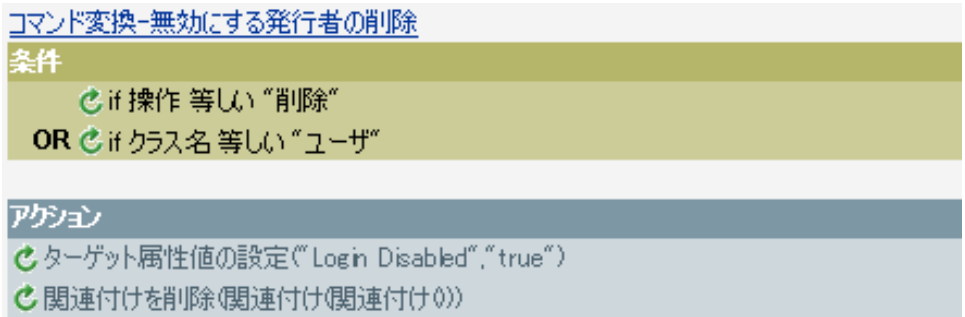
設定する属性値の構文を選択します。

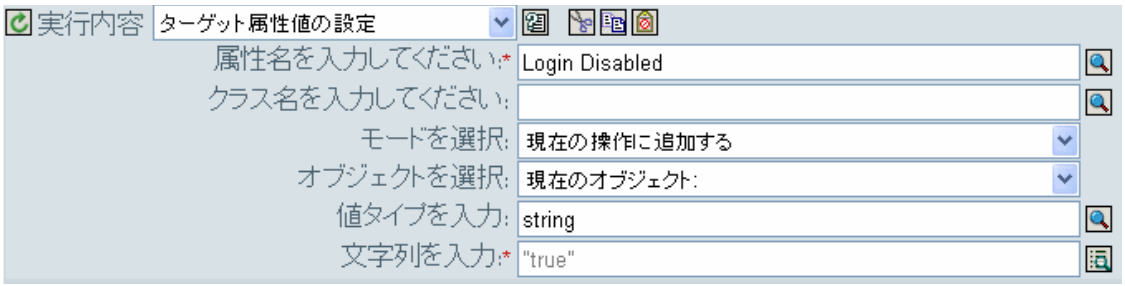
値

設定する属性値を指定します。

例

この例では、削除操作を代用してユーザオブジェクトを無効にします。このルールは、Identity Manager 3.0 に付属している事前定義されたルールです。詳細については、231 ページの 「コマンド変換 - 無効にする発行者の削除」を参照してください。





このルールでは、「ログインの無効化」の属性値を True に設定します。このルールでは、引数ビルダを使用して、この属性値としてテキスト「True」を追加します。このビルダの詳細については、221 ページの 「引数ビルダ」を参照してください。

3.6.34 ターゲットパスワードの設定

ターゲットデータストアにある現在のオブジェクトのパスワードを設定します。

フィールド

モード

このアクションを現在の操作の前と後のどちらに追加するか、または目的のデータストアへ直接書き込むかを選択します。

オブジェクト

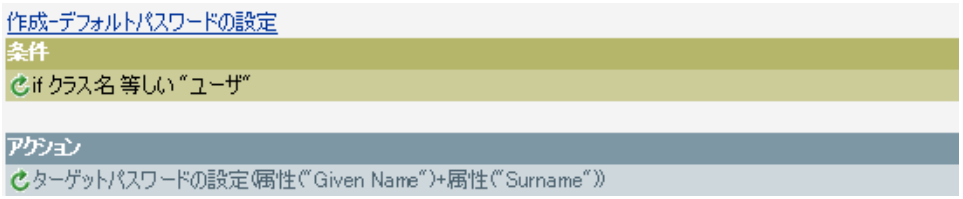
ターゲットオブジェクトを選択します。このオブジェクトは現在のオブジェクトにすることも、DN または関連付けによって指定することもできます。

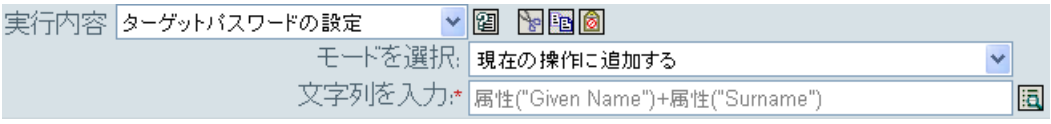
文字列

設定するパスワードを指定します。

例

この例では、作成されるユーザオブジェクトのデフォルトのパスワードを設定します。このルールは、Identity Manager 3.0 に付属している事前定義されたルールです。詳細については、235 ページの 「作成 - デフォルトパスワードの設定」を参照してください。





ユーザオブジェクトが作成され場合、パスワードは、名前属性に名字属性を加えたもの設定されます。

3.6.35 ローカル変数の設定

ローカル変数を設定します。

フィールド

変数名

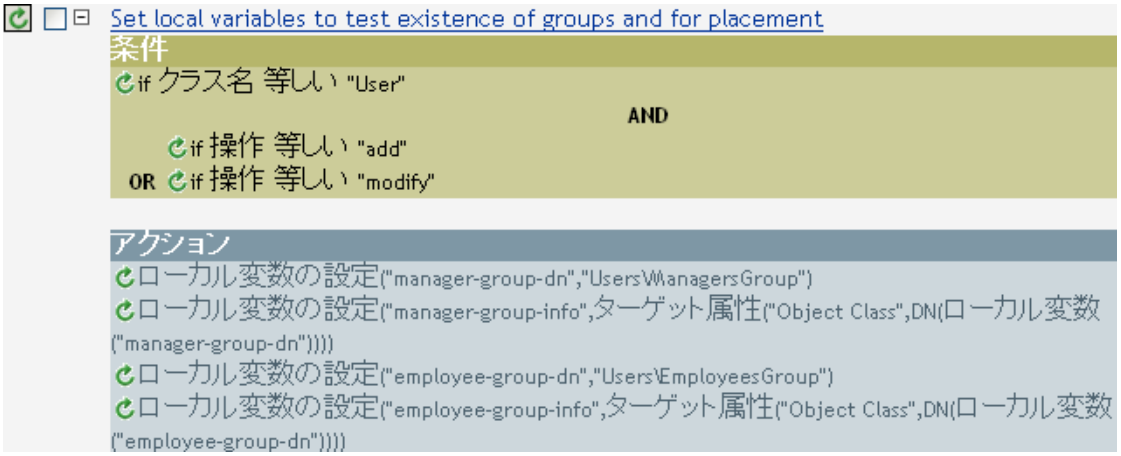
新しいローカル変数の名前を指定します。

変数タイプ

ローカル変数のタイプを選択します。文字列、XPath 1.0 ノードセット、または Java オブジェクトにできます。

例

この例では、役職に基づいて、ユーザオブジェクトを適切なグループ (従業員またはマネージャ) に追加します。必要に応じてグループも作成し、そのグループに同等セキュリティを設定します。これは「Govern Groups for User Based on Title (役職に基づくユーザグループの管理)」という名前のポリシーで、Novell のサポート Web サイトからダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。



ローカル変数は、ユーザオブジェクトのターゲット属性 (オブジェクトクラスとローカル変数 manager-group-info) の値に設定されます。引数ビルダは、ローカル変数の作成に使用されます。詳細については、221 ページの 「引数ビルダ」を参照してください。

3.6.36 操作関連付けの設定

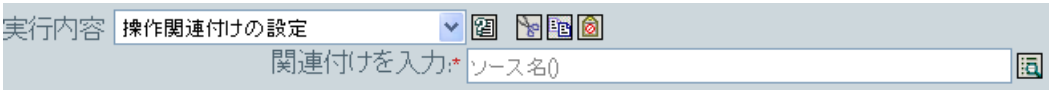
現在の操作に関連付けの値を設定します。

フィールド

関連付け

新しい関連付けの値を指定します。

例



3.6.37 操作クラス名の設定

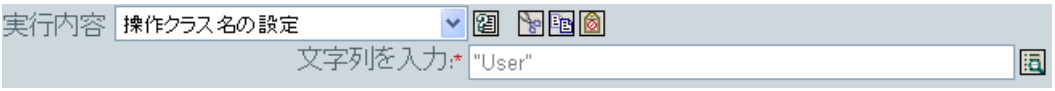
現在の操作のオブジェクトクラス名を設定します。

フィールド

文字列

新しいクラス名を指定します。

例



3.6.38 操作ターゲット DN の設定

現在の操作のターゲット DN を設定します。

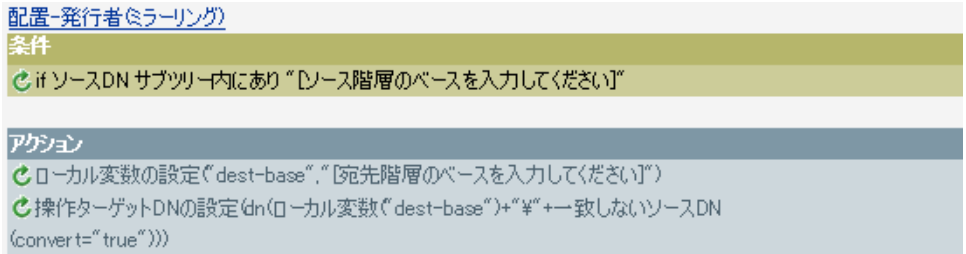
フィールド

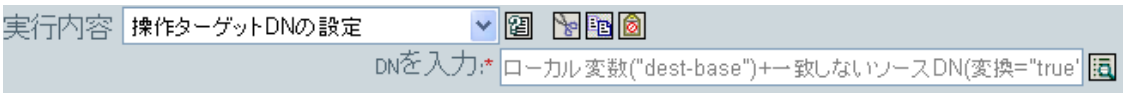
DN

新しいターゲット DN を指定します。

例

この例では、接続システムからミラー化された構造を使用して、識別ボールト内にオブジェクトを配置します。ソースおよびターゲットのデータストアで、ミラー化を開始するポイントを定義する必要があります。このルールは、Identity Manager 3.0 に付属している事前定義されたルールです。詳細については、81 ページの 「作成 - デフォルト属性値の設定」を参照してください。





このルールでは、操作ターゲット DN をターゲットのベースロケーションとソース DN のローカル変数として設定します。

3.6.39 操作プロパティの設定

操作プロパティを設定します。操作プロパティは、操作内に保存される名前付きの値です。一般に、操作の結果を処理するポリシーで必要になる可能性がある追加のコンテキストを提供するために使用されます。

フィールド

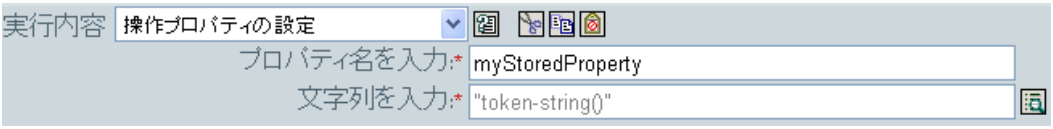
プロパティ名

操作プロパティの名前を指定します。

文字列

操作プロパティの名前を指定します。

例



3.6.40 操作ソース DN の設定

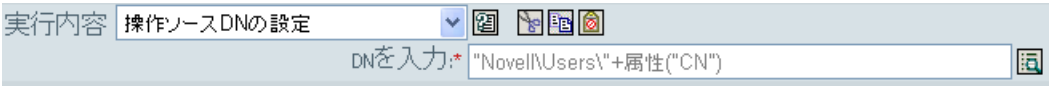
現在の操作のソース DN を設定します。

フィールド

DN

新しいソース DN を指定します。

例



3.6.41 操作テンプレート DN の設定

現在の操作のテンプレート DN を、指定した値に設定します。このアクションは、現在の操作が「追加」の場合のみ有効です。

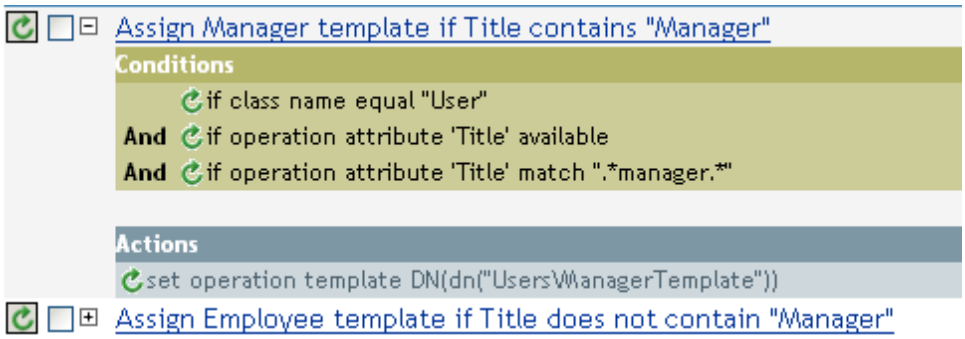
フィールド

DN

テンプレート DN を指定します。

例

この例では、役職属性に「Manager」という語句が含まれている場合に、Manager テンプレートを適用します。これは「Policy: Assign Template to User Based on Title (ポリシー : 役職に基づくユーザへのテンプレート割り当て)」という名前のポリシーで、Novell のサポート Web サイトからダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。



テンプレート「Manager Template」は、使用可能な役職属性を持っていて、役職名のどこかに「manager」という語句が含まれているユーザオブジェクトに適用されます。このポリシーでは、一致するすべてのものを検索する正規表現を使用しています。

3.6.42 ソース属性値の設定

ソースデータストアにあるオブジェクトの属性に値を追加し、その属性に設定されている他の値をすべて削除します。

フィールド

属性名

属性の名前を指定します。

クラス名

(オプション) ソースデータストア内のターゲットオブジェクトのクラス名を指定します。現在のオブジェクトのクラス名を使用するには、空白のままにします。

オブジェクト

ターゲットオブジェクトを選択します。このオブジェクトは現在のオブジェクトにすることも、DN または関連付けによって指定することもできます。

値のタイプ

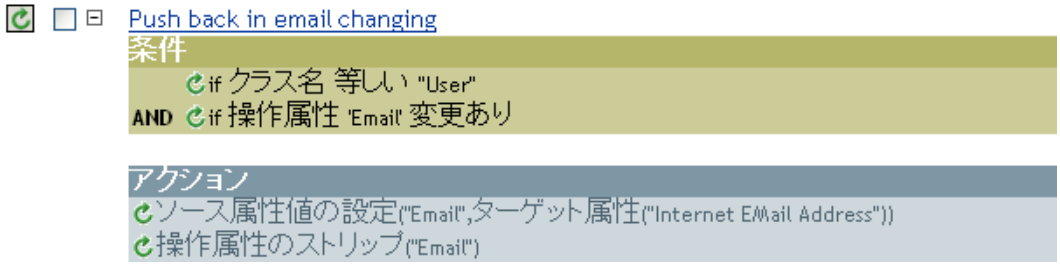
属性値の構文を選択します。

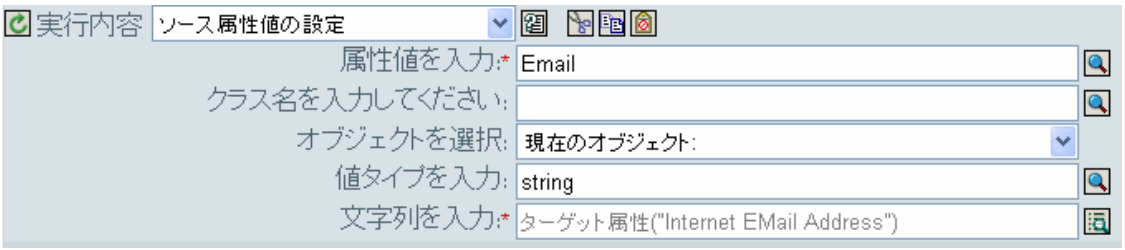
値

設定する属性値を指定します。

例

この例では、電子メールアドレスが変更されたことを検出し、変更内容を元の状態に戻します。これは「Policy: Reset Value of the E-mail Attribute (ポリシー : 電子メール属性値のリセット)」という名前のポリシーで、Novell のサポート Web サイトからダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。





このアクションでは、ターゲット属性「Internet EMail Address」の値を取得し、電子メールのソース属性をこの値と同じに設定します。

3.6.43 ソースパスワードの設定

ソースデータストアにある現在のオブジェクトのパスワードを設定します。

フィールド

文字列

設定するパスワードを指定します。

例

3.6.44 SSO 資格情報の設定

ユーザオブジェクトの作成またはパスワードの変更が実施されるときの、SSO 資格情報を設定します。このアクションは、資格情報のプロビジョニングポリシーの一部です。詳細については、333 ページの第 4 章「Novell 資格情報プロビジョニングポリシー」を参照してください。

フィールド

資格情報ストアオブジェクトの **DN**

リポジトリオブジェクトの DN を指定します。

ターゲットユーザの **DN**

ターゲットユーザの DN を指定します。

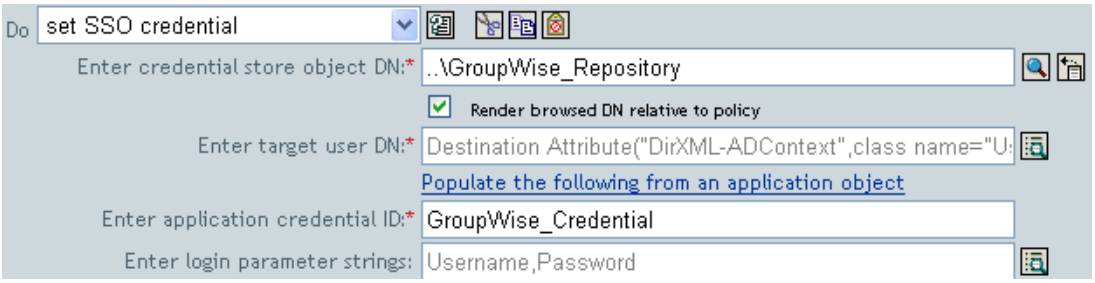
アプリケーションのアクティベーションキー **ID**

アプリケーションオブジェクト内に格納されるアプリケーションの資格情報を指定します。

ログインパラメータの文字列

アプリケーションのログインパラメータを指定します。ログインパラメータとは、アプリケーションオブジェクト内に格納されている認証キーです。

例



3.6.45 SSO パスフレーズの設定

ユーザオブジェクトがプロビジョニングされるときの Novell SecureLogin® のパスフレーズおよび回答を設定します。このアクションは、資格情報のプロビジョニングポリシーの一部です。詳細については、333 ページの第 4 章「Novell 資格情報プロビジョニングポリシー」を参照してください。

フィールド

資格情報ストアオブジェクトの **DN**

リポジトリオブジェクトの DN を指定します。

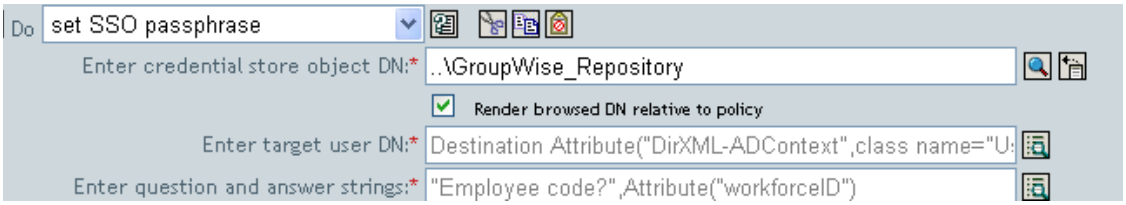
ターゲットユーザの **DN**

ターゲットユーザの DN を指定します。

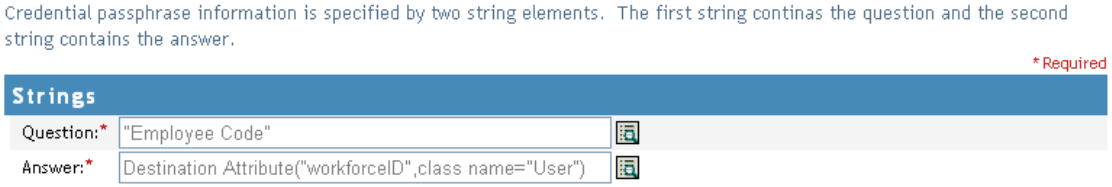
質問と回答の文字列

SecureLogin パスフレーズの質問と回答を指定します。

例



SecureLogin パスフレーズの質問と回答は、ポリシー内に文字列として保存されます。［これらの文字列を編集します］アイコンをクリックして、文字列ビルダを起動します。パスフレーズの質問と回答を指定します。



3.6.46 XML 属性の設定

XPath 式で選択された要素のセットに XML を設定します。

フィールド

名前

XML 属性の名前を指定します。この名前には、前にこのポリシーで定義されているネームスペースプリフィックスを含めることができます。

XPATH 式

XML 属性の設定先になる要素を含むノードセットを返す XPath 1.0 の式。

文字列

XML 属性の値を指定します。

例

3.6.47 SSO 資格情報の設定

ユーザオブジェクトの作成またはパスワードの変更が実施されるときの、SSO 資格情報を設定します。このアクションは、資格情報のプロビジョニングポリシーの一部です。詳細については、333 ページの第 4 章「Novell 資格情報プロビジョニングポリシー」を参照してください。

フィールド

資格情報ストアオブジェクトの **DN**

リポジトリオブジェクトの DN を指定します。

ターゲットユーザの **DN**

ターゲットユーザの DN を指定します。

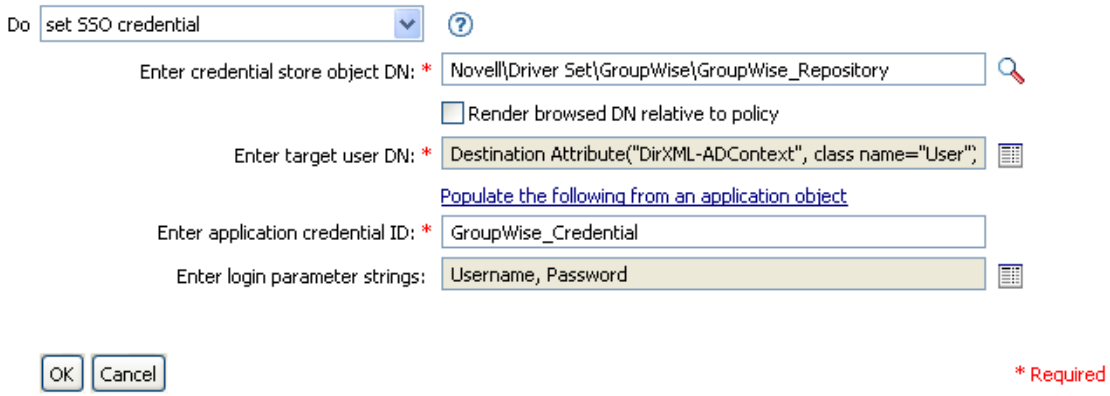
アプリケーションのアクティベーションキー **ID**

アプリケーションオブジェクト内に格納されるアプリケーションの資格情報を指定します。

ログインパラメータの文字列

アプリケーションのログインパラメータを指定します。ログインパラメータとは、アプリケーションオブジェクト内に格納されている認証キーです。

例



3.6.48 ステータス

ステータス通知を生成します。

フィールド

レベル

通知のステータスレベルを指定します。

メッセージ

引数ビルダを使用してステータスメッセージを指定できます。

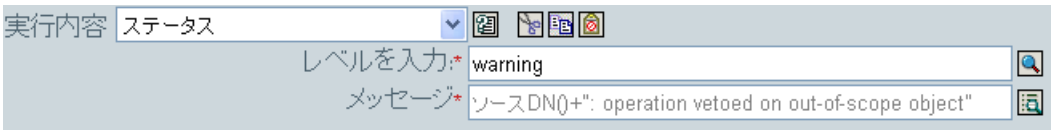
備考

レベルが「再試行」である場合、ポリシーは入力ドキュメントの処理をただちに中止して、現在処理中のイベントの再試行をスケジュールします。

レベルが「致命的エラー」である場合、ポリシーは入力ドキュメントの処理をただちに中止して、ドライバのシャットダウンを開始します。

現在の操作にイベント ID が割り当てられている場合、そのイベント ID がステータス通知に使用されます。割り当てられていない場合は、イベント ID はレポートされません。

例



3.6.49 操作属性のストリップ

現在の操作から属性に行ったすべての内容を除去します。

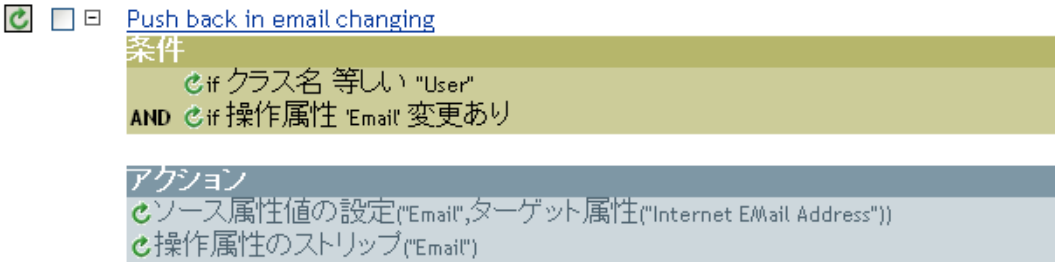
フィールド

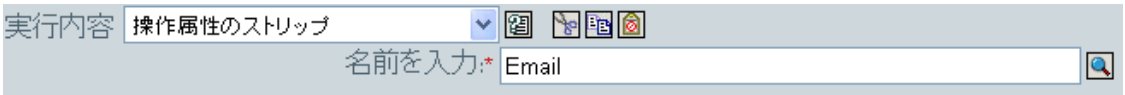
名前

除去する属性の名前を指定します。

例

この例では、電子メールアドレスが変更されたことを検出し、変更内容を元の状態に戻します。これは「Policy: Reset Value of the E-mail Attribute (ポリシー : 電子メール属性値のリセット)」という名前のポリシーで、Novell のサポート Web サイトからダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。





このアクションでは、電子メールの属性を除去します。保持されている値は、ターゲットの電子メール属性内にあったものです。

3.6.50 XPath のストリップ

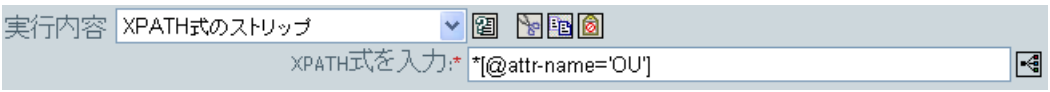
XPath 1.0 の式で選択されたノードを除去します。

フィールド

XPATH 式

除去されるノードを含むノードセットを返す XPath 1.0 の式を指定します。

例



3.6.51 メッセージのトレース

DSTRAC へメッセージを送信します。

フィールド

レベル

メッセージのトレースレベルを指定します。デフォルトのレベルは 0 です。メッセージは、指定したトレースレベルがドライバで設定されているトレースレベル以下である場合にのみ表示されます。

ドライバのトレースレベルの設定方法についての詳細は、『*Novell Identity Manager 3.0 管理ガイド*』の「“”Identity Manager のプロセスの表示」を参照してください。

色

トレースメッセージの色を選択します。

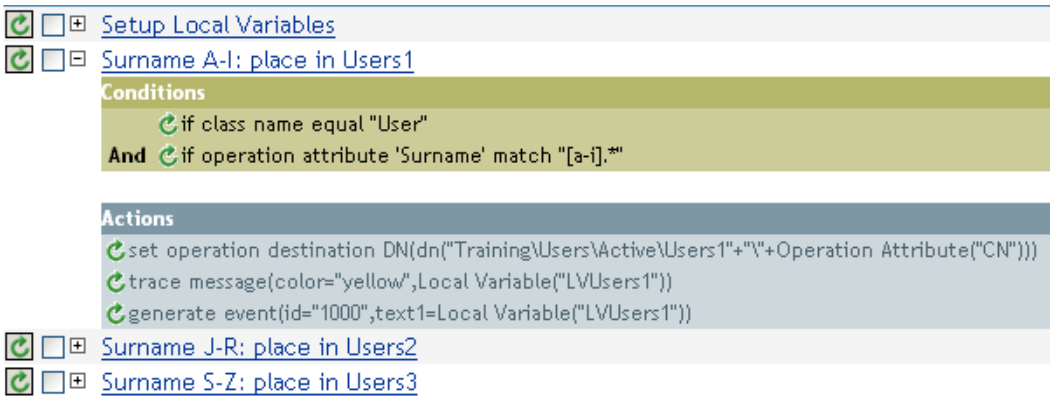
文字列

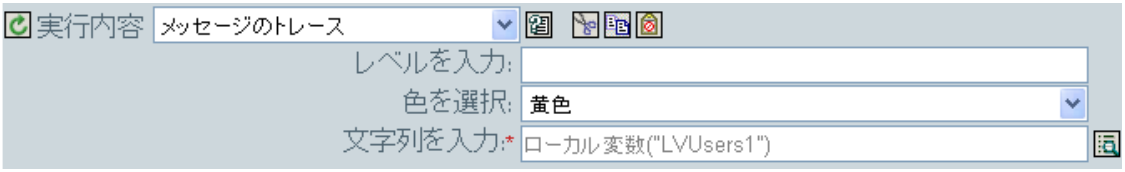
トレースメッセージの値を指定します。

例

この例には 4 つのルールがあり、これらのルールでは名字属性の最初の文字に基づいてユーザオブジェクトに配置ポリシーを実装し、トレースメッセージおよびカスタムの Novell Audit イベントの両方を生成します。メッセージのトレースアクションは、DSTRACE へのトレースメッセージを送信する場合に使用されます。これは、「Policy to Place by Surname (名字で配置するためのポリシー)」という名前のポリシーで、Novell の

サポート Web サイトからダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。





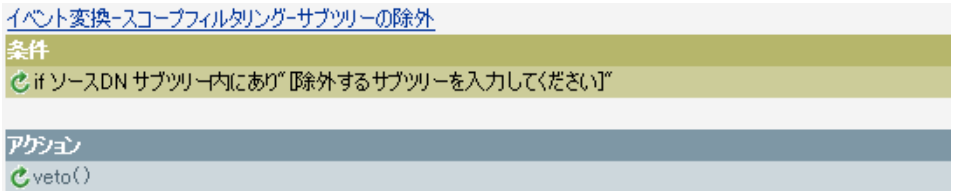
DSTRAC へトレースメッセージを送信します。ローカル変数の内容は LVUsers1 で、DSTRACE では黄色で表示されます。

3.6.52 拒否

現在の操作を拒否します。

例

この例では、指定されたサブツリーからのイベントをすべて除外します。このルールは、Identity Manager 3.0 に付属している事前定義されたルールです。詳細については、237 ページの 「イベント変換 - スコープフィルタリング - サブツリーの除外」を参照してください。



このアクションでは、指定されたサブツリーからのイベントをすべて拒否します。

3.6.53 操作属性値がない場合は拒否

現在の操作内の属性の使用状況に基づき、条件付きで現在の操作をキャンセルして現在のポリシーの処理を終了します。

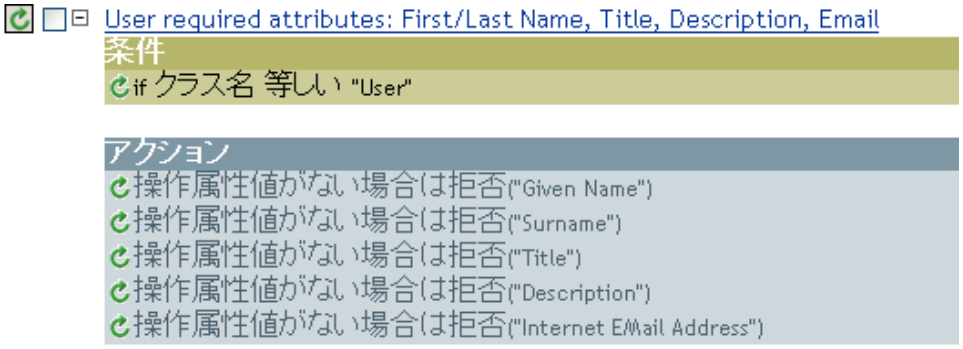
フィールド

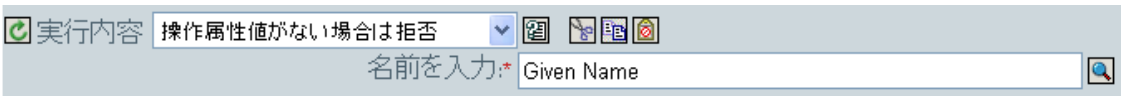
名前

属性の名前を指定します。

例

この例では、属性「名前」、「名字」、「役職」、「説明」、および「インターネット電子メールアドレス」が使用できない場合、ユーザオブジェクトは作成されません。これは「Policy to Enforce the Presences of Attributes (属性の存在を強制するポリシー)」という名前のポリシーで、Novell のサポート Web サイトからダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。





このアクションでは、属性「名前」、「名字」、「役職」、「説明」、「インターネット電子メールアドレス」が使用できない場合、操作を拒否します。

3.7 名詞トークン

この節では、ポリシービルダインタフェースで使用できるすべての名詞トークンについて、詳しく説明します。

3.7.1 追加されたエンタイトルメント

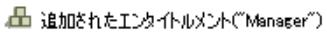
現在の操作で付与されたエンタイトルメントの値に展開します。

フィールド

名前

エンタイトルメントの名前。

例



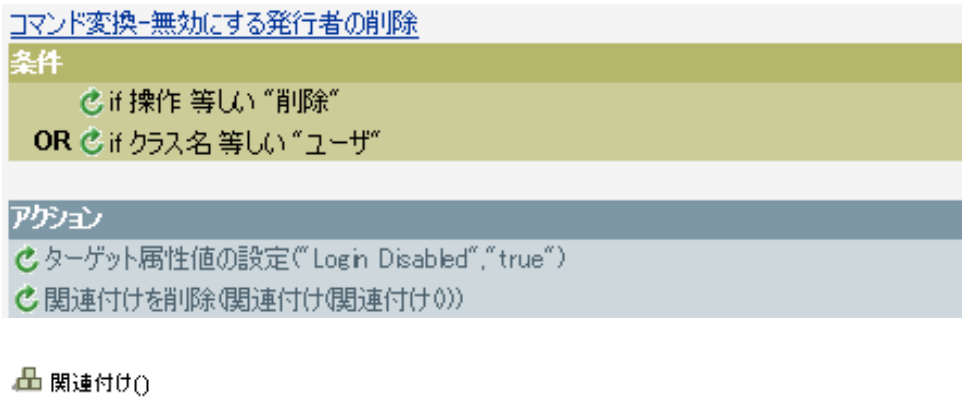
3.7.2 関連付け

現在の操作から関連付けの値に展開します。

例

この例は、Identity Manager 3.0 に付属している事前定義されたルールからのものです。事前定義されたルールの詳細については、231 ページの 「コマンド変換 - 無効にする発行者の削除」を参照してください。

関連付けを削除するアクションでは、関連付けトークンを使用して、現在の操作から値を取得します。このルールでは、ユーザオブジェクトから関連付けを削除することで、新しいイベントが発生してもユーザオブジェクトに影響を与えないようにします。



3.7.3 属性

現在の操作およびソースデータストア内の現在のオブジェクトからの属性値に展開します。これは、論理的には、操作属性のトークンとソース属性のトークンの結合と考えることができます。変更操作で削除された値は含まれません。

フィールド

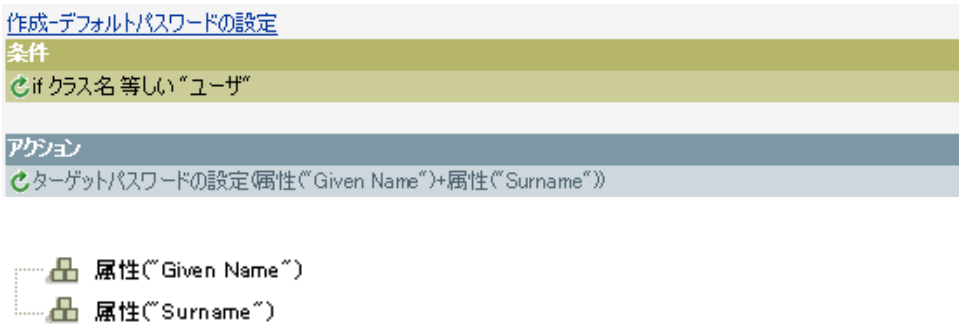
名前

属性の名前を指定します。

例

この例は、Identity Manager 3.0 に付属している事前定義されたルールからのものです。詳細については、235 ページの 「作成 - デフォルトパスワードの設定」を参照してください。

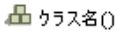
ターゲットパスワードの設定のアクションでは、属性トークンを使用してパスワードを作成します。パスワードは、名前属性と名字属性から作成されます。引数ビルダのエディタから、使用する属性を参照して選択します。



3.7.4 クラス名

現在の操作からオブジェクトクラス名に展開します。

例



3.7.5 ターゲット属性

ターゲットデータストアの現在のオブジェクト、DN、または関連付けの指定した属性値に展開します。

フィールド

クラス名

(オプション) ターゲットオブジェクトのクラス名を指定します。現在のオブジェクトのクラス名を使用するには、空白のままにします。

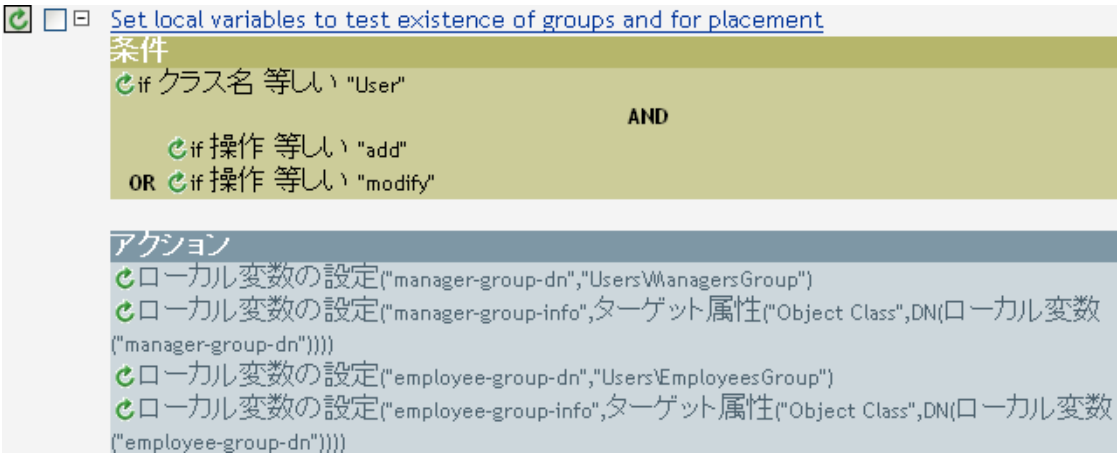
名前

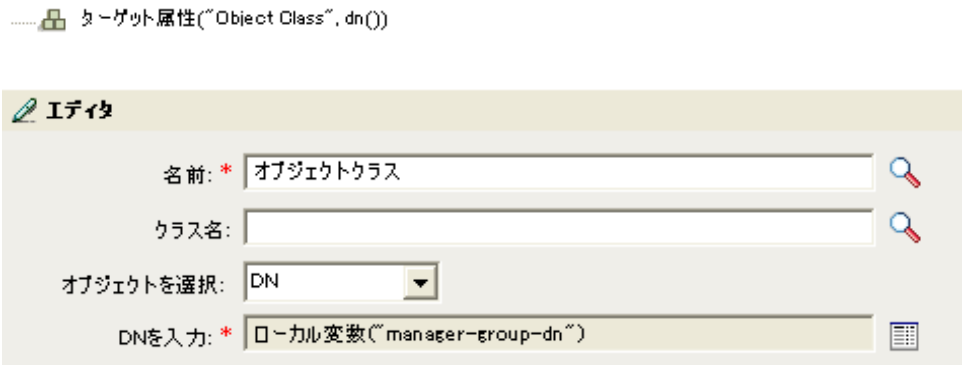
属性の名前。

例

この例は「Govern Groups for User Based on Title (役職に基づくユーザグループの管理)」ポリシーからのもので、Novell のサポート Web サイトからダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。

このポリシーでは、引数ビルダを使用してターゲット属性を作成します。ローカル変数の設定のアクションには、ターゲット属性のトークンが含まれています。





ターゲット属性はエディタを使用して作成します。この例では、オブジェクトクラスの属性が設定されます。DN は、オブジェクトの選択に使用されます。DN の値は、ローカル変数 manager-group-dn です。

3.7.6 ターゲット DN

現在の操作で指定されたターゲット DN に展開します。

フィールド

変換

DN をソースデータストアで使用される形式に変換するかどうかを選択します。

開始

開始の RDN インデックスを指定します。

- インデックス 0 はルートに最も近い RDN
- 正のインデックスはルートに最も近い RDN からのオフセット
- インデックス -1 はリーフに最も近いセグメント
- 負のインデックスは、リーフに最も近い RDN からルートに最も近い RDN 方向へのオフセット

長さ

含める RDN の数を指定します。負の数は (セグメント総数 + 長さ) +1 のように解釈されます (たとえば、セグメント数が 5 の DN では、長さが -1 の場合は -1 = (5 + (-1)) + 1 = 5、長さが -2 の場合は -2 = (5 + (-2)) + 1 = 4)。

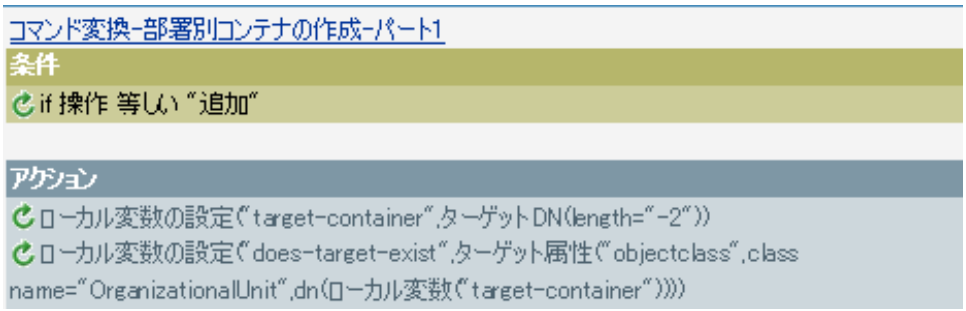
備考

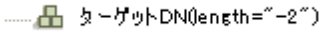
「開始」または「長さ」がデフォルト値 {0、-1} に設定されている場合は、DN 全体が使用されます。それ以外の場合は、「開始」および「長さ」で指定された DN の部分が使用されます。

例

この例では、ターゲット DN のトークンを使用して、ローカル変数 target-container の値を設定します。このポリシーでは、ユーザオブジェクトの部署別コンテナがない場合に、そのコンテナを作成します。このポリシーは、Identity Manager 3.0 に付属している事前定義

されたルールからのものです。詳細については、229 ページの 「コマンド変換 - 部署別コンテナの作成 - パート 1 とパート 2」を参照してください。

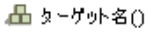




3.7.7 ターゲット名

現在の操作で指定されたターゲット DN の非修飾の相対識別名 (RDN) に展開します。

例



3.7.8 エンタイトルメント

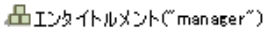
現在のオブジェクトから付与されたエンタイトルメントの値に展開します。

フィールド

名前

エンタイトルメントの名前。

例



3.7.9 グローバル構成値

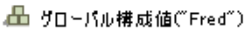
グローバル構成変数の値に展開します。

フィールド

名前

グローバル構成値の名前。

例



3.7.10 ローカル変数

ローカル変数の値に展開します。

フィールド

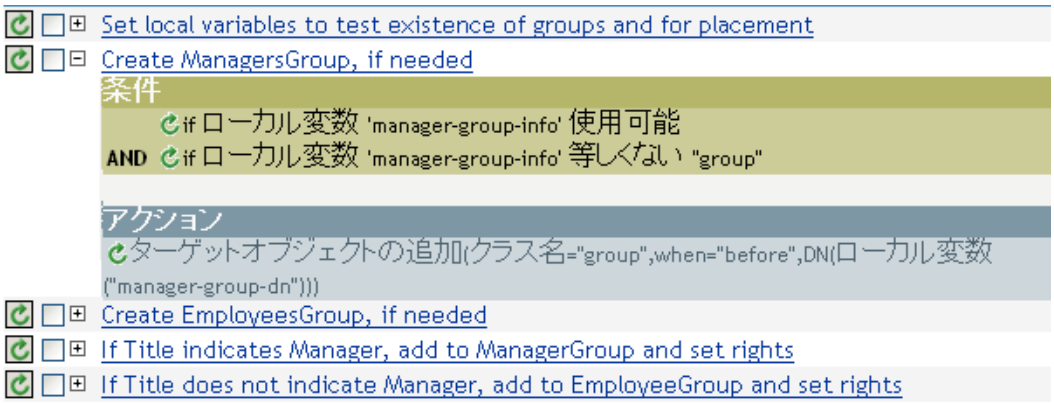
名前

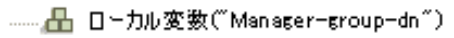
ローカル変数の名前を指定します。

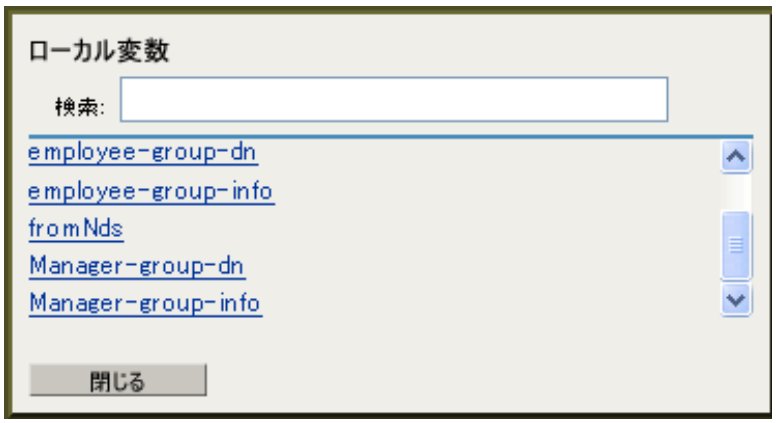
例

この例は「Govern Groups for User Based on Title (役職に基づくユーザグループの管理)」ポリシーからのもので、Novell のサポート Web サイトからダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。

ターゲットオブジェクトの追加アクションでは、ローカル変数のトークンを使用します。







ローカル変数は、ローカル変数の設定アクションがポリシーで以前使用されていた場合にのみ使用されます。ローカル変数に保存される値を設定します。エディタで参照アイコンをクリックすると、定義済みのすべてのローカル変数がリストされます。正しいローカル変数を選択します。

ローカル変数の値は、group-manager-dn です。これは 1 つ前のルール、マネージャのグループ Users\ManagersGroup の DN として group-manager-dn が定義されたローカル変数の設定アクションです。

3.7.11 名前付きパスワード

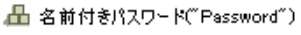
ドライバの名前付きパスワードに展開します。

フィールド

名前

パスワードの名前。

例



3.7.12 操作

現在の操作の名前に展開します。

例

3.7.13 操作属性

現在の操作から属性の値に展開します。変更操作で削除された値は含まれません。

フィールド

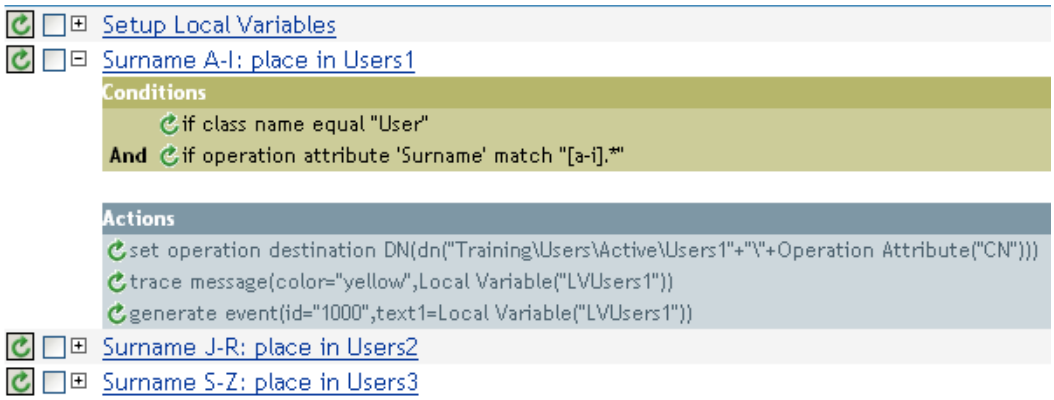
名前

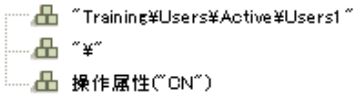
属性の名前を指定します。

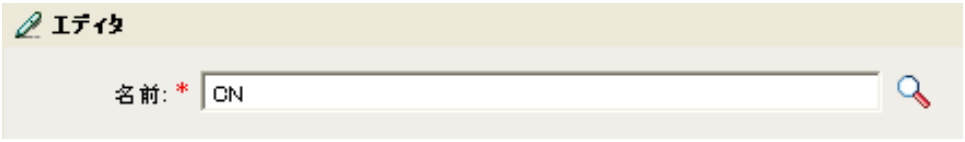
例

この例には 4 つのルールがあり、これらのルールでは名字属性の最初の文字に基づいてユーザオブジェクトに配置ポリシーを実装し、トレースメッセージおよびカスタムの Novell Audit イベントの両方を生成します。これは、「Policy to Place by Surname (名字で配置するためのポリシー)」という名前のポリシーで、Novell のサポート Web サイトからダ

ウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。







操作ターゲット DN の設定アクションには、操作属性のトークンが含まれています。操作属性のトークンは、ターゲット DN を CN 属性に設定します。このルールでは、Training\Users\Active\Users のコンテキストを取得して、\ および CN 属性の値を追加します。

3.7.14 操作プロパティ

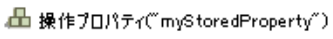
現在の操作の指定された操作プロパティの値に展開します。

フィールド

名前

操作プロパティの名前を指定します。

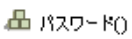
例



3.7.15 パスワード

現在の操作で指定されたパスワードに展開します。

例



3.7.16 削除された属性

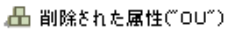
現在の操作で削除されている、指定した属性の値に展開します。変更操作の場合にのみ適用されます。

フィールド

名前

属性の名前を指定します。

例



3.7.17 削除されたエンタイトルメント

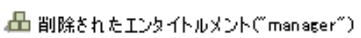
現在の操作で取り消されたエンタイトルメントの値に展開します。

フィールド

名前

エンタイトルメントの名前を指定します。

例



3.7.18 ソース属性

ソースデータストア内の 1 つのオブジェクトからの属性値に展開します。

フィールド

クラス名

(オプション) ターゲットオブジェクトのクラス名を指定します。現在のオブジェクトのクラス名を使用するには、空白のままにします。

名前

属性の名前。

例

3.7.19 ソース DN

現在の操作からソース DN に展開します。

フィールド

変換

DN をターゲットデータストアで使用される形式に変換するかどうかを選択します。

開始

開始の RDN インデックスを指定します。

- インデックス 0 はルートに最も近い RDN
- 正のインデックスはルートに最も近い RDN からのオフセット
- インデックス -1 はリーフに最も近いセグメント
- 負のインデックスは、リーフに最も近い RDN からルートに最も近い RDN 方向へのオフセット

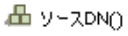
長さ

含める RDN のセグメントの数です。負の数は (セグメント総数 + 長さ) +1 のように解釈されます (たとえば、セグメント数が 5 の DN では、長さが -1 の場合は -1 = (5 + (-1)) + 1 = 5、長さが -2 の場合は -2 = (5 + (-2)) + 1 = 4)。

備考

「開始」または「長さ」がデフォルト値 {0、-1} に設定されている場合は、DN 全体が使用されます。それ以外の場合は、「開始」および「長さ」で指定された DN の部分が使用されます。

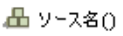
例



3.7.20 ソース名

現在の操作で指定されたソース DN の非修飾の相対識別名 (RDN) に展開します。

例



3.7.21 テキスト

テキストに展開します。

フィールド

テキスト

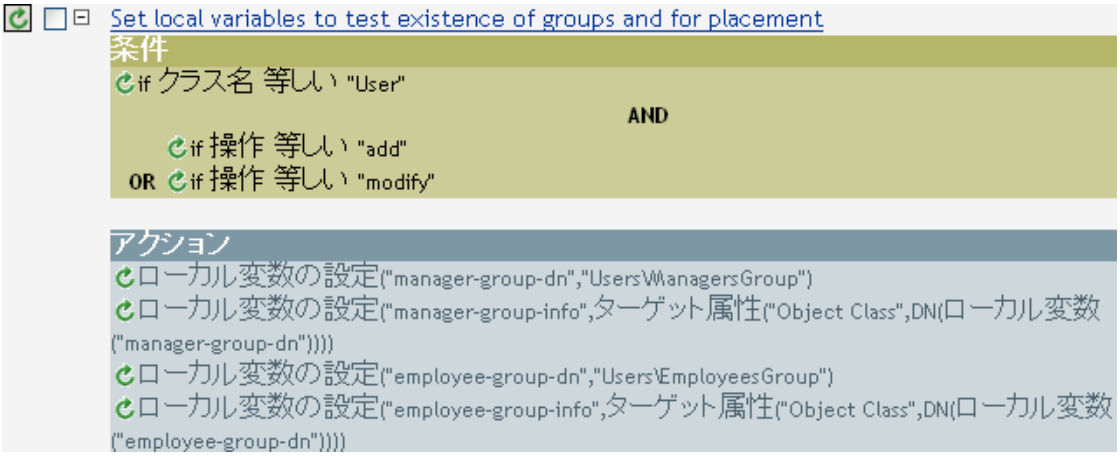
テキストを指定します。

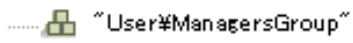
例

この例は「Govern Groups for User Based on Title (役職に基づくユーザグループの管理)」ポリシーからのもので、Novell のサポート Web サイトからダウンロードできます。詳細

については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。

テキストトークンは、マネージャのグループの DN を定義するため、ローカル変数の設定アクションで使用されます。テキストトークンには、オブジェクトまたはプレーンテキストを含められます。





テキストトークンには、マネージャのグループの DN が含まれます。使用するオブジェクトを参照するか、またはエディタに情報を入力します。

3.7.22 一意の名前

指定された条件に従って、ターゲットデータストアで一意の、パターンに基づいた名前に展開します。

フィールド

名前

一意性をチェックする属性の名前を指定します。

スコープ

一意性をチェックするスコープを指定します。

検索の開始

検索を開始するポイントを選択します。開始ポイントは、データストアのルートにするか、DN で指定するか、または関連付けにすることができます。

パターン

引数ビルダを使用して一意の値を生成する場合に使用するパターンを指定します。

カウンタの開始

一意の名前を検索する必要がある場合に使用する、カウンタを開始する数値を指定します。

桁

カウンタの桁数を指定します。デフォルトは 1 です。桁数に満たない値の場合、桁数が一致するように値の前に「0」のカウンタが付加されます。たとえば、桁数を 3 に設定すると、1 桁の値には 001、002 などのように 0 が付加されます。

備考

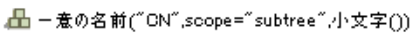
指定されたパターンごとに、ターゲットデータストアに対してのクエリが実行されます。このとき、指定された属性名、スコープおよび検索の開始値が使用されます。指定された各パターンは、見つかったオブジェクトを返さない値が検出されるまで、順に試行されます。

指定されたパターンがすべてなくなった場合は、最後のパターンにカウンタが追加され、クエリがインスタンスを返さなくなるまで、そのパターンが繰り返し試行されます (カウンタが毎回増えます)。

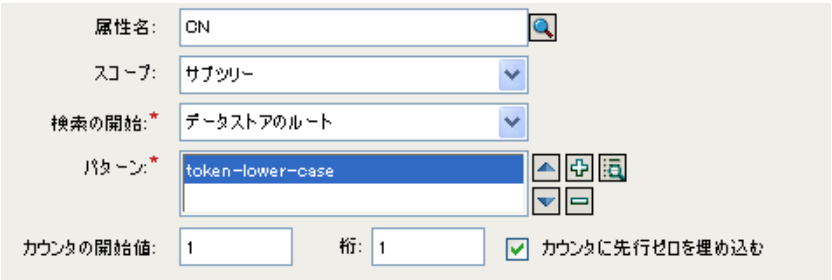
開始番号として別の番号を設定するには、[カウンタの開始] フィールドを使用します。カウンタは、[桁] フィールドで指定された桁数を使用します。桁数が指定された桁数より少ない場合、カウンタは右に詰められ、0 でパディングされます。桁数が指定された桁数より多い場合、一意の名前は生成されず、トークンで指定しているルールがエラーステータスを返します。

ターゲットデータストアが識別ボールトであり、[名前] フィールドが空白のままである場合は、擬似属性「[Entry].rdn」に対して検索が実行されます。これは、命名属性が何であるかにかかわらず、オブジェクトの RDN を示します。ターゲットデータストアが接続アプリケーションの場合、[名前] フィールドは必須です。

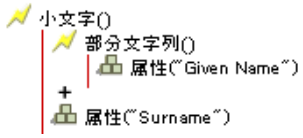
例



次に示すのは、一意の名前引数を作成するときの [エディタ] ペインの例です。



次のパターンは、一意の名前を提供するために作成されました。



このパターンで一意の名前を生成しない場合は、数値が 1 つ追加され、指定された桁数になるまで増分されます。この例では、エラーが発生するまで、数字を追加することで一意の名前が 9 つ生成されます (パターン 1 からパターン 9)。

3.7.23 一致しないソース DN

If ソース DN 条件との最後の検索で一致しなかった DN の一部に対応する、現在の操作に含まれるソース DN の一部分に展開します。

フィールド

変換

ターゲットデータストアで使用される DN のフォーマットに変換するかどうかを選択します。

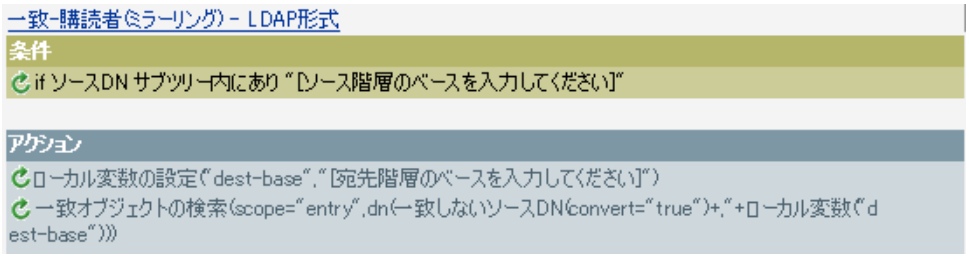
備考

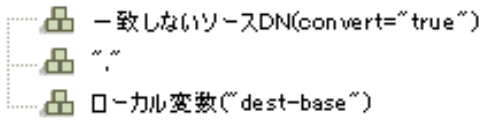
一致するものがなかった場合は、DN 全体が使用されます。

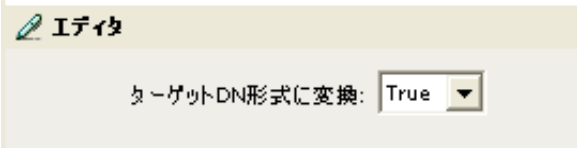
例

この例は、Identity Manager 3.0 に付属している事前定義されたルールからのものです。詳細については、241 ページの 「一致 - 購読者 (ミラーリング) - LDAP 形式」を参照してください。

一致オブジェクトの検索アクションでは、一致しないソース DN トークンを使用して、一致情報を LDAP 形式で作成します。ソース DN の一致しなかった部分を使用して、一致作業を行います。







3.7.24 XPath

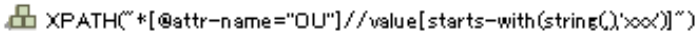
XPath 1.0 の式の評価結果に展開します。

フィールド

式

評価する XPath 1.0 の式。

例



3.8 動詞トークン

この節では、ポリシービルダインタフェースで使用できるすべての動詞について、詳しく説明します。

- 324 ページのセクション 3.8.1「ターゲット DN のエスケープ」
- 325 ページのセクション 3.8.2「ソース DN のエスケープ」
- 325 ページのセクション 3.8.3「小文字」
- 326 ページのセクション 3.8.4「DN の解析」
- 328 ページのセクション 3.8.5「すべて置換」
- 329 ページのセクション 3.8.6「最初を置換」
- 330 ページのセクション 3.8.7「部分文字列」
- 331 ページのセクション 3.8.8「大文字」

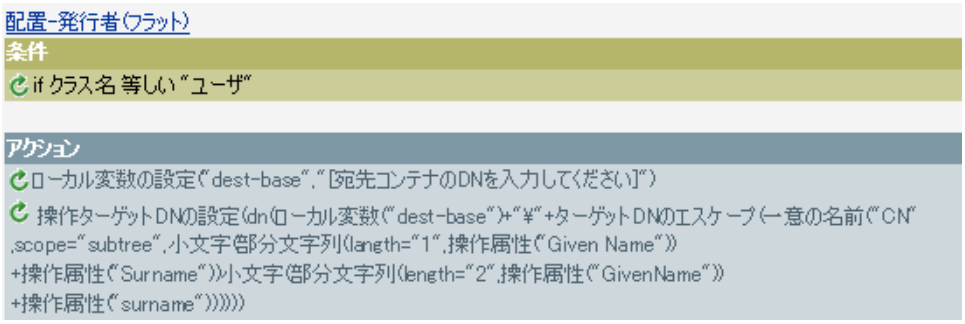
3.8.1 ターゲット DN のエスケープ

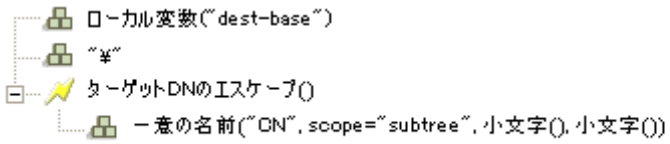
ターゲットデータストアの DN フォーマットのルールに従って文字列をエスケープします。

例

この例は、Identity Manager 3.0 に付属している事前定義されたルールからのものです。詳細については、99 ページの 「配置 - 発行者 (フラット)」を参照してください。

操作ターゲット DN の設定アクションでは、ターゲット DN のエスケープトークンを使用して、ユーザオブジェクトのターゲット DN を作成します。



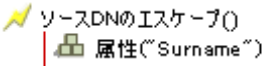


ターゲット DN のエスケープでは、一意の名前の値を取得して、これをターゲット DN の形式に設定します。

3.8.2 ソース DN のエスケープ

ソースデータストアの DN フォーマットのルールに従って文字列をエスケープします。

例



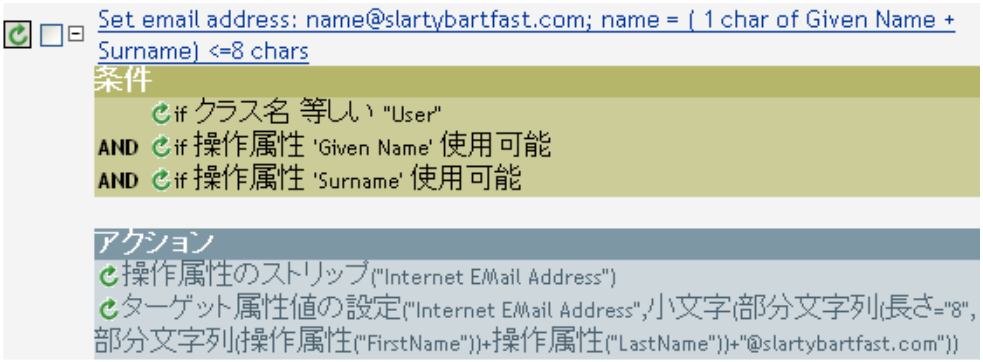
3.8.3 小文字

文字列内の文字を小文字に変換します。

例

この例では、電子メールアドレスを「name@slartybartfast.com」に設定します。name の部分は、名前と名字の最初の文字になります。これは「Policy: Create E-mail from Given Name and Surname (ポリシー : 名前と名字から電子メールを作成)」という名前のポリシー

で、Novell のサポート Web サイトでダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。



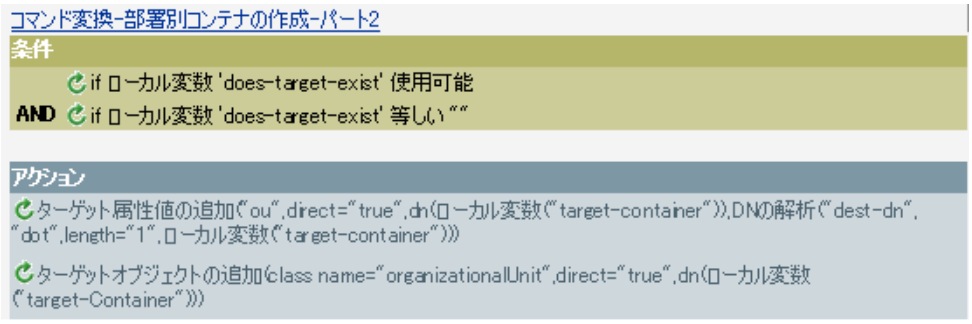
小文字トークンは、ターゲット属性値の設定アクションの情報を、すべて小文字に設定します。

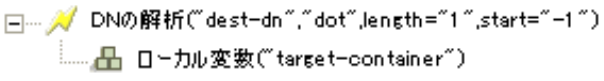
3.8.4 DN の解析

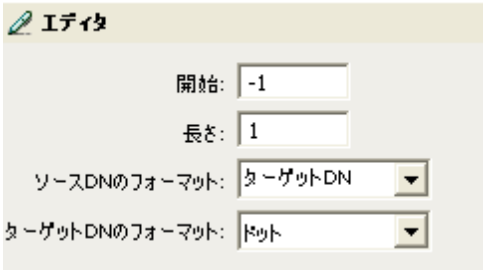
DN を別の形式に変換します。.

例

この例では、DN の解析トークンを使用して、ターゲット属性値の追加アクションの値を作成します。この例は、Identity Manager 3.0 に付属している事前定義されたルールからのものです。詳細については、229 ページの 「コマンド変換 - 部署別コンテナの作成 - パート 1 とパート 2」を参照してください。







DN の解析トークンは、ソース DN から情報を取得し、これをドット表記に変更しています。DN の解析からの情報は、OU の属性値に保存されます。

フィールド

開始

開始の RDN インデックスを指定します。

- インデックス 0 はルートに最も近い RDN
- 正のインデックスはルートに最も近い RDN からのオフセット
- インデックス -1 はリーフに最も近いセグメント
- 負のインデックスは、リーフに最も近い RDN からルートに最も近い RDN 方向へのオフセット

長さ

含める RDN の数です。負の数は (セグメント総数 + 長さ) +1 のように解釈されます (たとえば、セグメント数が 5 の DN では、長さが -1 の場合は -1 = (5 + (-1)) + 1 = 5、長さが -2 の場合は -2 = (5 + (-2)) + 1 = 4)。

ソース **DN** のフォーマット

ソース DN の解析に使用されるフォーマットを指定します。

ターゲット **DN** のフォーマット

解析された DN の出力に使用されるフォーマットを指定します。

ソース **DN** 区切り文字

ソース DN のフォーマットが［カスタム］に設定されている場合に、カスタムのソース DN 区切り文字を指定します。

ターゲット **DN** 区切り文字

ターゲット DN のフォーマットが［カスタム］に設定されている場合に、カスタムのターゲット DN 区切り文字を指定します。

備考

「開始」または「長さ」がデフォルト値 {0、-1} に設定されている場合は、DN 全体が使用されます。それ以外の場合は、「開始」または「長さ」で指定された DN の一部分が使用されます。

カスタムの DN フォーマットを指定する場合、区切り文字を構成する 8 文字は次のように定義されます。

1. タイプ付きの名前のブールフラグ : 0 は名前がタイプなし、1 はタイプ付きであることを示します。

2. Unicode No-Map 文字のブールフラグ :0 は、マップ不可能な Unicode 文字を出力しない、または ¥FEFF などのエスケープ処理された 16 進数字の文字列として変換しないことを意味します。eDirectory では、Unicode 文字の 0xfeff、0xfffe、0xfffd、および 0xffff は使用できません。

3. 相対 RDN 区切り文字

4. RDN 区切り文字

5. 名前ディバイダ

6. 名前の値の区切り文字

7. ワイルドカード文字

8. エスケープ文字

RDN 区切り文字と相対 RDN 区切り文字が同じ文字である場合、名前の向きは右から左、それ以外の場合は左から右になります。

区切り文字セットが 8 文字を超える場合、超過した文字はエスケープ処理が必要な文字と見なされるだけで、それ以外の特別な意味は考慮されません。

3.8.5 すべて置換

文字列内の正規表現と一致したものをすべて置換します。

フィールド

正規表現

置換される部分文字列と一致させる正規表現を指定します。

置換文字列

置換する文字列を指定します。

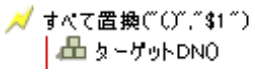
備考

正規表現の作成についての詳細は、次を参照してください。

- Sun の Java Web サイト (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html)
- Sun の Java Web サイト (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Matcher.html#replaceAll (java.lang.String))

［パターン］のオプションには CASE_INSENSITIVE、DOTALL、および UNICODE_CASE が使用されますが、適切な埋め込みエスケープを使用して逆の意味を指定することができます。

例



3.8.6 最初を置換

文字列内の正規表現と最初に一致したものを置換します。

フィールド

正規表現

置換する部分文字列を示す正規表現を指定します。

置換文字列

置換する文字列を指定します。

備考

一致したインスタンスは、[置換文字列] フィールドで指定された値で指定された文字列に置き換えられます。

正規表現の作成についての詳細は、次を参照してください。

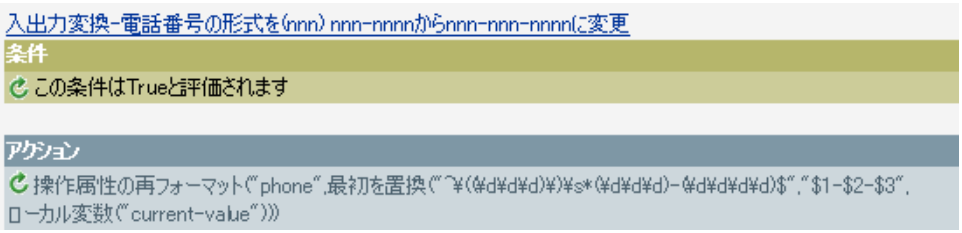
- Sun の Web サイト (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html)
- Sun の Web サイト (java.lang.String) (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Matcher.html#replaceAll (java.lang.String))

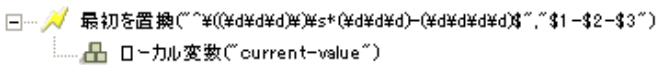
[パターン] のオプションには CASE_INSENSITIVE、DOTALL、および UNICODE_CASE が使用されますが、適切な埋め込みエスケープを使用して逆の意味を指定することができます。

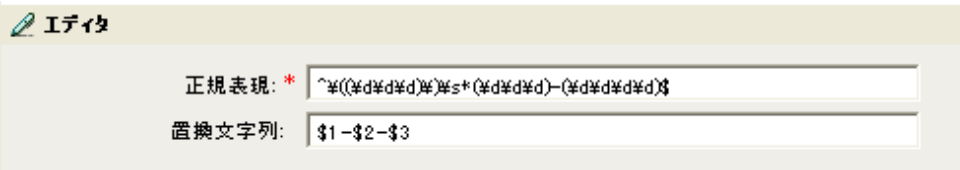
例

この例では、電話番号 (nnn)-nnn-nnnn を nnn-nnn-nnnn に再フォーマットします。このルールは、Identity Manager 3.0 に付属している事前定義されたルールです。詳細については、238 ページの 「入出力変換 - 電話番号の形式を (nnn) nnn-nnnn から nnn-nnn-nnnn に変更」を参照してください。

[最初を置換] トークンは、[操作属性の再フォーマット] アクションで使用されます。







正規表現 ^\((\d\d\d)\)\s*(\d\d\d)-(\d\d\d\d)$ は、(nnn) nnn-nnnn を、正規表現 $1-$2-$3 は nnn を示しています。このルールでは、電話番号の形式を (nnn) nnn-nnnn から nnn-nnn-nnnn に変更します。

3.8.7 部分文字列

文字列の一部を抽出します。

フィールド

開始

開始文字のインデックスを指定します。

- インデックス 0 は 1 文字目です。
- 正のインデックスは文字列の先頭からのオフセットです。
- インデックス -1 は最後の文字です。
- 負のインデックスは、最後の文字から文字列の先頭方向へのオフセットです。

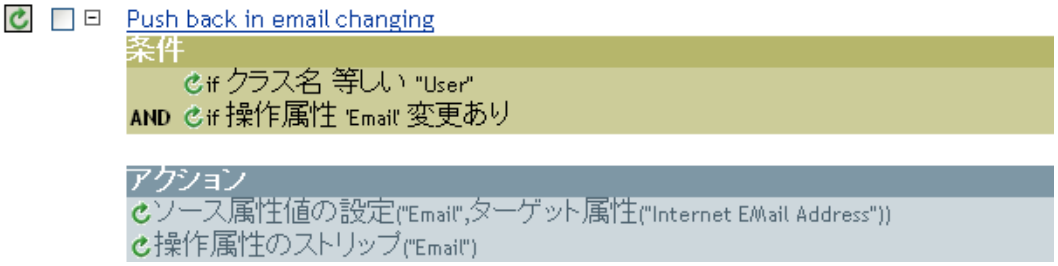
たとえば、開始が -2 に設定されると、最後の文字から読み込みが開始されます。-3 が指定されると、最後から 2 文字目で開始されます。

長さ

部分文字列に含める、開始位置からの文字数。負の数は (文字総数 + 長さ) + 1 のように解釈されます。たとえば、-1 の場合は全長または元の文字列を表します。-2 が指定されると、「全体の長さ -1」になります 5 文字の文字列の場合、長さが -1 の場合は -1= (5 + (-1)) + 1 = 5、長さが -2 の場合は -2 = (5 + (-2)) + 1 = 4 になります。.

例

この例では、電子メールアドレスを「name@slartybartfast.com」に設定します。name の部分は、名前と名字の最初の文字になります。これは「Policy: Create E-mail from Given Name and Surname (ポリシー : 名前と名字から電子メールを作成)」という名前のポリシーで、Novell のサポート Web サイトでダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。



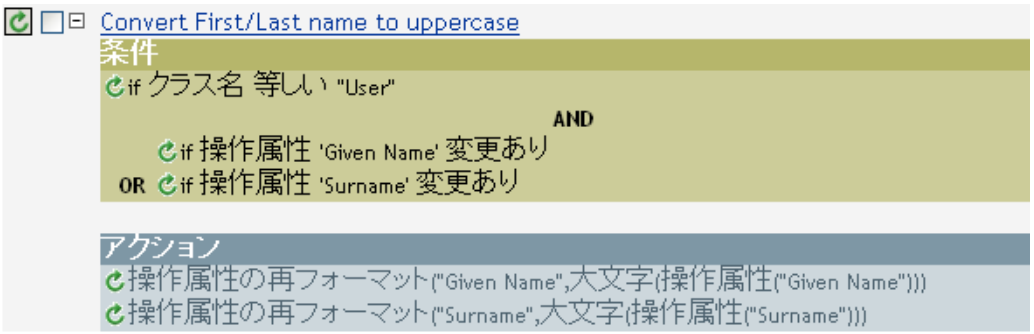
部分文字列トークンは、ターゲット属性値の設定アクションで 2 度使用されます。名前属性の最初の文字列を取得し、名字属性の 8 文字を追加して、1 つの部分文字列を作成します。

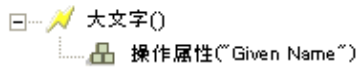
3.8.8 大文字

文字列内の文字を大文字に変換します。

例

この例では、ユーザオブジェクトの名前と名字の属性を大文字に変換します。これは「Policy: Convert First/Last Name to Upper Case (ポリシー : 名前と名字を大文字に変換)」という名前のポリシーで、Novell のサポート Web サイトでダウンロードできます。詳細については、33 ページの 「ダウンロード可能な Identity Manager ポリシー」を参照してください。





3.9 値

この節では、ポリシービルダに共通の値を一覧表示しています。

3.9.1 比較モード

モード	説明
case (大文字と小文字の区別あり)	1 文字ずつ比較する (大文字と小文字の区別あり)。
nocase (大文字と小文字の区別なし)	1 文字ずつ比較する (大文字と小文字の区別なし)。
正規表現	文字列全体を正規表現で比較する。デフォルトでは大文字と小文字は区別されませんが、式でエスケープして変更できます。 Sun の Java Web サイト (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html) および Sun の Java Web サイト (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Matcher.html#matches()) を参照してください。 ［パターン］のオプションには CASE_INSENSITIVE、DOTALL、および UNICODE_CASE が使用されますが、適切な埋め込みエスケープを使用して逆の意味を指定することができます。
src-dn (ソース DN)	ソースデータストアの DN のフォーマットに対する適切なセマンティックを使用して比較します。
dest-dn (ターゲット DN)	ターゲットデータストアの DN のフォーマットに対する適切なセマンティックを使用して比較します。
数字	数字を比較します。
octet (オクテット)	オクテット値 (Base64 でエンコード) で比較します。
構造	属性の構造構文の比較ルールに従って、構造属性を比較します。

Novell 資格情報プロビジョニングポリシー

4

Identity Manager 3 の Novell® 資格情報プロビジョニングポリシーは、アプリケーション資格情報を Novell SecretStore® および Novell SecureLogin 資格情報のリポジトリに同時にプロビジョニングする機能を実現することにより、すべての Identity Manager ドライバのユーザプロビジョニング機能が拡張されています。加えて、この製品では、否認防止が必要な環境で SecureLogin パスフレーズの質問と回答をプロビジョニングできます。

これらの機能によりユーザのシングルサインオン (SSO) の操作性を向上させ、SSO 技術への投資に対する見返りを増やすには、SecureLogin アカウント情報の初期設定をなくし、アプリケーション資格情報のセキュリティを高め、ユーザの SSO 資格情報ストアのプロビジョニングに通常関連する作業の反復を減らします。また、資格情報プロビジョニングポリシーで Identity Manager ポリシーを使用することで、アプリケーション資格情報を自動的にプロビジョニング解除し、アプリケーションデータへのアクセスを防ぐことができます。

- 333 ページのセクション 4.1「Novell SecureLogin による資格情報プロビジョニングポリシー」
- 336 ページのセクション 4.2「SecureLogin による資格情報プロビジョニングポリシーの実装」
- 357 ページのセクション 4.3「Novell SecretStore による資格情報プロビジョニングポリシー」
- 360 ページのセクション 4.4「SecretStore による資格情報プロビジョニングポリシーの実装」

4.1 Novell SecureLogin による資格情報プロビジョニングポリシー

資格情報プロビジョニングポリシーにより、SecureLogin がサポートするアプリケーション資格情報を自動的にプロビジョニングできます。このトピックでは、Identity Manager 内のオブジェクトとポリシーを設定するために必要な手順について記載しています。SecureLogin コンポーネントの展開および設定についての情報は含まれていません。SecureLogin のマニュアルは、「Novell SecureLogin 6.0 (http://www.novell.com/documentation/securelogin60/index.html)」のマニュアルを参照してください。

SecureLogin を用いて資格情報のプロビジョニングを実装するには、リポジトリオブジェクト、アプリケーションオブジェクトおよびポリシーが必要です。リポジトリとアプリケーションのオブジェクトには、Identity Manager が使用できるように SecureLogin の情報が格納されます。ポリシーは、ドライバで資格情報プロビジョニングを使用できるようにするために使用されます。詳細については、336 ページのセクション 4.2「SecureLogin による資格情報プロビジョニングポリシーの実装」を参照してください。

次のオプションも設定できます。

- 資格情報プロビジョニングは、発行者チャネル、購読者チャネル、または両方のチャネルで使用できます。

- SecureLogin の同期は、アプリケーションのパスワード同期の一部として実行したり、他のイベントにトリガさせたりすることができます。
- Web サービスの資格情報は、アプリケーションのアカウントをプロビジョニングしなくてもプロビジョニングできます。
- SecureLogin パスフレーズの初期の質問と回答をプロビジョニングできます。

335 ページの 図 4-1 は、一般的なシナリオを簡略に示したものです。このシナリオでは、Finance 部の SAP* Finance アプリケーションの新規ユーザに対し、SecureLogin 資格情報をプロビジョニングしています。SAP アプリケーションでは、通常のアプリケーションで指定する一般的なユーザ名とパスワードのほかにもログインパラメータが必要なため、SAP ユーザのプロビジョニングが使用されます。

この部署では、SAP HR システムと Identity Manager を使用して、アイデンティティボールト内に新しいユーザをプロビジョニングします。組織の情報に基づき、ユーザオブジェクトは Active Directory 内に実装された部署の認証ツリー内にプロビジョニングされます。ここが新しいユーザがネットワークに対して認証される場所であり、SecureLogin 資格情報のリポジトリの場所になります。続いて、Identity Manager によって、ユーザはさまざまな Finance アプリケーションに対しプロビジョニングされ、それらのシステムの資格情報は、Active Directory 内の SecureLogin ストアに同期されます。

図 4-1 は、ユーザ Glen の認証資格情報がプロビジョニングされているところを示しています。Glen が自分の部署の Active Directory 認証ドメインに対して認証を実行すると、

SecureLogin クライアントが起動し、SAP Finance のアカウントへシングルサインオンできます。このとき、SAP のパスワードを入力したり、記憶している必要さえありません。

図 *4-1* *SecureLogin* による資格情報プロビジョニング

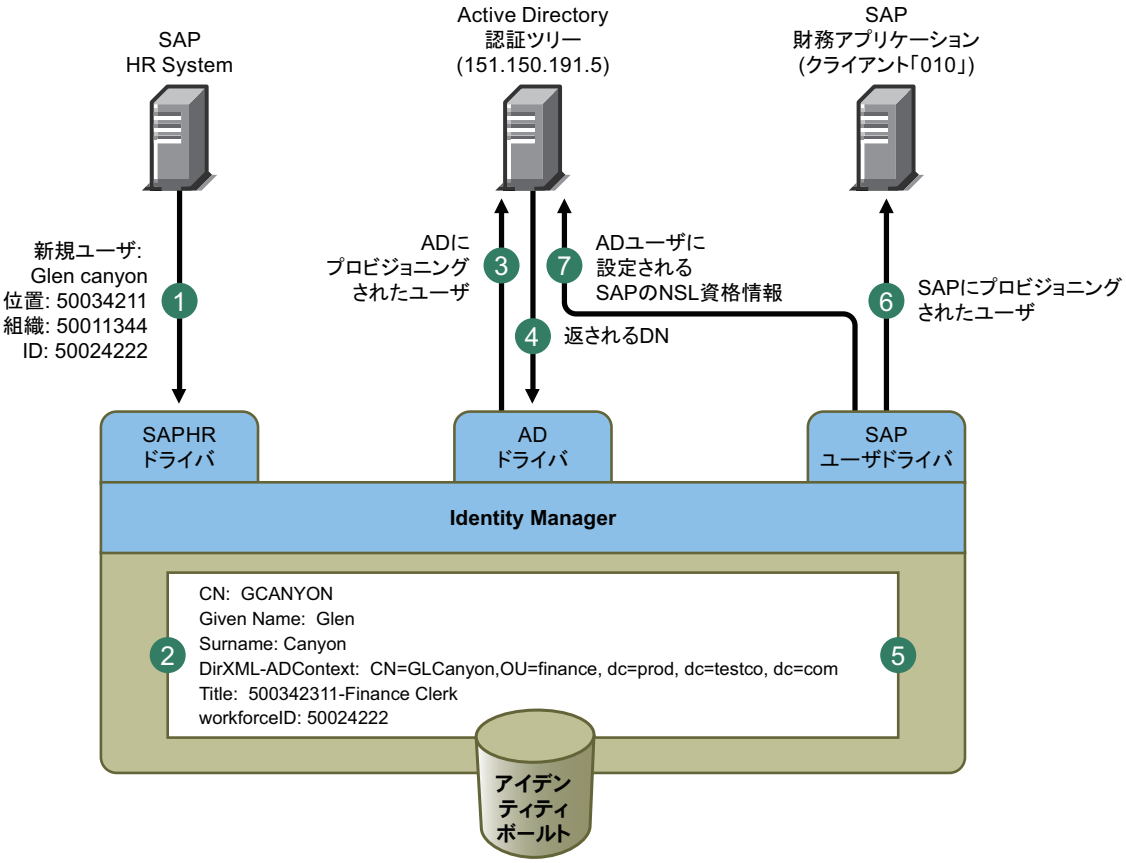


図 4-1 は、次の手順を示しています。

1. SAP HR システムが、新入社員 Glen Canyon のデータを発行します。Identity Manager の SAP HP ドライバが、このデータを処理します。
2. CN 値「GCANYON」および workforceID 値「50024222」をもつ新しいユーザオブジェクトが、アイデンティティボールト内に作成されます。このユーザは、会社の Finance 組織に割り当てられているため、ドメイン finance.prod.testco.com domain にある Finance 部の Active Directory サーバで認証を受ける必要があります。ドメインを同期する Identity Manager の Active Directory ドライバは、アイデンティティボールトの情報を使用するようになりました。
3. Glen は、Finance 部の Active Directory サーバにプロビジョニングされます。
4. ドライバは、Glen の LDAP 形式の完全識別名を取得するように設定されます。CN=GLCanyon,OU=finace,dc=prod,dc=testco,dc=com.
5. ドライバは、この名前をアイデンティティボールト内の GCANYON ユーザの DirXML-ADContext 属性に配置します。

   これで、アイデンティティボールト内で必要な属性が使用できるようになったので、SAP ユーザ管理ドライバによって、GCANYON オブジェクトの属性が処理されます。

6. Glen は Finance 組織に所属するため、ドライバは SAP Finance サーバ上ある SAP ユーザアカウントの GCANYON に対してプロビジョニングを行います。
7. アカウントの作成が成功すると、SAP ユーザ管理ドライバのポリシーによって、Glen の SAP 認証資格情報がこのユーザの AD ユーザアカウントにプロビジョニングされます。コマンドが「追加」操作であるため、ポリシーは SecureLogin パスフレーズの質問と回答もプロビジョニングします。

4.2 SecureLogin による資格情報プロビジョニングポリシーの実装

SecureLogin による資格情報プロビジョニングポリシーの実装は、柔軟にカスタマイズできます。実装手順は、SecureLogin がインストールされているプラットフォーム、プロビジョニング対象のアプリケーション、使用する Identity Manager ドライバによって異なります。

SecureLogin による資格情報プロビジョニングポリシーを実装するには、次のトピックを参照してください。

- 336 ページのセクション 4.2.1「Novell SecureLogin による資格情報プロビジョニングポリシーの要件」
- 337 ページのセクション 4.2.2「Novell SecureLogin の LDAP スキーマの拡張」
- 337 ページのセクション 4.2.3「Novell SecureLogin の展開環境設定パラメータの決定」
- 340 ページのセクション 4.2.4「Novell SecureLogin のリポジトリオブジェクトの作成」
- 347 ページのセクション 4.2.5「Novell SecureLogin のアプリケーションオブジェクトの作成」
- 353 ページのセクション 4.2.6「Novell SecureLogin の資格情報プロビジョニングポリシーの環境設定」

4.2.1 Novell SecureLogin による資格情報プロビジョニングポリシーの要件

SecureLogin による資格情報プロビジョニングポリシーを使用するには、次の要件を満たす必要があります。

- Support Pack 1 が適用された Identity Manager 3.0
  - eDirectory™ 8.7x がインストールされている必要があります。eDirectory 8.8 はサポートされていません。
  - jsso.jar、idmcp.jar および jnet.jar が Identity Manager Java ライブラリの標準の場所にあることを確認します。
- Novell SecureLogin 6.0 以降

要件が満たされていることを確認したら、337 ページのセクション 4.2.2「Novell SecureLogin の LDAP スキーマの拡張」に進んでください。

4.2.2 Novell SecureLogin の LDAP スキーマの拡張

SecureLogin を eDirectory サーバ上に展開する場合、ndsschema.exe というツールを使用して、SecureLogin の属性セットをもつ eDirectory スキーマを拡張します。これらの属性は、暗号化された資格情報、ポリシーなどをユーザおよびコンテナのオブジェクトに保存するのに使用されます。属性を次に示します。

- Prot:SSO Auth
- Prot:SSO Entry
- Prot:SSO Entry Checksum
- Prot:SSO Profile
- Prot:SSO Security Prefs
- Prot:SSO Security Prefs Checksum

これらの属性は eDirectory に特有のもので、SecureLogin 製品を機能させるために必要です。Identity Manager 3.0 Support Pack 1 に付属しているプロビジョニング API では、LDAP ネームスペースを使用してその機能を実行することで、あらゆる SecureLogin 資格情報ストアと連動できるようにします。上記の属性への LDAP マッピングを行うには、SecureLogin 製品に付属している 2 つめのツールを使用します。このツールの名前は ldapschema.exe で、eDirectory 属性へ LDAP ネームスペースをマッピングするため、eDirectory 環境で使用されます。

ldapschema.exe を実行したら、iManager で LDAP グループ属性を確認することで、マッピングを確認します。

1. iManager で、［LDAP］>［LDAP Options (LDAP オプション)］の順にクリックします。
2. SecureLogin をホストする eDirectory サーバに関連付けられた LDAP グループを選択します。
3. ［LDAP Group (LDAP グループ)］プロパティページから、［Attribute Map (属性マップ)］オプションを選択し、上記の属性が次の［Primary LDAP Attributes (プライマリ LDAP 属性)］にマップされていることを確認します。
   - protocom-SSO-Auth-Data
   - protocom-SSO-Entries
   - protocom-SSO-Entries-Checksum
   - protocom-SSO-Profile
   - protocom-SSO-Security-Prefs
   - protocom-SSO-Security-Prefs-Checksum

スキーマを拡張したら、337 ページのセクション 4.2.3「Novell SecureLogin の展開環境設定パラメータの決定」に進んでください。

4.2.3 Novell SecureLogin の展開環境設定パラメータの決定

図 4-1 の展開シナリオで説明されている同期機能を提供するには、まず、Identity Manager および SecureLogin 環境に関連したすべてのビジネスプロセス情報を収集します。338 ページの ï 4-1§ 「SecureLogin の資格情報プロビジョニングポリシーワークシート」を印刷して、情報を記録するためのワークシートとして使用してください。

表 **4-1** *SecureLogin* の資格情報プロビジョニングポリシーワークシート

必要な環境設定情報	情報
1) SecureLogin シングルサインオンのプロビジョニング用に設定するアプリケーション。	
2) SecureLogin アプリケーションの定義が認証サーバであらかじめ設定されていて、その内容をこれらのシステムにプロビジョニングされる新規ユーザが継承可能であることを確認する。	
3) SecureLogin リポジトリサーバの DNS 名または IP アドレス。	
4) SecureLogin リポジトリサーバの SSL LDAP ポート。	
5) SecureLogin リポジトリサーバ管理者の完全修飾された LDAP 識別名。	
6) SecureLogin リポジトリサーバの管理者のパスワード。	
7) SecureLogin サーバからエクスポートされる、SSL 証明書へのフルパスおよび証明書名。証明書は、Identity Manager サーバのローカルに配置する必要があります。	
8) 1 つの SecureLogin リポジトリを複数のドライバで使用するか、または各ドライバで専用のリポジトリを使用するかを決定する。	
9) SecureLogin アプリケーションごとのアプリケーション ID。	
10) アプリケーションごとに必要な認証キーを用意する。ユーザ名、パスワード、クライアントおよび言語など。これらはアプリケーションごとに異なる場合があります。	
11) 認証キーの値をスタティックな値に設定するかどうかを決定する。	
12) ユーザごとに異なる値である (または異なる値にできる) スタティックでない値の場合は、そのスタティックでない情報のソースを書き留める (イベント情報またはアイデンティティボールトの属性値)。	
13) ターゲットアプリケーションへのパスワードも同期しているドライバに SecureLogin のプロビジョニングを実装する場合、SecureLogin のプロビジョニングを、ターゲットアプリケーションのサーバにパスワードが設定される前と後のどちらで開始するかを決定する。	
14) リポジトリおよびアプリケーションのオブジェクトが格納されるドライバオブジェクトの名前 (格納先ドライバは別々に指定可能)。	

必要な環境設定情報	情報
15) ターゲットアプリケーションのユーザオブジェクトの DN を決定する。	
16) SecureLogin パスフレーズを実装する場合、パスフレーズの質問と回答を決定する。	質問 : 回答 :

プロビジョニング環境設定データの例

プロビジョニングシナリオを使用したサンプルデータを次に示します。ここでは、Finance 部の Active Directory 認証ツリー内のユーザに、SAP Finance サーバの SecureLogin 資格情報をプロビジョニングします。

表 *4-2* *SecureLogin* の資格情報プロビジョニングポリシーワークシートの例

必要な環境設定情報	情報
1) SecureLogin シングルサインオンのプロビジョニング用に設定するアプリケーション。	SAP Finance アプリケーション
2) SecureLogin アプリケーションの定義が認証サーバであらかじめ設定されていて、その内容をこれらのシステムにプロビジョニングされる新規ユーザが継承可能であることを確認する。	確認済み
3) SecureLogin リポジトリサーバの DNS 名または IP アドレス。	151.150.191.5
4) SecureLogin リポジトリサーバの SSL LDAP ポート。	636
5) SecureLogin リポジトリサーバ管理者の完全修飾された LDAP 識別名。	cn=admin,ou=prod,dc=testco,dc=.com
6) SecureLogin リポジトリサーバの管理者のパスワード。	dixml
7) SecureLogin サーバからエクスポートされる、SSL 証明書へのフルパスおよび証明書名。証明書は、Identity Manager サーバのローカルに配置する必要があります。	c:\novell\nds\FinanceAD.cer
8) 1 つの SecureLogin リポジトリを複数のドライバで使用するか、または各ドライバで専用のリポジトリを使用するかを決定する。	この例では、リポジトリは 1 つだけにします。
9) SecureLogin アプリケーションごとのアプリケーション ID。	SAP - 151.150.191.27
10) アプリケーションごとに必要な認証キーを用意する。ユーザ名、パスワード、クライアントおよび言語など。これらはアプリケーションごとに異なる場合があります。	SAP Client 010 ログインパラメータクライアント SAP Client 010 ログインパラメータ言語 SAP Client 010 ログインパラメータユーザ名 SAP Client 010 ログインパラメータパスワード
11) 認証キーの値をスタティックな値に設定するかどうかを決定する。	SAP Client 010 ログインパラメータクライアント : 「010」SAP Client 010 ログインパラメータ言語 : 「EN」

必要な環境設定情報	情報
12) ユーザごとに異なる値である (または異なる値にできる) スタティックでない値の場合は、そのスタティックでない情報のソースを書き留める (イベント情報またはアイデンティティボールトの属性値)。	SAP Client 010 ログインパラメータユーザ名 : アイデンティティボールト属性「sapUsername」 SAP Client 010 ログインパラメータパスワード : イベント <password>
13) ターゲットアプリケーションへのパスワードも同期しているドライバに SecureLogin のプロビジョニングを実装する場合、SecureLogin のプロビジョニングを、ターゲットアプリケーションのサーバにパスワードが設定される前と後のどちらで開始するかを決定する。	後
14) リポジトリおよびアプリケーションのオブジェクトが格納されるドライバオブジェクトの名前。(格納先ドライバは別々に指定可能)	SAP ドライバ
15) ターゲットアプリケーションのユーザオブジェクトの DN を決定する。	アイデンティティボールトの属性 “DirXML-ADContext”
16) SecureLogin パスフレーズをプロビジョニングする場合、パスフレーズの質問と回答を決定する。	質問 :「従業員コードは ?」回答 : アイデンティティボールトの属性の「workforceID」

その他の環境設定情報 :

- Finance部のADツリーは、すべてのFinanceアプリケーションのSecureLoginリポジトリとして動作します。
- Finance 部関連のプロビジョニングドライバは、すべて「Finance Drivers」という名前のドライバセット内に設定されます。
- アイデンティティボールトの属性 employeeStatus の値が「I」に設定された場合、SAP ユーザアカウントを削除して、その SAP ユーザアカウントの SecureLogin 資格情報も Active Directory ユーザから削除する必要があります。

すべての環境設定データを決定したら、340 ページのセクション 4.2.4「Novell SecureLogin のリポジトリオブジェクトの作成」に進んでください。

4.2.4 Novell SecureLogin のリポジトリオブジェクトの作成

リポジトリオブジェクトには、SecureLogin のスタティックな環境設定情報が保存されます。リポジトリの情報は、アプリケーション資格情報を使用するアプリケーションからは独立しています。この情報は、接続システム (SAP、PeopleSoft*、Notes* など) に関係なく、すべてのプロビジョニングイベントに適用されます。リポジトリオブジェクトは、Designer または iManager で作成できます。

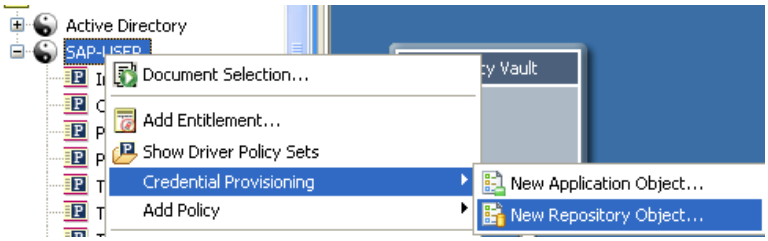
- 340 ページの 「Designer での Novell SecureLogin のリポジトリオブジェクトの作成」
- 344 ページの 「iManager での Novell SecureLogin のリポジトリオブジェクトの作成」

Designer での Novell SecureLogin のリポジトリオブジェクトの作成

次に示すのは、Designer でリポジトリオブジェクトを作成する方法のうちの 1 つです。

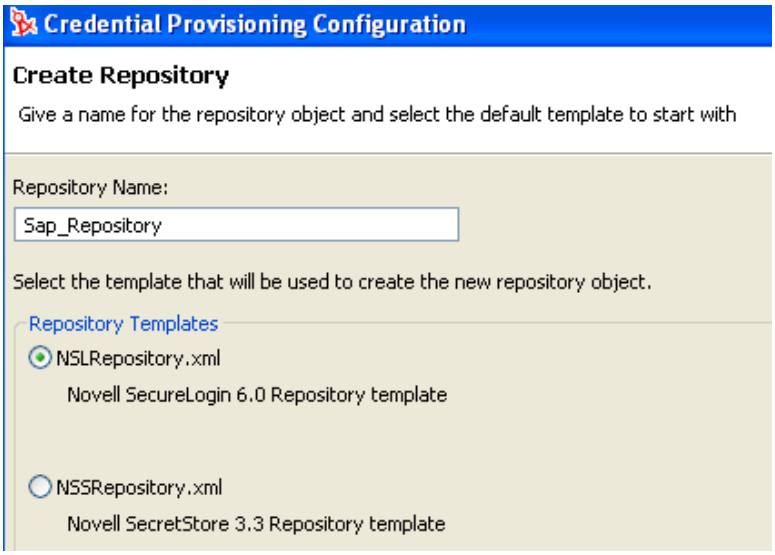
1 アウトラインビューで、リポジトリオブジェクトを格納するドライバオブジェクトを右クリックします。

2 ［資格情報のプロビジョニング］>［New Repository Object (新規リポジトリオブジェクト)］の順にクリックします。



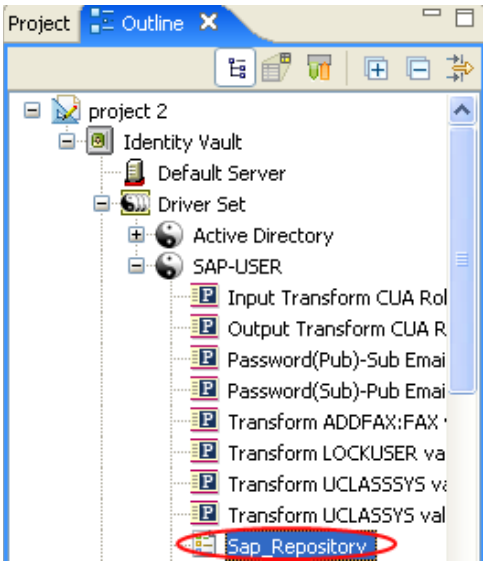
3 リポジトリオブジェクトの名前を指定します。

4 SecureLogin テンプレートを使用するため、［NSLRepository.xml］を選択します。

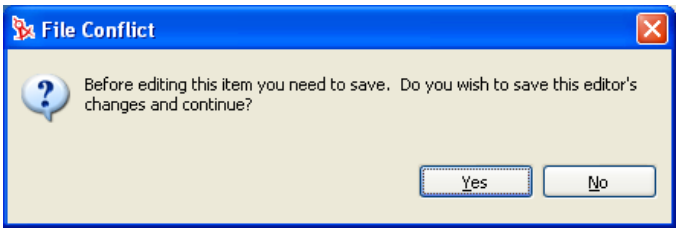


5 ［OK］をクリックします。

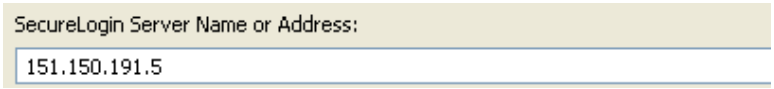
6 環境設定情報を追加するため、アウトラインビューでリポジトリオブジェクトをダブルクリックします。



7 [はい] をクリックして、新しいリポジトリオブジェクトを保存します。

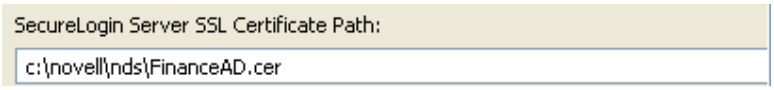


8 SecureLogin サーバの DNS 名または IP アドレスを指定します。ワークシート項目の 3). を参照してください。



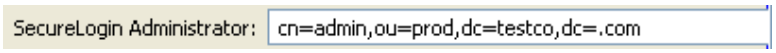
9 SecureLogin サーバの SSL ポートを指定します。ワークシート項目の 4) を参照してください。

10 SecureLogin サーバからエクスポートされる SSL 証明書へのフルパスを指定します。このパスには証明書名を含め、Identity Manager サーバのローカルに配置する必要があります。ワークシート項目の 7) を参照してください。



SecureLogin サーバは、複数のタイプのプラットフォーム上で実行できます。SSL 証明書のエクスポート方法については、プラットフォームごとのマニュアルを参照してください。

11 SecureLogin 管理者の完全修飾された LDAP 識別名を指定します。ワークシート項目の 5) を参照してください。

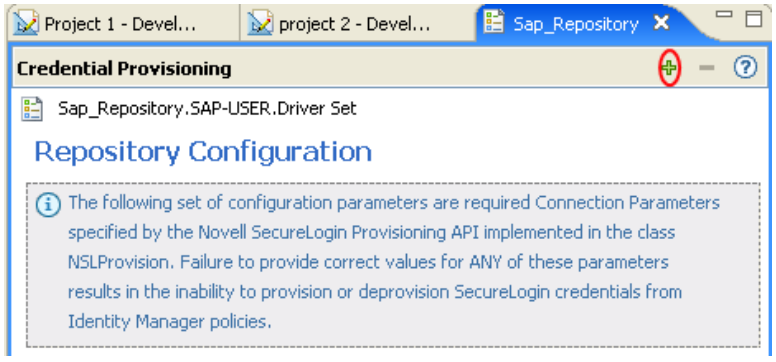


12 ［パスワードを設定する］をクリックします。

13 SecureLogin 管理者のパスワードを 2 回入力し、［OK］をクリックします。ワークシート項目の 6) を参照してください。

14 情報を確認し、［保存］アイコンをクリックして情報を保存します。

15 (オプション) リポジトリオブジェクトに対する他の環境設定パラメータを作成する場合は、［Add new item (新しい項目の追加)］アイコンをクリックします。

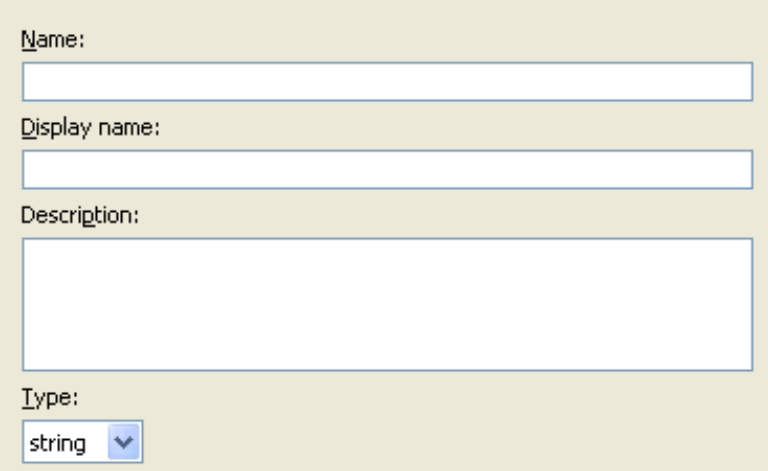


15a パラメータの名前を指定します。

15b パラメータの表示名を指定します。

15c 参照情報として、パラメータの説明を入力します。

パラメータは文字列で保存されます。



15d ［OK］をクリックします。

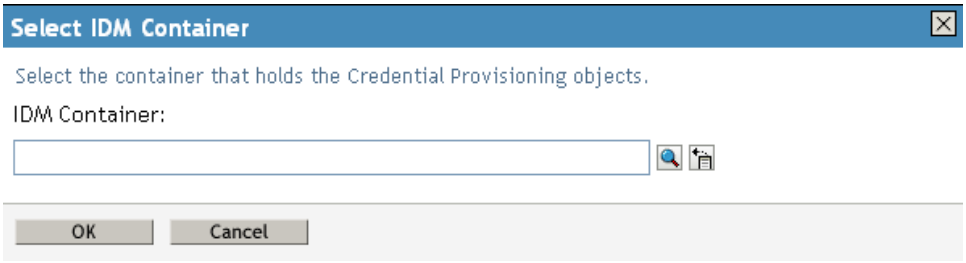
15e ［保存］アイコンをクリックして、リポジトリオブジェクトを保存します。

リポジトリオブジェクトが作成されたら、347 ページの 「Novell SecureLogin のアプリケーションオブジェクトの作成」に進んでください。

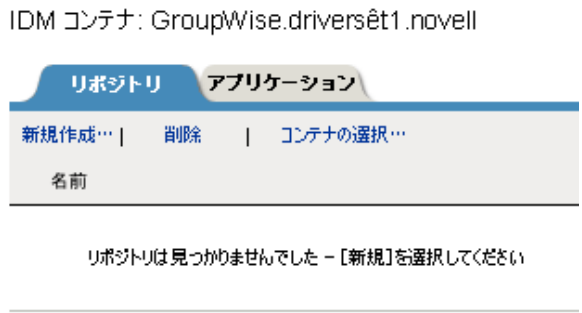
iManager での Novell SecureLogin のリポジトリオブジェクトの作成

1 iManager で、［資格情報のプロビジョニング］>［環境設定］の順に選択します。

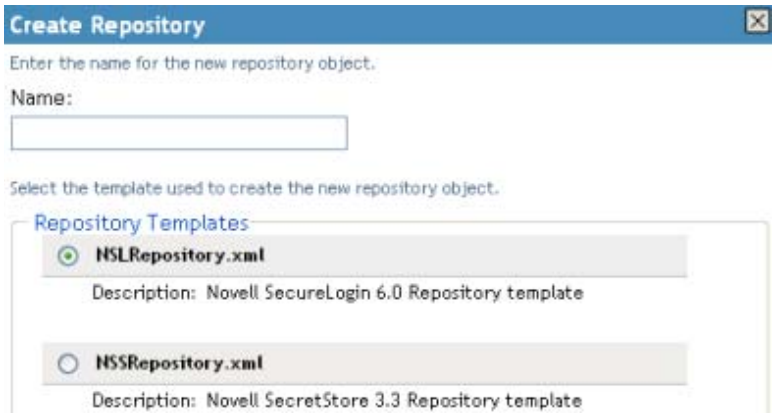
2 リポジトリオブジェクトを保存するドライバオブジェクトを参照して選択し、［OK］をクリックします。



3 ［新規作成］をクリックしてリポジトリを作成します。

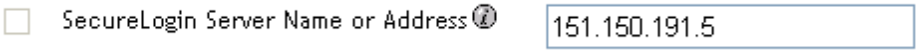


4 リポジトリオブジェクトの名前を指定したら、SecureLogin テンプレートを使用してリポジトリを作成するため、［NSLRepository.xml］を選択します。

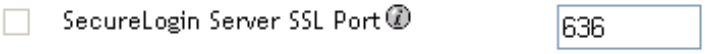


5 ［OK］をクリックします。

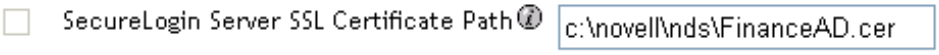
6 SecureLogin サーバの DNS 名または IP アドレスを指定します。ワークシート項目の 3) を参照してください。



7 SecureLogin サーバの SSL ポートを指定します。ワークシート項目の 4) を参照してください。



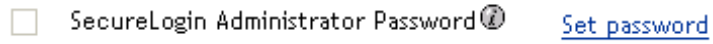
8 SecureLogin サーバからエクスポートされる SSL 証明書へのフルパスを指定します。このパスには証明書名を含め、Identity Manager サーバのローカルに配置する必要があります。ワークシート項目の 7) を参照してください。



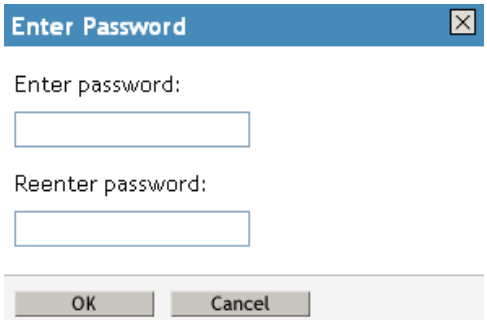
SecureLogin サーバは、複数のタイプのプラットフォーム上で実行できます。SSL 証明書のエクスポート方法については、プラットフォームごとのマニュアルを参照してください。

9 SecureLogin 管理者の完全修飾された LDAP 識別名を指定します。ワークシート項目の 5) を参照してください。

10 ［パスワードを設定する］をクリックします。

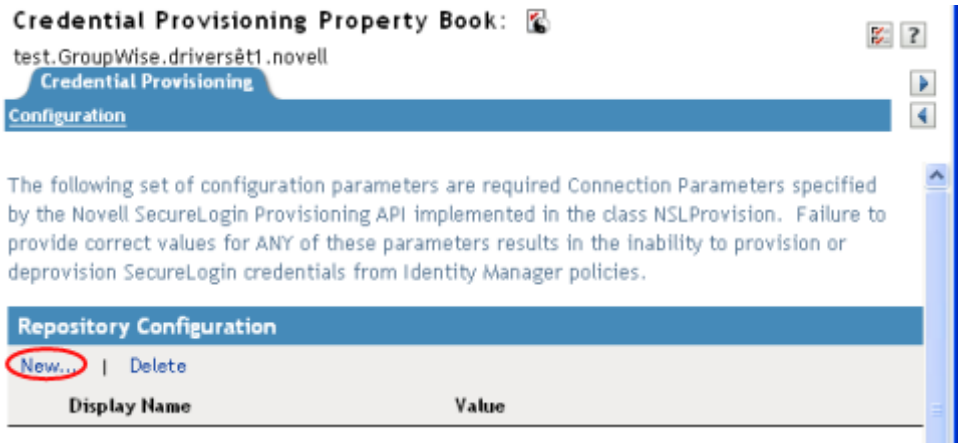


11 SecureLogin 管理者のパスワードを 2 回入力し、［OK］をクリックします。ワークシート項目の 6) を参照してください。



12 指定した値を確認し、［OK］をクリックします。

13 (オプション) リポジトリオブジェクトに対する他の環境設定パラメータを作成する必要がある場合は、［新規作成］をクリックします。

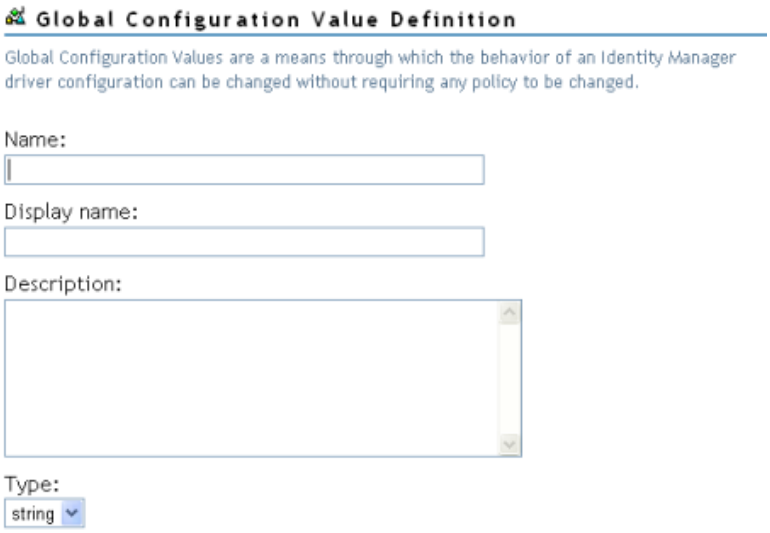


13a パラメータの名前を指定します。

13b パラメータの表示名を指定します。

13c 参照情報として、パラメータの説明を入力します。

パラメータは文字列で保存されます。



13d ［OK］をクリックします。

リポジトリオブジェクトが作成されたら、350 ページの 「iManager での Novell SecureLogin のアプリケーションオブジェクトの作成」に進んでください。

4.2.5 Novell SecureLogin のアプリケーションオブジェクトの作成

アプリケーションオブジェクトには、SecureLogin のアプリケーション認証パラメータ値が保存されます。アプリケーション情報は、そのアプリケーションの資格情報を使用しているアプリケーションに特有のものです (GroupWise® クライアントの情報または SAP データベースクライアントの情報など)。アプリケーションオブジェクトは、Designer または iManager で作成できます。

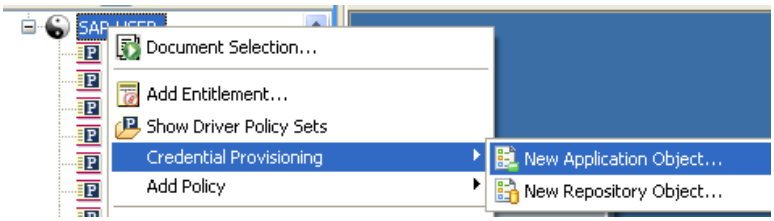
- 347 ページの「Designer での Novell SecureLogin のアプリケーションオブジェクトの作成」
- 350 ページの「iManager での Novell SecureLogin のアプリケーションオブジェクトの作成」

Designer での Novell SecureLogin のアプリケーションオブジェクトの作成

次に示すのは、Designer でアプリケーションオブジェクトを作成する方法のうちの 1 つです。

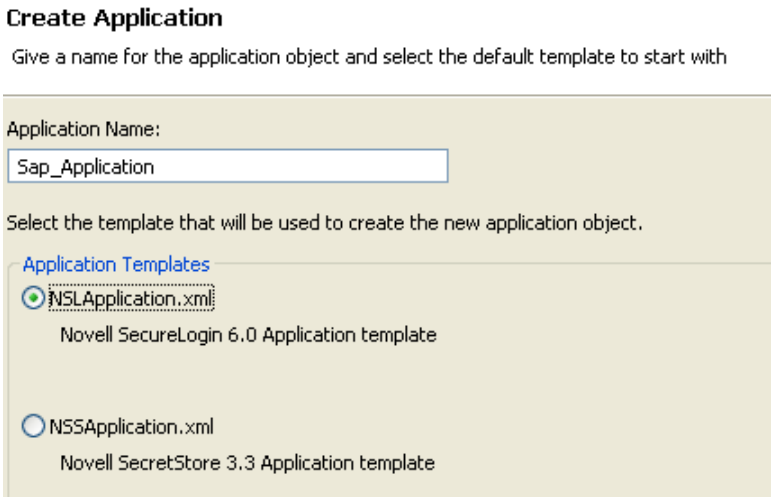
1 アウトラインビューで、アプリケーションオブジェクトを格納するドライバオブジェクトを右クリックします。

2 ［資格情報のプロビジョニング］>［New Application Object (新規アプリケーションオブジェクト)］の順にクリックします。



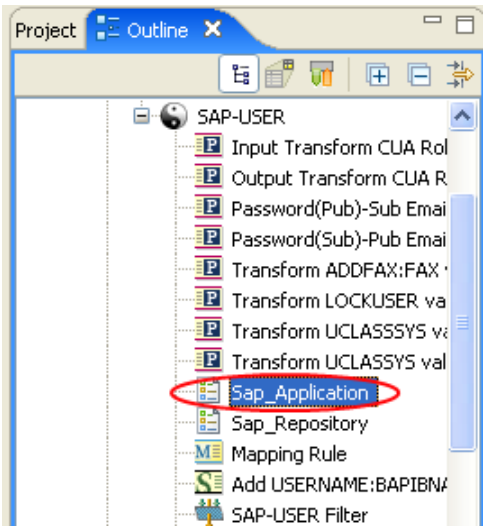
3 アプリケーションオブジェクトの名前を指定します。

4 SecureLogin テンプレートを使用するため、［NSLApplication.xml］を選択します。

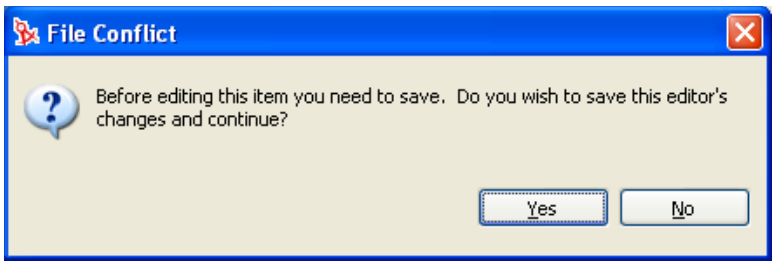


5 ［OK］をクリックします。

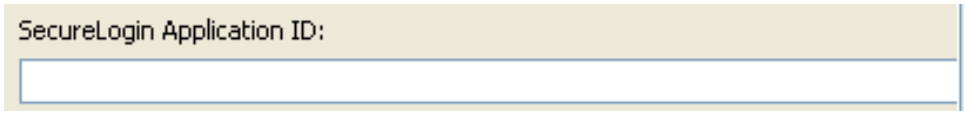
6 環境設定情報を追加するため、アウトラインビューでアプリケーションオブジェクトをダブルクリックします。



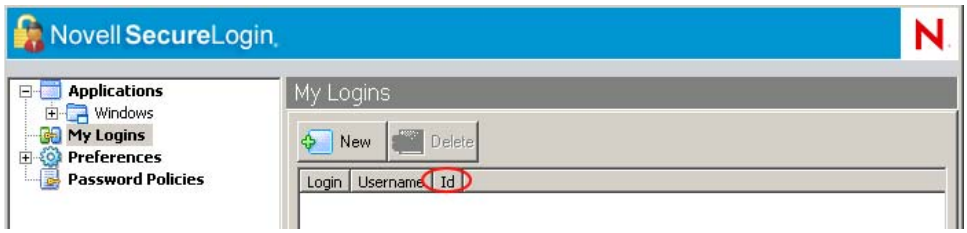
7 ［はい］をクリックして、新しいアプリケーションオブジェクトを保存します。



8 SecureLogin のアプリケーション ID を指定します。ワークシート項目の 9) を参照してください。

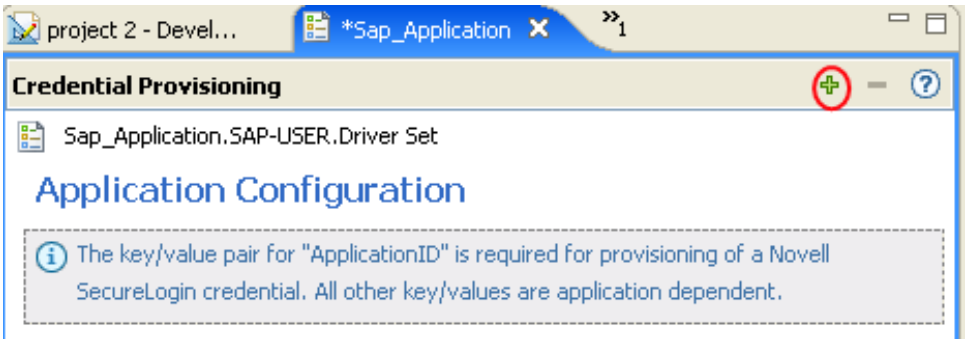


SecureLogin のアプリケーション ID を見つけるには、［My Logins (マイログイン)］をクリックします。アプリケーション ID は、［Id (ID)］フィールドに保存されています。



9 ［保存］アイコンをクリックして、アプリケーションを保存します。

10 アプリケーションに必要な認証キーを追加するため、［Add new item (新しい項目の追加)］アイコンをクリックします。

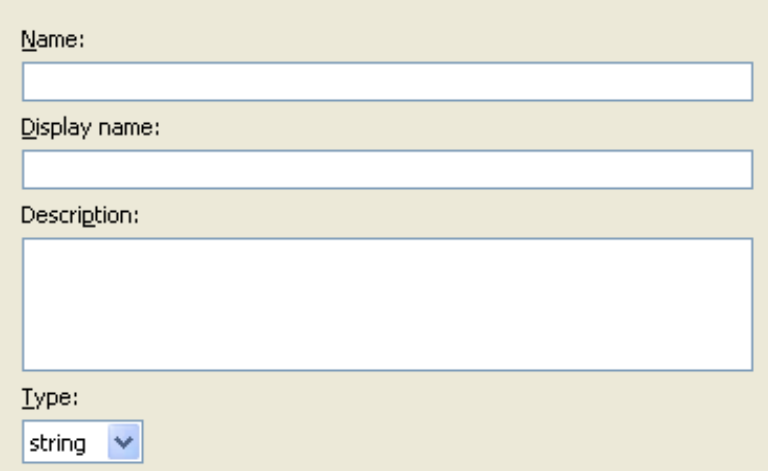


10a 認証キーの名前を指定します。

10b 認証キーの表示名を指定します。

10c 参照情報として、認証キーの説明を入力します。

認証キーは文字列で保存されます。



10d ［OK］をクリックします。

10e 入力が必要な新規認証キーごとに、ステップ 10 を繰り返します。

アプリケーションの認証キーを見つけるには、そのアプリケーションのユーザに対し、SecureLogin 資格情報を手動作成し、そのユーザでログインします。ユーザがログインすると、SecureLogin の管理ウィンドウ内の［My Logins (マイログイン)］に、認証キー情報が表示されます。

11 認証キーがすべてのユーザ資格情報で共有するスタティックな値である場合、その値を指定します。

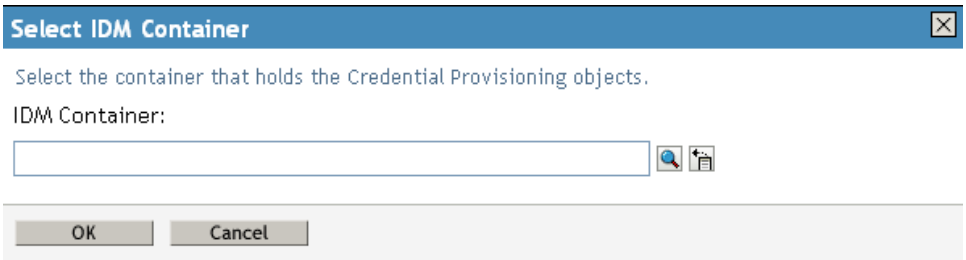
12 ［保存］アイコンをクリックして、アプリケーションを保存します。

アプリケーションオブジェクトが作成されたら、353 ページの 「Novell SecureLogin の資格情報プロビジョニングポリシーの環境設定」に進んでください。

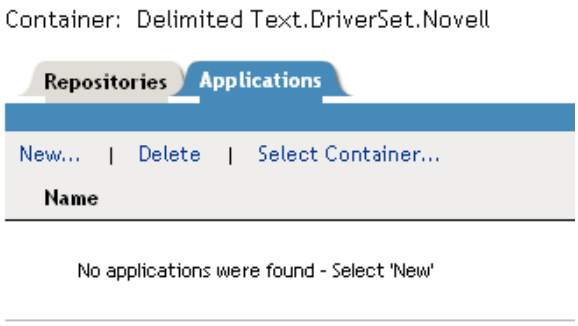
iManager での Novell SecureLogin のアプリケーションオブジェクトの作成

1 iManager で、［資格情報のプロビジョニング］>［環境設定］の順に選択します。

2 アプリケーションオブジェクトを保存するドライバオブジェクトを参照して選択し、［OK］をクリックします。

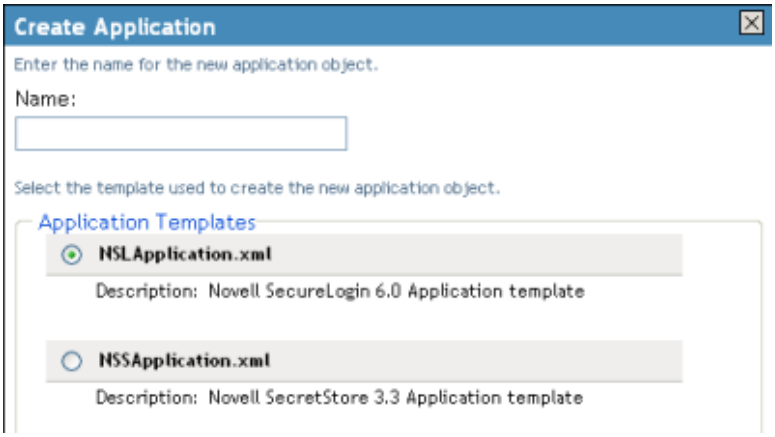


3 ［アプリケーション］タブを選択し、［新規作成］をクリックします。



4 アプリケーションオブジェクトの名前を指定します。

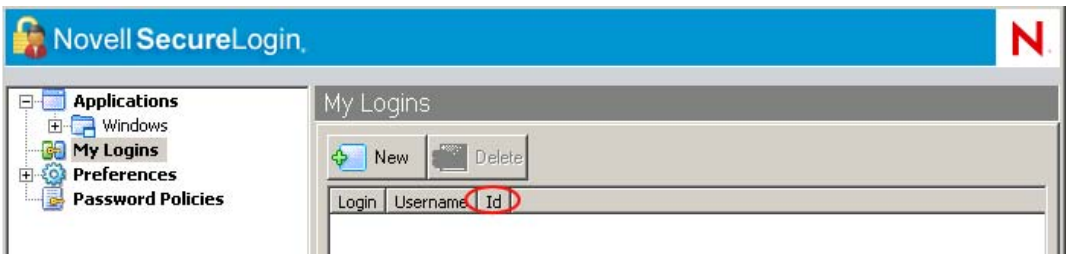
5 SecureLogin テンプレートを使用してアプリケーションを作成するため、［NSLApplication.xml］を選択します。



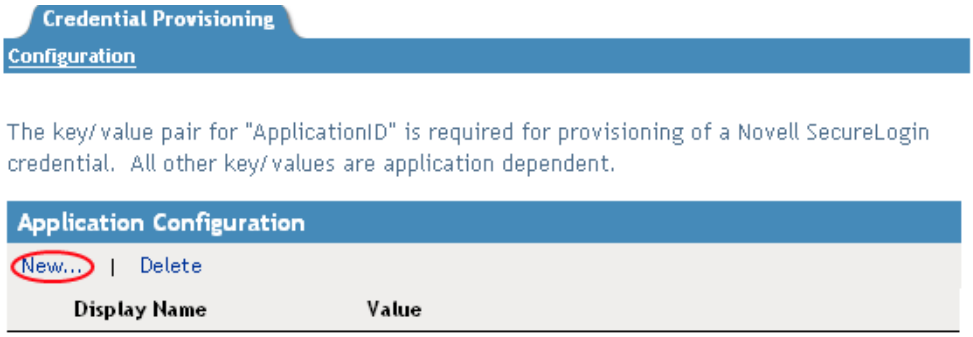
6 ［OK］をクリックします。

7 ［SecureLogin アプリケーション ID］を指定します。ワークシート項目の 9) を参照してください。

SecureLogin のアプリケーション ID を見つけるには、［My Logins (マイログイン)］をクリックします。アプリケーション ID は、［Id (ID)］フィールドに保存されています。



8 認証キーパラメータを作成するため、［新規作成］をクリックします。ワークシート項目の 10) を参照してください。

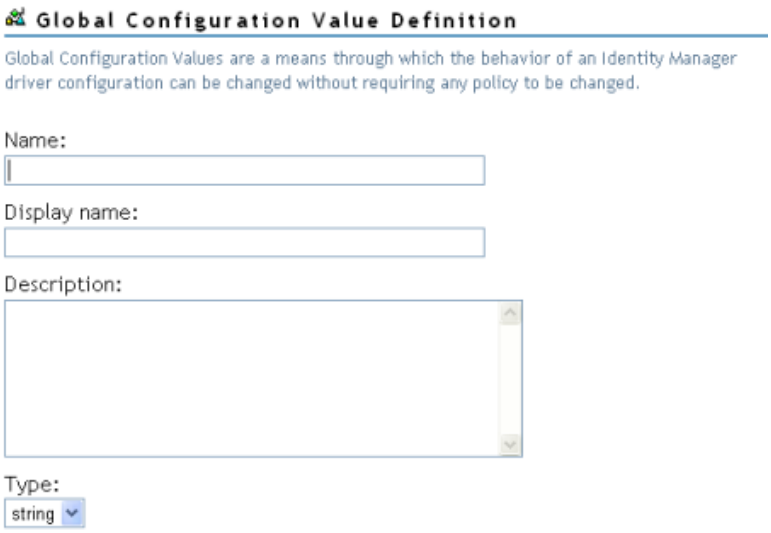


8a 認証キーの名前を指定します。

8b 認証キーの表示名を指定します。

8c 参照情報として、認証キーの説明を入力します。

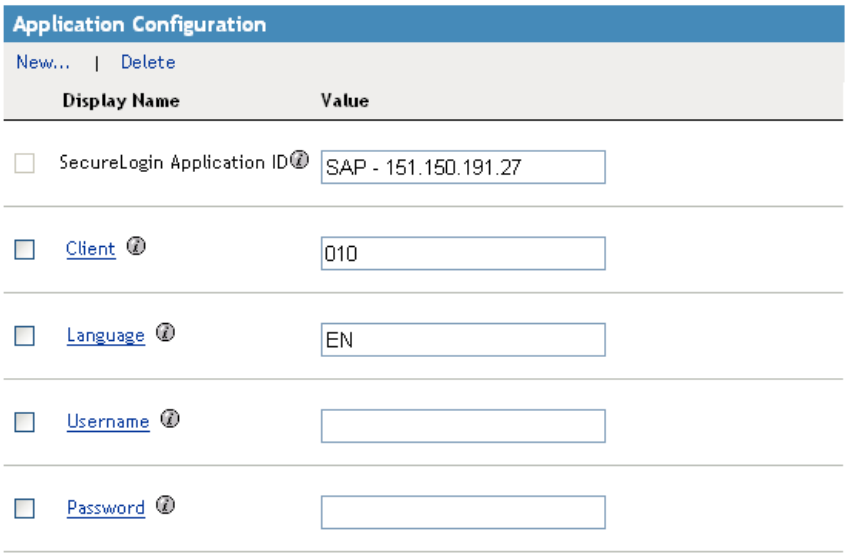
認証キーは文字列で保存されます。



アプリケーションの認証キーを見つけるには、そのアプリケーションのユーザに対し、SecureLogin 資格情報を手動作成し、そのユーザでログインします。ユーザがログインすると、SecureLogin の管理ウィンドウ内の［My Logins (マイログイン)］に、認証キー情報が表示されます。

8d ［OK］をクリックします。

8e 認証キー値を指定し、その値がスタティックである場合は、続いて［OK］をクリックします。



アプリケーションオブジェクトが作成されたら、353 ページの 「Novell SecureLogin の資格情報プロビジョニングポリシーの環境設定」に進んでください。

4.2.6 Novell SecureLogin の資格情報プロビジョニングポリシーの環境設定

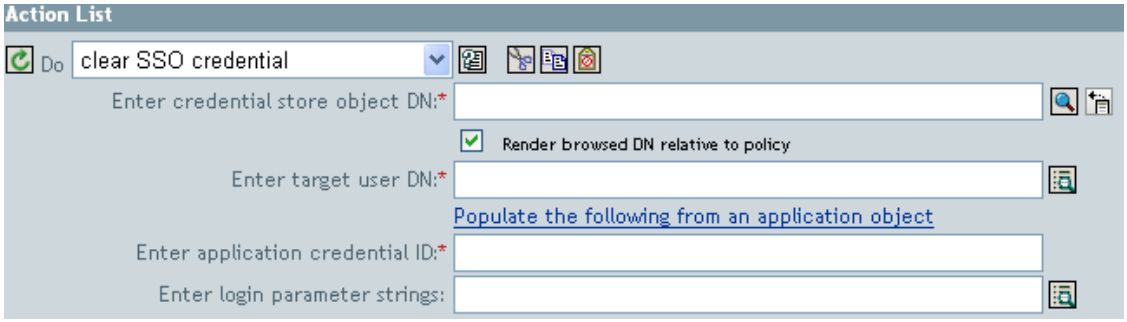
リポジトリとアプリケーションのオブジェクトを作成したら、ポリシーを作成して SecureLogin 情報をプロビジョニングする必要があります。ポリシーは、リポジトリとアプリケーションのオブジェクトに格納された情報を使用します。ポリシービルダの 3 つのアクションにより、SecureLogin 資格情報をプロビジョニングできるようにします。

- 354 ページの 「SSO 資格情報のクリア」
- 354 ページの 「SSO 資格情報の設定」
- 355 ページの 「SSO パスフレーズの設定」

SSO 資格情報のクリア

［SSO 資格情報のクリア］アクションにより、SSL 資格情報をクリアすることでオブジェクトのプロビジョニングを解除できます。

図 *4-2* *SSO* 資格情報のクリア

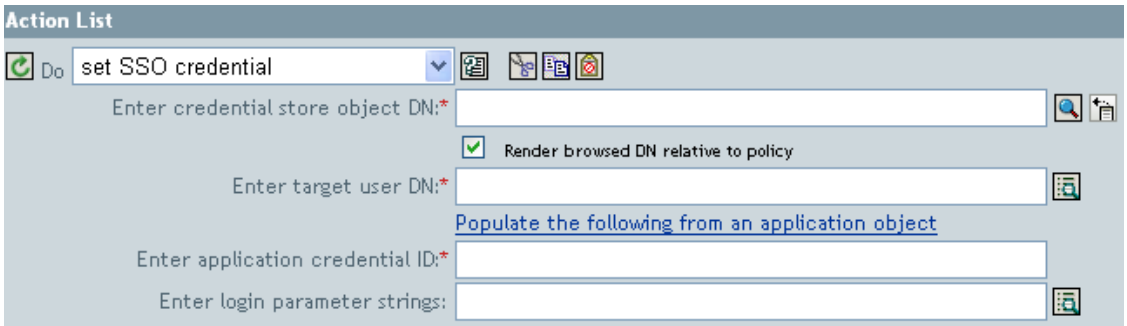


- 資格情報ストアオブジェクトの **DN** を入力 **:** リポジトリオブジェクトを参照して選択します。
- ターゲットユーザの **DN** を入力 **:** 引数ビルダを使用してターゲットユーザの DN を作成します。ワークシート項目の 15) を参照してください。
- アプリケーションのアクティベーションキー **ID** を入力 **:** アプリケーション ID を指定します。ワークシート項目の 9) を参照してください。
- ログインパラメータの文字列を入力 **:** 文字列ビルダを起動して、アプリケーションの認証キーを入力します。ワークシート項目の 10) を参照してください。

SSO 資格情報の設定

［SSO 資格情報の設定］アクションは、ユーザオブジェクトの作成またはパスワードの変更が実施されるときに、SSO 資格情報を設定できるようにします。

図 *4-3* *SSO* 資格情報の設定



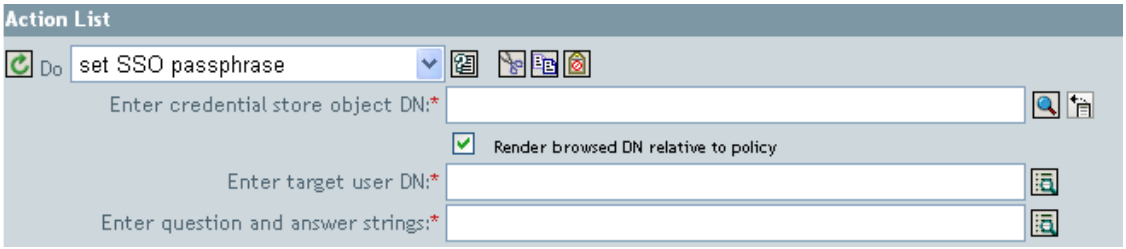
- 資格情報ストアオブジェクトの **DN** を入力 **:** リポジトリオブジェクトを参照して選択します。
- ターゲットユーザの **DN** を入力 **:** 引数ビルダを使用してターゲットユーザの DN を作成します。ワークシート項目の 15) を参照してください。
- アプリケーションのアクティベーションキー **ID** を入力 **:** アプリケーション ID を指定します。ワークシート項目の 9) を参照してください。

- ログインパラメータの文字列を入力 **:** 文字列ビルダを起動して、アプリケーションの認証キーを入力します。ワークシート項目の 10) を参照してください。

SSO パスフレーズの設定

［SSO パスフレーズの設定］アクションにより、SecureLogin パスフレーズを作成して、ユーザオブジェクトがプロビジョニングされる場合に回答できるようにします。

図 *4-4* *SSO* パスフレーズの設定



- 資格情報ストアオブジェクトの **DN** を入力 **:** リポジトリオブジェクトを参照して選択します。
- ターゲットユーザの **DN** を入力 **:** 引数ビルダを使用してターゲットユーザの DN を作成します。ワークシート項目の 15) を参照してください。
- 質問と回答の文字列を入力 **:** 文字列ビルダを起動して、パスフレーズの質問と回答を入力します。ワークシート項目の 16) を参照してください。

資格情報プロビジョニングポリシーの例

プロビジョニングポリシーは、各自の環境の要件を満たすように実装およびカスタマイズできます。次の例では、335 ページの 図 4-1 で示したシナリオのポリシーの設定方法について説明します。

Finance 部のシナリオでは、SecureLogin のプロビジョニングは、SAP 内にパスワードが設定された後に実行されます。必要なパラメータの多くはスタティックに設定されており、リポジトリやアプリケーションのオブジェクトを介してすべてのポリシーで使用可能です。ただし、スタティックでないデータパラメータ (sapUsername、password、DirXML-ADContext および workforceID) もあります。これらのパラメータは、SAP 管理ドライバの <add> または <modify-password> コマンドが実行され、<output> ステータスドキュメントが SAP ユーザ管理ドライバシムから返された後にのみ使用可能です。<ouput> ドキュメントには、購読者チャネルの操作属性が含まれていないため、コマンドのユーザコンテキストは失われ、その結果、オブジェクトへのクエリが阻まれます。そのため、次のことを実行する必要があります。

- SAP ユーザドライバの購読者作成ポリシーによって、スタティックでないデータパラメータの存在を強制するようにします。
- 購読者コマンドを SAP ユーザドライバシムへ発行する前に、プロビジョニング操作に必要なスタティックでないパラメータをキャッシュします。
- コマンドが正常に実行された後は、SecureLogin のプロビジョニングで使用するため、キャッシュされたデータを取得します。

注 **:** Identity Manager 3.0 Support Pack 1 のメディアには、XML 形式で使用可能なサンプルポリシーがあります。ファイル名は、SampleInputTransform.xml、

SampleSubCommandTransform.xml および SampleSubEventTransform.xml です。これらのファイルは、次のディレクトリにあります (プラットフォーム別に示します)。

- linux\setup\utilities\cred_prov
- nt\dirxml\utilities\cred_prov
- nw\dirxml\utilities\cred_prov

これらのファイルは、ユーティリティのインストール時に資格情報プロビジョニングのサンプルポリシーを選択すると、Identity Manager サーバにインストールされます。サンプルポリシーは、次の場所にインストールされます (プラットフォーム別に示します)。

- Windows:C:\Novell\NDS\DirXMLUtilities (デフォルト。インストール時に変更可)
- NetWare®:SYS:\System\DirXmlUtilities
- Linux (eDir 8.7):/usr/lib/dirxml/rules/credprov

サンプルポリシーは、各自の環境で機能するポリシーを開発するための開始ポイントとして使用できます。

操作データキャッシング

必須の操作データキャッシングに使用できるメカニズムは、<operation-data> 要素です。SecureLogin アカウントは <add> または <modify-password> コマンドのいずれかからプロビジョニングする必要があるため、スタティックでないデータキャッシングポリシーを実装する論理的な場所は、購読者コマンド変換ポリシー内になります。次の例に、一般的な SecureLogin のプロビジョニングにおける <operation-data> 要素を示します。

<operation-data> <nsl-sync-data> <nsl-target-user-dn> cn=GLCANYON,ou=finance,dc=prod,dc=testco,dc=com </nsl-target-user-dn> <nsl-app-username>GCANYON</nsl-app-username> <password><!-- content suppressed --></password> <nsl-passphrase-answer>50024222</nsl-passphrase-answer> </nsl-sync-data> </operation-data>

335 ページの 図 4-1 で示したサンプルの Finance 部のシナリオでは、操作データのペイロードを入力するのに次の値が必要です。

- <nsl-target-user-dn> 要素は、アイデンティティボールトの DirXML-ADContext 属性を使用して入力されます。この属性は、Active Directory ドライバによって設定されたものです。AD ドライバによって値が設定されたときに SAP ユーザドライバに通知されるようにするには、DirXML-ADContext を購読者フィルタに通知属性として追加してください。
- <nsl-app-username> 要素は、sapUsername 属性によって入力されます。<add> コマンドは、SAP ユーザドライバの作成ポリシーによって生成されるため、操作属性として使用できます。SAP ユーザドライバを使用すると、SAP ユーザ名の値が関連付けの値の一部になります。このことは、パスワード変更イベントの場合、関連付けから名前が解析されるということ意味します。
- パスワード要素には、<add> または <modify-password> コマンド内の <password> 要素の値が入力されます。
- <nsl-passphrase-answer> 要素には、アイデンティティボールトから workforceID 属性の値が入力されます。この属性は、SAP HR ドライバによって設定されたものです。この値は、アイデンティティボールトへの初めてのプロビジョニングで設定されますが、workforceID を通知属性として購読者フィルタに追加することを推奨します。

SecureLogin のプロビジョニング

シナリオでは、SecureLogin 資格情報プロビジョニング用に操作データが取得され、使用される最初の場所は、ドライバの入力変換ポリシー内に指定されています。サンプルのシナリオでは、次の 3 つのポリシーが実装されています。

- パスワードが正常に同期された後の SecureLogin 資格情報の設定
- SecureLogin パスフレーズと回答の設定
- アプリケーションユーザが削除された場合の SecureLogin 資格情報の削除 (アイデンティティボールトのオブジェクトは削除されない)

注 **:** SampleInputTransform.xml ファイルには、SecureLogin 資格情報をパスワード同期が成功した後に設定するためのサンプルポリシーがあります。このファイルは、Identity Manager 3.0 Support Pack 1 メディアの「Credential Provisioning」フォルダにあります。

SecureLogin 資格情報の設定ポリシーでは、プロビジョニングが実行されるのは、返されたコマンドステータスが「成功」で、以前に設定した <operation-data> が存在する場合のみに限定してください。

SecureLogin のプロビジョニング解除

「接続システムのユーザアカウントは削除し、アイデンティティボールトのアカウントは残す」というポリシーを使用する状況は数多く考えられます。Finance のシナリオでは、ユーザのアイデンティティボールトの employeeStatus 属性値が「I」に設定された場合に、SAP ユーザのアカウントを削除して、SecureLogin 資格情報のプロビジョニングを解除します。この状況を処理するため、SAP ユーザドライバの購読者イベント変換に、変更属性値をオブジェクトの削除に変換するポリシーが含まれています。Active Directory アカウントの名前は、削除コマンドが実行された後も必要なため、<operation-data> イベントを <delete> コマンドに設定して、入力変換ポリシー内の SecureLogin のプロビジョニング解除ポリシーで使用できるようにする必要があります。

<operation-data> <nsl-sync-data> <nsl-target-user-dn> cn=GLCANYON,ou=finance,dc=prod,dc=testco,dc=com </nsl-targer-user-dn> </nsl-sync-data> </operation-data>

<modify> イベントを <delete> に変換してこの要素を作成するポリシーは、SampleSubEventTransform.xml ファイル内のサンプル資格情報プロビジョニングポリシーにあります。

4.3 Novell SecretStore による資格情報プロビジョニングポリシー

資格情報プロビジョニングポリシーにより、アプリケーション資格情報を Novell SecretStore 内のユーザオブジェクトにプロビジョニングできます。アプリケーションサーバおよびユーザ資格情報を、通常の Identity Manager プロビジョニングシナリオの一部としてプロビジョニングできるため、より安全で同期された Web シングルサインオン機能をユーザに提供できます。

このドキュメントでは、Identity Manager 内のオブジェクトとポリシーを設定するために必要な手順について記載しています。SecretStore コンポーネントの展開および設定についての情報は含まれていません。SecretStore のマニュアルは、「Novell SecretStore 3.3.3 のマ

ニュアル (http://www.novell.com/documentation/secretstore33/index.html)」を参照してください。

SecretStore で資格情報のプロビジョニングを実装するには、リポジトリオブジェクト、アプリケーションオブジェクトおよびポリシーの作成が必要です。リポジトリとアプリケーションのオブジェクトには、Identity Manager が使用できるように SecretStore の情報が格納されます。ポリシーは、ドライバが資格情報プロビジョニングを使用できるようにするために使用されます。次のオプションも設定できます。

- 資格情報プロビジョニングは、発行者チャネル、購読者チャネル、または両方のチャネルで設定できます。
- SecretStore の同期は、アプリケーションのパスワード同期の一部として発生させたり、他のイベントによってトリガすることができます。
- Web サービスの資格情報は、アプリケーションのアカウントをプロビジョニングしなくてもプロビジョニングできます。

図 4-5 は、一般的なシナリオを簡略に示したものです。このシナリオでは、シングルサインオンの資格情報を GroupWise® の新規ユーザにプロビジョニングしています。この部署では、SAP HR システムと Identity Manager を使用して、アイデンティティボールト内に新しいユーザをプロビジョニングします。組織の情報に基づき、ユーザは、eDirectory 内に実装された部署の認証ツリー内にプロビジョニングされます。ここが新しいユーザがネットワークに対して認証される場所であり、会社のファイアウォールの外から安全なシングルサインオン機能を提供するために、Novell iChain® または Access Manager® によって使用される GroupWise セキュリティ資格情報のリポジトリになります。ユーザは続いて、

Identity Manager によって GroupWise にプロビジョニングされ、それらのシステムの資格情報は、認証ツリー内の SecretStore 属性に同期されます。

図 *4-5* *SecretStore* による資格情報プロビジョニング

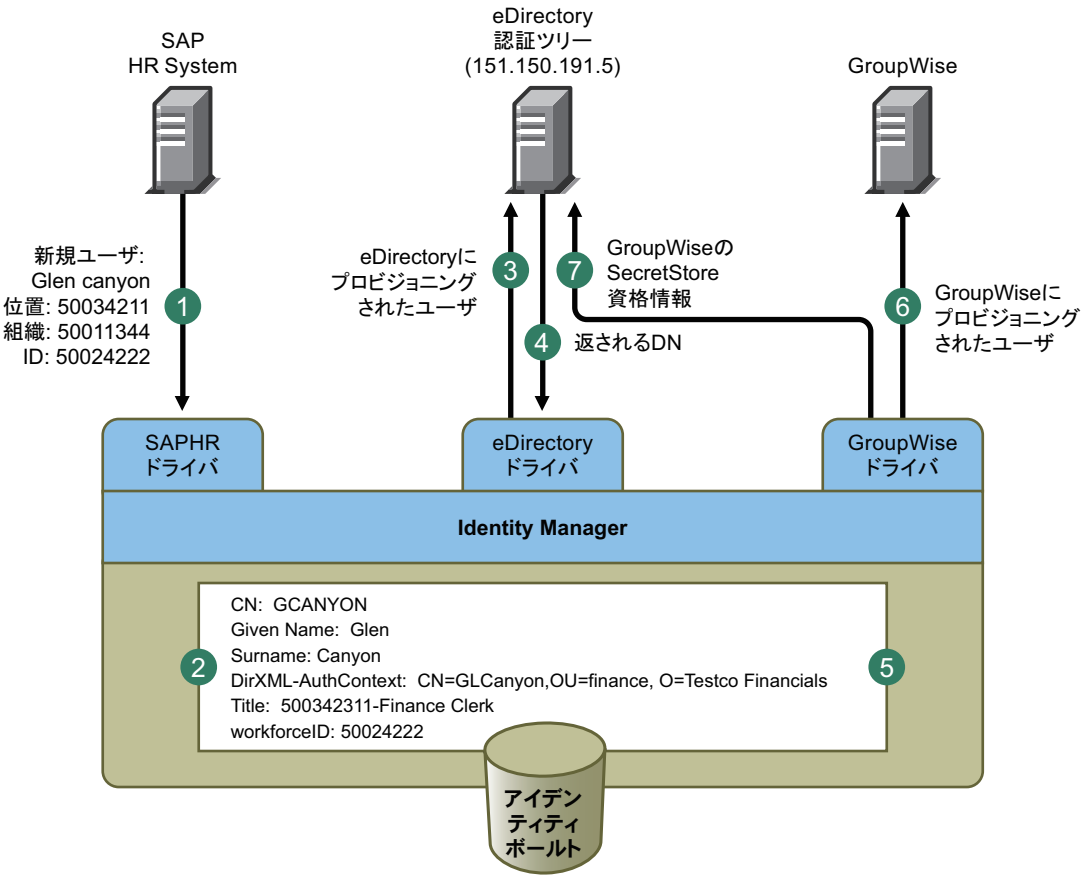


図 4-5 は、次のプロビジョニング手順を示しています。

1. SAP HR システムが、新入社員 Glen Canyon のデータを発行します。Identity Manager の SAP HP ドライバが、このデータを処理します。
2. 新しいユーザオブジェクトが、CN 値「GCANYON」および workforceID 値「50024222」を使用して、アイデンティティボールト内に作成されます。このユーザは、会社の Finance 組織に割り当てられているため、Finance 部の eDirectory サーバに認証される必要があります。Identity Manager の eDirectory ドライバがドメインを同期すると、アイデンティティボールトの情報を使用できるようになります。
3. Glen は、Finance 部の eDirectory サーバにプロビジョニングされます。
4. ドライバは、Glen の LDAP 形式の完全識別名を取得するように設定されます :CN=GLCanyon,OU=finance,O=Testco Financials
5. この LDAP 名は、アイデンティティボールト内の GCANYON ユーザの DirXML-AuthContext (ユーザオブジェクトの拡張属性、DirXML-ADContext のコピー) 属性に配置されます。

   これで、アイデンティティボールト内で必要な属性が使用できるようになったので、GroupWise ドライバによって、GCANYON オブジェクトの属性が処理されます。

6. Glen は Finance 組織に所属するため、ドライバは GCANYON の GroupWise アカウントを Finance 部の GroupWise ドメインサーバ上にプロビジョニングします。
7. アカウントの作成が成功すると、GroupWise ドライバのポリシーによって、Glen の GroupWise 認証資格情報がこのユーザの eDirectory ユーザアカウントの SecretStore にプロビジョニングされます。

Glen がインターネットから会社の Web サイトに認証される場合、iChain サーバが SecretStore 資格情報を使用して、このユーザの安全な GroupWise 電子メールアカウントへの認証情報を入力するので、GroupWise 資格情報を自分で入力する必要も、会社のリソースにセキュリティを追加設定する必要もありません。

4.4 SecretStore による資格情報プロビジョニングポリシーの実装

SecretStore による資格情報プロビジョニングポリシーの実装は、柔軟にカスタマイズできます。実装手順は、SecretStore がインストールされているプラットフォーム、プロビジョニング対象のアプリケーション、使用する Identity Manager ドライバによって異なります。

SecretStore による資格情報プロビジョニングポリシーを実装するには ,、次の手順に従います。

- 360 ページのセクション 4.4.1「Novell SecretStore による資格情報プロビジョニングポリシーの要件」
- 361 ページのセクション 4.4.2「Novell SecretStore の展開環境設定パラメータの決定」
- 364 ページのセクション 4.4.3「Novell SecretStore のリポジトリオブジェクトの作成」
- 371 ページのセクション 4.4.4「Novell SecretStore のアプリケーションオブジェクトの作成」
- 378 ページのセクション 4.4.5「Novell SecretStore の資格情報プロビジョニングポリシーの環境設定」

4.4.1 Novell SecretStore による資格情報プロビジョニングポリシーの要件

SecretStore による資格情報プロビジョニングポリシーを使用するには、次の要件を満たす必要があります。

- Support Pack 1 が適用された Identity Manager 3.0
  - eDirectory 8.7x, がインストールされている必要があります。eDirectory 8.8 はサポートされていません。
  - +jsso.jar、idmcp.jar および jnet.jar が Identity Manager Java ライブラリの標準の場所にあることを確認します。
- SecretStore 3.3 以降

要件が満たされていることを確認したら、361 ページのセクション 4.4.2「Novell SecretStore の展開環境設定パラメータの決定」に進んでください。

4.4.2 Novell SecretStore の展開環境設定パラメータの決定

図 4-5 の展開シナリオで説明されている同期機能を提供するには、まず、Identity Manager および SecretStore 環境に関連したすべてのビジネスプロセス情報を収集します。361 ページの ï 4-3§ 「SecretStore の資格情報プロビジョニングポリシーワークシート」を印刷して、情報を記録するためのワークシートとして使用してください。

表 4-3 *SecretStore* の資格情報プロビジョニングポリシーワークシート

必要な環境設定情報	情報
1) Web シングルサインオンのプロビジョニング用に設定するアプリケーション。	
2) SecretStore リポジトリサーバの DNS 名または IP アドレス。	
3) SecretStore リポジトリサーバの SSL LDAP ポート。	
4) SecretStore リポジトリサーバ管理者の完全修飾された LDAP 識別名。	
5) SecretStore リポジトリサーバの管理者のパスワード。	
6) SecretStore サーバからエクスポートされる、SSL 証明書へのフルパスおよび証明書名。証明書は、Identity Manager サーバのローカルに配置する必要があります。	
7) 1 つの SecretStore リポジトリを複数のドライバで使用するか、または各ドライバで専用のリポジトリを使用するかを決定する。	
8) 使用される SecretStore のシークレットタイプを記録する。	サポートされるシークレットタイプには次に示す 2 つの種類があります。 ◆ A: アプリケーションのシークレット (SS_App:prefix) ◆ C: 資格情報セットのシークレット (SS_CredSet:prefix)
9) プロビジョニング対象アプリケーションごとのアプリケーション ID または資格情報セット。	
10) ユーザ名やパスワードなどのアプリケーションごとに必要な認証キーを用意する。これらはアプリケーションごとに異なる場合があります。	
11) 認証キーの値をスタティックな値に設定するかどうかを決定する。	
12) ユーザごとに異なる値である (または異なる値にできる) スタティックでない値の場合は、そのスタティックでない情報のソースを書き留める (イベント情報またはアイデンティティボールトの属性値)。	

必要な環境設定情報	情報
13) ターゲットアプリケーションへのパスワードも同期しているドライバに SecretStore のプロビジョニングを実装する場合、SecretStore のプロビジョニングを、ターゲットアプリケーションのサーバにパスワードが設定される前と後のどちらで開始するかを決定する。	
14) リポジトリおよびアプリケーションのオブジェクトが格納されるドライバオブジェクトの名前。(格納先ドライバは別々に指定可能)	
15) ターゲットアプリケーションのユーザオブジェクトの DN を決定する。	

プロビジョニング環境設定データの例

プロビジョニングシナリオを使用したサンプルデータを次に示します。ここでは、Finance eDirectory 認証ツリー内のユーザに、Finance 部の GroupWise ドメインサーバの SecretStore 資格情報をプロビジョニングします。

SecretStore のリポジトリ情報

表 *4-4* *SecretStore* の資格情報プロビジョニングポリシーワークシートの例

必要な環境設定情報	情報
1) Web シングルサインオンのプロビジョニング用に設定するアプリケーション。	GroupWise
2) SecretStore リポジトリサーバの DNS 名または IP アドレス。	151.150.191.5
3) SecretStore リポジトリサーバの SSL LDAP ポート。	636
4) SecretStore リポジトリサーバ管理者の完全修飾された LDAP 識別名。	cn=admin,ou=finance,o=Tesetco Financials
5) SecretStore リポジトリサーバの管理者のパスワード。	dixml
6) SecretStore サーバからエクスポートされる、SSL 証明書へのフルパスおよび証明書名。証明書は、Identity Manager サーバのローカルに配置する必要があります。	c:\novell\nds\FinanceAD.cer
7) 1 つの SecretStore リポジトリを複数のドライバで使用するか、または各ドライバで専用のリポジトリを使用するかを決定する。	この例では、リポジトリは 1 つだけにします。

必要な環境設定情報	情報
8) 使用される SecretStore のシークレットタイプを記録する。	サポートされるシークレットタイプには次に示す 2 つの種類があります。 ◆ A: アプリケーションのシークレット (SS_App:prefix) ◆ **C: 資格情報セットのシークレット (SS_CredSet:prefix)**
9) プロビジョニング対象アプリケーションごとのアプリケーション ID または資格情報セット。	GroupWise_Credentials
10) ユーザ名やパスワードなどのアプリケーションごとに必要な認証キーを用意する。これらはアプリケーションごとに異なる場合があります。	ユーザ名 パスワード
11) 認証キーの値をスタティックな値に設定するかどうかを決定する。	このシナリオではスタティックな情報は使用しません。
12) ユーザごとに異なる値である (または異なる値にできる) スタティックでない値の場合は、そのスタティックでない情報のソースを書き留める (イベント情報またはアイデンティティボールトの属性値)。	ユーザ名 : アイデンティティボールト属性の「CN」パスワード : イベントの <password>
13) ターゲットアプリケーションへのパスワードも同期しているドライバに SecretStore のプロビジョニングを実装する場合、SecretStore のプロビジョニングを、ターゲットアプリケーションのサーバにパスワードが設定される前と後のどちらで開始するかを決定する。	後
14) リポジトリおよびアプリケーションのオブジェクトが格納されるドライバオブジェクトの名前。(格納先ドライバは別々に指定可能)	GroupWise-Finance ドライバ
15) ターゲットアプリケーションのユーザオブジェクトの DN を決定する。	アイデンティティボールト属性の「DirXML-ADContext」

その他の環境設定情報 :

- Finance 部の eDirectory ツリーは、すべての Finance アプリケーションの SecretStore リポジトリとして動作します。
- Finance 部関連のプロビジョニングドライバは、すべて「Finance Drivers」という名前のドライバセット内に設定されます。
- アイデンティティボールトの属性 employeeStatus の値が「I」に設定された場合、GroupWise ユーザアカウントを削除して、その GroupWise ユーザアカウントの SecretStore 資格情報も eDirectory ユーザから削除する必要があります。

収集したデータから見ると、SecretStore リポジトリの情報は、Finance 部のアプリケーションをプロビジョニングするすべてのドライバに対してグローバルです。また、すべてのプロビジョニング情報は、GroupWise のログインパラメータであるユーザ名、パスワードおよびターゲットユーザの DN を除き、スタティックに設定されています。

すべてのパラメータを決定したら、364 ページのセクション 4.4.3「Novell SecretStore のリポジトリオブジェクトの作成」に進んでください。

4.4.3 Novell SecretStore のリポジトリオブジェクトの作成

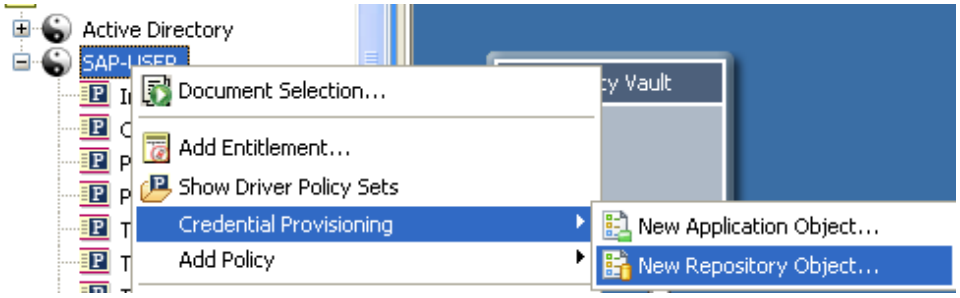
リポジトリオブジェクトには、SecretStore のスタティックな環境設定情報が保存されます。リポジトリの情報は、アプリケーション資格情報を使用するアプリケーションからは独立しています。この情報は、接続システム (SAP、PeopleSoft、Notes など) に関係なく、すべてのプロビジョニングイベントに適用されます。リポジトリオブジェクトは、Designer または iManager で作成できます。

- 364 ページの 「Designer での Novell SecretStore のリポジトリオブジェクトの作成」
- 367 ページの 「iManager での Novell SecretStore のリポジトリオブジェクトの作成」

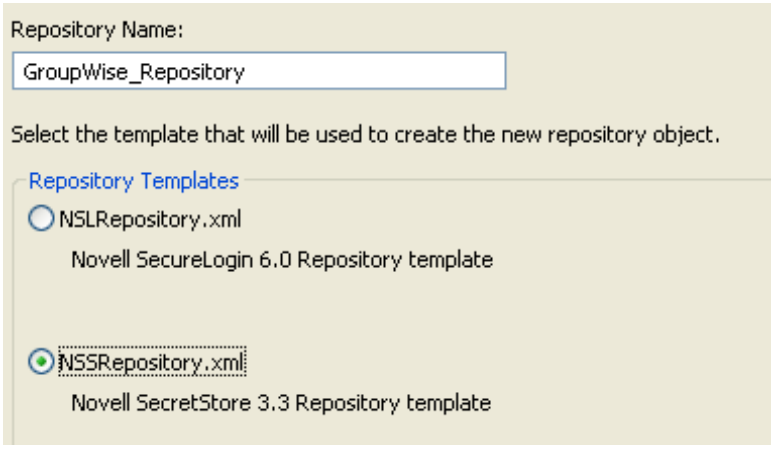
Designer での Novell SecretStore のリポジトリオブジェクトの作成

次に示すのは、Designer でリポジトリオブジェクトを作成する方法のうちの 1 つです。

1. アウトラインビューで、リポジトリオブジェクトを格納するドライバオブジェクトを右クリックします。
2. ［資格情報のプロビジョニング］>［New Repository Object (新規リポジトリオブジェクト)］の順にクリックします。

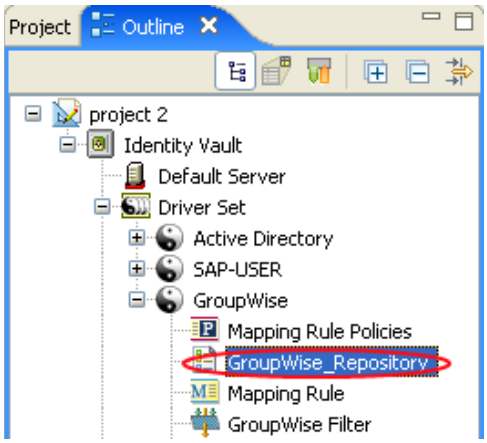


3. リポジトリオブジェクトの名前を指定します。
4. SecretStore テンプレートを使用するため、［NSSRepository.xml］を選択します。

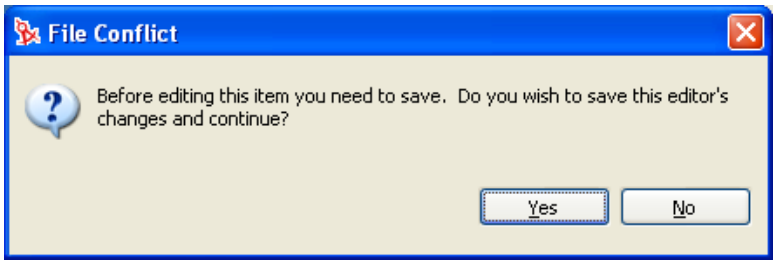


5. ［OK］をクリックします。

6 環境設定情報を追加するため、アウトラインビューでリポジトリオブジェクトをダブルクリックします。



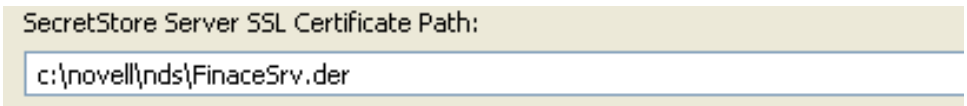
7 [はい] をクリックして、新しいリポジトリオブジェクトを保存します。



8 SecretStore サーバの DNS 名または IP アドレスを指定します。ワークシート項目の 2) を参照してください。

9 SecretStore サーバの SSL ポートを指定します。ワークシート項目の 3) を参照してください。

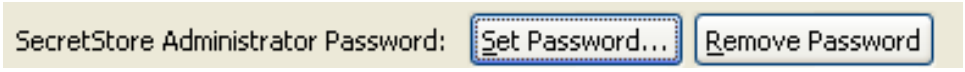
10 SecretStore サーバからエクスポートされる SSL 証明書へのフルパスを指定します。このパスには証明書名を含め、Identity Manager サーバのローカルに配置する必要があります。ワークシート項目の 6) を参照してください。



注 : SSL 証明書のエクスポート方法については、iManager のマニュアルを参照してください。

11 SecretStore 管理者の完全修飾された LDAP 識別名を指定します。ワークシート項目の 4) を参照してください。

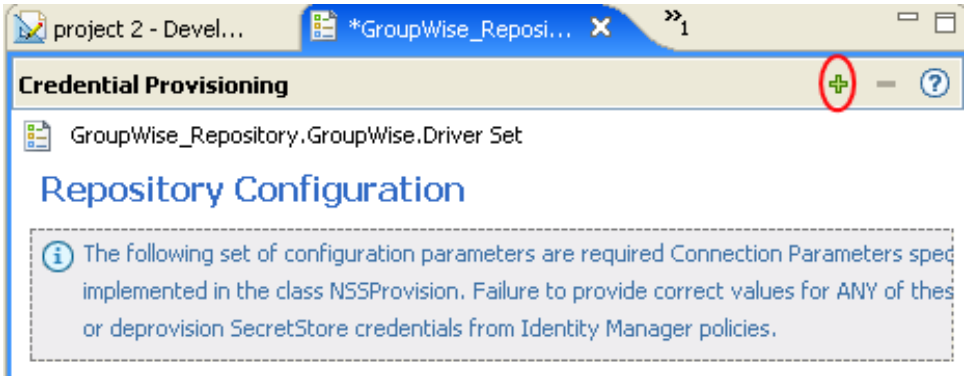
12 ［パスワードを設定する］をクリックします。



13 SecretStore 管理者のパスワードを 2 回入力し、［OK］をクリックします。ワークシート項目の 5) を参照してください。

14 情報を確認し、［保存］アイコンをクリックして情報を保存します。

15 (オプション) リポジトリオブジェクトに対する他の環境設定パラメータを作成する場合は、［Add new item (新しい項目の追加)］アイコンをクリックします。

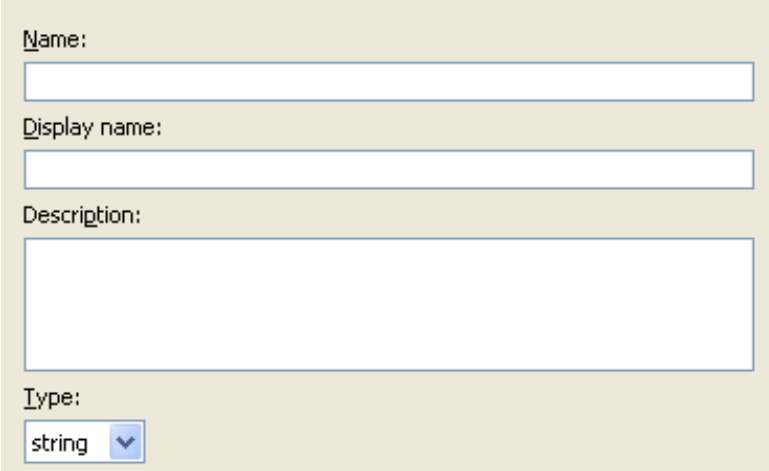


15a パラメータの名前を指定します。

15b パラメータの表示名を指定します。

15c 参照情報として、パラメータの説明を入力します。

パラメータは文字列で保存されます。



15d ［OK］をクリックします。

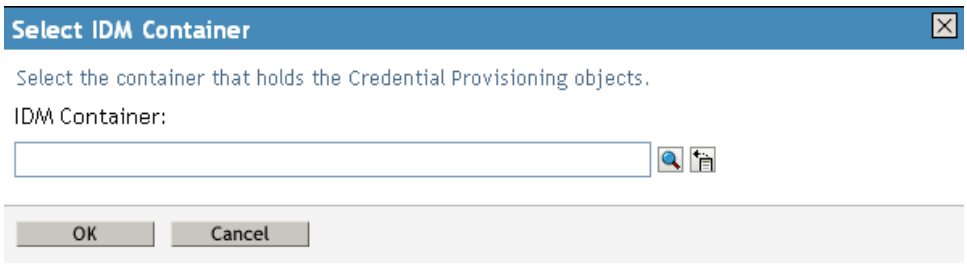
15e ［保存］アイコンをクリックして、リポジトリオブジェクトを保存します。

リポジトリオブジェクトが作成されたら、347 ページの 「Designer での Novell SecureLogin のアプリケーションオブジェクトの作成」に進んでください。

iManager での Novell SecretStore のリポジトリオブジェクトの作成

1 iManager で、［資格情報のプロビジョニング］>［環境設定］の順に選択します。

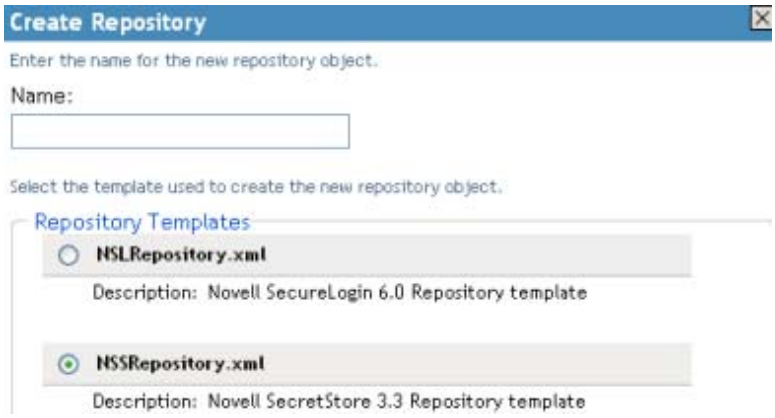
2 リポジトリオブジェクトを保存するドライバオブジェクトを参照して選択し、［OK］をクリックします。



3 ［新規作成］をクリックしてリポジトリを作成します。

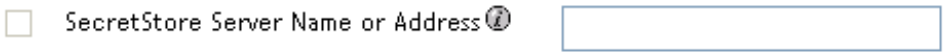
4 リポジトリオブジェクトの名前を指定します。

5 SecretStore テンプレートを使用してリポジトリを作成するため、［NSSRepository.xml］を選択します。

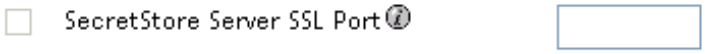


6 ［OK］をクリックします。

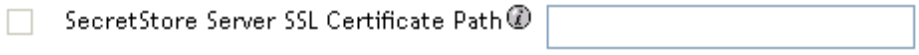
7 SecretStore サーバの DNS 名または IP アドレスを指定します。ワークシート項目の 2) を参照してください。



8 SecretStore サーバの SSL ポートを指定します。ワークシート項目の 3) を参照してください。



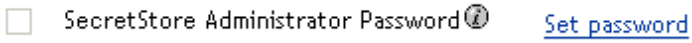
9 SecretStore サーバからエクスポートされる SSL 証明書へのフルパスを指定します。このパスには証明書名を含め、Identity Manager サーバのローカルに配置する必要があります。ワークシート項目の 6) を参照してください。



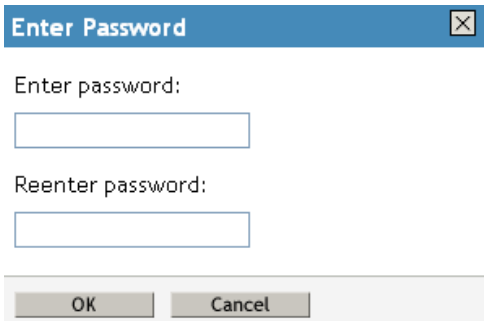
注 : SSL 証明書のエクスポート方法については、iManager のマニュアルを参照してください。

10 SecretStore 管理者の完全修飾された LDAP 識別名を指定します。ワークシート項目の 4) を参照してください。

11 ［パスワードを設定する］をクリックします。

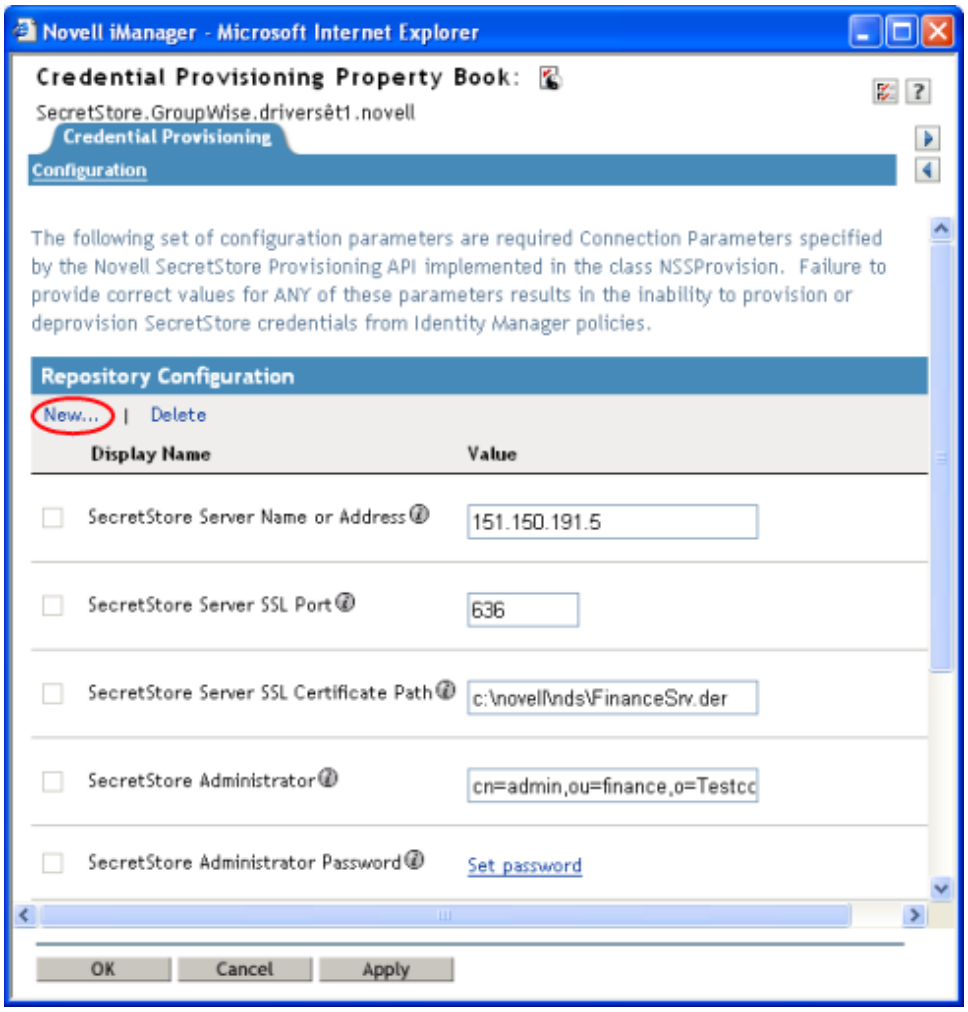


12 SecretStore 管理者のパスワードを 2 回入力し、［OK］をクリックします。ワークシート項目の 5) を参照してください。



13 指定した値を確認し、［OK］をクリックします。

14 (オプション) リポジトリオブジェクトに対する他の環境設定パラメータを作成する場合は、［新規作成］をクリックします。



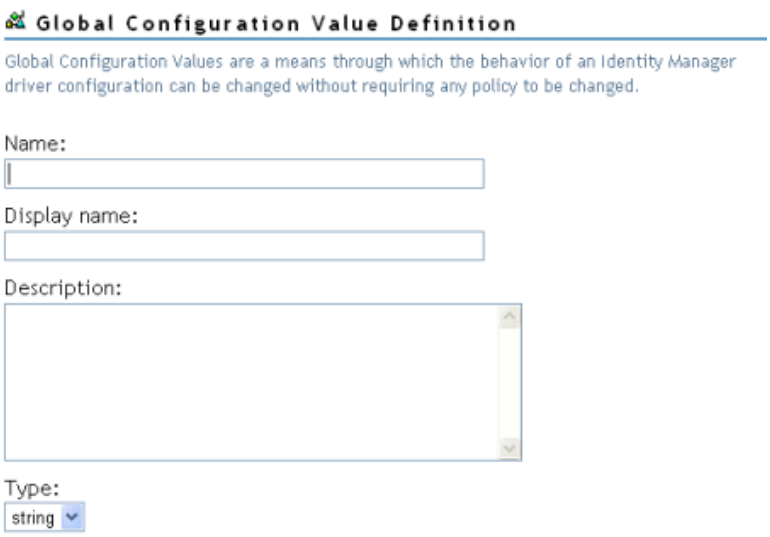
この例は、335 ページの 図 4-1 のシナリオに記載されています。

14a パラメータの名前を指定します。

14b パラメータの表示名を指定します。

14c 参照情報として、パラメータの説明を入力します。

パラメータは文字列で保存されます。



14d ［OK］をクリックします。

リポジトリオブジェクトが作成されたら、350 ページの 「iManager での Novell SecureLogin のアプリケーションオブジェクトの作成」に進んでください。

4.4.4 Novell SecretStore のアプリケーションオブジェクトの作成

アプリケーションには、SecretStore のスタティックな環境設定パラメータ値が保存されます。アプリケーション情報は、そのアプリケーションの資格情報を使用しているアプリケーションに特有のものです (GroupWise クライアントの情報、SAP データベースクライアントの情報など)。アプリケーションオブジェクトは、Designer または iManager で作成できます。

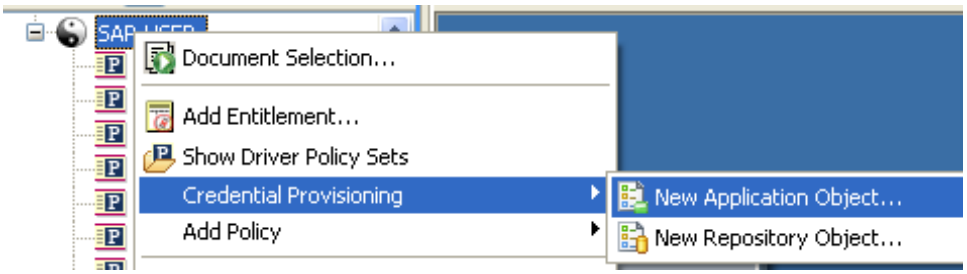
- 371 ページの 「Designer での Novell SecretStore のアプリケーションオブジェクトの作成」
- 375 ページの 「iManager での Novell SecretStore のアプリケーションオブジェクトの作成」

Designer での Novell SecretStore のアプリケーションオブジェクトの作成

次に示すのは、Designer でアプリケーションを作成する方法のうちの 1 つです。

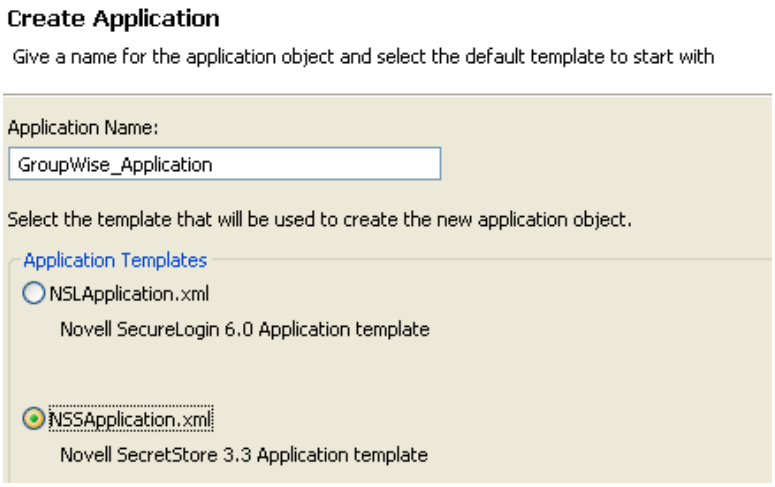
1 アウトラインビューで、アプリケーションオブジェクトを格納するドライバオブジェクトを右クリックします。

2 ［資格情報のプロビジョニング］>［New Application Object (新規アプリケーションオブジェクト)］の順にクリックします。



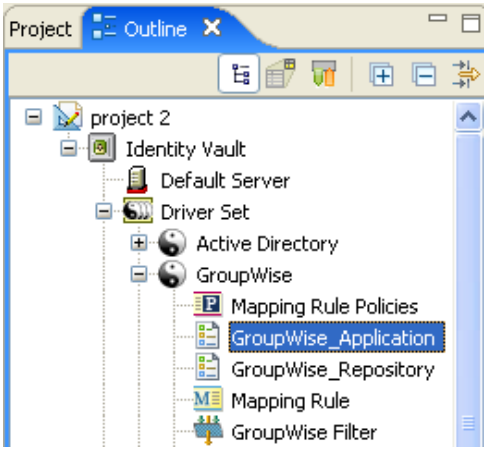
3 アプリケーションオブジェクトの名前を指定します。

4 SecretStore テンプレートを使用するため、［NSSApplication.xml］を選択します。

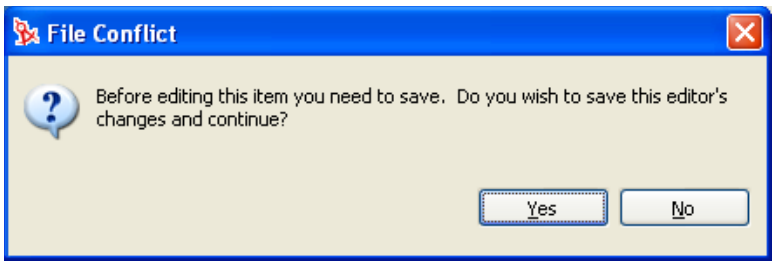


5 ［OK］をクリックします。

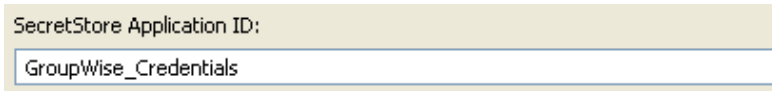
6 環境設定情報を追加するため、アウトラインビューでアプリケーションオブジェクトをダブルクリックします。



7 ［はい］をクリックして、新しいアプリケーションオブジェクトを保存します。



8 SecretStore のアプリケーション ID を指定します。ワークシート項目の 9) を参照してください。

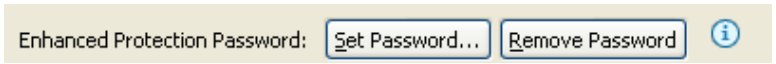


9 ［SecretStore のシークレットタイプ］を選択します。ワークシート項目の 8) を参照してください。

10 ［SecretStore の共有シークレットタイプ］を選択します。ワークシート項目の 8) を参照してください。

11 SecretStore の［拡張保護フラグの使用］で、［使用不可］または［使用可能］を選択します。

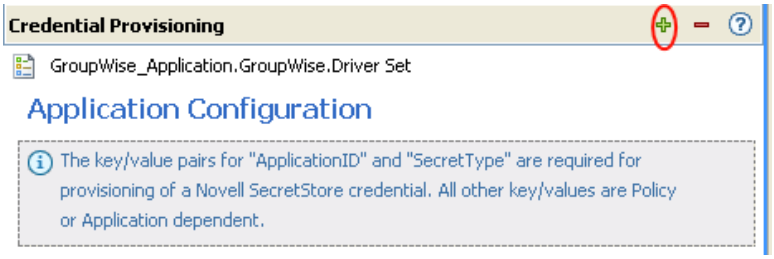
12 有効である場合、［パスワードを設定する］をクリックして、［拡張保護パスワード］を設定します。



13 パスワードを 2 回入力し、［OK］をクリックします。

14 ［保存］アイコンをクリックして、アプリケーションを保存します。

15 アプリケーションに必要な認証キーを追加するため、［Add new item (新しい項目の追加)］アイコン をクリックします。

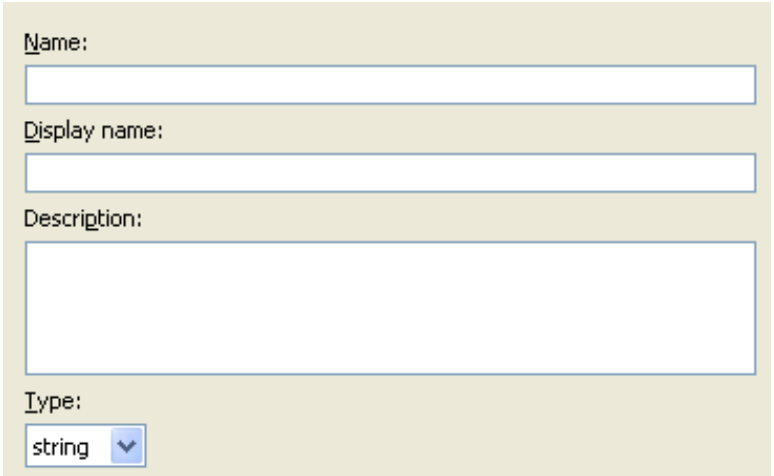


15a 認証キーの名前を指定します。

15b 認証キーの表示名を指定します。

15c 参照情報として、認証キーの説明を入力します。

認証キーは文字列で保存されます。



15d ［OK］をクリックします。

15e 入力が必要な新規認証キーごとに、ステップ 15 を繰り返します。

16 認証キーがすべてのユーザ資格情報で共有するスタティックな値である場合、その値を指定します。

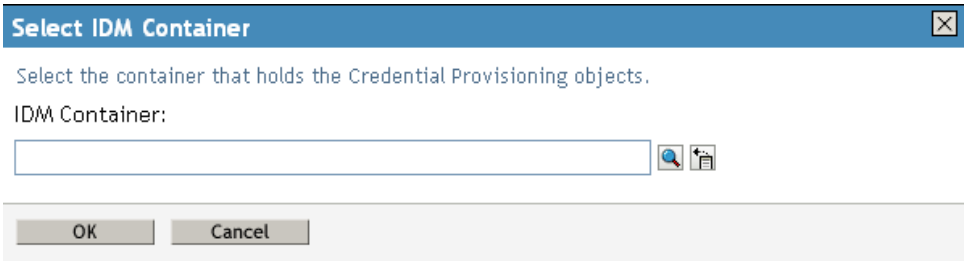
17 ［保存］アイコンをクリックして、アプリケーションを保存します。

アプリケーションオブジェクトが作成されたら、378 ページのセクション 4.4.5「Novell SecretStore の資格情報プロビジョニングポリシーの環境設定」に進んでください。

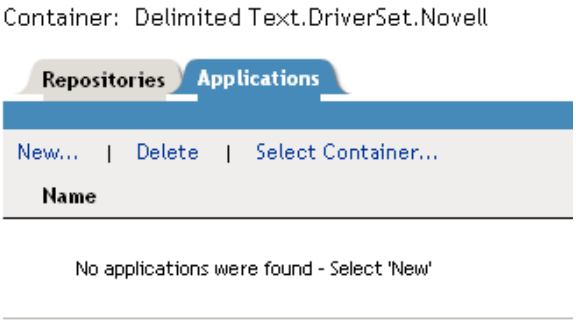
iManager での Novell SecretStore のアプリケーションオブジェクトの作成

1 iManager で、［資格情報のプロビジョニング］>［環境設定］の順に選択します。

2 アプリケーションオブジェクトを保存するドライバオブジェクトを参照して選択し、［OK］をクリックします。

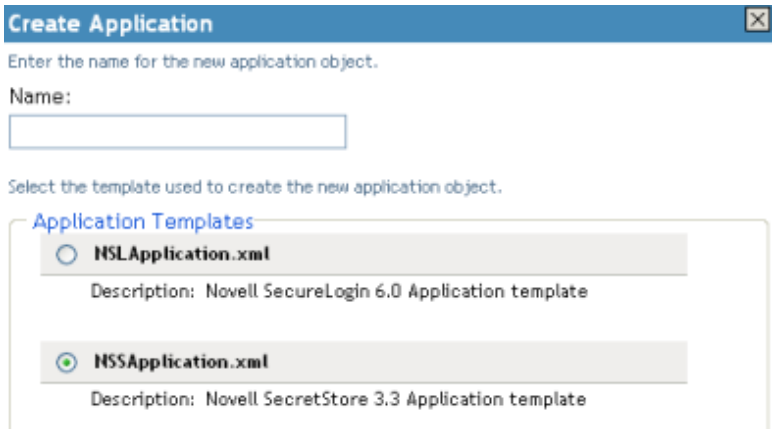


3 ［アプリケーション］タブを選択し、［新規作成］をクリックします。



4 アプリケーションオブジェクトの名前を指定します。

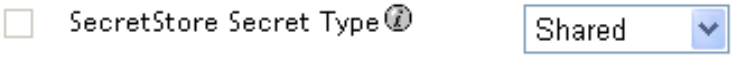
5 SecretStore テンプレートを使用してアプリケーションを作成するため、［NSSApplication.xml］を選択します。



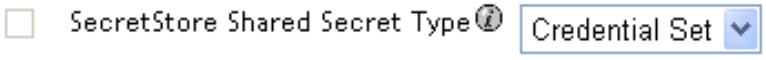
6 ［OK］をクリックします。

7 ［SecretStore アプリケーション ID］を指定します。ワークシート項目の 9) を参照してください。

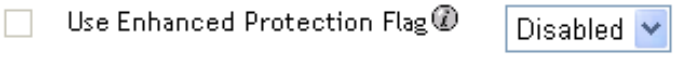
8 ［SecretStore のシークレットタイプ］を選択します。ワークシート項目の 7) を参照してください。SecretStore のタイプは［Shared (共有)］または［Not Shared (共有なし)］です。



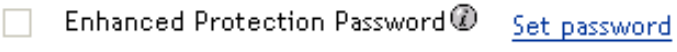
9 ［SecretStore の共有シークレットタイプ］を選択します。ワークシート項目の 8) を参照してください。共有される場合の SecretStore のタイプは、［Credential Set (資格情報セット)］または［アプリケーション］です。



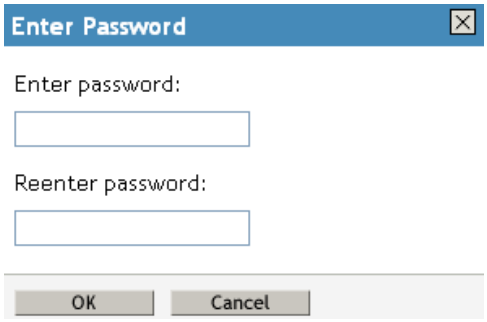
10 SecretStore の［拡張保護フラグの使用］で、［無効］または［有効］を選択します。



11 有効である場合、［パスワードを設定する］をクリックして、［拡張保護パスワード］を設定します。



12 パスワードを 2 回入力し、［OK］をクリックします。



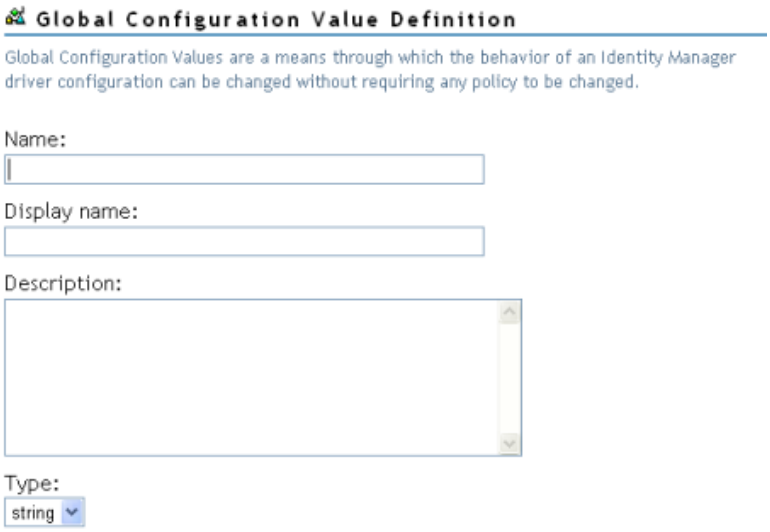
13 アプリケーションで必要な認証キーを作成するため、［新規作成］をクリックします。ワークシート項目の 10) を参照してください。

13a 認証キーの名前を指定します。

13b 認証キーの表示名を指定します。

13c 参照情報として、認証キーの説明を入力します。

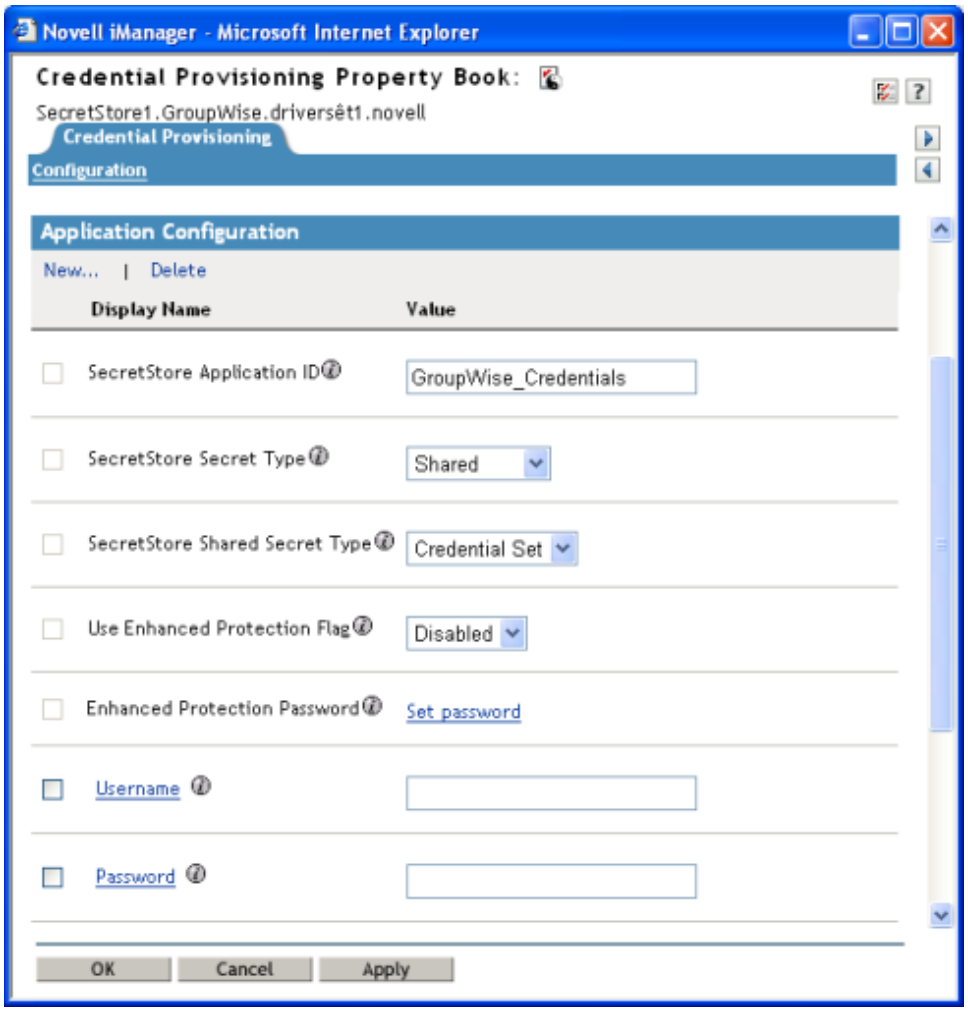
認証キーは文字列で保存されます。



13d ［OK］をクリックします。

13e アプリケーションが必要とする認証キーごとに、ステップ 13 を繰り返します。

14 認証キー値を指定し、その値がスタティックである場合は、続いて［OK］をクリックします。



アプリケーションオブジェクトが作成されたら、378 ページのセクション 4.4.5「Novell SecretStore の資格情報プロビジョニングポリシーの環境設定」に進んでください。

4.4.5 Novell SecretStore の資格情報プロビジョニングポリシーの環境設定

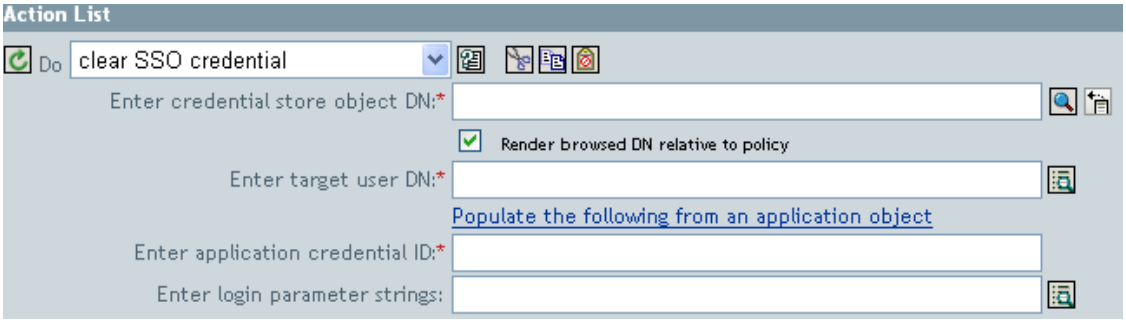
リポジトリとアプリケーションのオブジェクトを作成したら、ポリシーを作成して SecretStore 情報をプロビジョニングする必要があります。ポリシーは、リポジトリとアプリケーションのオブジェクトに格納された情報を使用します。ポリシービルダの 2 つのアクションにより、SecretStore 資格情報をプロビジョニングできるようにします。

- 354 ページの 「SSO 資格情報のクリア」
- 354 ページの 「SSO 資格情報の設定」

SSO 資格情報のクリア

［SSO 資格情報のクリア］アクションにより、SSL 資格情報をクリアすることでオブジェクトのプロビジョニングを解除できます。

図 *4-6* *SSO* 資格情報のクリア

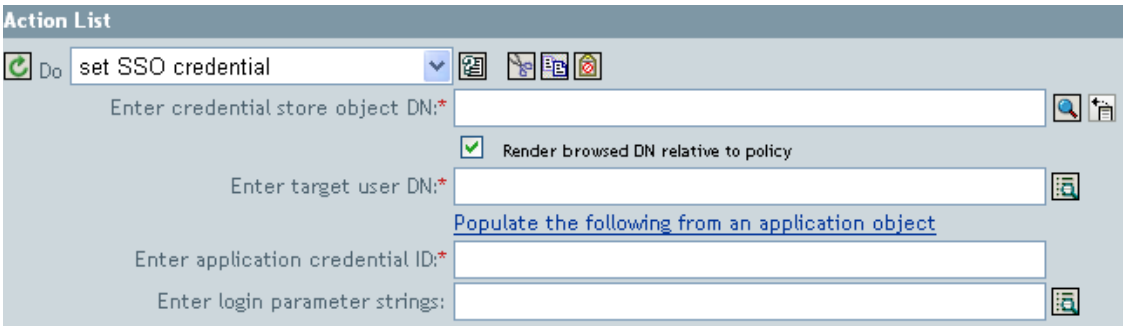


- 資格情報ストアオブジェクトの **DN** を入力 **:** リポジトリオブジェクトを参照して選択します。
- ターゲットユーザの **DN** を入力 **:** 引数ビルダを使用してターゲットユーザの DN を作成します。ワークシート項目の 15) を参照してください。
- アプリケーションのアクティベーションキー **ID** を入力 **:** アプリケーション ID を指定します。ワークシート項目の 9) を参照してください。
- ログインパラメータの文字列を入力 **:** 文字列ビルダを起動して、アプリケーションの認証キーを入力します。ワークシート項目の 10) を参照してください。

SSO 資格情報の設定

［SSO 資格情報の設定］アクションは、ユーザオブジェクトの作成またはパスワードの変更が実施されるときの、SSO 資格情報を設定できるようにします。

図 *4-7* *SSO* 資格情報の設定



- 資格情報ストアオブジェクトの **DN** を入力 **:** リポジトリオブジェクトを参照して選択します。
- ターゲットユーザの **DN** を入力 **:** 引数ビルダを使用してターゲットユーザの DN を作成します。ワークシート項目の 15) を参照してください。
- アプリケーションのアクティベーションキー **ID** を入力 **:** アプリケーション ID を指定します。ワークシート項目の 9) を参照してください。

- ログインパラメータの文字列を入力 **:** 文字列ビルダを起動して、アプリケーションの認証キーを入力します。ワークシート項目の 10) を参照してください。

資格情報プロビジョニングポリシーの例

資格情報プロビジョニングポリシーは、各自の環境の要件を満たすように実装およびカスタマイズできます。次の例では、359 ページの 図 4-5 で示したシナリオのポリシーの設定方法について説明します。

Finance 部のシナリオでは、SecretStore のプロビジョニングは、GroupWise 内にパスワードが設定された後に実行されます。必要なパラメータの多くはスタティックに設定されており、リポジトリやアプリケーションのオブジェクトを介してすべてのポリシーで使用可能です。ただし、スタティックでないデータパラメータ (CN、password、および DirXML-ADContext) もあります。これらのパラメータは、GroupWise ユーザの <add> または <modify-password> コマンドが実行され、<output> ドキュメントが GroupWise ドライバシムから返された後にのみ使用可能です。<ouput> ドキュメントには、購読者の操作属性が含まれていないため、コマンドのユーザコンテキストは失われ、その結果、オブジェクトへのクエリが阻まれます。そのため、次のことを実行する必要があります。

- GroupWise ドライバの購読者作成ポリシーによって、スタティックでないデータパラメータの存在を強制するようにします。
- 購読者コマンドを GroupWise ドライバシムへ発行する前に、プロビジョニング操作に必要なスタティックでないパラメータをキャッシュします。
- コマンドが正常に実行された後は、SecretStore のプロビジョニングで使用するため、キャッシュされたデータを取得します。

注 **:** Identity Manager 3.0 Support Pack 1 のメディアには、XML 形式で使用可能なサンプルポリシーがあります。ファイル名は、SampleInputTransform.xml、SampleSubCommandTransform.xml および SampleSubEventTransform.xml です。これらのファイルは、次のディレクトリにあります。

- linux\setup\utilities\cred_prov
- nt\dirxml\utilities\cred_prov
- nw\dirxml\utilities\cred_prov

これらのファイルは、ユーティリティのインストール時に資格情報プロビジョニングのサンプルポリシーを選択すると、Identity Manager サーバにインストールされます。サンプルポリシーは、次の場所にインストールされます (プラットフォーム別に示します)。

- Windows:C:\Novell\NDS\DirXMLUtilities (デフォルト。インストール時に変更可)
- NetWare: SYS:\System\DirXmlUtilities
- Linux (eDir 8.7):/usr/lib/dirxml/rules/credprov

サンプルポリシーは、各自の環境で機能するポリシーを開発するための開始ポイントとして使用できます。

操作データキャッシング

必須の操作データキャッシングに使用できるメカニズムは、<operation-data> 要素です。SecretStore アカウントは <add> または <modify-password> コマンドのいずれかからプロビジョニングする必要があるため、スタティックでないデータキャッシングポリシーを実装

する論理的な場所は、購読者コマンド変換ポリシー内になります。次の例に、一般的な SecretStore のプロビジョニングにおける要素を示します。

<operation-data> <nss-sync-data> <nss-target-user-dn>cn=GLCANYON,ou=finance,o=Testco Financials </nss-target-user-dn> <nss-app-username>GCANYON</nsl-app-username> <password><!-- content suppressed --></password> <nss-passphrase-answer>50024222</nsl-passphrase-answer> </nss-sync-data> </operation-data>

359 ページの 図 4-5 で示したサンプルの Finance 部のシナリオでは、操作データのペイロードを入力するのに次の値が必要です。

- <nss-target-user-dn> 要素は、アイデンティティボールトの DirXML-ADContext 属性を使用して入力されます。この属性は、eDirectory ドライバによって設定されたものです。eDirectory ドライバによって値が設定されたときに GroupWise ドライバに通知されるようにするには、DirXML-ADContext を購読者フィルタに通知属性として追加してください。
- <nss-app-username> 要素は、アイデンティティボールト内の CN 属性値によって入力されます。
- パスワード要素には、<add> または <modify-password> コマンド内の <password> 要素の値が入力されます。

SecretStore のプロビジョニング

サンプルのシナリオでは、SecretStore 資格情報プロビジョニング用の操作データが取得され、使用される最初の場所は、ドライバの入力変換ポリシー内にあります。サンプルのシナリオでは、2 つのポリシーが実装されています。

- パスワードが正常に同期された後の SecretStore 資格情報の設定。
- アプリケーションユーザが削除された場合のSecretStore資格情報の削除(アイデンティティボールトのオブジェクトは削除されない)

注 **:** SampleInputTransform.xml ファイルには、SecretStore 資格情報をパスワード同期が成功した後に設定するためのサンプルポリシーがあります。このファイルは、Identity Manager 3.0 Support Pack 1 メディアの utilities ディレクトリの cred_prov フォルダにあります。

SecretStore 資格情報の設定ポリシーでは、プロビジョニングが実行されるのは、返されたコマンドステータスが「成功」で、以前に設定した <operation-data> が存在する場合のみに限定してください。

SecretStore のプロビジョニング解除

「接続システムのユーザアカウントは削除し、アイデンティティボールトのアカウントは残す」というポリシーを使用する状況は数多く考えられます。Finance のシナリオでは、ユーザのアイデンティティボールトの employeeStatus 属性値が「I」に設定された場合に、GroupWise アカウントを削除して、SecretStore 資格情報のプロビジョニングを解除します。この状況を処理するため、GroupWise ドライバの購読者イベント変換に、変更属性値をオブジェクトの削除に変換するポリシーが含まれています。eDirectory アカウントの名前は、削除コマンドが実行された後も必要なため、<operation-data> イベントを <delete> コマンドに設定して、入力変換ポリシー内の SecretStore のプロビジョニング解除ポリシーで使用できるようにする必要があります。

<operation-data> <nss-sync-data> <nss-target-user-dn>cn=GLCANYON,ou=finance,o=Testco Financials </nss-targer-user-dn> </nss-sync-data> </operation-data>

<modify> イベントを <delete> に変換してこの要素を作成するポリシーは、dentity Manager 3.0 Support Pack 1 メディア上の utilities ディレクトリの cred_prov フォルダにある SampleSubEventTransform.xml という名前のファイルに XML 形式で入っています。

XSLT スタイルシートを使用したポリシーの定義

5

ポリシーは、XSLT スタイルシートとして実装できます。XSLT は、XML ドキュメントを変換する標準的な言語です。メタディレクトリエンジン内の XSLT プロセッサは、1999 年 11 月の W3C 勧告に準拠しています。関連の仕様については、次を参照してください。

- XSL 変換 (XSLT) (http://www.w3.org/TR/1999/REC-xslt-19991116)
- XML Path 言語 (XPath) (http://www.w3.org/TR/1999/REC-xpath-19991116)

次の節では、Identity Manager によるスタイルシートを使用した実装方法について説明します。

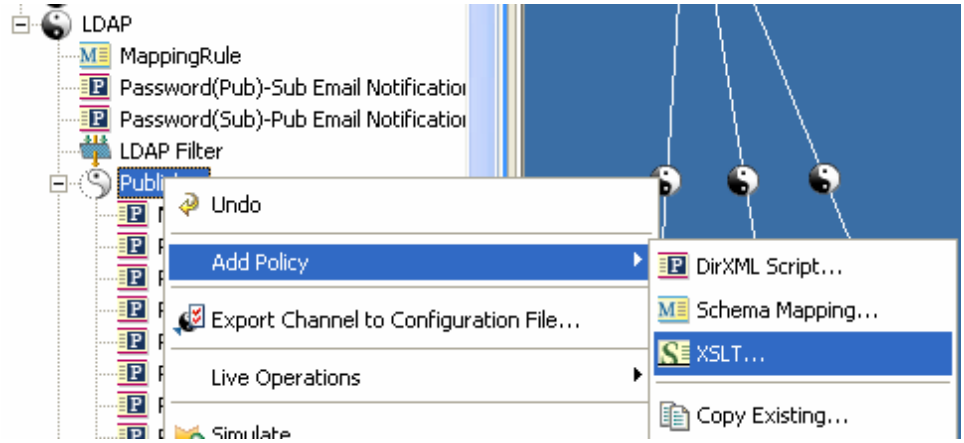
5.1 Designer による XSLT スタイルシートの管理

XSLT ポリシーのスタイルシートは、Designer を使用して追加、変更、削除されます。次の項では、Designer による XSLT スタイルシートの使用方法について詳しく説明します。

5.1.1 Designer による XSLT ポリシーの追加

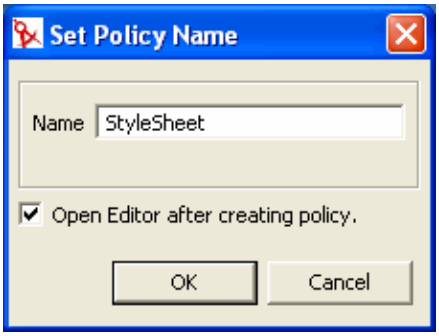
1. Designer 内でプロジェクトを開き、［Outline (アウトライン)］タブを選択します。
2. スタイルシートを指定するドライバと場所を選択します。

3 右クリックして、[[Add Policy (ポリシーの追加)] > [SXLT] の順に選択します。

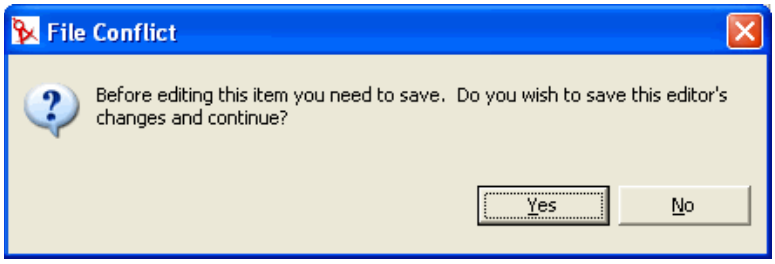


4 スタイルシートの名前を指定します。

5 [Open Editor after creating policy (ポリシーの作成後にエディタを開く)] を選択し、[OK] をクリックします。



6 [はい] を選択して、新しいポリシーを編集する前にプロジェクトを保存します。



7 「add your custom templates here (ここにカスタムテンプレートを追加)」の下に、スタイルシート情報を追加します。

8 ［ファイル］>［保存］の順に選択して、スタイルシートを保存します。

5.2 iManager による XSLT スタイルシートの管理

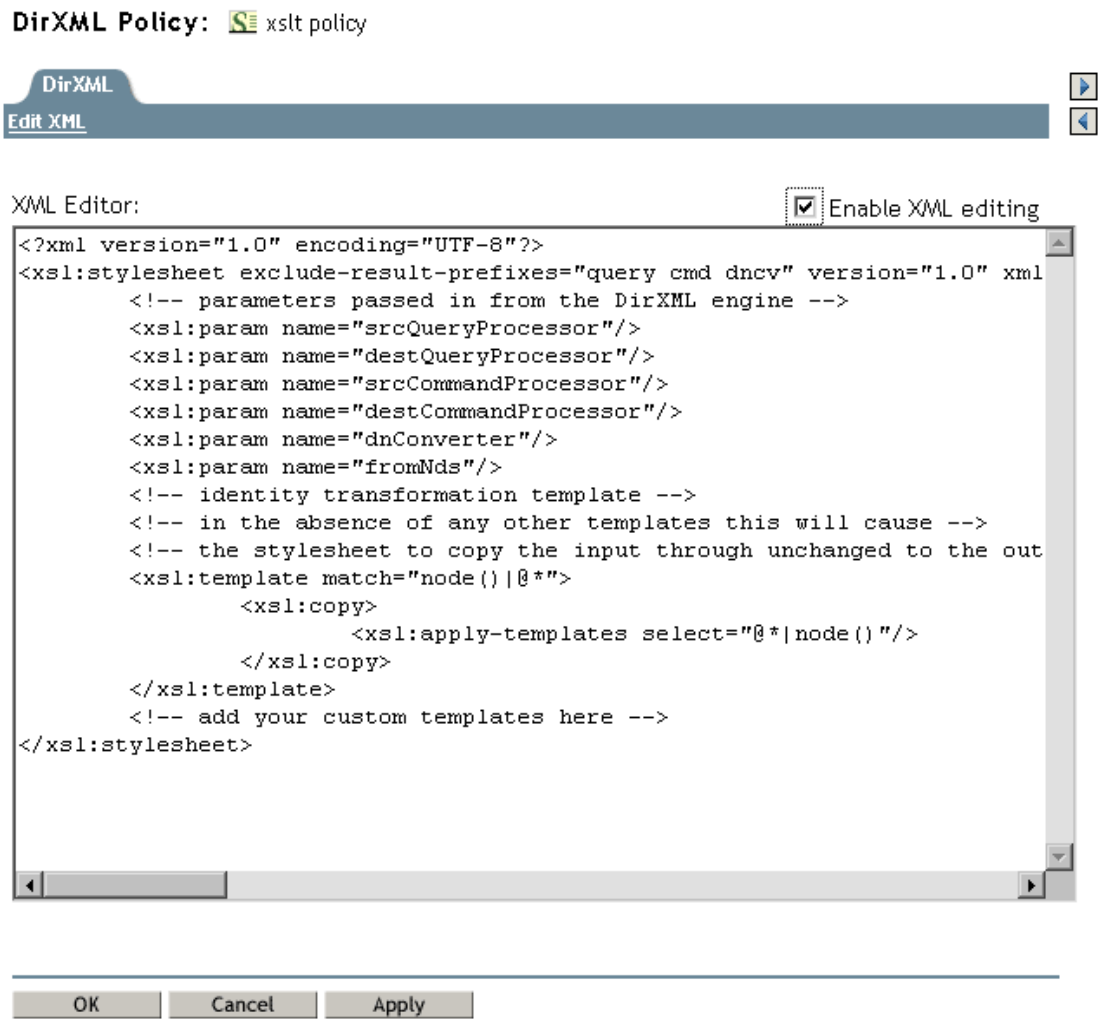
XSLT ポリシーのスタイルシートは、iManager を使用して追加、変更、削除されます。次の項では、iManager による XSLT スタイルシートの使用方法について詳しく説明します。

- 385 ページのセクション 5.2.1「iManager による XSLT ポリシーの追加」

5.2.1 iManager による XSLT ポリシーの追加

1 管理するドライバの［Identity Manager ドライバの概要］を開きます。
2 定義するポリシーを示しているアイコンをクリックします。
3 ［挿入］をクリックします。
4 新しいポリシーの名前を入力して、［Enter (入力)］をクリックします。

5 XSLT ポリシーを定義し、［OK］をクリックします。



5.3 識別情報の変換の開始

iManager または Designer を使用してスタイルシートを新規作成すると、識別情報の変換を実装するスタイルシートで値が事前入力されます。他にテンプレートがない場合は、識別情報の変換によって、入力 XML ドキュメントを通過させることができます。その際、スタイルシートは変更されません。ポリシーは通常、テンプレートを追加することで、変更を実施する XML でだけ動作するように実装します。スタイルシートがドキュメントと、XDS とは異なる XML ボキャブラリ間での変換に使用される場合、識別情報テンプレートを削除しなければならない場合があります (SOAP 用入出力変換ポリシーや区切りテキストドライバなど)。

5.4 Identity Manager から受け取るパラメータの使用

メタディレクトリエンジンは、ポリシーのスタイルシートに、スタイルシートが使用できる次のパラメータを渡します。

- srcQueryProcessor—XdsQueryProcessor インタフェースを実装する Java オブジェクト。これにより、スタイルシートがソースデータベースに対し、より多くの情報を照会できるようになります。
- destQueryProcessor—XdsQueryProcessor インタフェースを実装する Java オブジェクト。これにより、スタイルシートがターゲットデータベースに対し、より多くの情報を照会できるようになります。
- srcCommandProcessor—XdsCommandProcessorインタフェースを実装するJavaオブジェクト。これにより、スタイルシートがイベントソースに対し、コマンドをライトバックできるようになります。DirXML 1.0 では使用できません。
- destCommandProcessor—XdsCommandProcessor インタフェースを実装する Java オブジェクト。これにより、スタイルシートがコマンドを発行して、ターゲットデータストアにコマンドを直接送信できるようになります。
- dnConverter—XdsCommandProcessor インタフェースを実装する Java オブジェクト。これにより、スタイルシートがアイデンティティボールトのオブジェクト DN を、ある形式から他の形式に変換できるようになります。詳細については、「Interface DNCoverter (http://developer.novell.com/ndk/doc/dirxml/dirxmlbk/api/com/novell/nds/dirxml/driver/DNConverter.html)」を参照してください。
- fromNds— ソースデータストアがアイデンティティボールトである場合には「True」、接続システムである場合には「False」であるブール値です。

iManager または Designer で新しいスタイルシートを作成する場合、これらのパラメータへの宣言を含むスタイルシートで事前入力されます。

スキーママッピングポリシー、入力変換ポリシーおよび出力変換ポリシーを使用してクエリやコマンドパラメータを使用する場合、次の制限が適用されます。

- アプリケーションシムに発行されたクエリは、そのアプリケーションシムによって予測された形式である必要があります。つまり、スキーマ名はアプリケーションのネームスペース内にあり、シムによってどのような XML ボキャブラリがネイティブで使用されているかをクエリで確認する必要があります。クエリには関連付けの参照は追加されません。
- アプリケーションシムからの応答は、そのシムの形式で返されます。変更やスキーママッピング、関連付けの参照の解決は行われません。
- eDirectory™ に発行されたクエリは、eDirectory™ によって予測された形式である必要があります。つまり、スキーマ名は eDirectory ネームスペース内にあり、クエリは XDS である必要があります。関連付けの参照は解決されません。
- アプリケーションシムからの応答は、そのシムの形式で返されます。変更やスキーママッピングは行われません。

クエリプロセッサ

クエリプロセッサの使用は、拡張機能の Novell® XSLT を実装するかどうかによって決まります。クエリを作成するには、XdsQueryProcessor インタフェースに対してネームス

ペースを宣言する必要があります。この作業は、次の内容をスタイルシートの <xsl:stylesheet> または <xsl:transform> 要素に追加することで行います。

```
xmlns:query="http://www.novell.com/nxsl/java/
com.novell.nds.dirxml.driver.XdsQueryProcessor"
```

iManager または Designer で新しいスタイルシートを作成する場合、ネームスペース宣言で事前入力されます。クエリプロセッサについての詳細は、「Class XdsQueryProcessor (http://developer.novell.com/ndk/doc/dirxml/dirxmlbk/api/com/novell/nds/dirxml/driver/XdsQueryProcessor.html)」を参照してください。

次の例では、クエリプロセッサの 1 つを使用しています (長すぎる行はラップされており、< では開始されません)。

```
<!-- Query object name queries NDS for the passed object name -->

<xsl:template name="query-object-name">
   <xsl:param name="object-name"/>

<!-- build an xds query as a result tree fragment -->
   <xsl:variable name="query">
            <query>
               <search-class class-name="{ancestor-or-self:
                     :add/@class-name}"/>

<!-- NOTE: depends on CN being the naming attribute -->
               <search-attr attr-name="CN">
                   <value><xsl:value-of select="$object-name"/
                        ></value>
               </search-attr>
<!-- put an empty read attribute in so that we don't get -->
<!-- the whole object back                                -->
               <read-attr/>
            </query>
   </xsl:variable>

<!-- query NDS -->
<xsl:variable name="result" select="query:query($destQuery
     Processor,$query)"/>

<!-- return an empty or non-empty result tree fragment -->
<!-- depending on result of query                      -->
   <xsl:value-of select="$result//instance"/>
</xsl:template>
```

他の例です。

```
<?xml version="1.0"?>
<xsl:transform
            version="1.0"
       xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
       xmlns:cmd="http://www.novell.com/nxsl/java
```

```
       com.novell.nds.dirxml.driver.XdsCommandProcessor"
>
<xsl:param name="srcCommandProcessor"/>

<xsl:template match="node()|@*">
     <xsl:copy>
            <xsl:apply-templates select="@*|node()"/>
      </xsl:copy>
</xsl:template>

<xsl:template match="add">
      <xsl:copy>
            <xsl:apply-templates select="@*|node()"/>
      </xsl:copy>

      <!-- on a user add, add Engineering department to the source
object -->
      <xsl:variable name="dummy">
            <modify class-name="{@class-name} "dest-dn="{@src-dn}">
                  <xsl-copy-of select="association"/>
                  <modify-attr attr-name="OU">
                  <add-value>
                              <value type="string">Engineering</value>
                        </add-value>
                  </modify-attr>
            </modify>
      </xsl:variable>
      <xsl:variable name="dummy2"
         select="cmd:execute($srcCommandProcessor, $dummy)"/>
</xsl:template>

</xsl:transform>
```

5.5 拡張機能の使用

XSLT は、ある種の変換を実行するには優れたツールですが、ゼロでない文字列操作や反復処理などの変換では、十分な機能が備わっているとはいえません。Novell XSLT プロセッサには、スタイルシートから Java(広い意味では JNI でアクセス可能な他の言語) で実装された関数を呼び出すことができる、拡張機能が実装されています。

具体例としては、クエリプロセッサによる上記の例や、Java による文字列操作を示した次の例を参照してください (長すぎる行はラップされており、< では開始されません)。

```
<!-- get-dn-prefix places the part of the passed dn that -->
<!-- precedes the last occurrence of '\' in the passed dn -->
<!-- in a result tree fragment meaning that it can be    -->
<!-- used to assign a variable value                     -->

<xsl:template name="get-dn-prefix" xmlns:jstring="http://
     www.novell.com/nxsl/java/java.lang.String">

   <xsl:param name="src-dn"/>
```

```
<!-- use java string stuff to make this much easier -->
   <xsl:variable name="dn" select="jstring:new($src-dn)"/>
   <xsl:variable name="index" select="jstring:lastIndexOf
       ($dn,'\')"/>
   <xsl:if test="$index != -1">
      <xsl:value-of select="jstring:substring($dn,0,$index)
          "/>
   </xsl:if>
</xsl:template>
```

5.6 パスワードの作成例：作成ポリシー

次のスタイルシートは、作成ポリシーで使用できます。ユーザを作成し、そのユーザの名字と CN の属性からパスワードを生成し、識別情報の変換 (中断と変換を試行しているイベントを除き、ドキュメント内のすべてで渡される) を実行します。

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!--   This stylesheet has an example of how to replace a create rule
with
        an XSLT stylesheet and supply an initial password for "User"
objects. -->

<xsl:transform xmlns:xsl="http://www.w3.org/1999/XSL/Transform
     "version="1.0">

<!-- ensure we have required NDS attributes -->
<xsl:template match="add">
   <xsl:if test="add-attr[@attr-name='Surname'] and
                 add-attr[@attr-name='CN']">
      <!-- copy the add through -->
      <xsl:copy>
         <xsl:apply-templates select="@*|node()"/>
         <!-- add a <password> element -->
         <xsl:call-template name="create-password"/>
      </xsl:copy>
   </xsl:if>

<!-- if the xsl:if fails, we don't have all the required attributes
      so we won't copy the add through, and the create rule will veto
the add -->

</xsl:template>

<xsl:template name="create-password">
   <password>
      <xsl:value-of select="concat(add-attr[@attr-name='Surname']/
value,
           '-',add-attr[@attr-name='CN']/value)"/>
   </password>
</xsl:template>
```

```
<!-- identity transform for everything we don’t want to change -->

<xsl:template match="@*|node()">
    <xsl:copy>
        <xsl:apply-templates select="@*|node()"/>
    </xsl:copy>
</xsl:template>

</xsl:transform>
```

5.7 eDirectory ユーザの作成例 : 作成ポリシー

次のスタイルシートは、作成ポリシーで使用できます。ここでは、外部アプリケーションで作成されたエントリから eDirectory ユーザを作成する方法を示します。この例は、人事部のデータベースに新入社員を作成し、これをネットワーク上に移すという作業が基になっています。ユーザの名前と名字を取得して、一意の CN を eDirectory ツリー内に生成します。eDirectory では、一意の CN を用いる必要があるのは CN が属すコンテナ内に限られますが、このスタイルシートでは eDirectory ツリー内のすべてのコンテナで一意になるようにします。

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!--  This stylesheet is an example of how to replace a create rule
with an
      XSLT stylesheet and that creates the User name from the Surname
and
      given Name attributes -->

<xsl:transform
   xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"
   xmlns:query="http://www.novell.com/nxsl/java/
com.novell.nds.dirxml.driver.
         XdsQueryProcessor"
   >

<!-- This is for testing the stylesheet outside of Identity Manager so
things
     are pretty to look at -->
<xsl:strip-space elements="*"/>
<xsl:preserve-space elements="value,component"/>
<xsl:output method="xml" indent="yes"/>

<!-- Identity Manager always passes two stylesheet parameters to an
XSLT rule:
     an inbound and outbound query processor -->
<xsl:param name="srcQueryProcessor"/>
<xsl:param name="destQueryProcessor"/>

<!-- match <add> elements -->
<xsl:template match="add">
```

```
   <!-- ensure we have required NDS attributes we need for the name -->
   <xsl:if test="add-attr[@attr-name='Surname'] and
                 add-attr[@attr-name='Given Name']">

      <!-- copy the add through -->
      <xsl:copy>
         <!-- copy any attributes through except for the src-dn -->
         <!-- we'll construct the src-dn below so that the placement
rule will work -->
         <xsl:apply-templates select="@*[string(.) != 'src-dn']"/>

        <!-- call a template to construct the object name and place the
result in a variable -->
         <xsl:variable name="object-name">
            <xsl:call-template name="create-object-name"/>
         </xsl:variable>

         <!-- now create the src-dn attribute with the created name -->
         <xsl:attribute name="src-dn">
            <xsl:variable name="prefix">
               <xsl:call-template name="get-dn-prefix">
                  <xsl:with-param name="src-dn" select="string(@src-
dn)"/>
               </xsl:call-template>
            </xsl:variable>
            <xsl:value-of select="concat($prefix,'\',$object-name)"/>
         </xsl:attribute>

         <!-- if we have a "CN" attribute, set it to the constructed
name -->
         <xsl:if test="./add-attr[@attr-name='CN']">
            <add-attr attr-name="CN">
               <value type="string"><xsl:value-of select="$object-
name"/></value>
            </add-attr>
         </xsl:if>

         <!-- copy the rest of the stuff through, except for what we
have already copied -->
         <xsl:apply-templates select="*[name() != 'add-attr' or @attr-
name != 'CN'] |
                                      comment() |
                                       processing-instruction() |
                                        text()"/>

         <!-- add a <password> element -->
         <xsl:call-template name="create-password"/>

      </xsl:copy>
   </xsl:if>
   <!-- if the xsl:if fails, it means we don't have all the required
attributes
       so we won't copy the add through, and the create rule will veto
the add -->
```

```
</xsl:template>

<!-- get-dn-prefix places the part of the passed dn that precedes the
-->
<!-- last occurrance of '\' in the passed dn in a result tree fragment
-->
<!-- meaning that it can be used to assign a variable value
-->
<xsl:template name="get-dn-prefix" xmlns:jstring="http://
www.novell.com/nxsl/java/java.lang.String">
   <xsl:param name="src-dn"/>

   <!-- use java string stuff to make this much easier -->
   <xsl:variable name="dn" select="jstring:new($src-dn)"/>
   <xsl:variable name="index" select="jstring:lastIndexOf($dn,'\')"/>
   <xsl:if test="$index != -1">
      <xsl:value-of select="jstring:substring($dn,0,$index)"/>
   </xsl:if>
</xsl:template>

<!-- create-object-name creates a name for the user object and places
the   -->
<!-- result in a result tree fragment
-->
<xsl:template name="create-object-name">

   <!-- first try is first initial followed by surname -->
   <xsl:variable name="given-name" select="add-attr[@attr-name='Given
Name']/value"/>
   <xsl:variable name="surname" select="add-attr[@attr-
name='Surname']/value"/>
   <xsl:variable name="prefix" select="substring($given-name,1,1)"/>
   <xsl:variable name="object-name" select="concat($prefix,$surname)"/
>

   <!-- then see if name already exists in NDS -->
   <xsl:variable name="exists">
      <xsl:call-template name="query-object-name">
         <xsl:with-param name="object-name" select="$object-name"/>
      </xsl:call-template>
   </xsl:variable>

   <!-- if exists, then try 1st fallback, else return result -->
   <xsl:choose>
      <xsl:when test="$exists != ''">
         <xsl:call-template name="create-object-name-2"/>
      </xsl:when>
      <xsl:otherwise>
         <xsl:value-of select="$object-name"/>
      </xsl:otherwise>
   </xsl:choose>

</xsl:template>
```

```
<!-- create-object-name-2 is the first fallback if the name created by
-->
<!-- create-object-name already exists
-->
<xsl:template name="create-object-name-2">

   <!-- first try is first name followed by surname -->
   <xsl:variable name="given-name" select="add-attr[@attr-name='Given
Name']/value"/>
   <xsl:variable name="surname" select="add-attr[@attr-
name='Surname']/value"/>
   <xsl:variable name="object-name" select="concat($given-
name,$surname)"/>

   <!-- then see if name already exists in NDS -->
   <xsl:variable name="exists">
      <xsl:call-template name="query-object-name">
         <xsl:with-param name="object-name" select="$object-name"/>
      </xsl:call-template>
   </xsl:variable>

   <!-- if exists, then try last fallback, else return result -->
   <xsl:choose>
      <xsl:when test="$exists != ''">
         <xsl:call-template name="create-object-name-fallback"/>
      </xsl:when>
      <xsl:otherwise>
         <xsl:value-of select="$object-name"/>
      </xsl:otherwise>
   </xsl:choose>

</xsl:template>

<!-- create-object-name-fallback recursively tries a name created by
-->
<!-- concatenating the surname and a count until NDS doesn't find
-->
<!-- the name. There is a danger of infinite recursion, but only if
-->
<!-- there is a bug in NDS                                          -
->
<xsl:template name="create-object-name-fallback">
   <xsl:param name="count" select="1"/>

   <!-- construct the a name based on the surname and a count -->
   <xsl:variable name="surname" select="add-attr[@attr-
name='Surname']/value"/>
   <xsl:variable name="object-name" select="concat($surname,'-
',$count)"/>

   <!-- see if it exists in NDS -->
   <xsl:variable name="exists">
      <xsl:call-template name="query-object-name">
         <xsl:with-param name="object-name" select="$object-name"/>
```

```
      </xsl:call-template>
   </xsl:variable>

   <!-- if exists, then try again recursively, else return result -->
   <xsl:choose>
      <xsl:when test="$exists != ''">
         <xsl:call-template name="create-object-name-fallback">
            <xsl:with-param name="count" select="$count + 1"/>
         </xsl:call-template>
      </xsl:when>
      <xsl:otherwise>
         <xsl:value-of select="$object-name"/>
      </xsl:otherwise>
   </xsl:choose>

</xsl:template>

<!-- query object name queries NDS for the passed object-name. Ideally,
this would   -->
<!-- not depend on "CN": to do this, add another parameter that is the
name of the   -->
<!-- naming attribute.
-->
<xsl:template name="query-object-name">
   <xsl:param name="object-name"/>

   <!-- build an xds query as a result tree fragment -->
   <xsl:variable name="query">
      <nds ndsversion="8.5" dtdversion="1.0">
         <input>
            <query>
              <search-class class-name="{ancestor-or-self::add/@class-
name}"/>
               <!-- NOTE: depends on CN being the naming attribute -->
               <search-attr attr-name="CN">
                  <value><xsl:value-of select="$object-name"/></value>
               </search-attr>
              <!-- put an empty read attribute in so that we don't get
the whole object back -->
               <read-attr/>
            </query>
         </input>
      </nds>
   </xsl:variable>

   <!-- query NDS -->
   <xsl:variable name="result"
select="query:query($destQueryProcessor,$query)"/>

   <!-- return an empty or non-empty result tree fragment depending on
result of query -->
   <xsl:value-of select="$result//instance"/>
</xsl:template>
```

```
<!-- create an initial password -->
<xsl:template name="create-password">
   <password>
      <xsl:value-of select="concat(add-attr[@attr-name='Surname']/
value,'-',add-attr[@attr-name='CN']/value)"/>
   </password>
</xsl:template>

<!-- identity transform for everything we don't want to mess with -->
<xsl:template match="@*|node()">
    <xsl:copy>
        <xsl:apply-templates select="@*|node()"/>
    </xsl:copy>
</xsl:template>

</xsl:transform>
```

フィルタの管理

6

フィルタエディタにより、フィルタの管理ができます。フィルタエディタでは、クラスおよび属性が発行者チャネルおよび購読者チャネルによって処理される方法を定義します。

この節では、次のフィルタ関連のトピックついて説明します。

6.1 Designer でのフィルタタスク

この節では、Designer における一般的なフィルタ関連のタスクを実行する手順を説明します。

6.1.1 フィルタエディタへのアクセス方法

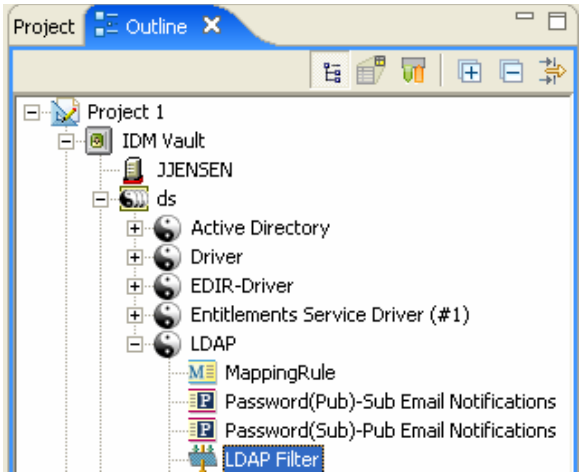
フィルタエディタにより、フィルタの編集ができます。フィルタエディタへのアクセス方法には、モデルアウトライン、ポリシーフローおよびポリシーセットビューのそれぞれを使用する 3 つの方法があります。

［Model Outline (モデルアウトライン)］ビュー

1. 開いているプロジェクトで、［Outline (アウトライン)］タブをクリックします。
2. ［Show Model Outline (モデルアウトラインの表示)］アイコンをクリックします。
3. フィルタを管理するドライバを選択し、右側のプラス記号をクリックします。
4. ［フィルタ］アイコンをダブルクリックして、フィルタエディタを起動します。

   または

右クリックして、［編集］を選択します。

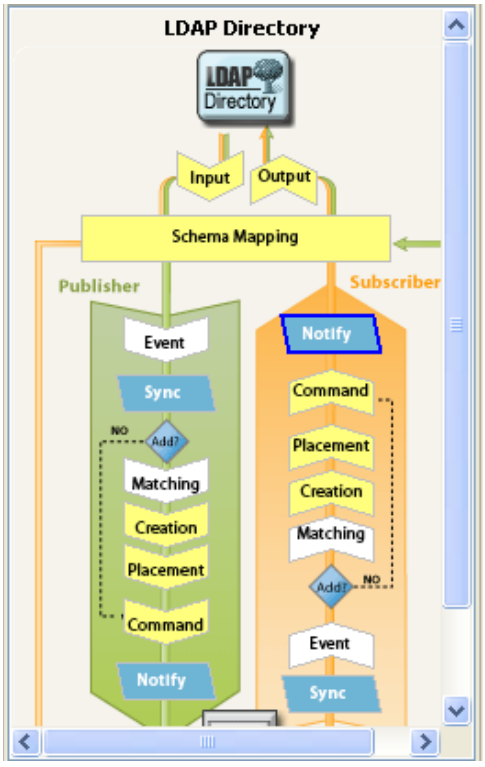


［Policy Flow (ポリシーフロー)］ビュー

1 開いているプロジェクトで、［Outline (アウトライン)］タブをクリックします。

2 ［Show Policy Flow (ポリシーフローの表示)］アイコンを選択します。

3 ［Sync (同期)］アイコンまたは［Notify (通知)］アイコンをダブルクリックして、フィルタエディタを起動します。



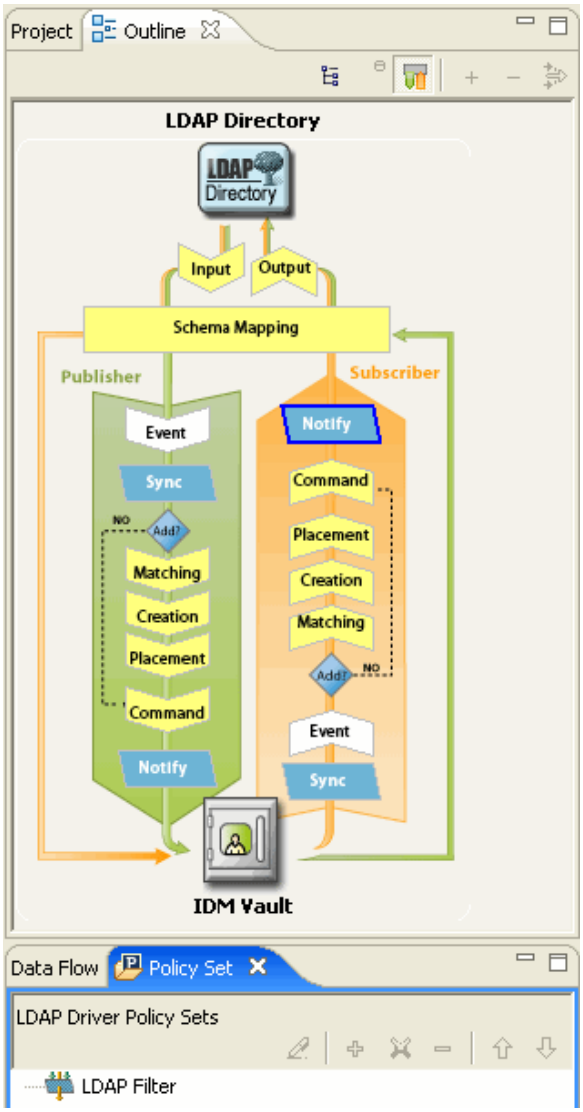
4 ポリシーフローの下にあるポリシーセットマネージャにフィルタが表示されている場合は、このフィルタをダブルクリックしてフィルタエディタを起動します。

または

右クリックして、[[Edit Policy (ポリシーの編集)] > [フィルタ] の順に選択します。

ポリシーセットビュー

1 フィルタポリシーをダブルクリックします。



キーボード操作

表 *6-1* フィルタエディタでのキーボード操作

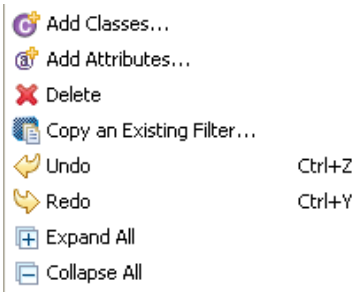
アクション	説明
上矢印	フィルタエディタ内でカーソルを上方向に移動します。
下矢印	フィルタエディタ内でカーソルを下方向に移動します。

アクション	説明
左矢印	表示されている情報を縮小表示します
右矢印	表示されている情報を展開します。
Insert	クラスを追加します。
Ctrl+Insert	属性を追加します。
Delete	選択された項目を削除します。
Enter	編集モードに切り替えます。変更を確定するには、Enter を 2 回押します。
Esc	編集モードを終了します。

6.1.2 フィルタの編集

フィルタエディタにより、フィルタの作成と編集ができます。コンテキストに応じたメニューを表示するには、項目を右クリックします。

図 *6-1* フィルタオプション



- 400 ページの 「クラスと属性の削除または追加」
- 401 ページの 「複数の属性の変更」
- 401 ページの 「既存のフィルタのコピー」
- 402 ページの 「属性のデフォルト値の設定」
- 402 ページの 「フィルタ設定の変更」

クラスと属性の削除または追加

クラスと属性を削除または追加することで、接続されたデータストアとアイデンティティボールトとの間で同期するオブジェクトを指定できます。

クラスまたは属性の削除

クラスまたは属性を同期しないようにする場合は、クラスまたは属性をフィルタから完全に削除するのが最善の方法です。フィルタから属性とクラスを追加または削除するには、次の 2 つの方法があります。

- 削除するクラスまたは属性を右クリックし、[削除] を選択します。
- 削除するクラスまたは属性を選択し、右上隅の [削除] アイコン ✖ をクリックします。

クラスの追加

1 フィルタエディタ内で右クリックし、[Add Classes (クラスの追加)] をクリックします。

または

右上隅の [Add Classes (クラスの追加)] アイコン をクリックします。

2 追加するクラスを参照して選択し、[OK] をクリックします。

3 情報を同期するオプションを変更します。

4 変更を保存するため、[ファイル] > [保存] の順にクリックします。

属性の追加

1 フィルタエディタ内で右クリックし、[属性の追加] をクリックします。

または

右上隅の [属性] アイコン をクリックします。

2 追加する属性を参照して選択し、[OK] をクリックします。

3 情報を同期するオプションを変更します。

4 変更を保存するには、[ファイル] > [保存] の順にクリックします。

複数の属性の変更

フィルタエディタでは、複数の属性を一度に変更できます。キーを押しながら複数の属性を選択します。オプションを変更すると、選択したすべての属性でそれが変更されます。

既存のフィルタのコピー

既存のフィルタを別のドライバからコピーし、現在作業しているドライバで使用できます。

1 [Copy an Existing Filter (既存のフィルタのコピー)] アイコン をクリックします。

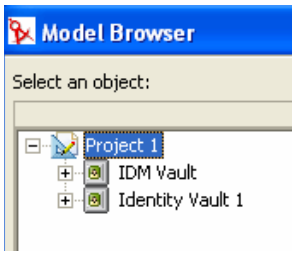
または

フィルタエディタ内で右クリックし、[Copy an Existing Filter (既存のフィルタのコピー)] をクリックします。

2 コピーするフィルタオブジェクトを参照して選択し、[OK] をクリックします。

プロジェクトに複数のアイデンティティボールトがある場合は、他のアイデンティティボールトからフィルタをコピーできます。他のオブジェクトを参照して選択する

場合は、他のアイデンティティボールトを参照し、そこに保存されているフィルタを使用できます。



属性のデフォルト値の設定

新しい属性をフィルタに追加する場合は、そのデフォルト値を定義できます。

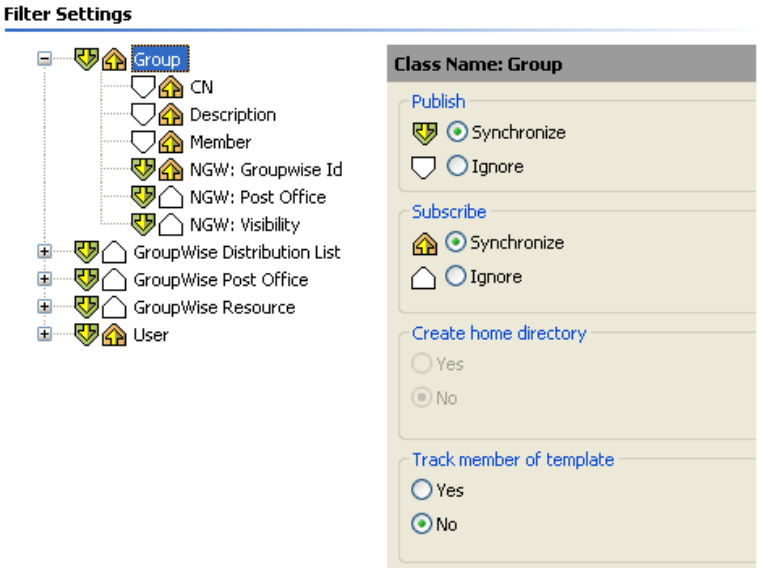
1 右上隅の［Set Default Values for New Attributes (新しい属性のデフォルト値の設定)］アイコン をクリックします。

2 新しい属性に指定するオプションを選択し、［OK］をクリックします。

フィルタ設定の変更

フィルタエディタには、アイデンティティボールトと接続システム間の情報の同期方法を変更するオプションがあります。フィルタの設定は、クラスと属性で異なります。

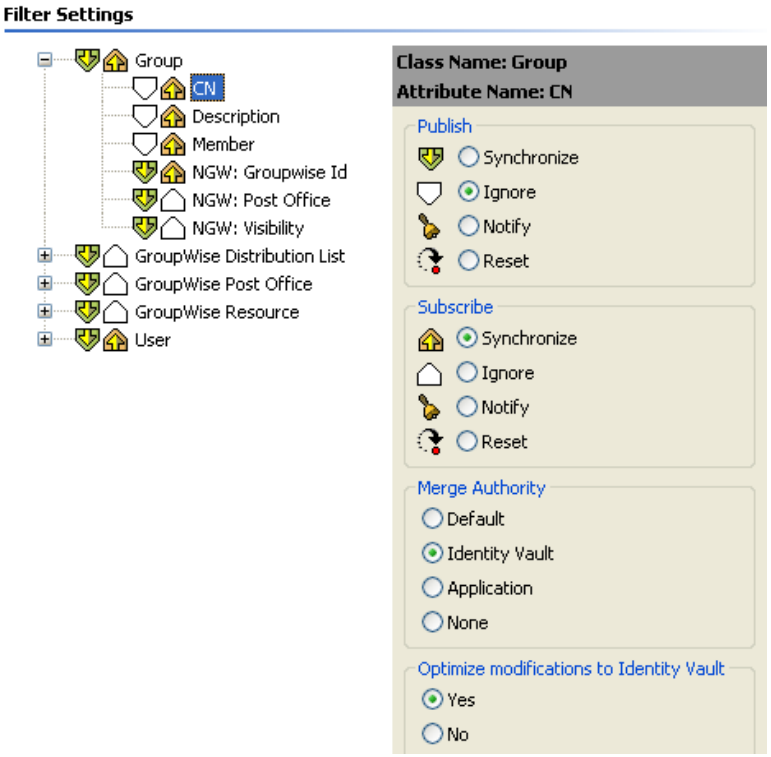
1 フィルタエディタで、クラスを選択します。



2 選択したクラスのフィルタ設定を変更します。

オプション	定義
発行者	◆ **[同期]**：接続システムからアイデンティティボールト方向でクラスを同期できます。 ◆ **[無視]**：接続システムからアイデンティティボールト方向でクラスを同期しません。
購読者	◆ **[同期]**：アイデンティティボールトから接続システム方向でクラスを同期できます。 ◆ **[無視]**：アイデンティティボールトから接続システム方向でクラスを同期しません。
ホームディレクトリの作成	◆ **[はい]**：ホームディレクトリを自動的に作成します。 ◆ **[いいえ]**：ホームディレクトリを作成しません。
テンプレートのメンバを追跡	◆ **[はい]**：発行者チャネルがテンプレートからオブジェクトを作成するときに、「テンプレートのメンバ」属性を保持しているかどうかを調べます。 ◆ **[いいえ]**：「テンプレートのメンバ」属性を追跡しません。

3 属性を選択します。



4 選択した属性のフィルタ設定を変更します。

オプション	定義
発行者	◆ **［同期］：** このオブジェクトに対する変更は、レポートされ、自動的に同期化されます。 ◆ **［無視］：** このオブジェクトに対する変更は、レポートも自動的な同期化もされません。 ◆ **［通知］：** このオブジェクトに対する変更はレポートされますが、自動的には同期化されません。 ◆ **［リセット］：** オブジェクト値を、もう一方のチャネルで指定された値にリセットします (この値は、発行者と購読者の一方のチャネルだけに設定できます。両方には設定できません)。
購読者	◆ **［同期］：** このオブジェクトに対する変更は、レポートされ、自動的に同期化されます。 ◆ **［無視］：** このオブジェクトに対する変更は、レポートも自動的な同期化もされません。 ◆ **［通知］：** このオブジェクトに対する変更はレポートされますが、自動的には同期化されません。 ◆ **［リセット］：** オブジェクト値を、もう一方のチャネルで指定された値にリセットします (この値は、発行者と購読者の一方のチャネルだけに設定できます。両方には設定できません)。
マージ権限	◆ **［Default Behavior (デフォルトの動作)］：** 属性がどちらのチャネルでも同期されていない場合、マージは行われません。 属性が一方のチャネルだけで同期されている場合は、そのチャネルのターゲットに既存の値がすべて削除され、そのチャネルのソースの値と置き換えられます。ソースに複数の値があるが、ターゲットでは値を 1 つしか受け取れない場合は、値のうち 1 つだけがターゲット側で使用されます。 属性が両方のチャネルで同期され、いずれの側も値を 1 つしか受け取れない場合、接続されているアプリケーションがアイデンティティボールト値を取得します。ただし、アイデンティティボールトに値がない場合を除きます。この場合は、アイデンティティボールトが、接続されているアプリケーションから (存在する場合は) 値を取得します。 属性が両方のチャネルで同期され、一方だけが複数の値を受け入れられる場合は、単一値しか受入れられない方の値が複数値側に存在していない場合に限り、複数値側に追加されます。単一の側に値がない場合は、単一の側に追加する値を選択できます。 これは常に有効な動作です。 ◆ **［アイデンティティボールト］：** 属性が発行者チャネルではなく購読者チャネルで同期されている場合のデフォルトの動作と同じ動作です。 これは、購読者チャネルで同期されている場合に有効な動作です。 ◆ **［アプリケーション］：** 属性が購読者チャネルではなく発行者チャネルで同期されている場合のデフォルトの動作と同じ動作です。 これは、発行者チャネルで同期されている場合に有効な動作です。 ◆ ［なし］：同期に関係なく、マージされません。

オプション	定義
Identity Manager に対する変更の最適化	◆ ［はい］：アイデンティティボールトで行われる変更を最小限にするために、発行者チャネルでこの属性の変更を調べます。 ◆ ［いいえ］：変更を調べません。

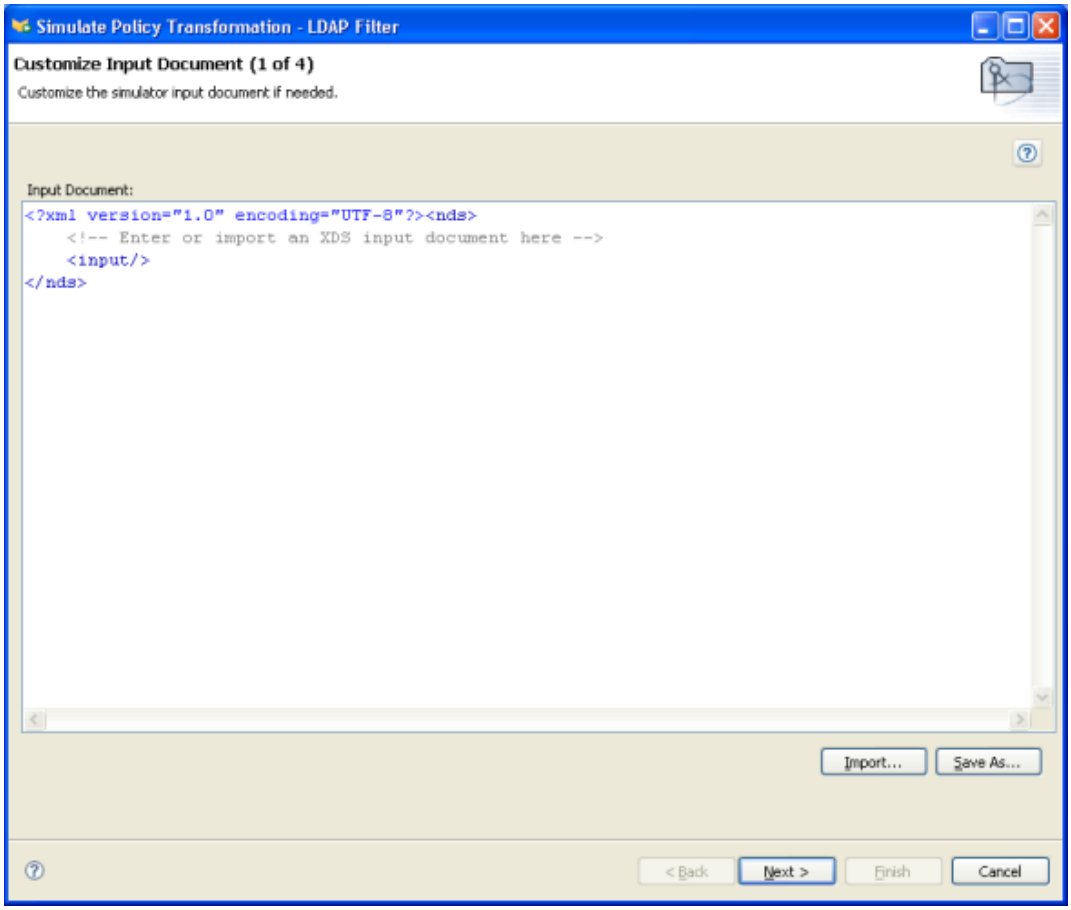
5 ［保存］アイコン をクリックして、変更を保存します。

6.1.3 フィルタのテスト

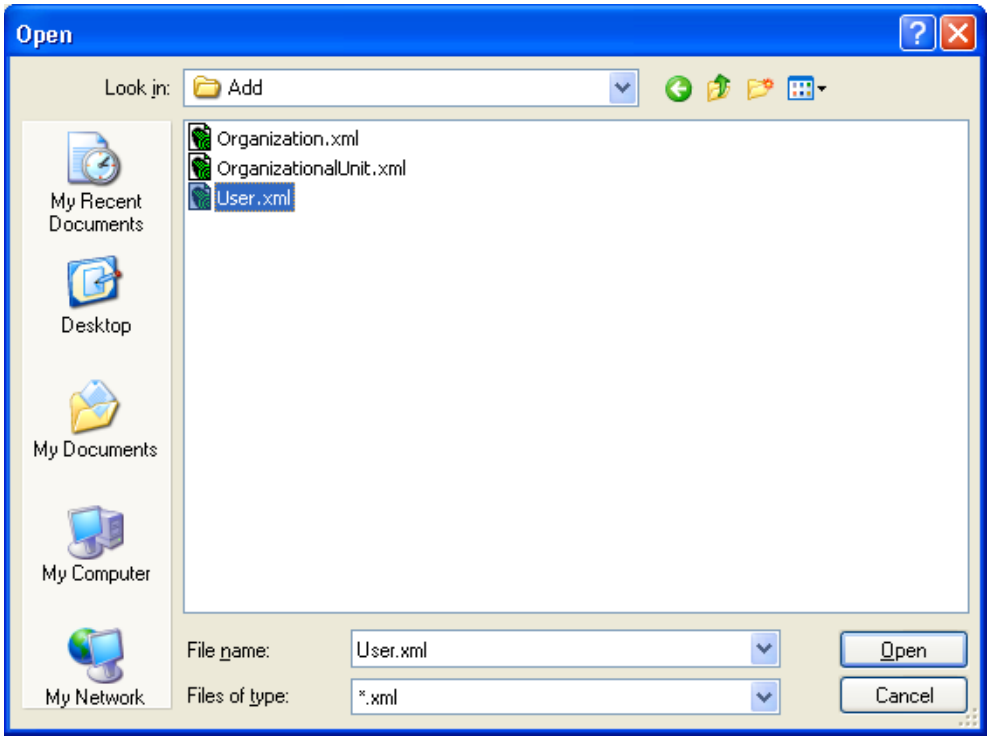
Designer には、ポリシーシミュレータと呼ばれるツールが付属しています。このツールを使用すると、運用環境に実装しなくてもポリシーをテストできます。ポリシーシミュレータをフィルタエディタから起動して、変更後のポリシーをテストできます。

1 ツールバーの［Launch Policy Simulator (ポリシーシミュレータの起動)］アイコン をクリックします。

2 ［Import (インポート)］を選択し、イベントをシミュレートするファイルを参照して選択します。

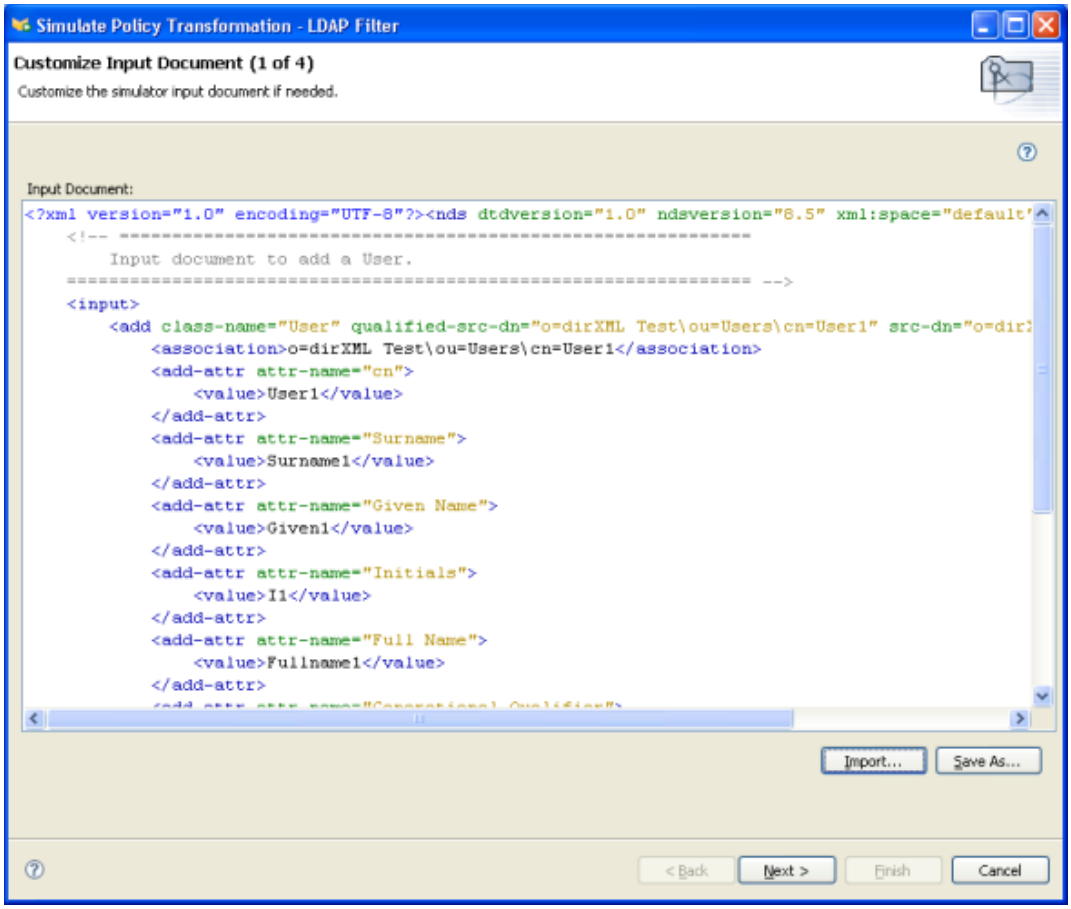


3 ファイルを選択して、［開く］をクリックします。この例では、com.novell.designer.idm.policy\simulation\add\User.xml ファイルを使用して、ユーザオブジェクトの「追加」イベントをシミュレートします。



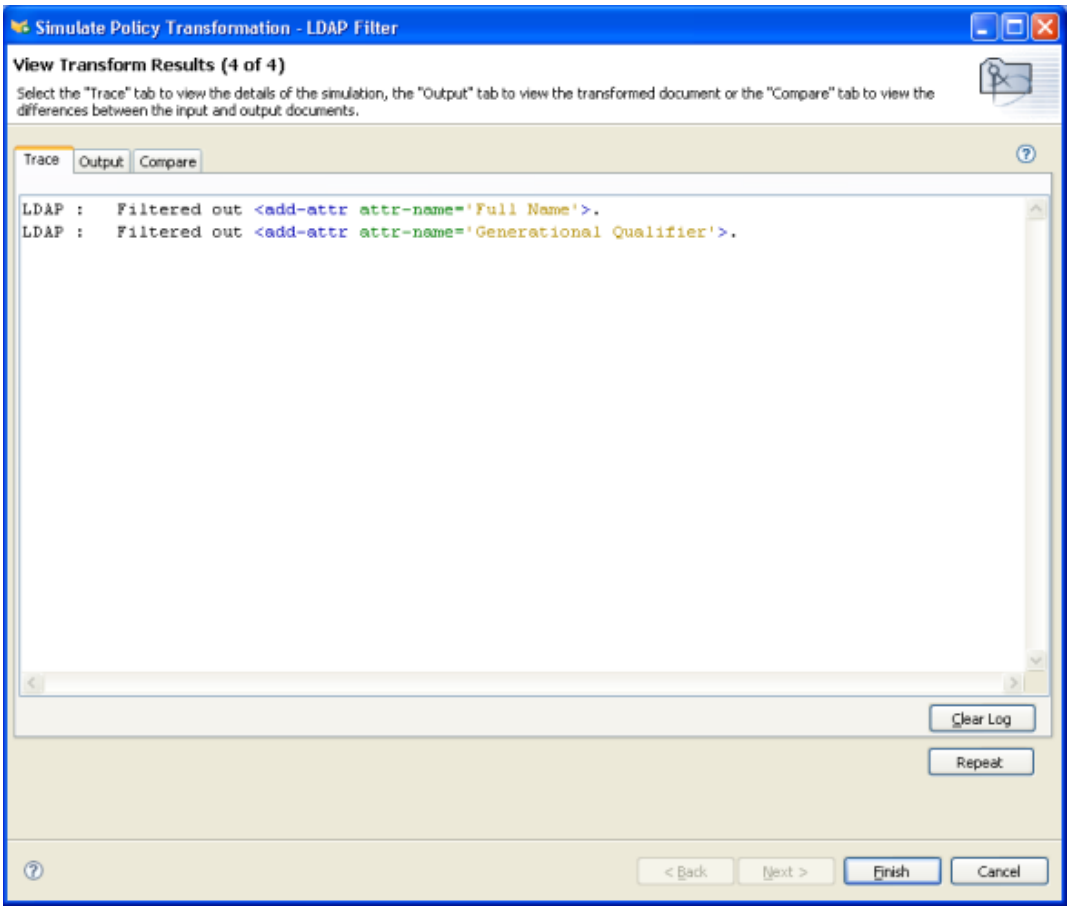
ポリシーシミュレータに、ユーザの「追加」イベントの入力ドキュメントが表示されます。

4［次へ］をクリックして、シミュレーションを開始します。

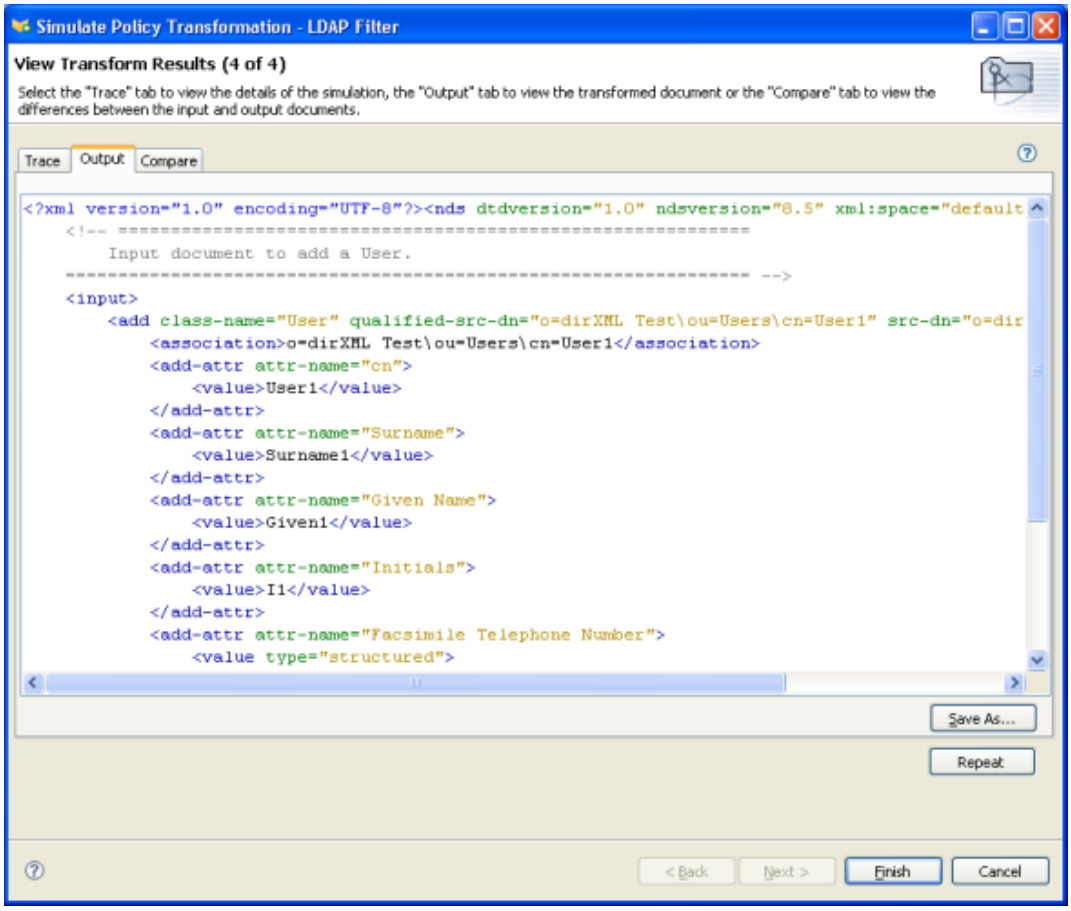


ポリシーシミュレータに、追加イベントのログ、出力ドキュメント、および入力ドキュメントと生成された出力ドキュメントの比較が表示されます。

5 ［トレース］タブを選択して、DSTRACE に表示されるとおりの追加イベントの結果を表示します。

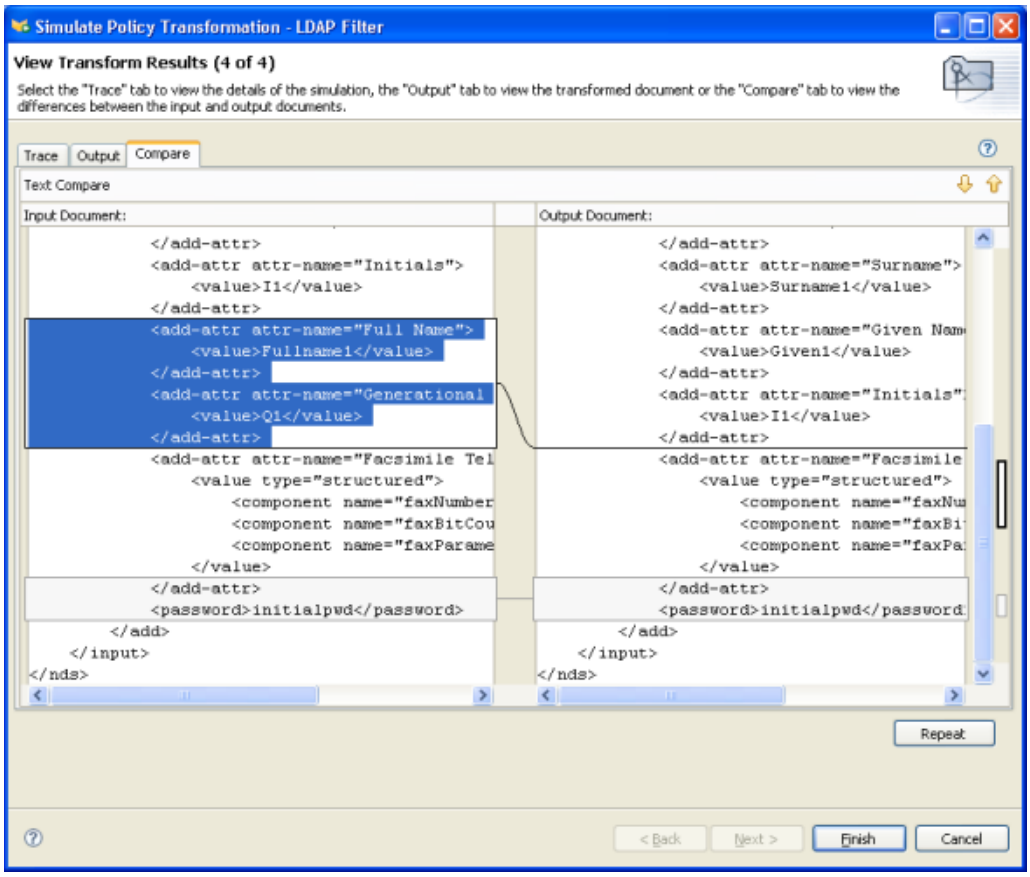


6［出力］タブを選択し、入力ドキュメントに対してフィルタが実行されたときに生成される出力ドキュメントを表示します。入力ドキュメントはユーザの「追加」イベントです。



入力ドキュメントおよび出力ドキュメントを編集できます。変更を保持する場合は、［名前を付けて保存］をクリックします。

7 [Compare (比較)] タブを選択して、入力ドキュメントのテキストと、生成された出力ドキュメントを比較します。



8 [Repeat (繰り返し)] をクリックして、別の入力ドキュメントを選択し、イベントの結果を表示します。

9 フィルタのテストが済んだら、[終了] をクリックして、ポリシーシミュレータを閉じます。

6.1.4 フィルタの XML ソースの表示

Designer では、XML エディタまたはテキストエディタを使用して、XML を表示、編集、および検証できます。

- 410 ページの 「XML ソースの表示」
- 413 ページの 「XML ソースの編集」
- 416 ページの 「XML ソースの検証」

XML ソースの表示

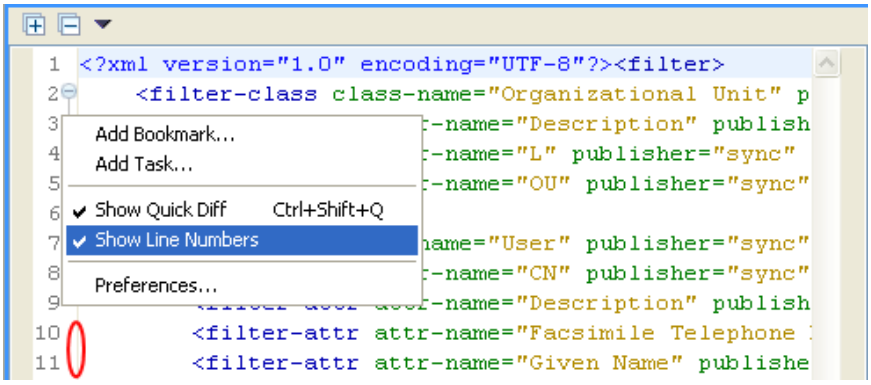
XML ソースは、XML 形式または XML ツリー形式で表示できます。

XML ソースビューを開くには

1 フィルタエディタのワークスペースの下部にある［XML Source (XML ソース)］をクリックします。

XML エディタに行番号が表示されます。行番号を表示するには、左の余白を右クリックし、［Show Line Numbers (行番号の表示)］を選択します。

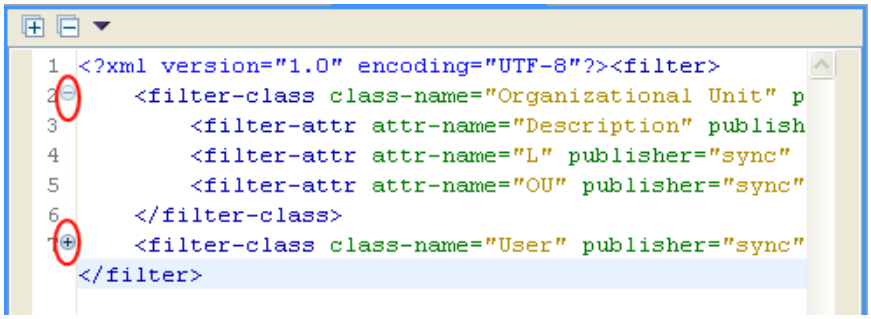
図 6-2 「*Show Line Numbers (行番号の表示)*」フィルタ



XML エディタは、XML を機能別に展開または縮小します。多くの XML を含む機能が複数ある場合は、左上隅のマイナスアイコンをクリックして、XML を縮小できます。XML 機能をすべて展開するには、左上隅のプラスアイコンをクリックします。

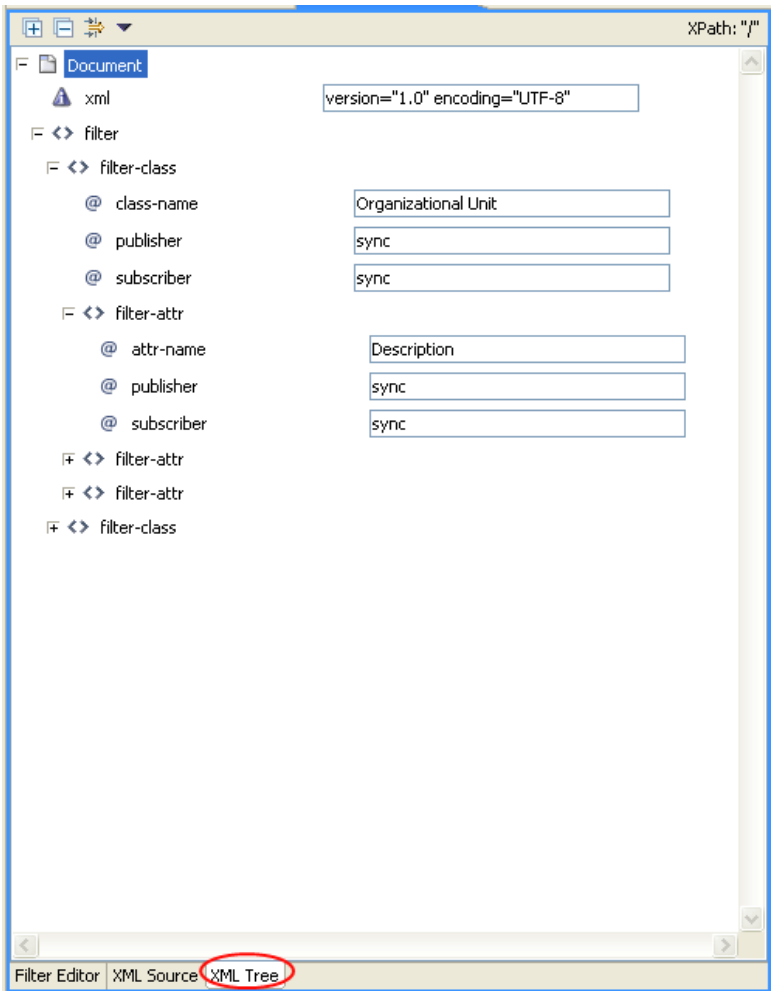
各要素には、左の余白にそれぞれのプラスまたはマイナスアイコンがあります。

図 *6-3* *フィルタの XML のプラスまたはマイナスアイコン*



XML をツリー形式で表示するには

1 フィルタエディタのワークスペースの下部にある［XML Tree (XML ツリー)］をクリックします。

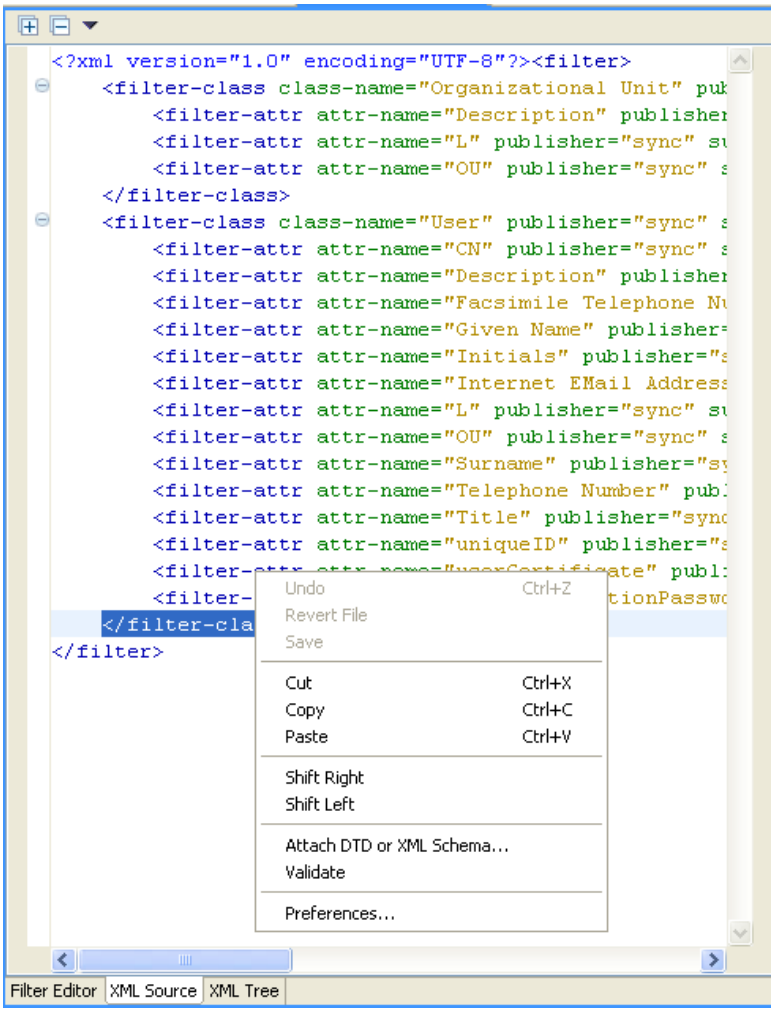


ツリー全体を表示するには、一覧表示されている個々の項目を展開します。

XML ソースの編集

XML は XML エディタで編集できます。GUI を使用する場合と同様、XML エディタで変更することもできます。

図 6-4 フィルタの *XML* ソースの編集



ロードされるデフォルトエディタは、.xml ファイルのタイプに関連付けられています。デフォルトエディタが見つからない場合は、システムのテキストエディタがロードされます。XML ソースビューの機能は、ロードされるエディタに基づきます。

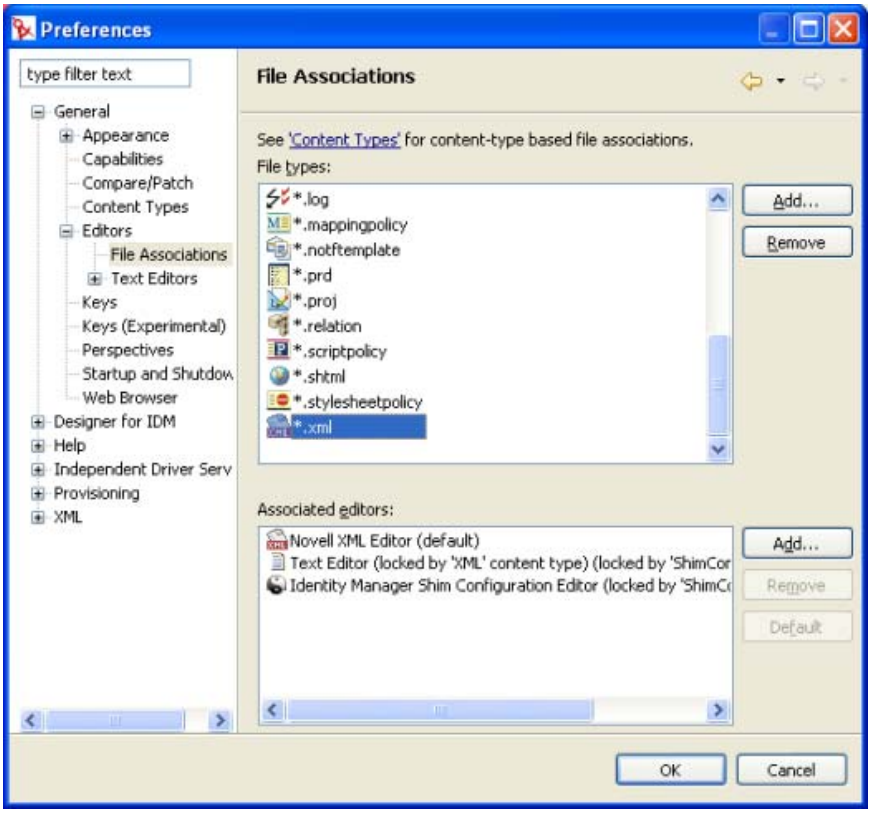
右クリックすると、XML エディタに含まれる機能のリストが表示されます。

- ［元に戻す］：最後のアクションを元に戻します。
- ［**Revert File (** ファイルを戻す **)**］：ファイルを、保存されていたバージョンに戻します。
- ［保存］：ファイルを保存します。
- ［切り取り］：選択された情報を切り取ります。

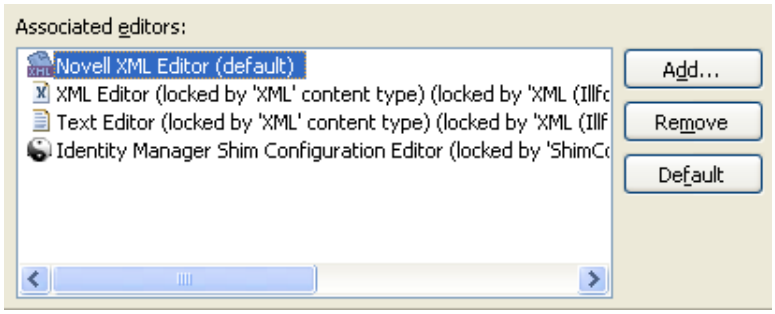
- ［コピー］：選択された情報をクリップボードにコピーします。
- ［貼り付け］：情報をドキュメントに貼り付けます。
- ［**Shift Right (** 右にシフト **)**］：行を右にインデントします。
- ［**Shift Left (** 左にシフト **)**］：行を左にインデントします。
- ［**Attache DTD or XML Schema (DTD** または **XML** スキーマを添付 **)**］：ポリシーの検証のために、DTD または XML スキーマファイルを添付します。
- ［検証］：XML コードを検証します。
- ［初期設定］：XML エディタの初期設定を指定します。

XML ソースビュー用に、別の XML エディタを選択するには

1. メインメニューの［Window (ウィンドウ)］>［初期設定］の順にクリックします。
2. ［一般］>［Editor (エディタ)］>［File Associations (ファイルの関連付け)］の順にクリックします。
3. ファイルタイプのリストから［*.xml］を選択します。



4 ［Associated editors (関連付けられているエディタ)］で、エディタ (たとえば、［Novell XML Editor (Novell XML エディタ)］) を選択します (適切なエディタがリストにない場合は、［追加］をクリックしてリストに追加します)。



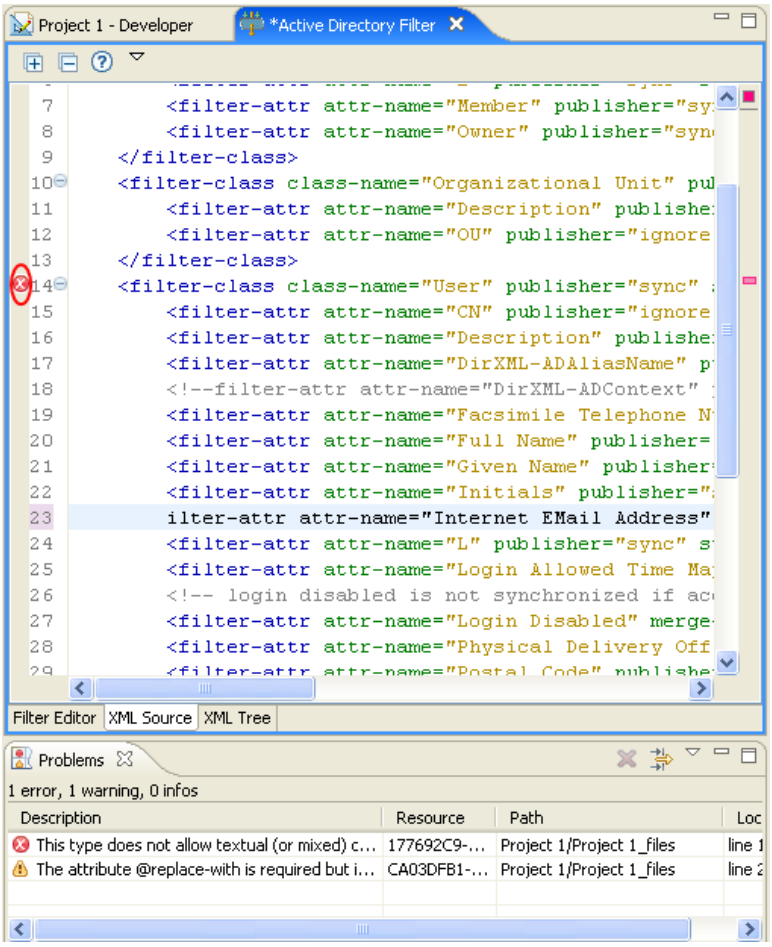
5 ［OK］をクリックします。

6 フィルタエディタをいったん閉じて再度開きます。［XML ソース］ビューにデフォルトエディタがロードされます。

XML ソースの検証

XML エディタは、XML コードを検証します。右クリックし、[検証] を選択します。エラーがある場合は、その行に赤の「x」が表示されます。ウィンドウの下部の説明に、問題についての詳しい情報が示されます。

図 *6-5* フィルタの検証



この例では、<filter-attr> の開始タグと最初の 1 文字がありません。

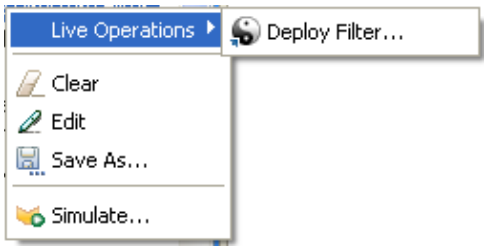
6.1.5 追加のフィルタオプション

フィルタオブジェクトを右クリックすると、[Outline (アウトライン)] ビュー、[Policy Flow (ポリシーフロー)] ビュー、および [Policy Set (ポリシーセット)] ビューで、複数のオプションが表示されます。

- 417 ページの 「[Outline (アウトライン)] ビューの追加オプション」
- 417 ページの 「[Policy Flow (ポリシーフロー)] ビューの追加オプション」
- 418 ページの 「[Policy Set (ポリシーセット)] ビューの追加オプション」

[Outline (アウトライン)] ビューの追加オプション

1 [Outline (アウトライン)] ビューで、フィルタオブジェクトを右クリックします。



- [**Live Operations (** ライブ操作 **)**] > [**Deploy Filter (** フィルタの展開 **)**]：アイデンティティボールトにフィルタを展開します。
- [**Clear (** クリア **)**]：フィルタポリシーからすべての内容を削除します。ただしオブジェクトは残します。
- [編集]：フィルタエディタを起動します。詳細については、400 ページのセクション 6.1.2「フィルタの編集」を参照してください。
- [名前を付けて保存]：フィルタを .xml ファイルとして保存します。
- [**Simulate (** シミュレート **)**]：ポリシーシミュレータを起動します。詳細については、405 ページのセクション 6.1.3「フィルタのテスト」を参照してください。

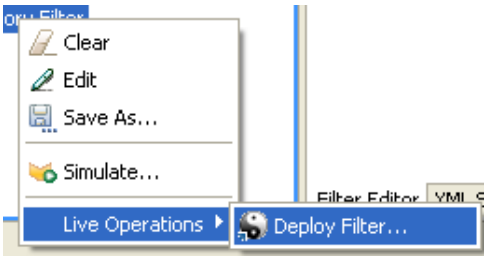
[Policy Flow (ポリシーフロー)] ビューの追加オプション

1 [Policy Flow (ポリシーフロー)] ビューで、フィルタオブジェクトを右クリックします。

- [**Edit Policy (** ポリシーの編集 **)**] > [フィルタ]：フィルタエディタを起動します。詳細については、400 ページのセクション 6.1.2「フィルタの編集」を参照してください。
- [**Simulate (** シミュレート **)**]：ポリシーシミュレータを起動します。詳細については、405 ページのセクション 6.1.3「フィルタのテスト」を参照してください。

［**Policy Set (ポリシーセット)**］ビューの追加オプション

1 ［Policy Set (ポリシーセット)］ビューで、フィルタオブジェクトを右クリックします。



- ［**Clear (クリア)**］：フィルタポリシーからすべての内容を削除します。ただしオブジェクトは残します。
- ［編集］：フィルタエディタを起動します。詳細については、419 ページのセクション 6.2.2「フィルタの編集」を参照してください。
- ［保存］：フィルタを .Xml ファイルとして保存します。
- ［**Simulate (シミュレート)**］：ポリシーシミュレータを起動します。詳細については、405 ページのセクション 6.1.3「フィルタのテスト」を参照してください。
- ［**Live Operations (ライブ操作)**］> ［**Deploy Filter (フィルタの配置)**］：アイデンティティボールトにフィルタを展開できます。

6.2 iManager でのフィルタタスク

この節では、iManager における一般的なフィルタ関連のタスクを実行する手順を説明します。

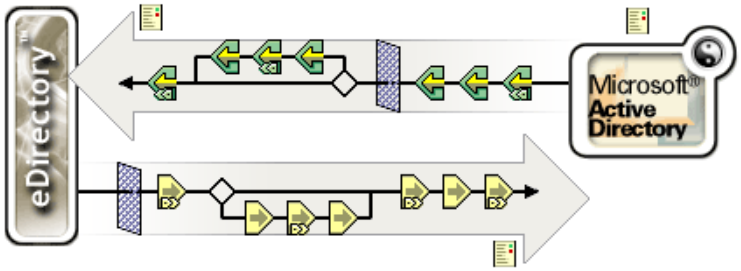
6.2.1 フィルタへのアクセス

1 iManager で、［Identity Manager］役割を展開し、［Identity Manager の概要］をクリックします。

2 ［ツリー全体を検索する］、または［次のコンテナ内を検索する］を選択し、［検索］をクリックします。

3 フィルタにアクセスするドライバをクリックします。［Identity Manager ドライバの概要］が開きます。

図 *6-6* ドライバの概要



4 発行者チャネルまたは購読者チャネルのフィルタアイコンをクリックします。いずれの場合も同じオブジェクトです。

6.2.2 フィルタの編集

フィルタエディタには、アイデンティティボールトと接続システム間の情報の同期方法を編集するオプションがあります。次に、フィルタを編集する場合の最も一般的なタスクのリストを示します。

- 419 ページの 「フィルタからのクラスまたは属性の削除」
- 419 ページの 「クラスの追加」
- 419 ページの 「属性の追加」
- 420 ページの 「フィルタのコピー」
- 420 ページの 「テンプレートの設定」
- 420 ページの 「フィルタ設定の変更」

フィルタからのクラスまたは属性の削除

1 クラスまたは属性を選択して、［削除］をクリックします。

クラスの追加

1 ［Add Class (クラスの追加)］をクリックします。
2 情報を同期するオプションを変更します。
3 ［適用］をクリックします。

属性の追加

1 ［属性の追加］をクリックします。
2 情報を同期するオプションを変更します。

3 ［適用］をクリックします。

フィルタのコピー

既存のドライバから現在作業しているドライバにフィルタをコピーできます。

1 ［Copy Filter From (フィルタのコピー元)］をクリックします。
2 フィルタのコピー元のドライバを参照して選択し、［OK］をクリックします。

テンプレートの設定

フィルタに追加する属性のデフォルト値を設定できます。

1 ［Set Template (テンプレートの設定)］をクリックします。
2 新しい属性に設定するオプションを選択し、［OK］をクリックします。

値は、属性を作成した後に変更することもできます。

フィルタ設定の変更

フィルタエディタには、アイデンティティボールトと接続システム間の情報の同期方法を変更するオプションがあります。フィルタの設定は、クラスと属性で異なります。

1 フィルタエディタで、クラスを選択します。
2 選択したクラスのフィルタ設定を変更します。

オプション	定義
発行者	◆ **［同期］：** 接続システムからアイデンティティボールト方向でクラスを同期できます。 ◆ **［無視］：** 接続システムからアイデンティティボールト方向でクラスを同期しません。
購読者	◆ **［同期］：** アイデンティティボールトから接続システム方向でクラスを同期できます。 ◆ **［無視］：** アイデンティティボールトから接続システム方向でクラスを同期しません。
ホームディレクトリの作成	◆ **［はい］：** ホームディレクトリを自動的に作成します。 ◆ **［いいえ］：** ホームディレクトリを作成しません。
テンプレートのメンバを追跡	◆ **［はい］：** 発行者チャネルがテンプレートからオブジェクトを作成するときに、「テンプレートのメンバ」属性を保持しているかどうかを調べます。 ◆ **［いいえ］：**「テンプレートのメンバ」属性を追跡しません。

3 属性を選択します。
4 選択した属性のフィルタ設定を変更します。

オプション	定義
発行者	◆ **［同期］：** このオブジェクトに対する変更は、レポートされ、自動的に同期化されます。 ◆ **［無視］：** このオブジェクトに対する変更は、レポートも自動的な同期化もされません。 ◆ **［通知］：** このオブジェクトに対する変更はレポートされますが、自動的には同期化されません。 ◆ **［リセット］：** オブジェクト値を、もう一方のチャネルで指定された値にリセットします (この値は、発行者と購読者の一方のチャネルだけに設定できます。両方には設定できません)。
購読者	◆ **［同期］：** このオブジェクトに対する変更は、レポートされ、自動的に同期化されます。 ◆ **［無視］：** このオブジェクトに対する変更は、レポートも自動的な同期化もされません。 ◆ **［通知］：** このオブジェクトに対する変更はレポートされますが、自動的には同期化されません。 ◆ **［リセット］：** オブジェクト値を、もう一方のチャネルで指定された値にリセットします (この値は、発行者と購読者の一方のチャネルだけに設定できます。両方には設定できません)。
マージ権限	◆ **［Default Behavior (デフォルトの動作)］：** 属性がどちらのチャネルでも同期されていない場合、マージは行われません。 属性が一方のチャネルだけで同期されている場合は、そのチャネルのターゲットに既存の値がすべて削除され、そのチャネルのソースの値と置き換えられます。ソースに複数の値があるが、ターゲットでは値を 1 つしか受け取れない場合は、値のうち 1 つだけがターゲット側で使用されます。 属性が両方のチャネルで同期され、いずれの側も値を 1 つしか受け取れない場合、接続されているアプリケーションがアイデンティティボールト値を取得します。ただし、アイデンティティボールトに値がない場合を除きます。この場合は、アイデンティティボールトが、接続されているアプリケーションから (存在する場合は) 値を取得します。 属性が両方のチャネルで同期され、一方だけが複数の値を受け入れられる場合は、単一値しか受入れられない方の値が複数値側に存在していない場合に限り、複数値側に追加されます。単一の側に値がない場合は、単一の側に追加する値を選択できます。 これは常に有効な動作です。 ◆ **［アイデンティティボールト］：** 属性が発行者チャネルではなく購読者チャネルで同期されている場合のデフォルトの動作と同じ動作です。 これは、購読者チャネルで同期されている場合に有効な動作です。 ◆ **［アプリケーション］：** 属性が購読者チャネルではなく発行者チャネルで同期されている場合のデフォルトの動作と同じ動作です。 これは、発行者チャネルで同期されている場合に有効な動作です。 ◆ ［なし］：同期に関係なく、マージされません。

オプション	定義
Identity Manager に対する変更の最適化	◆ ［はい］：アイデンティティボールトで行われる変更を最小限にするために、発行者チャネルでこの属性の変更を調べます。 ◆ ［いいえ］：変更を調べません。

5 ［OK］をクリックし、変更を保存します。

スキーママッピングポリシーの管理 7

スキーママッピングポリシーは、クラス名および属性名をアイデンティティボールトのネームスペースとアプリケーションのネームスペースとの間でマップします。同じスキーママッピングポリシーが両方向に適用されます。メタディレクトリエンジンとアプリケーションシム間で渡されるドキュメントは、いずれのチャネルでも、またいずれの方向でも、すべてのドキュメントがスキーママッピングポリシーを通過します。

ドライバごとに 1 つのスキーママッピングポリシーがあります。

この節では、次のフィルタ関連のトピックついて説明します。

7.1 Designer におけるスキーママッピングポリシーのタスク

この節では、Designer における一般的なスキーママッピングポリシー関連のタスクを実行する手順を説明します。

7.1.1 スキーママップエディタへのアクセス

スキーママップエディタでは、スキーママッピングポリシーを編集できます。Designer でスキーママップエディタにアクセスする方法には、[Outline (アウトライン)] ビュー、[Policy Flow (ポリシーフロー)] ビュー、または [Policy Set (ポリシーセット)] ビューのそれぞれを使用する 3 つの方法があります。

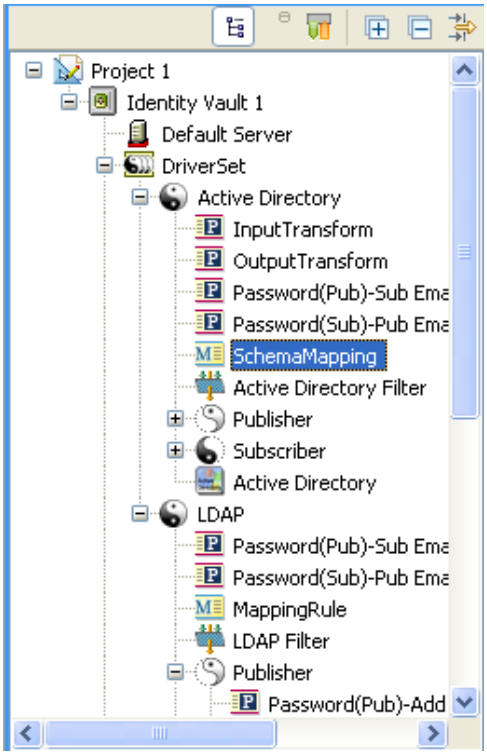
[Outline (アウトライン)] ビュー

1. 開いているプロジェクトで、[Outline (アウトライン)] タブをクリックします。
2. [Show Model Outline (モデルアウトラインの表示)] アイコンをクリックします。
3. スキーママッピングポリシーを管理するドライバを選択し、右側のプラス記号をクリックします。

4 ［Schema Map (スキーママップ)］アイコンをダブルクリックして、スキーママップエディタを起動します。

または

右クリックして、［編集］を選択します。



［Policy Flow (ポリシーフロー)］ビュー

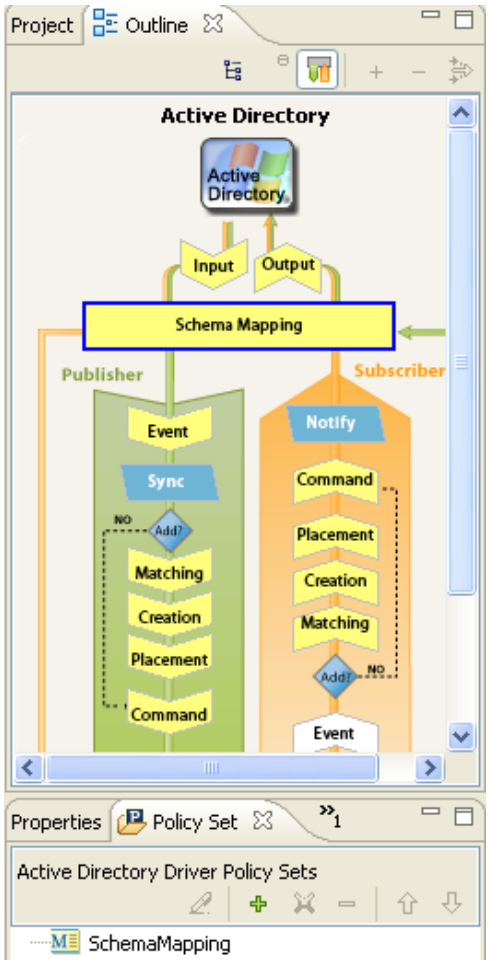
1 開いているプロジェクトで、［Outline (アウトライン)］タブをクリックします。

2 ［Show Policy Flow (ポリシーフローの表示)］アイコン をクリックします。

3 スキーママッピングポリシーをダブルクリックして、スキーママップエディタを起動します。

または

右クリックして［Edit Policy (ポリシーの編集)］を選択して、スキーママップエディタを起動します。

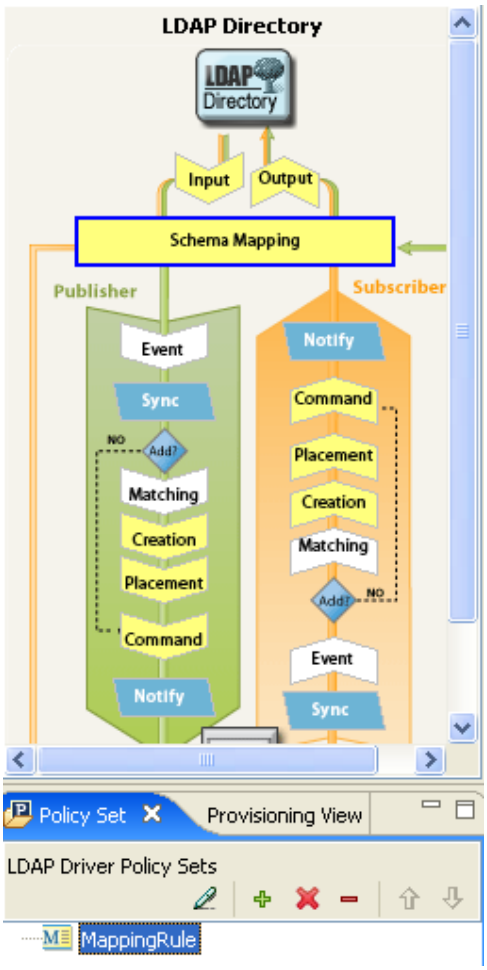


ポリシーセットビュー

1 スキーママップポリシーをダブルクリックして、［Policy Set (ポリシーセット)］ビューを起動します。

または

スキーママップポリシーを右クリックして、［編集］を選択します。



キーボード操作

表 *7-1* スキーママップエディタでのキーボード操作

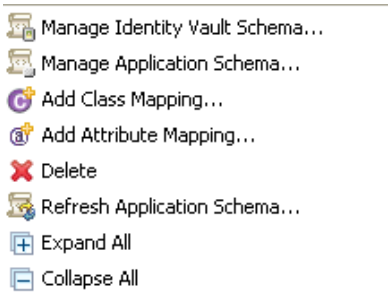
アクション	説明
上矢印	スキーママップエディタ内でカーソルを上方向に移動します。
下矢印	スキーママップエディタ内でカーソルを下方向に移動します。
左矢印	表示されている情報を縮小表示します
右矢印	表示されている情報を展開します。
Insert	クラスを追加します。
Ctrl+Insert	属性を追加します。
Delete	選択された項目を削除します。

アクション	説明
Enter	編集モードに切り替えます。変更を確定するには、Enter を 2 回押します。
Esc	編集モードを終了します。

7.1.2 スキーママッピングポリシーの編集

スキーママップエディタでは、スキーママッピングポリシーを作成および編集できます。コンテキストに応じたメニューを表示するには、項目を右クリックします。

図 7-1 スキーママップエディタのコンテキストに応じたメニュー



- 427 ページの 「クラスと属性の削除または追加」
- 429 ページの 「アプリケーションスキーマのリフレッシュ」
- 429 ページの 「項目の編集」
- 429 ページの 「項目のソート」
- 430 ページの 「スキーマの管理」

クラスと属性の削除または追加

- 427 ページの 「クラスまたは属性の削除」
- 428 ページの 「クラスの追加」
- 428 ページの 「属性の追加」

クラスまたは属性の削除

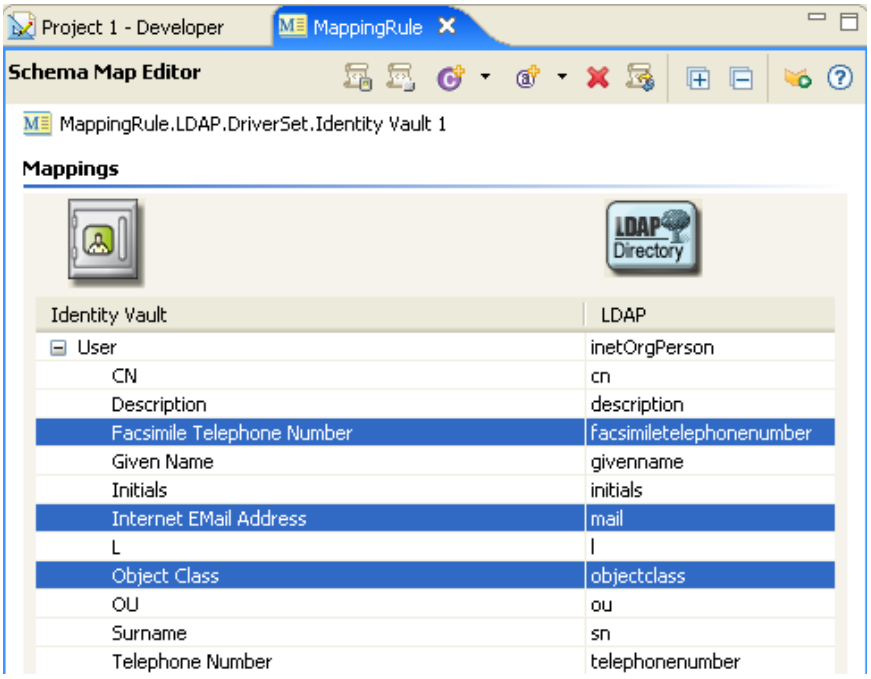
クラスまたは属性を接続システム上のクラスまたは属性にマップしないようにする場合、クラスまたは属性をスキーママッピングポリシーから完全に削除するのが最善の方法です。スキーママッピングポリシーで属性とクラスを追加または削除するには、次の 3 つの方法があります。

- 削除するクラスまたは属性を選択して右クリックし、[削除] をクリックします。
- 削除するクラスまたは属性を選択し、右上隅の [削除] アイコン ✖ をクリックします。
- 削除するクラスまたは属性を選択し、Delete キーを押します。

複数のクラスまたは属性を選択して、一度に削除することができます。

1 キーを押しながら、各項目をマウスで選択します。

2 Delete キーを押して項目を削除します。



クラスの追加

1 スキーママップエディタ内で右クリックし、[Add Class Mapping (クラスマッピングの追加)] をクリックします。

または

右上隅の [Add Class Mapping (クラスマッピングの追加)] アイコン をクリックします。

2 アイデンティティボールトのドロップダウンリストで、追加するクラスを選択します。

3 接続システムのドロップダウンリストで、追加するクラスを選択します。

4 変更を保存するため、[ファイル] > [保存] の順にクリックします。

属性の追加

1 スキーママップエディタ内で右クリックし、[Add Attribute Mapping (属性マッピングの追加)] をクリックします。

または

右上隅の [Add Attribute Mapping (属性マッピングの追加)] アイコン をクリックします。

2 アイデンティティボールトのドロップダウンリストで、追加する属性を選択します。

3 接続システムのドロップダウンリストで、追加する属性を選択します。

4 変更を保存するため、[ファイル] > [保存] の順にクリックします。

アプリケーションスキーマのリフレッシュ

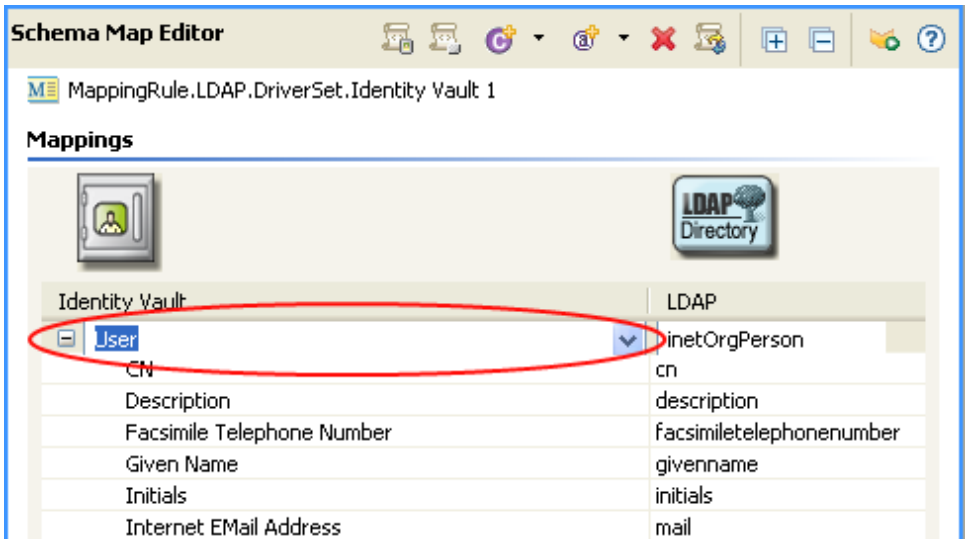
アプリケーションでスキーマを変更している場合は、それらの変更をスキーママッピングポリシーに反映させる必要があります。新しいスキーマを使用できるようにするには、ツールバーの［アプリケーションスキーマのリフレッシュ］アイコン をクリックします。

新しいクラスまたは属性のマッピングを作成する場合は、接続されているアプリケーションのドロップダウンリストに新しいスキーマを表示できます。

項目の編集

マッピングを編集するには、選択した行をダブルクリックします。インプレースエディタが表示されるので、マッピングを編集できます。

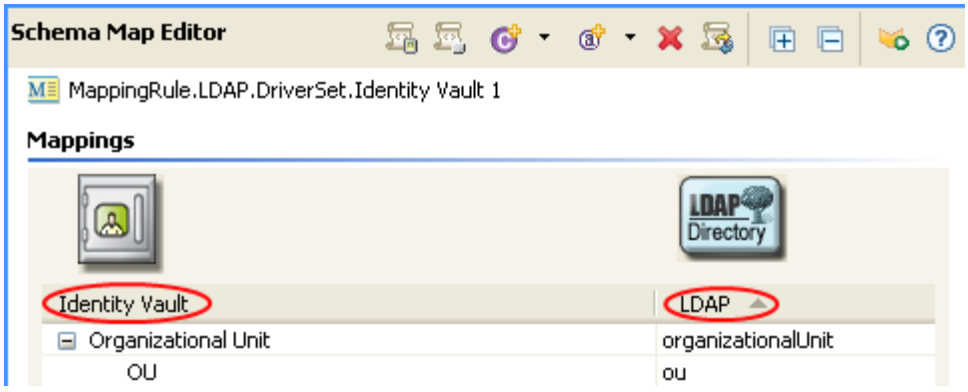
図 7-2 スキーママップエディタ



項目のソート

スキーマエディタでは、Identity Manager または接続システムに基づいて、項目を昇順にソートできます。ソートするには、いずれかの列見出しをクリックします。

図 7-3 スキーママップエディタでの項目のソート



スキーマの管理

Designer では、アイデンティティボールトスキーマおよび任意の接続システムのスキーマを管理できます。スキーマをインポートして変更し、それをアイデンティティボールトまたは接続システムに戻すことができます。アイデンティティボールトスキーマを管理するには、スキーママップエディタ内を右クリックし、[Manage Identity Vault Schema (アイデンティティボールトスキーマの管理)] をクリックします。接続システムのスキーマを管理するには、スキーママップエディタ内を右クリックし、[Manage Application Schema (アプリケーションスキーマの管理)] をクリックします。スキーマの管理方法については、『Designer for Identity Manager 3: 管理ガイド』の「“ スキーマの管理 ”」を参照してください。

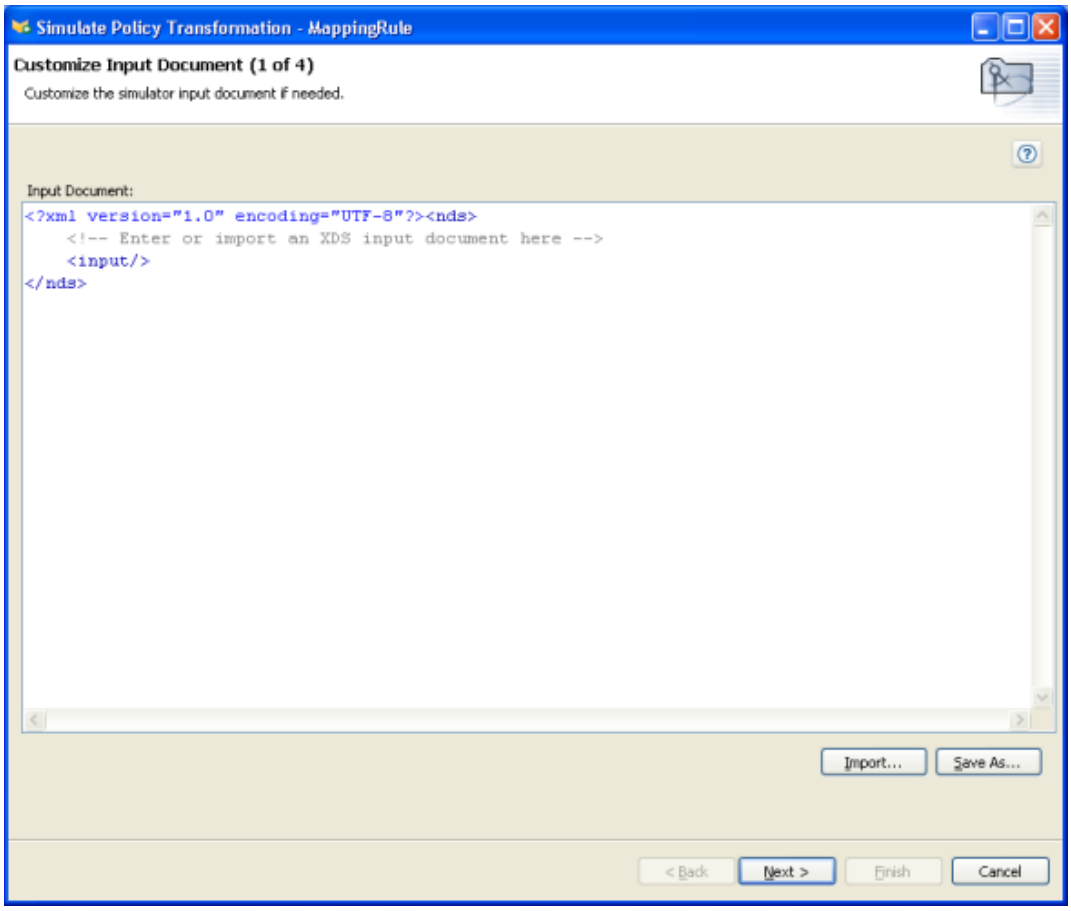
7.1.3 スキーママッピングポリシーのテスト

Designer には、ポリシーシミュレータと呼ばれるツールが付属しています。このツールを使用すると、運用環境に実装しなくてもポリシーをテストできます。スキーママッピングエディタからポリシーシミュレータを起動して、変更後のポリシーをテストできます。

ポリシーシミュレータにアクセスしてスキーママッピングポリシーをテストするには

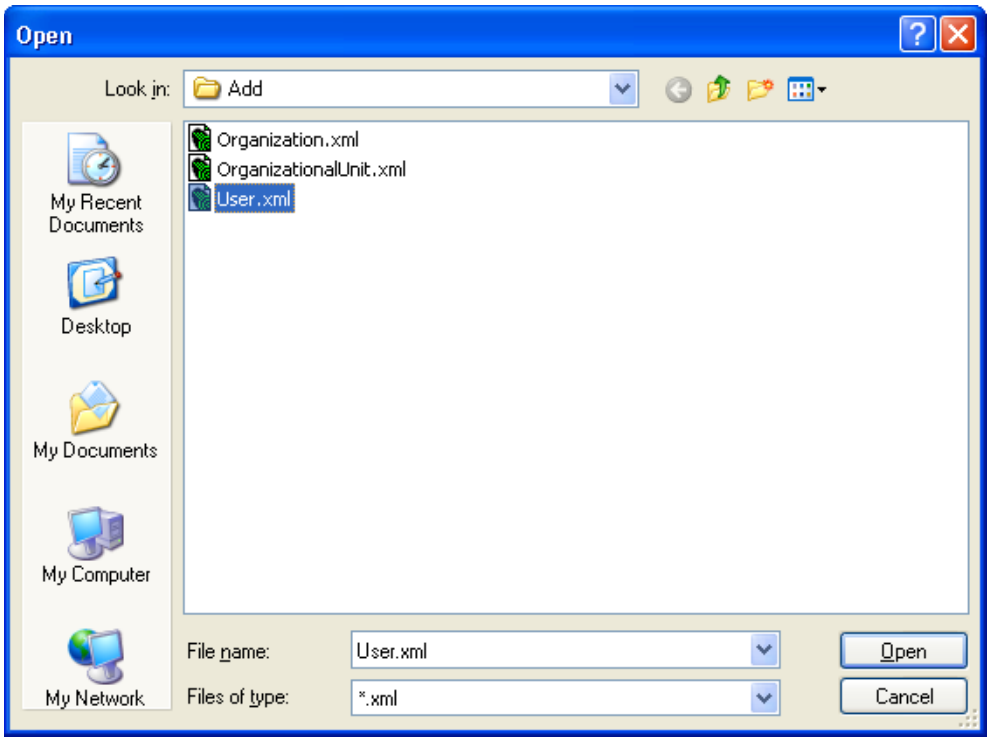
1 ツールバーの [Launch Policy Simulator (ポリシーシミュレータの起動)] アイコン をクリックします。

2 ［Import (インポート)］を選択し、イベントをシミュレートするファイルを参照して選択します。



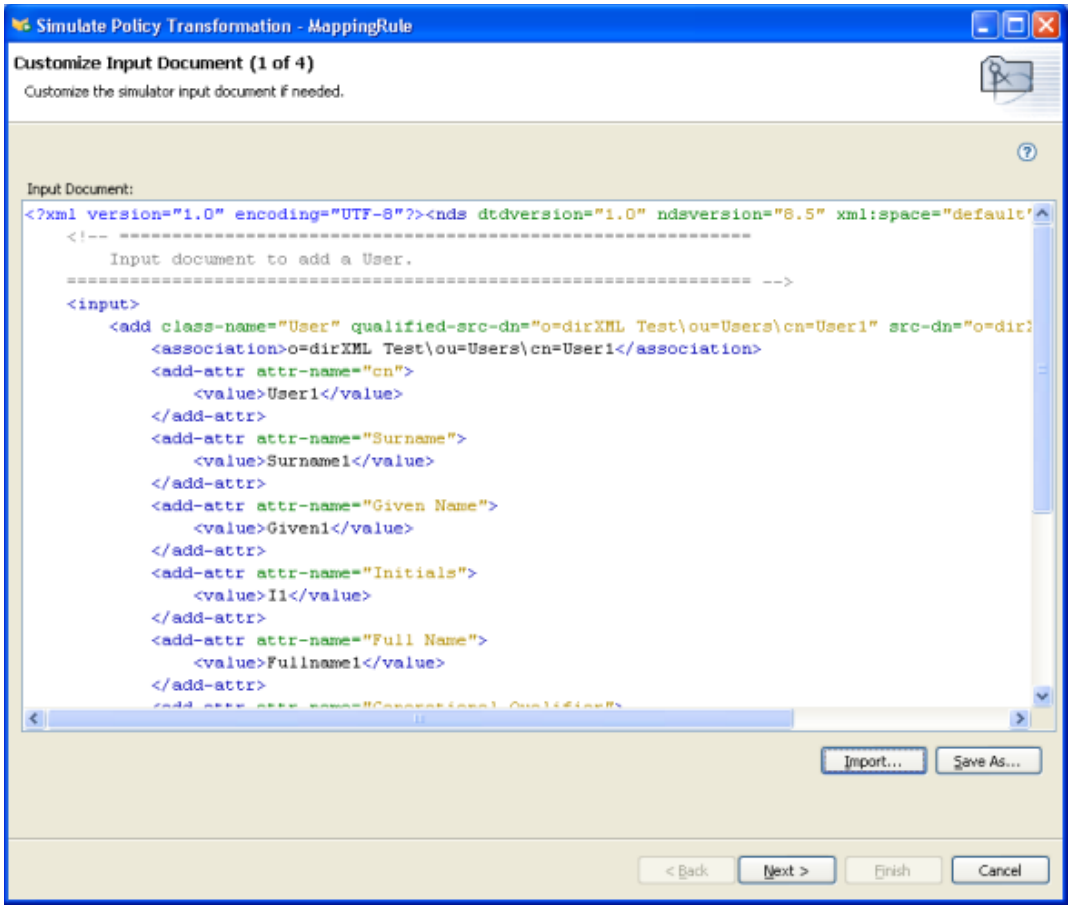
3 ファイルを選択して、［開く］をクリックします。

この例では、com.novell.designer.policy\simulation\add\user.xml ファイルを使用して、ユーザオブジェクトの「追加」イベントをシミュレートします。

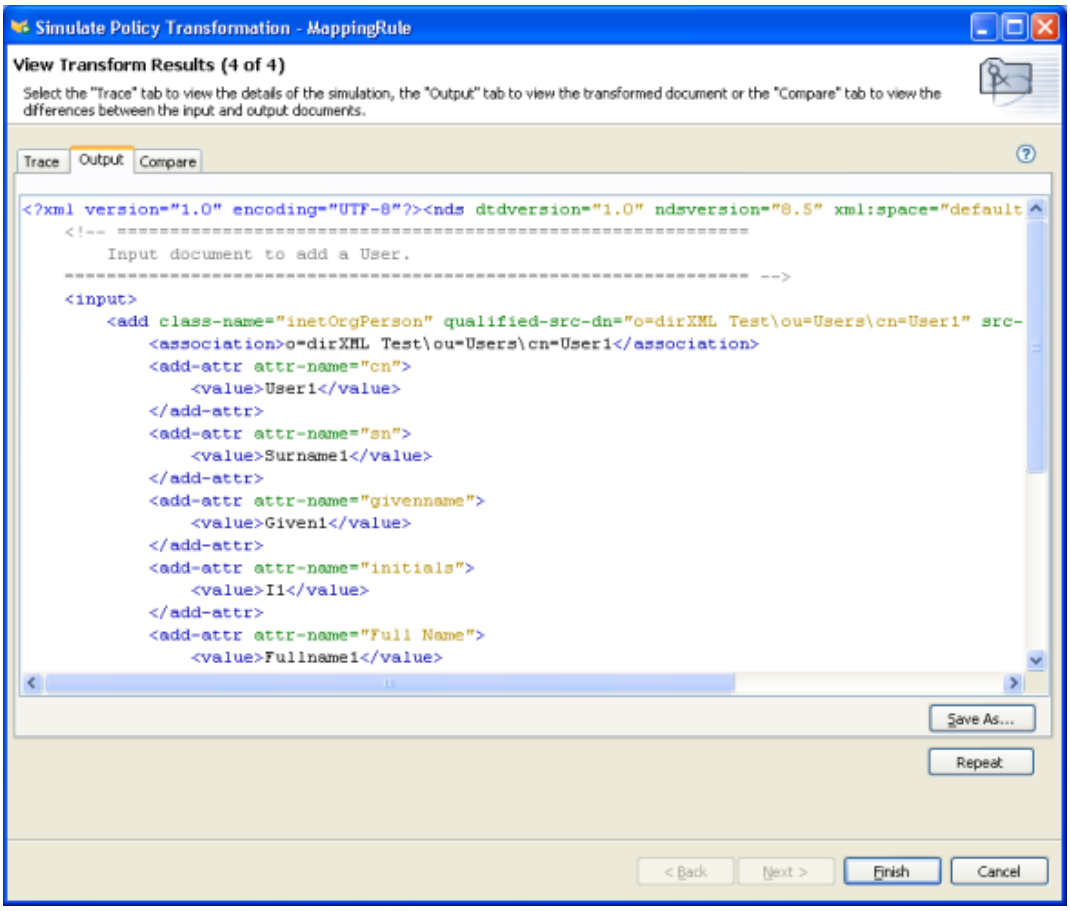


ポリシーシミュレータに、ユーザの「追加」イベントの入力ドキュメントが表示されます。

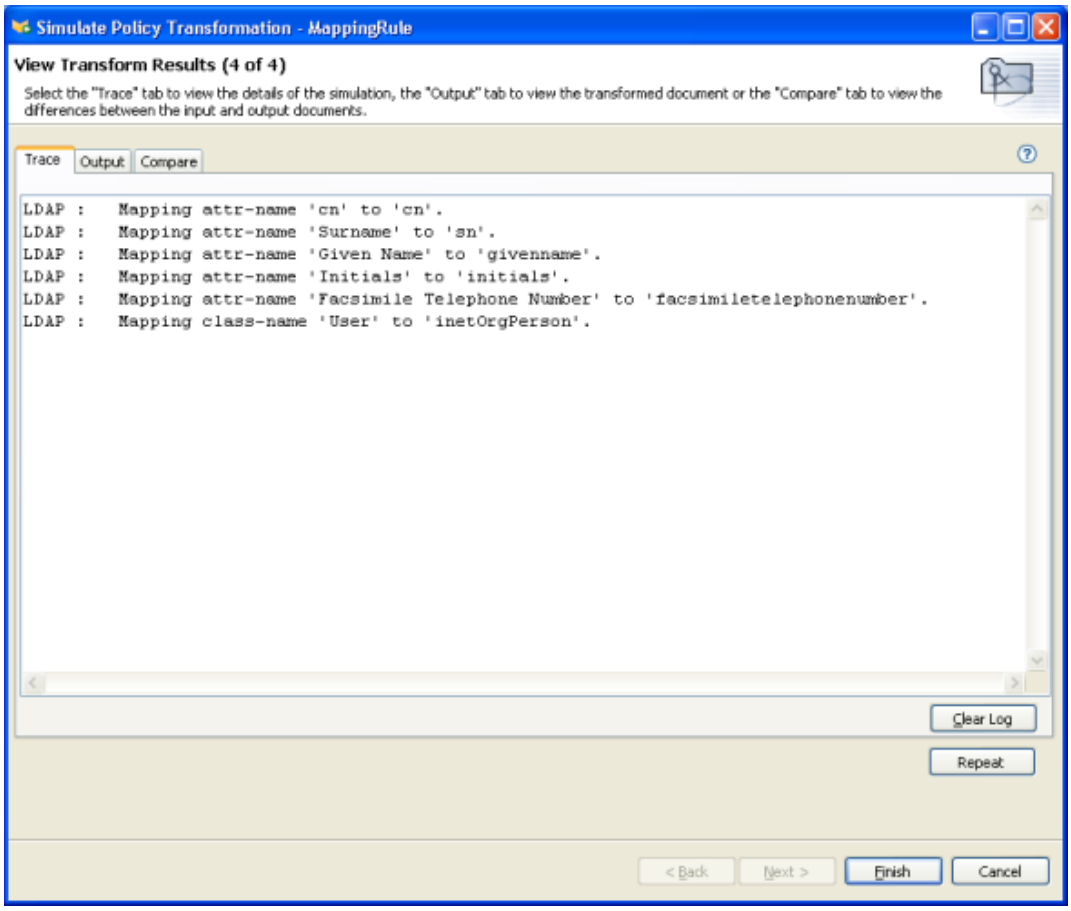
4［次へ］をクリックして、シミュレーションを開始します。



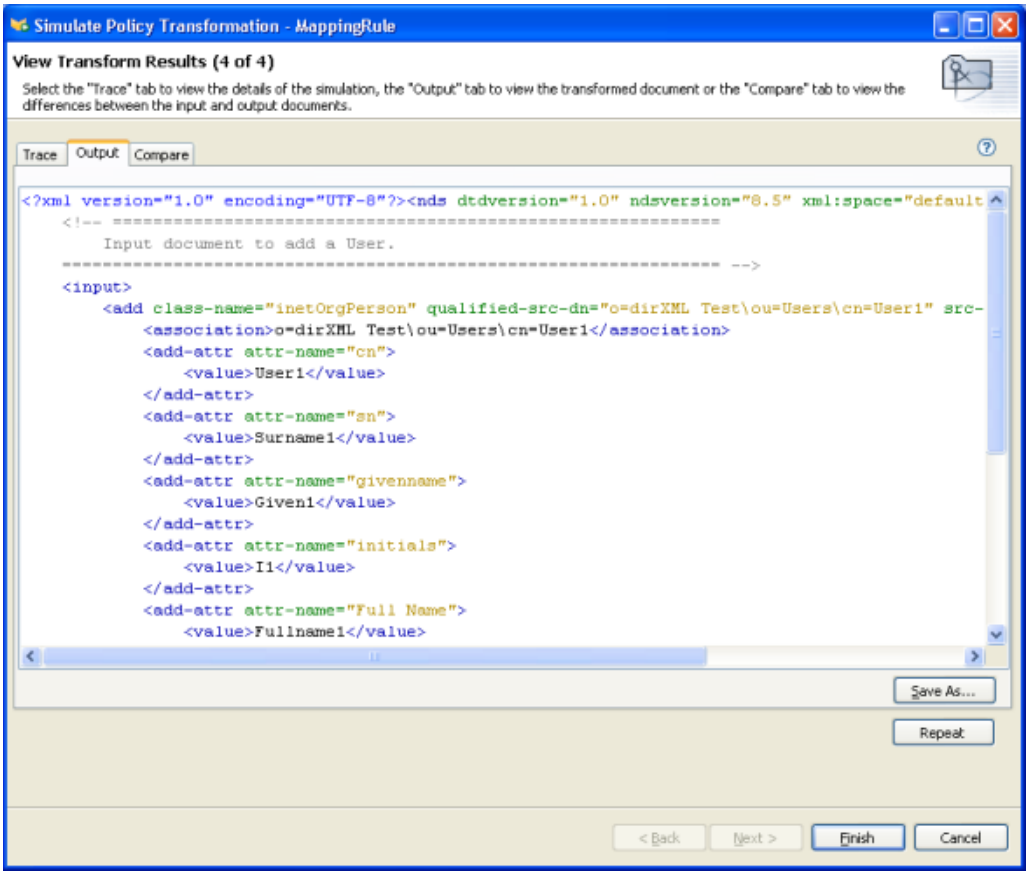
ポリシーシミュレータに、追加イベントのログ、出力ドキュメント、および入力ドキュメントと生成された出力ドキュメントの比較が表示されます。



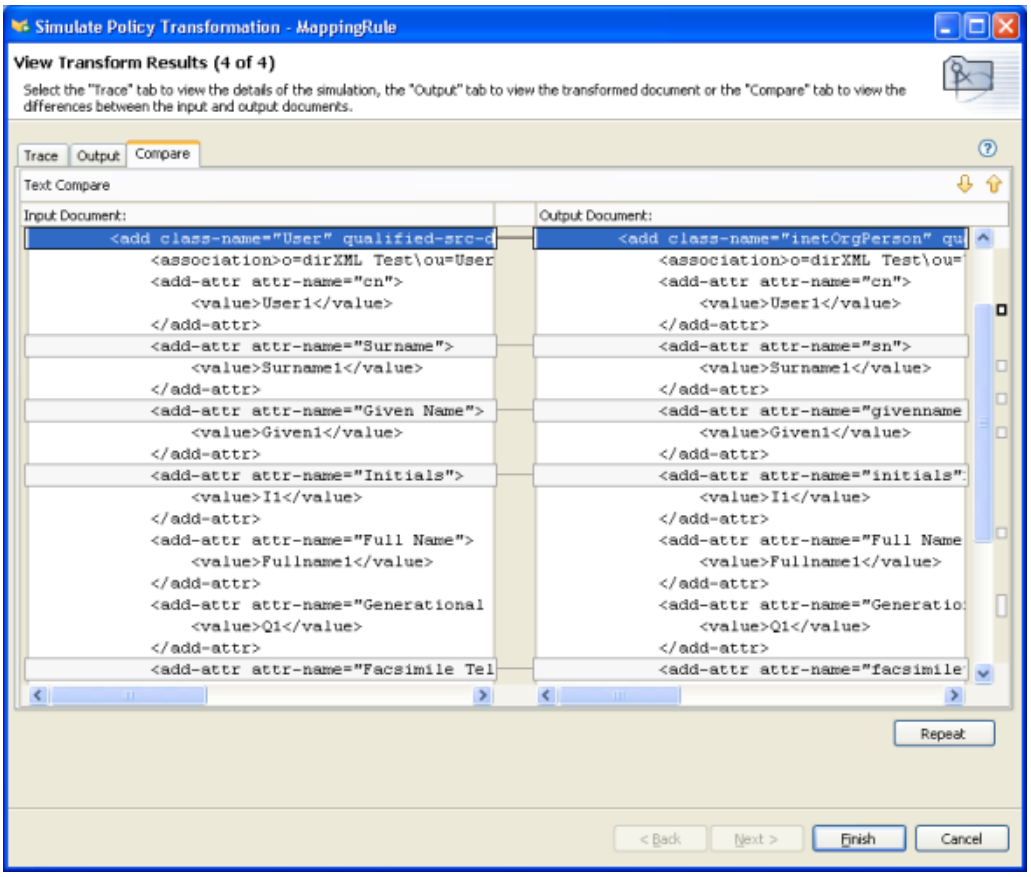
5 ［トレース］タブを選択して、DSTRACE に表示されるとおりの追加イベントの結果を表示します。



6 ［出力］タブを選択し、入力ドキュメントに対して実行されるスキーママップポリシーから生成される出力ドキュメントを表示します。この例では、ユーザの追加イベントです。



7 [Compare (比較)] タブを選択して、入力ドキュメントのテキストと、生成されたドキュメント (出力ドキュメント) を比較します。



8 [Repeat (繰り返し)] をクリックして、別の入力ドキュメントを選択し、イベントの結果を表示します。

9 スキーママッピングポリシーのテストが済んだら、[終了] をクリックして、ポリシーシミュレータを閉じます。

7.1.4 スキーママッピングポリシー XML へのアクセス

Designer では、XML エディタまたはテキストエディタを使用して、XML を表示、編集、および検証できます。

- 437 ページの 「XML ソースの表示」
- 441 ページの 「XML ソースの編集」
- 443 ページの 「XML ソースの検証」

XML ソースの表示

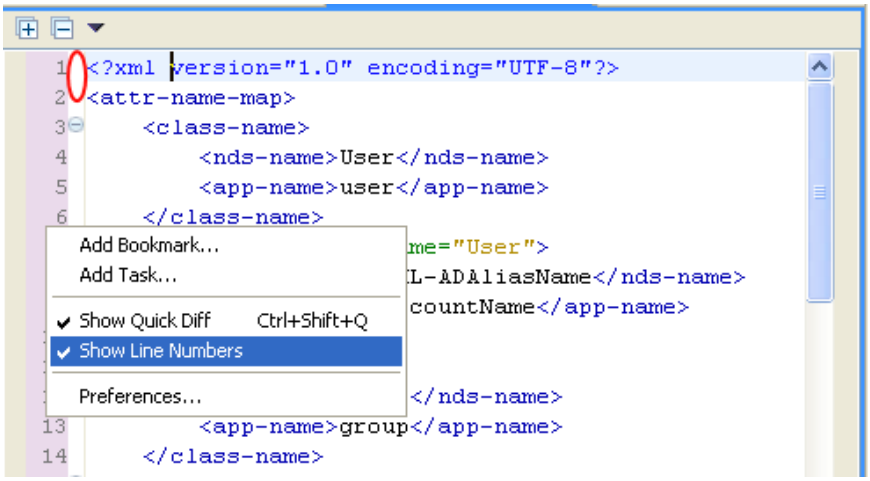
XML ソースは、XML 形式または XML ツリー形式で表示できます。

XML ソースビューを開くには

1 スキーママップエディタのワークスペースの下部にある［> XML Source (XML ソース)］をクリックします。

XML エディタに行番号が表示されます。行番号を表示するには、左の余白を右クリックし、［Show Line Numbers (行番号の表示)］を選択します。

図 7-4 スキーママップポリシーの行番号



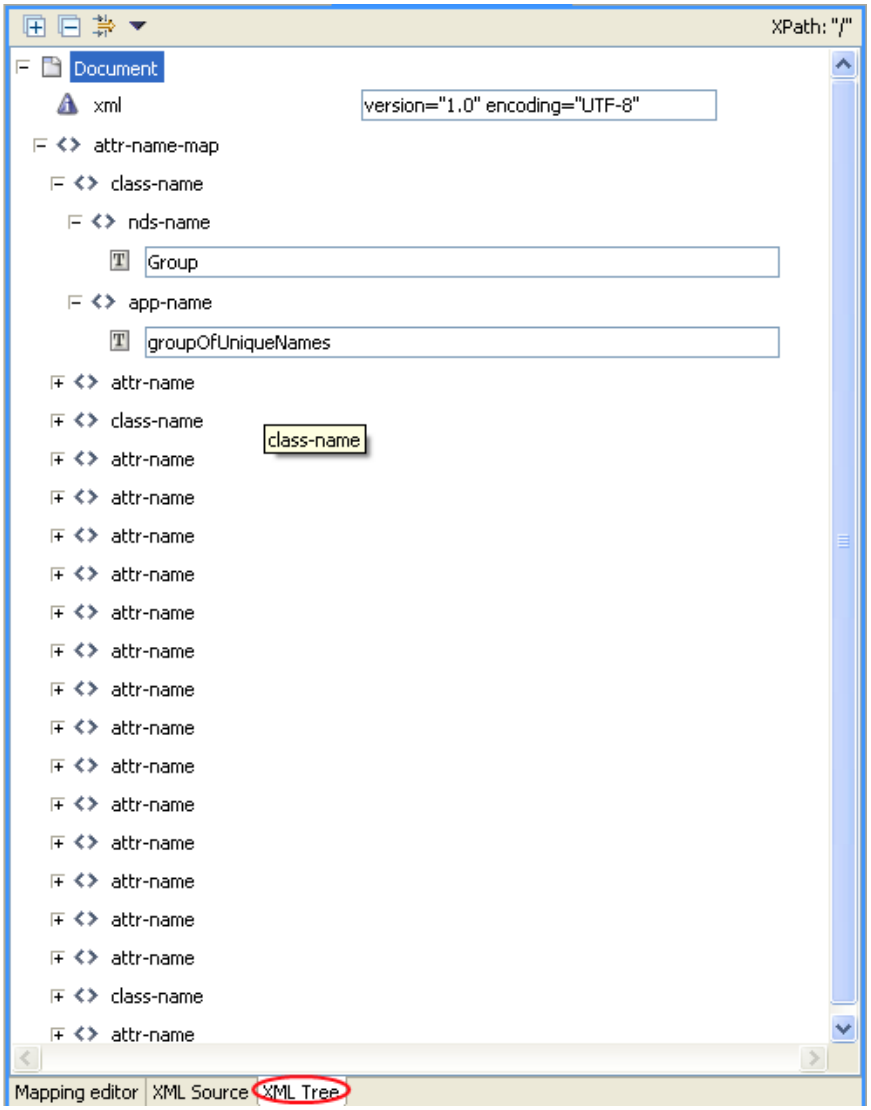
XML エディタは、XML を機能別に展開または縮小します。多くの XML を含む機能が複数ある場合は、左上隅のマイナスアイコンをクリックして、XML を縮小できます。XML 機能をすべて展開するには、左上隅のプラスアイコンをクリックします。

各要素には、左の余白にそれぞれのプラスまたはマイナスアイコンがあります。

図 7-5 スキーママップポリシー *XML* のプラスまたはマイナスアイコン

XML をツリー形式で表示するには

1 スキーママップエディタのワークスペースの下部にある［XML Tree (XML ツリー)］をクリックします。

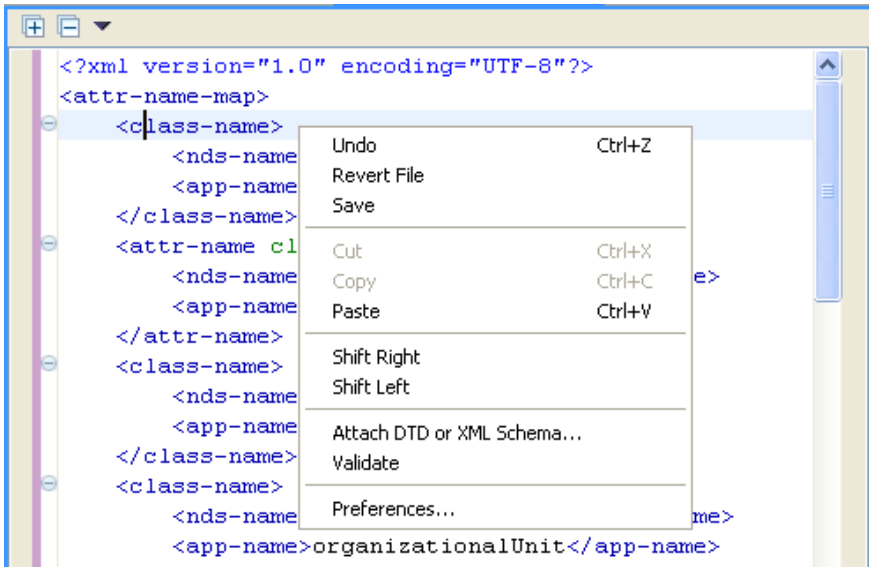


ツリー全体を表示するには、一覧表示されている個々の項目を展開します。

XML ソースの編集

XML は XML エディタで編集できます。GUI を使用する場合と同様、XML エディタで変更することもできます。

図 7-6 スキーママップポリシーの *XML* ソースの編集



ロードされるデフォルトエディタは、.xml ファイルのタイプに関連付けられています。デフォルトエディタが見つからない場合は、システムのテキストエディタがロードされます。XML ソースビューの機能は、ロードされるエディタに基づきます。

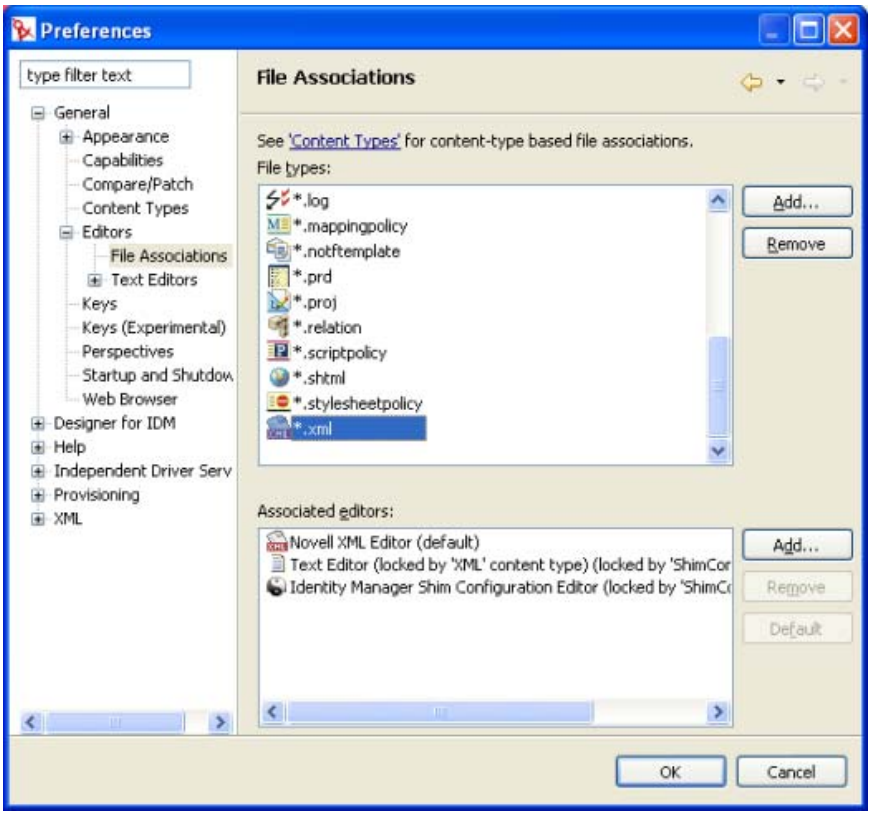
右クリックすると、XML エディタに含まれる機能のリストが表示されます。

- ［元に戻す］：最後のアクションを元に戻します。
- ［**Revert File (** ファイルを戻す **)**］：ファイルを、保存されていたバージョンに戻します。
- ［保存］：ファイルを保存します。
- ［切り取り］：選択された情報を切り取ります。
- ［貼り付け］：情報をドキュメントに貼り付けます。
- ［**Shift Right (** 右にシフト **)**］：行を右にインデントします。
- ［**Shift Left (** 左にシフト **)**］：行を左にインデントします。
- ［**Attach DTD or XML Schema (DTD** または **XML** スキーマを添付 **)**］：ポリシーの検証のために、DTD または XML スキーマファイルを添付します。
- ［検証］：XML コードを検証します。
- ［初期設定］：XML エディタの初期設定を指定します。

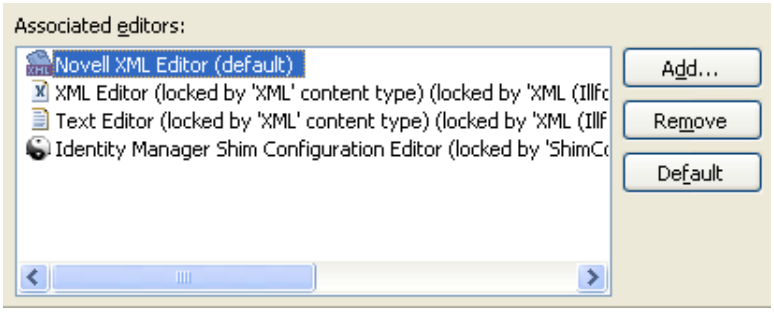
XML ソースビュー用に、別の XML エディタを選択するには

1. メインメニューの［Window (ウィンドウ)］>［初期設定］の順にクリックします。
2. ［一般］>［エディタ］>［ファイルの関連付け］の順にクリックします。

3 ファイルタイプのリストから［*.xml］を選択します。



4 ［Associated editors (関連付けられているエディタ)］で、エディタ (たとえば、［Novell XML Editor (Novell XML エディタ)］) を選択します。適切なエディタがリストにない場合は、［追加］をクリックしてリストに追加できます。



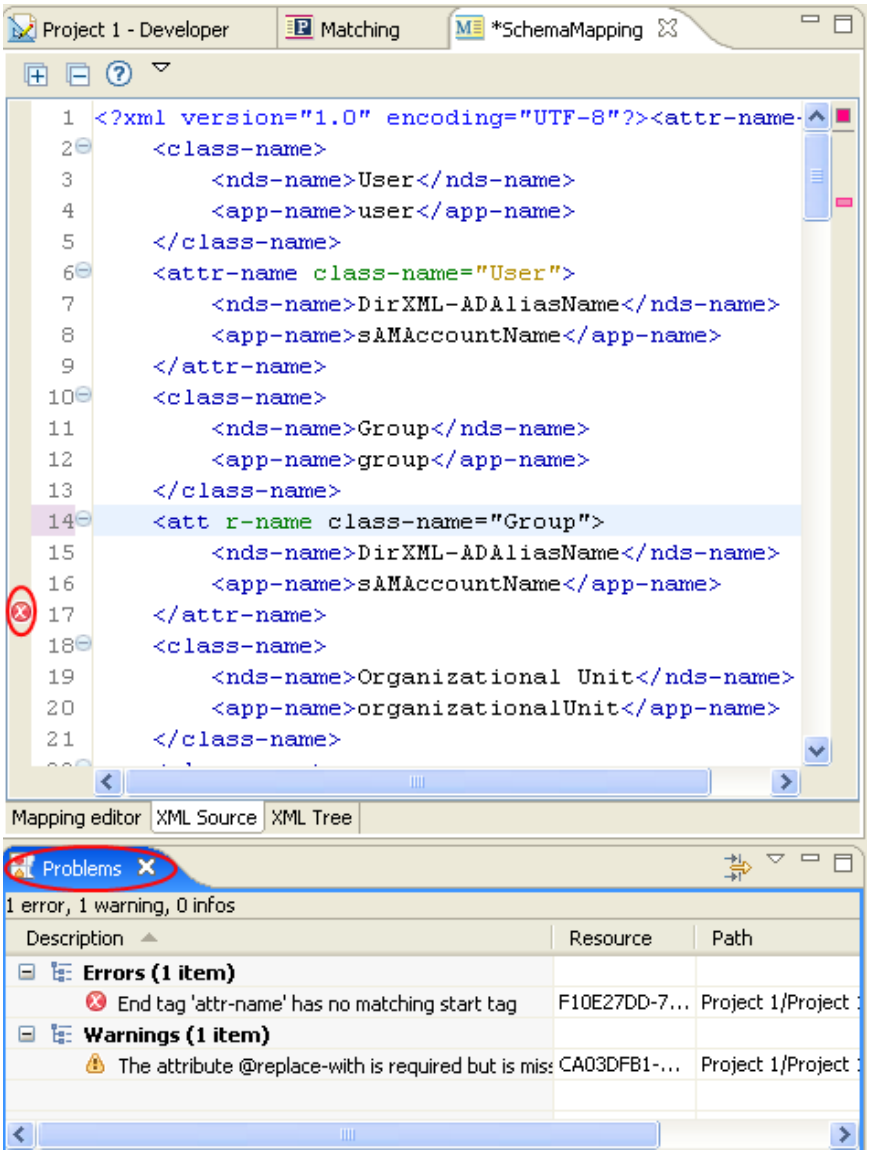
5 ［OK］をクリックします。

6 スキーママップエディタをいったん閉じて再度開きます。XML ソースビューにデフォルトエディタがロードされます。

XML ソースの検証

XML エディタは、XML コードを検証します。右クリックし、[検証] を選択します。エラーがある場合は、その行に赤の「x」が表示されます。ウィンドウの下部の説明に、問題についての詳しい情報が示されます。

図 7-7 スキーママップポリシーの検証



この例では、<attr-name> の終了タグに対応する開始タグがありません。

7.1.5 追加のスキーママップポリシーオプション

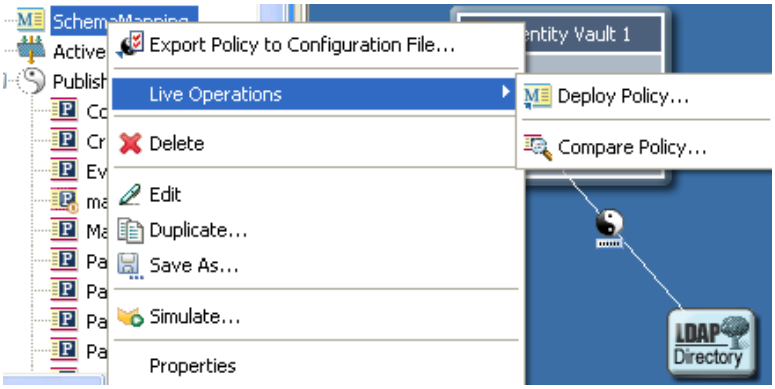
スキーママップポリシーを右クリックすると、[Outline (アウトライン)] ビュー、[Policy Flow (ポリシーフロー)] ビュー、および [Policy Set (ポリシーセット)] ビューで、複数のオプションが表示されます。

- 444 ページの 「[Outline (アウトライン)] ビューの追加オプション」

- 445 ページの 「[Policy Flow (ポリシーフロー)] の追加オプション」
- 447 ページの 「[Policy Set (ポリシーセット)] ビューの追加オプション」

[Outline (アウトライン)] ビューの追加オプション

1 [Outline (アウトライン)] ビューでスキーママップポリシーを右クリックします。

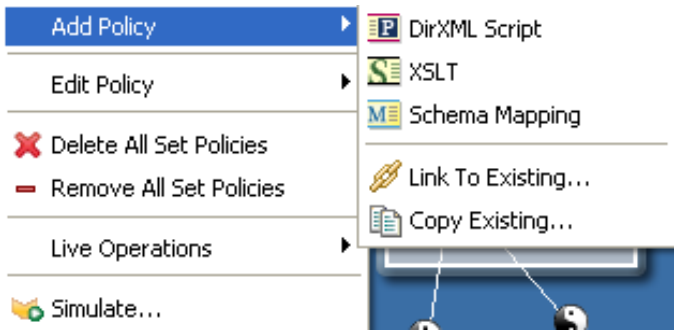


- [**Export Policy to Configuration File (** 環境設定ファイルへのポリシーのエクスポート **)**]：スキーママップポリシーを .xml ファイルとして保存します。
- [**Live Operations (** ライブ操作 **)**] > [**Deploy Policy (** ポリシーの展開 **)**]：アイデンティティボールトにスキーママップポリシーを展開します。
- [**Live Operations (** ライブ操作 **)**] > [**Compare Policy (** ポリシーの比較 **)**]：

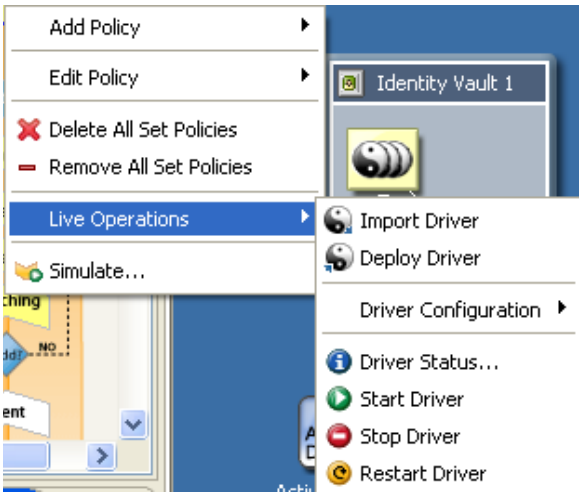
  Designer のスキーママップポリシーとアイデンティティボールトのスキーママップポリシーを比較します。
- [削除]：スキーママップポリシーを削除します。
- [編集]：スキーママップエディタを起動します。詳細については、427 ページのセクション 7.1.2「スキーママッピングポリシーの編集」を参照してください。
- [**Duplicate (** 複製 **)**]：スキーママップポリシーのコピーを作成します。
- [名前を付けて保存]：スキーママップポリシーを .xml ファイルとして保存します。
- [**Simulate (** シミュレート **)**]：スキーママップポリシーをテストします。詳細については、430 ページのセクション 7.1.3「スキーママッピングポリシーのテスト」を参照してください。
- [プロパティ]：スキーママップポリシーの名前を変更できます。

［Policy Flow (ポリシーフロー)］の追加オプション

1 ［Policy Flow (ポリシーフロー)］ビューでスキーママップポリシーを右クリックします。



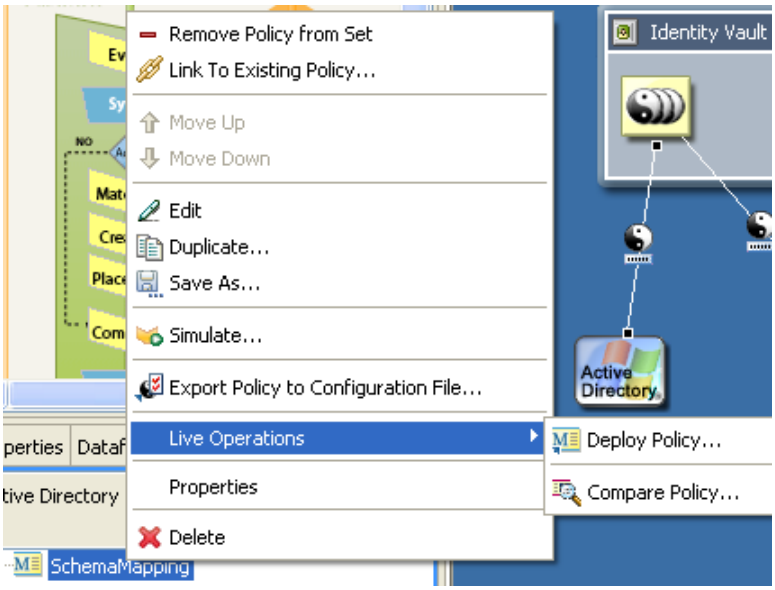
- ［**Add Policy (** ポリシーの追加 **)**］> ［**DirXML** スクリプト］: DirXML® スクリプトを使用して、新しいスキーママップポリシーを追加します。
- ［**Add Policy (** ポリシーの追加 **)**］> ［**XSLT**］: XSLT を使用して、新しいスキーママップポリシーを追加します。
- ［**Add Policy (** ポリシーの追加 **)**］> ［スキーマのマッピング］: 情報を含まない新しいスキーママップポリシーを追加します。
- ［**Add Policy (** ポリシーの追加 **)**］> ［**Link to Existing (** 既存へのリンク **)**］: 既存のスキーママップポリシーを参照して選択し、現在のスキーママップポリシーにリンクできます。
- ［**Add Policy (** ポリシーの追加 **)**］> ［**Copy Existing (** 既存のコピー **)**］: 既存のスキーママップポリシーを参照して選択し、現在のスキーママップポリシーにコピーできます。
- ［**Edit Policy (** ポリシーの編集 **)**］> ［スキーマのマッピング］: スキーママップエディタを起動します。詳細については、448 ページのセクション 7.2.2「スキーママッピングポリシーの編集」を参照してください。
- ［**Delete All Set Policies (** セット内のすべてのポリシーの削除 **)**］: 選択したポリシーセット内のすべてのポリシーを削除します。
- ［**Remove All Set Policies (** セット内のすべてのポリシーの除外 **)**］: 選択したポリシーセットからすべてのポリシーを除外しますが、既存のポリシーの削除はしません。



- ［**Live Operations (** ライブ操作 **)**］>［**Import Driver (** ドライバのインポート **)**］：アイデンティティボールトから既存のドライバをインポートします。
- ［**Live Operations (** ライブ操作 **)**］>［**Deploy Driver (** ドライバの展開 **)**］：アイデンティティボールトに既存のドライバを展開します。
- ［**Live Operations (** ライブ操作 **)**］>［ドライバ環境設定］>［**Import Attributes (** 属性のインポート **)**］：アイデンティティボールトから属性をインポートし、それを Designer 内の属性と比較できます。
- ［**Live Operations (** ライブ操作 **)**］>［ドライバ環境設定］>［**Import Attributes (** 属性の展開 **)**］：Designer からアイデンティティボールトに属性を展開し、Designer 内の属性とアイデンティティボールト内の属性を比較します。
- ［**Live Operations (** ライブ操作 **)**］>［**Driver Status (** ドライバステータス **)**］：ドライバのステータスを表示します。
- ［**Live Operations (** ライブ操作 **)**］>［ドライバの起動］：ドライバを起動します。
- ［**Live Operations (** ライブ操作 **)**］>［ドライバの停止］：ドライバを停止します。
- ［**Live Operations (** ライブ操作 **)**］>［ドライバの再起動］：ドライバを再起動します。
- ［**Simulate (** シミュレート **)**］：スキーママップポリシーをテストします。詳細については、430 ページのセクション 7.1.3「スキーママッピングポリシーのテスト」を参照してください。

［Policy Set (ポリシーセット)］ビューの追加オプション

1 ［Policy Set (ポリシーセット)］ビューでスキーママップポリシーを右クリックします。



- ［**Remove Policy from Set (** セットからポリシーを除外 **)**］：ポリシーセットからスキーママップポリシーを除外しますが、そのポリシー自体は削除しません。
- ［**Link to Existing Policy (** 既存のポリシーへのリンク **)**］：別のスキーママップポリシーを参照して選択し、既存のポリシーにリンクできます。
- ［上へ移動］：ポリシーの実行順で、スキーママップポリシーを上に移動します。
- ［下に移動］：ポリシーの実行順で、スキーママップポリシーを下に移動します。
- ［編集］：スキーママップエディタを起動します。詳細については、448 ページのセクション 7.2.2「スキーママッピングポリシーの編集」を参照してください。
- ［**Duplicate (** 複製 **)**］：スキーママップポリシーのコピーを作成します。
- ［名前を付けて保存］：スキーママップポリシーを .xml ファイルとして保存します。
- ［**Simulate (** シミュレート **)**］：スキーママップポリシーをテストします。詳細については、430 ページのセクション 7.1.3「スキーママッピングポリシーのテスト」を参照してください。
- ［**Export Policy to Configuration File (** 環境設定ファイルへのポリシーのエクスポート **)**］：スキーママップポリシーを .xml ファイルとして保存します。
- ［**Live Operations (** ライブ操作 **)**］> ［**Deploy the Policy (** ポリシーの展開 **)**］：アイデンティティボールトにスキーママップポリシーを展開します。
- ［**Live Operations (** ライブ操作 **)**］> ［**Compare Policy (** ポリシーの比較 **)**］：

  Designer のスキーママップポリシーとアイデンティティボールトのスキーママップポリシーを比較します。
- ［プロパティ］：スキーママップポリシーの名前を変更できます。
- ［削除］：スキーママップポリシーを削除します。

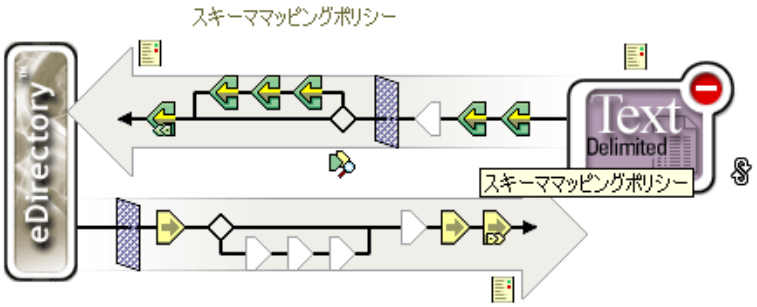
7.2 iManager におけるスキーママッピングポリシーのタスク

この節では、iManager におけるスキーママッピングポリシー関連のタスクを実行する手順を説明します。

- 448 ページのセクション 7.2.1「スキーママッピングポリシーへのアクセス」
- 448 ページのセクション 7.2.2「スキーママッピングポリシーの編集」

7.2.1 スキーママッピングポリシーへのアクセス

1. iManager で、［Identity Management］役割を展開し、［Identity Manager の概要］をクリックします。
2. ドライバセットで、［ツリー全体を検索する］、または［次のコンテナ内を検索する］を選択し、［検索］をクリックします。
3. スキーママッピングポリシーを管理するドライバをクリックします。［Identity Manager ドライバの概要］ページが開きます。



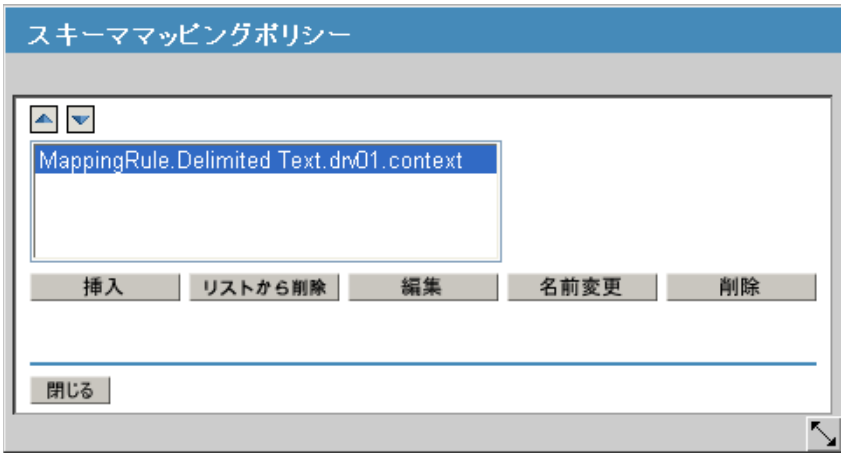
4. スキーママッピングポリシーをクリックします。
5. ［編集］をクリックします。

7.2.2 スキーママッピングポリシーの編集

スキーママッピングポリシーの編集には、2 つの作業があります。1 つ目は、ポリシーセットにあるポリシーの配置の編集です。2 つ目は、スキーママップエディタを使用したポリシー自体の編集です。

ポリシーの配置

スキーママッピングポリシーをクリックすると、オプションを含むウィンドウが表示されます。

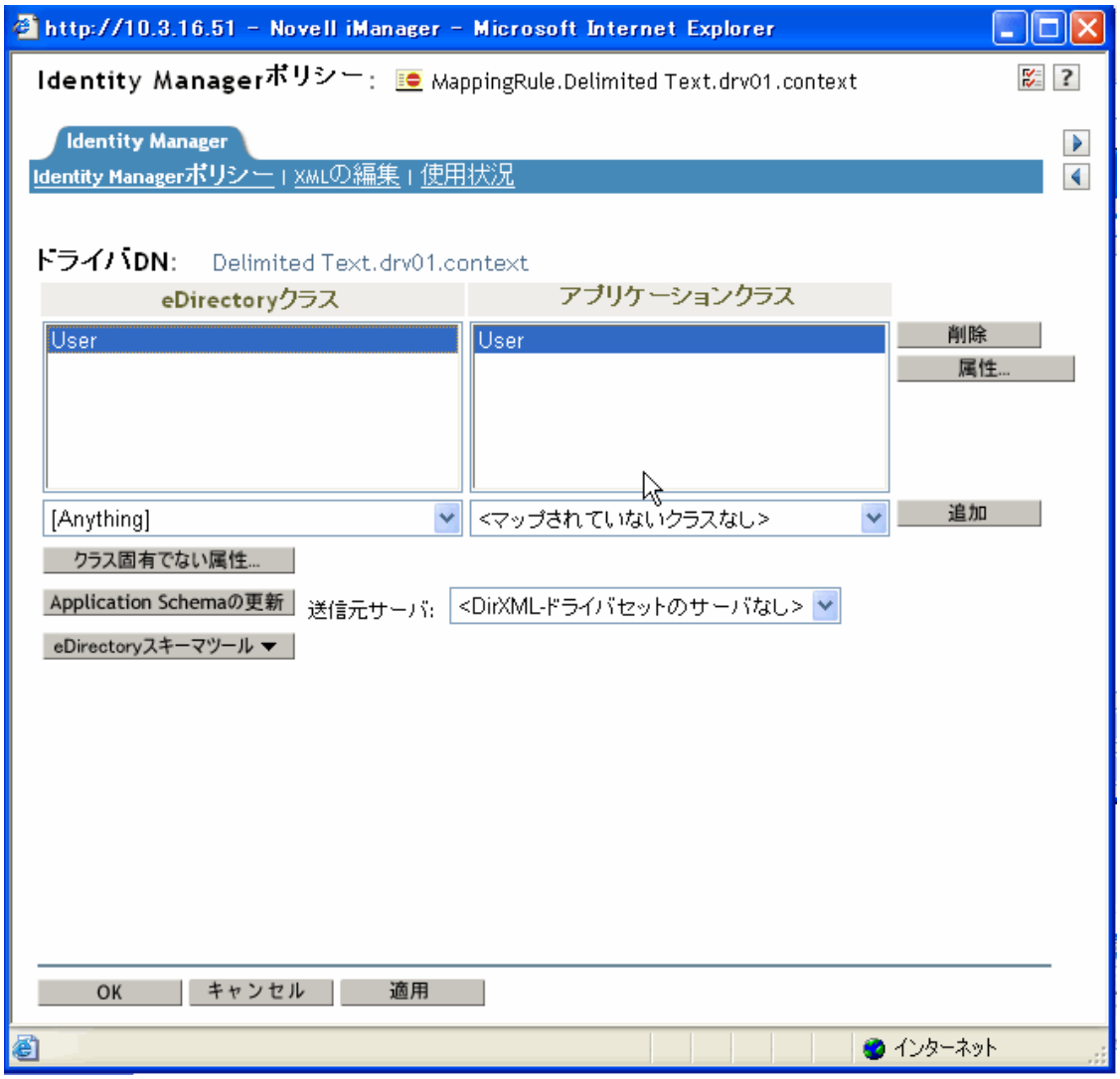


これらのオプションを使用して、現在作業しているポリシーを配置できます。次の表は、各オプションの説明です。

オプション	説明
ポリシーを上位に移動	ポリシーが複数ある場合に、選択したポリシーを上に移動します。
ポリシーを下位に移動	ポリシーが複数ある場合に、選択したポリシーを下に移動します。
挿入	新規または既存のポリシーを一覧表示されているポリシーに挿入します。
Remove (除外)	ポリシーセットから選択したポリシーを除外します。ただし、ポリシー自体は削除しません。
編集	スキーママップエディタを起動します。
名前変更	選択したポリシーの名前を変更します。
削除	選択したポリシーを削除します。

スキーママップエディタ

スキーママップエディタは、スキーママッピングポリシーを作成および管理するための機能を完備したグラフィカルインタフェースです。スキーママップエディタでは、XML を使用してポリシーを作成します。



スキーママップエディタには、次の 3 つのタブがあります。

- 450 ページの 「Identity Manager ポリシー」
- 451 ページの 「XML の編集」
- 452 ページの 「用途」

Identity Manager ポリシー

ほとんどの情報が含まれており、ここで GUI を使用してポリシーを編集します。スキーママップエディタでは、次のタスクを実行できます。

クラスと属性の削除	削除するクラスまたは属性を選択し、［削除］をクリックします。
クラスの追加	ドロップダウンリストから eDirectory クラスを選択し、次にドロップダウンリストからアプリケーションクラスを選択します。項目を選択した状態で、［追加］をクリックし、［適用］をクリックして、変更を保存します。
属性の追加	追加する属性のクラスを選択し、［属性］をクリックします。ドロップダウンリストから eDirectory 属性を選択し、次にドロップダウンリストからアプリケーション属性を選択します。項目を選択した状態で、［追加］をクリックし、［OK］をクリックして、変更を保存します。
クラスに固有でない属性の一覧表示	クラスに関連付けられていない属性がある場合は、［Non-specific Class Attributes (クラスに固有でない属性)］アイコンをクリックして、これらの属性をすべて一覧表示します。
アプリケーションスキーマのリフレッシュ	アプリケーションでスキーマが変更されている場合は、［アプリケーションスキーマのリフレッシュ］アイコンをクリックします。ウィザードが接続システムサーバにアクセスして新しいスキーマを取得します。スキーマは、更新された後にドロップダウンリストに表示されます。
eDirectory スキーマツールの使用	◆ ［属性の追加］: 既存の属性を、選択したクラスに追加します。 ◆ ［属性の作成］: 新しい属性を作成します。 ◆ ［クラスの作成］: 新しいクラスを作成します。 ◆ ［属性の削除］: 選択した属性を削除します。 ◆ ［クラスの削除］: 選択したクラスを削除します。 ◆ ［Refresh eDirectory Schema (eDirectory スキーマのリフレッシュ)］: eDirectory スキーマを変更した後、［Refresh eDirectory Schema (eDirectory スキーマのリフレッシュ)］をクリックすると、ドロップダウンリストが新しい情報で更新されます。

警告 **:** アイデンティティボールトで使用されているクラスまたは属性は削除しないでください。削除すると、オブジェクトが「不明」になります。

XML の編集

［XML 編集の有効化］をクリックすると、DirXML スクリプトポリシーを編集できます。DirXML スクリプトを変更し、［適用］をクリックして、変更を保存します。

用途

このポリシーを現在参照しているドライバのリストを表示します。リストでは、このポリシーのドライバセット内のポリシーだけを参照します。このポリシーが別のドライバセットから参照されていても、その参照はここには表示されません。