

Novell iManager のポリシー

3.5.1

September 28, 2007

IMANAGER のポリシー

www.novell.com



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、すべてのノベル製ソフトウェアについて、いかなる保証、表明または約束も行っていない。またノベル製ソフトウェアの商品性、および特定の目的への適合性について、いかなる黙示の保証も否認し、排除します。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。

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、すべての輸出規制を遵守し、製品の輸出、再輸出、または輸入に必要なすべての許可または等級を取得するものとします。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。ノベル製ソフトウェアの輸出については、「[Novell International Trade Services \(http://www.novell.com/info/exports/\)](http://www.novell.com/info/exports/)」の Web ページをご参照ください。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。

Copyright © 2007 Novell, Inc. All rights reserved. 本書の一部または全体を無断で複製、写真複写、検索システムへの登録、転載することは、その形態を問わず禁止します。

本書に記載された製品で使用されている技術に関連する知的財産権は、弊社に帰属します。これらの知的財産権は、「[Novell Legal Patents \(http://www.novell.com/company/legal/patents/\)](http://www.novell.com/company/legal/patents/)」の Web ページに記載されている 1 つ以上の米国特許、および米国ならびにその他の国における 1 つ以上の特許または出願中の特許を含む場合があります。

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

オンラインヘルプ: 本製品とその他の Novell 製品の最新のオンラインヘルプにアクセスする場合は、「[Novell Documentation \(http://www.novell.com/documentation\)](http://www.novell.com/documentation)」の Web ページをご覧ください。

Novell の商標

Novell の商標一覧については、「[商標とサービスの一覧 \(http://www.novell.com/company/legal/trademarks/tmlist.html\)](http://www.novell.com/company/legal/trademarks/tmlist.html)」を参照してください。

サードパーティ資料

サードパーティの商標は、それぞれの所有者に属します。

目次

このガイドについて	13
1 概要	15
1.1 ポリシー	15
2 ポリシービルダでポリシーを管理する	17
2.1 ポリシービルダへのアクセス	17
2.2 ポリシーの作成	18
2.2.1 ドライバ内にポリシーを作成する	18
2.2.2 ライブラリ内にポリシーを作成する	19
2.3 ポリシー内での各ルールの定義	21
2.4 ルール内で引数を作成する	23
2.5 ポリシーの変更	25
2.6 ポリシーの削除	25
2.7 ポリシーの名前変更	26
2.8 ポリシーの削除	26
2.9 XML ファイルからのポリシーのインポート	26
2.10 XML ファイルへのポリシーのエクスポート	26
2.11 ポリシーの参照の作成	26
3 追加ビルダの使用	29
3.1 引数アクションビルダ	29
3.2 引数ビルダ	30
3.2.1 引数ビルダの <i>Àf</i>	33
3.3 一致属性ビルダ	33
3.3.1 一致属性ビルダのヒント	34
3.4 アクションの引数コンポーネントビルダ	34
3.5 引数値リストビルダ	35
3.5.1 引数値リストビルダのヒント	35
3.6 名前付き文字列ビルダ	35
3.7 条件の引数コンポーネントビルダ	37
4 スキーママッピングポリシーの定義	39
4.1 スキーママッピングポリシーへのアクセス	39
4.2 スキーママッピングポリシーの編集	39
4.2.1 ポリシーの配置	39
4.2.2 スキーママップエディタ	40
5 フィルタを使用したオブジェクトのフローの制御	45
5.1 フィルタへのアクセス	45
5.2 フィルタの編集	45
5.2.1 フィルタからのクラスまたは属性の削除	46
5.2.2 クラスの追加	47
5.2.3 属性の追加	47

5.2.4	フィルタのコピー	47
5.2.5	テンプレートの設定	47
5.2.6	フィルタ設定の変更	47

6 事前定義されたルールの使用 51

6.1	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	52
6.1.1	ポリシーの作成	52
6.1.2	事前定義されたルールのインポート	52
6.1.3	ルールの動作	53
6.2	コマンド変換 - 無効にする発行者の削除	53
6.2.1	ポリシーの作成	54
6.2.2	事前定義されたルールのインポート	54
6.2.3	ルールの動作	54
6.3	作成 - 属性が必要	54
6.3.1	ポリシーの作成	55
6.3.2	事前定義されたルールのインポート	55
6.3.3	ルールの動作	55
6.4	作成 - 発行者 - テンプレートの使用	56
6.4.1	ポリシーの作成	56
6.4.2	事前定義されたルールのインポート	56
6.4.3	ルールの動作	57
6.5	作成 - デフォルト属性値の設定	57
6.5.1	ポリシーの作成	57
6.5.2	事前定義されたルールのインポート	57
6.5.3	ルールの動作	58
6.6	作成 - デフォルトパスワードの設定	58
6.6.1	ポリシーの作成	58
6.6.2	事前定義されたルールのインポート	58
6.6.3	ルールの動作	59
6.7	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	59
6.7.1	ポリシーの作成	59
6.7.2	事前定義されたルールのインポート	59
6.7.3	ルールの動作	60
6.8	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	60
6.8.1	ポリシーの作成	60
6.8.2	事前定義されたルールのインポート	60
6.8.3	ルールの動作	61
6.9	入出力変換 - 電話番号の形式を (nnn) nnn-nnnn から nnn-xxx-nnnn に変更	61
6.9.1	ポリシーの作成	61
6.9.2	事前定義されたルールのインポート	62
6.9.3	ルールの動作	62
6.10	入出力変換 - 電話番号の形式を nnn-xxx-nnnn から (nnn) nnn-nnnn に変更	62
6.10.1	ポリシーの作成	62
6.10.2	事前定義されたルールのインポート	63
6.10.3	ルールの動作	63
6.11	一致 - 発行者 (ミラーリング)	63
6.11.1	ポリシーの作成	63
6.11.2	事前定義されたルールのインポート	63
6.11.3	ルールの動作	64
6.12	一致 - 購読者 (ミラーリング) - LDAP 形式	64
6.12.1	ポリシーの作成	64
6.12.2	事前定義されたルールのインポート	65
6.12.3	ルールの動作	65
6.13	一致 - 属性値別	65
6.13.1	ポリシーの作成	66
6.13.2	事前定義されたルールのインポート	66

6.13.3	ルールの動作	66
6.14	配置 - 発行者 (ミラーリング)	67
6.14.1	ポリシーの作成	67
6.14.2	事前定義されたルールのインポート	67
6.14.3	ルールの動作	68
6.15	配置 - 購読者 (ミラーリング)-LDAP 形式	68
6.15.1	ポリシーの作成	68
6.15.2	事前定義されたルールのインポート	68
6.15.3	ルールの動作	69
6.16	配置 - 発行者 (フラット)	69
6.16.1	ポリシーの作成	69
6.16.2	事前定義されたルールのインポート	69
6.16.3	ルールの動作	70
6.17	配置 - 購読者 (フラット)-LDAP 形式	70
6.17.1	ポリシーの作成	70
6.17.2	事前定義されたルールのインポート	70
6.17.3	ルールの動作	71
6.18	配置 - 部署別発行者	71
6.18.1	ポリシーの作成	71
6.18.2	事前定義されたルールのインポート	71
6.18.3	ルールの動作	72
6.19	配置 - 部署別購読者 -LDAP 形式	72
6.19.1	ポリシーの作成	72
6.19.2	事前定義されたルールのインポート	72
6.19.3	ルールの動作	73
7	リソースオブジェクトに情報を保存する	75
7.1	ライブラリオブジェクト	75
7.1.1	ライブラリの管理	75
7.1.2	ライブラリにオブジェクトを追加する	77
7.1.3	ライブラリに保存されたポリシーを使用する	81
7.2	マッピングテーブルオブジェクト	81
7.2.1	マッピングテーブルオブジェクトを作成する	81
7.2.2	マッピングテーブルオブジェクトをポリシーに追加する	83
7.3	ECMA Script	84
7.4	アプリケーションオブジェクト	84
7.5	リポジトリオブジェクト	84
7.6	ソースオブジェクト	84
8	ポリシーでの ECMAScript の使用	85
8.1	ECMAScript の作成	85
8.1.1	ドライバ内に ECMAScript を作成する	85
8.1.2	ライブラリ内に ECMAScript を作成する	86
8.2	既存の ECMAScript の使用	88
8.2.1	既存の ECMAScript をドライバで使用する	88
8.2.2	既存の ECMAScript をライブラリで使用する	89
8.3	ポリシーを備えた ECMAScripts の例	90
8.3.1	ECMAScript 関数を呼び出す DirXML スクリプトポリシー	90
8.3.2	ECMAScript の関数をドライバレベルで呼び出す XSLT	91
8.3.3	ECMAScript 関数をスタイルシート内で呼び出す XSLT ポリシー	92
9	条件	93
	If 関連付け	94

フィールド 94	
例 95	
If 属性	96
フィールド 96	
例 97	
If クラス名	99
フィールド 99	
例 100	
If ターゲット属性	102
フィールド 102	
例 103	
If ターゲット DN	105
フィールド 105	
例 106	
If エンタイトルメント	107
フィールド 107	
例 109	
If グローバル構成値	110
注 110	
フィールド 110	
例 111	
If ローカル変数	112
フィールド 112	
例 113	
If 名前付きパスワード	115
フィールド 115	
例 115	
If 操作属性	116
フィールド 116	
例 118	
If 操作プロパティ	119
フィールド 119	
例 120	
If 操作	121
フィールド 121	
例 122	
If パスワード	124
フィールド 124	
例 125	
If ソース属性	127
フィールド 127	
例 128	
If ソース DN	129
フィールド 129	
例 129	
If XML 属性	131
フィールド 131	
例 132	

If XPath 式	133
フィールド 133	
備考 133	
例 133	
変数の拡張	135
注 135	
10 アクション	137
関連付けの追加	139
フィールド 139	
例 139	
ターゲット属性値の追加	140
フィールド 140	
例 140	
ターゲットオブジェクトの追加	142
フィールド 142	
備考 142	
例 142	
ソース属性値の追加	144
フィールド 144	
例 144	
ソースオブジェクトの追加	145
フィールド 145	
例 145	
XML 要素の追加	146
フィールド 146	
備考 146	
例 146	
XML テキストの追加	148
フィールド 148	
備考 148	
例 148	
中断	150
例 150	
ターゲット属性値のクリア	151
フィールド 151	
例 151	
操作プロパティのクリア	152
フィールド 152	
例 152	
SSO 資格情報のクリア	153
フィールド 153	
例 153	
ソース属性値のクリア	154
フィールド 154	
例 154	
XPath 式によるクローン	155

フィールド 155	
備考 155	
例 155	
操作属性のクローン	156
フィールド 156	
例 156	
ターゲットオブジェクトの削除	157
フィールド 157	
例 157	
ソースオブジェクトの削除	158
フィールド 158	
例 158	
一致オブジェクトの検索	159
フィールド 159	
備考 159	
例 159	
毎	161
フィールド 161	
備考 161	
例 161	
イベントの生成	162
フィールド 162	
備考 163	
例 163	
if	165
フィールド 165	
例 165	
エンタイトルメントの実装	167
フィールド 167	
例 167	
ターゲットオブジェクトの移動	168
フィールド 168	
例 168	
ソースオブジェクトの移動	170
フィールド 170	
例 170	
操作属性の再フォーマット	171
フィールド 171	
例 171	
関連付けの削除	172
フィールド 172	
例 172	
ターゲット属性値の削除	173
フィールド 173	
例 173	
ソース属性値の削除	174
フィールド 174	

例 174	
ターゲットオブジェクトの名前変更	175
フィールド 175	
例 175	
操作属性の名前変更	176
フィールド 176	
例 176	
ソースオブジェクトの名前変更	177
フィールド 177	
例 177	
電子メールの送信	178
フィールド 178	
例 179	
テンプレートから電子メールを送信	180
フィールド 180	
例 181	
デフォルト属性値の設定	182
フィールド 182	
例 182	
ターゲット属性値の設定	183
フィールド 183	
例 183	
ターゲットパスワードの設定	185
フィールド 185	
例 185	
ローカル変数の設定	186
フィールド 186	
例 186	
操作関連付けの設定	187
フィールド 187	
例 187	
操作クラス名の設定	188
フィールド 188	
例 188	
操作ターゲット DN	189
フィールド 189	
例 189	
操作プロパティの設定	190
フィールド 190	
例 190	
操作ソース DN の設定	191
フィールド 191	
例 191	
操作テンプレート DN の設定	192
フィールド 192	
例 192	
ソース属性値の設定	193
フィールド 193	

例 193	
ソースパスワードの設定	195
フィールド 195	
例 195	
SSO 資格情報の設定	196
フィールド 196	
例 196	
SSO パスフレーズの設定	197
フィールド 197	
例 197	
XML 属性の設定	198
フィールド 198	
例 198	
ステータス	199
フィールド 199	
備考 199	
例 199	
ワークフローの開始	200
リマーク 200	
フィールド 200	
例 201	
操作属性のストリップ	202
フィールド 202	
例 202	
XPath のストリップ	203
フィールド 203	
備考 203	
例 203	
メッセージのトレース	204
フィールド 204	
例 204	
拒否	206
例 206	
操作属性値がない場合は拒否	207
フィールド 207	
例 207	
While	209
フィールド 209	
例 209	
変数の拡張	210
リマーク 210	
11 名詞トークン	211
追加されたエンタイトルメント	212
フィールド 212	
備考 212	
例 212	

関連付け	213
例 213	
属性	214
フィールド 214	
備考 214	
例 214	
文字	215
備考 215	
フィールド 215	
例 215	
クラス名	216
例 216	
ターゲット属性	217
フィールド 217	
備考 217	
例 217	
ターゲット DN	219
フィールド 219	
備考 219	
例 219	
ターゲット名	221
例 221	
ドキュメント	222
フィールド 222	
例 222	
エンタイトルメント	223
フィールド 223	
備考 223	
例 223	
パスワードの生成	224
フィールド 224	
例 224	
グローバル構成値	225
フィールド 225	
例 225	
ローカル変数	226
フィールド 226	
例 226	
名前付きパスワード	228
フィールド 228	
例 228	
説明	230
例 230	
操作属性	231
フィールド 231	
例 231	
操作プロパティ	232
フィールド 232	

例 232	
[パスワード]	233
例 233	
クエリ	234
フィールド 234	
例 234	
削除された属性	235
フィールド 235	
備考 235	
例 235	
削除されたエンタイトルメント	236
フィールド 236	
備考 236	
例 236	
解決	237
フィールド 237	
例 237	
ソース属性	238
フィールド 238	
備考 238	
例 238	
ソース DN	239
フィールド 239	
備考 239	
例 239	
ソース名	240
例 240	
[時刻]	241
フィールド 241	
例 241	
テキスト	242
フィールド 242	
例 242	
一意の名前	243
フィールド 243	
備考 244	
例 244	
一致しないソース DN	246
フィールド 246	
備考 246	
例 246	
XPath	247
フィールド 247	
備考 247	
例 247	
変数の拡張	248
注 248	

12 動詞トークン	249
Base64 デコード	250
フィールド 250	
例 250	
Base64 エンコード	251
フィールド 251	
例 251	
変換時間	252
フィールド 252	
例 252	
ターゲット DN のエスケープ	254
例 254	
ソース DN のエスケープ	255
例 255	
参加	256
フィールド 256	
例 256	
小文字	257
例 257	
マップ	258
備考 258	
フィールド 258	
例 258	
DN の解析	259
フィールド 259	
備考 259	
例 260	
すべて置換	261
フィールド 261	
備考 261	
例 261	
最初を置換	262
フィールド 262	
備考 262	
例 262	
分割	264
フィールド 264	
例 264	
部分文字列	265
フィールド 265	
例 265	
大文字	267
例 267	
XML 解析	268
例 268	
XML シリアライズ	269
例 269	
変数の拡張	270

注 270

このガイドについて

Novell® Identity Manager 3.5.1 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、iManager でのポリシーの作成と管理について詳しく説明しています。

- ◆ 15 ページの第 1 章「概要」
- ◆ 17 ページの第 2 章「ポリシービルダでポリシーを管理する」
- ◆ 29 ページの第 3 章「追加ビルダの使用」
- ◆ 39 ページの第 4 章「スキーママッピングポリシーの定義」
- ◆ 45 ページの第 5 章「フィルタを使用したオブジェクトのフローの制御」
- ◆ 51 ページの第 6 章「事前定義されたルールの使用」
- ◆ 75 ページの第 7 章「リソースオブジェクトに情報を保存する」
- ◆ 85 ページの第 8 章「ポリシーでの ECMAScript の使用」
- ◆ 93 ページの第 9 章「条件」
- ◆ 137 ページの第 10 章「アクション」
- ◆ 211 ページの第 11 章「名詞トークン」
- ◆ 249 ページの第 12 章「動詞トークン」

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。

フィードバック

本マニュアルおよびこの製品に含まれているその他のマニュアルについて、皆様のご意見やご要望をお寄せください。オンラインマニュアル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 www.novell.com/documentation/feedback.html にアクセスしてコメントを記入してください。

マニュアルの更新

Policies in iManager の最新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/idm35>) を参照してください。

追加のマニュアル

Identity Manager ドライバのマニュアルについては、[Identity Manager マニュアル Web サイト \(http://www.novell.com/documentation/idm35drivers/index.html\)](http://www.novell.com/documentation/idm35drivers/index.html) を参照してください。

Novell iManager のマニュアルについては、[Novell iManager のマニュアルの Web サイト \(http://www.novell.com/documentation/imanager26/index.html\)](http://www.novell.com/documentation/imanager26/index.html) を参照してください。

Designer Designer 2.1 for Identity Manager 3.5.1 のマニュアルの Web サイト (<http://www.novell.com/documentation/designer21/>) を参照してください。

マニュアルの表記規則

Novell のマニュアルでは、「より大きい」記号 (>) を使用して手順内の操作と相互参照パス内の項目の順序を示します []

商標記号 (®、™ など) は、Novell の商標を示します。アスタリスク (*) は、サードパーティの商標を示します。

プラットフォームによっては、シングルパス名に円記号 (()) を使用できる場合とスラッシュ (/) を使用できる場合がありますが、パス名は円記号で表記されます。Linux*、UNIX* など、スラッシュを使う必要があるプラットフォームを使用しているユーザは、必要に応じてスラッシュを使用してください。

ポリシーにより、識別ボールドとリモートデータストアとの間で同期するデータを管理します。ポリシーはポリシーセット内に保存されます。詳細は『“[Understanding Policies for Identity Manager 3.5.1](#)”』で「[Understanding Types of Policies](#)」を参照してください。Identity Manager では、ポリシーを作成および管理するための iManager プラグインがインストールされます。このセクションでは、ポリシーを作成および使用する方法について説明します。

- ◆ [17 ページの第 2 章「ポリシービルダでポリシーを管理する」](#)
- ◆ [29 ページの第 3 章「追加ビルダの使用」](#)
- ◆ [39 ページの第 4 章「スキーママッピングポリシーの定義」](#)
- ◆ [45 ページの第 5 章「フィルタを使用したオブジェクトのフローの制御」](#)
- ◆ [51 ページの第 6 章「事前定義されたルールの使用」](#)
- ◆ [75 ページの第 7 章「リソースオブジェクトに情報を保存する」](#)
- ◆ [85 ページの第 8 章「ポリシーでの ECMAScript の使用」](#)

また、このガイドには DirXML[®] スクリプトにおけるすべての要素を詳細に示したセクションも含まれています。DirXML スクリプトの詳細については、『“[Understanding Policies for Identity Manager 3.5.1](#)”』で「[DirXML Script](#)」を参照してください。

- ◆ [93 ページの第 9 章「条件」](#)
- ◆ [137 ページの第 10 章「アクション」](#)
- ◆ [211 ページの第 11 章「名詞トークン」](#)
- ◆ [249 ページの第 12 章「動詞トークン」](#)

1.1 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 ([137 ページの第 10 章「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 ([93 ページの第 9 章「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、名詞と動詞の 2 つに分類できます。
 - ◆ 名詞トークン ([211 ページの第 11 章「名詞トークン」](#)を参照) 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 ([249 ページの第 12 章「動詞トークン」](#)を参照) 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 (『“ [Understanding Policies for Identity Manager 3.5.1](#) ”』の「*Regular Expressions*」を参照) および XPath 1.0 の式 (『“ [Understanding Policies for Identity Manager 3.5.1](#) ”』の「*XPath 1.0 Expressions*」を参照) は、一般的にはポリシーで目的の結果を生成するための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。要素は、Novell の `nds.dtd` の一部です。詳細については、『“ [Identity Manager 3.5.1 DTD Reference](#) ”』の「*NDS DTD*」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。

ポリシービルダでポリシーを管理する

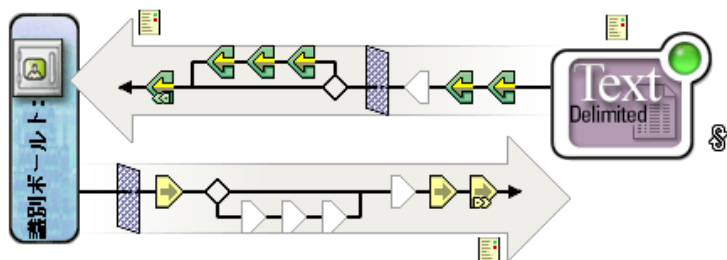
2

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。

- ◆ 17 ページのセクション 2.1 「ポリシービルダへのアクセス」
- ◆ 18 ページのセクション 2.2 「ポリシーの作成」
- ◆ 21 ページのセクション 2.3 「ポリシー内での各ルールの定義」
- ◆ 23 ページのセクション 2.4 「ルール内で引数を作成する」
- ◆ 25 ページのセクション 2.5 「ポリシーの変更」
- ◆ 25 ページのセクション 2.6 「ポリシーの削除」
- ◆ 26 ページのセクション 2.7 「ポリシーの名前変更」
- ◆ 26 ページのセクション 2.8 「ポリシーの削除」
- ◆ 26 ページのセクション 2.9 「XML ファイルからのポリシーのインポート」
- ◆ 26 ページのセクション 2.10 「XML ファイルへのポリシーのエクスポート」
- ◆ 26 ページのセクション 2.11 「ポリシーの参照の作成」

2.1 ポリシービルダへのアクセス

- 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 ドライバセットを指定し、[検索] をクリックします。
- 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] ページが開きます。



このビューからポリシーを管理します。

- 4 編集するポリシーをクリックして、ポリシービルダを開きます。





2.2 ポリシーの作成

ポリシーは、ドライバ内またはライブラリオブジェクト内に作成できます。

- 18 ページのセクション 2.2.1 「ドライバ内にポリシーを作成する」
- 19 ページのセクション 2.2.2 「ライブラリ内にポリシーを作成する」

2.2.1 ドライバ内にポリシーを作成する

- 1 管理するドライバの [Identity Manager ドライバの概要] ページを開きます。
- 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 [新しいポリシーの作成] を選択します。
- 5 新規ポリシーの名前を指定します。
- 6 ポリシーの実装方法を選択し、[OK] をクリックします。

● 新しいポリシーの作成

新しいポリシーで使用される名前を入力します。

配置

ポリシーを作成するコンテナを選択します。

Subscriber.GroupWise.driver_set.context

このポリシーをどのように実装しますか?

ポリシービルダ

XSLT

既存のポリシーからコピーを作成する

コピーするポリシーを選択します。

既存のポリシーを使用する



使用する既存のポリシーのDNを入力します。

OK

キャンセル

- 6a [ポリシービルダ] を選択すると、ポリシービルダが起動します。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、21 ページのセクション 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。
- 6b [XSLT] を選択すると、XML エディタが起動します。XSLT を使用してポリシーを定義する場合は、『“ [Understanding Policies for Identity Manager 3.5.1](#) ”』の「[Defining Policies by Using XSLT Style Sheets](#)」を参照してください。
- 6c [既存のポリシーからコピーを作成する] を選択する場合には、コピーするポリシーを参照して選択します。

既存のポリシーを使用して新しいポリシーを作成することもできます。

- 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 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 [既存のポリシーを使用する] を選択してから、使用する既存のポリシーを参照して選択します。
- 5 [OK] をクリックします。

2.2.2 ライブラリ内にポリシーを作成する

- 1 iManager で、[Identity Manager] > [Identity Manager Overview] の順に選択します。
- 2 [検索] をクリックして、ツリー内のすべてのドライバを検索します。
- 3 [Identity Manager の概要] で、[ライブラリオブジェクト] をクリックします。

Identity Managerの概要

1個のドライバセットが次の場所に見つかりました: ディレクトリ全体
0 ライブラリオブジェクト 検索された場所: ディレクトリ全体

- 4 ポリシーを追加するライブラリをクリックします。

ライブラリ管理

ライブラリで検索された場所:

[root] 🔍 🏠 ▶

新規作成... | 削除

<input type="checkbox"/> 名前	コンテナ
<input checked="" type="checkbox"/> library	context


- 5 プラスアイコンをクリックして、ライブラリにポリシーを追加します。

Identity Managerライブラリ

ライブラリ: library.context

ポリシー | マッピングテーブル | 資格情報のプロビジョニング

次のポリシーはこのドライバで見つかりました:
(ポリシーのルールリストを取得するには、ポリシー名の左のイメージをクリックします。)

☐ DirXML-Library 

削除

- 6 ポリシーの名前を指定します。
- 7 ポリシーの実装方法を選択し、[OK] をクリックします。

ポリシーの作成

新しいポリシーで使用される名前を入力します。

作成

ポリシーを作成するコンテナを選択します。

library.context

このポリシーをどのように実装しますか?

- ポリシービルダ
 - XSLT
 - ECMAScript
 - 既存のポリシーからコピーを作成する
- コピーするポリシーを選択します。

OK

キャンセル

7a [ポリシービルダ]、[XSLT]、または [ECMAScript] を選択する場合には、ライブラリにオブジェクトが作成され、表示されます。各オブジェクトを編集して、ポリシー情報をオブジェクトに追加する必要があります。

7b [既存のポリシーからコピーを作成する] を選択する場合には、ライブラリに保存するポリシーを参照して選択します。

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 2-1 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

②説明: 著者:
バージョン:
最終変更日時:

条件
条件構造を選択:
 OR 条件, AND グループ
 AND 条件, OR グループ

条件グループの追加 *必須

条件グループ 1

If <条件の選択>

アクション
アクションリスト

実行内容 <アクションを選択します>

OK キャンセル

ルールビルダのインターフェースでは、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、環境内でのオブジェクトの追加を拒否するルールを作成する必要がある場合は、「追加操作が実行された場合に操作を拒否する」というルールを定義します。

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 2-2 ルールビルダインターフェースにおけるユーザの移動条件

The screenshot shows two rows of configuration in the Rule Builder. The first row is for an 'If' condition with the label 'If 操作'. It has a dropdown menu set to '等しい' (Equal), a mode dropdown set to '大文字小文字の区別' (Case Sensitive), and a text input field containing 'move'. The second row is for an 'AND If' condition with the label 'AND If クラス名'. It also has a dropdown menu set to '等しい' (Equal), a mode dropdown set to '大文字小文字の区別' (Case Sensitive), and a text input field containing 'User'. Each row has a green checkmark icon on the left and a set of utility icons on the right.

次のアクションも選択します。

図 2-3 ルールビルダインターフェースにおける拒否アクション

The screenshot shows a single dropdown menu labeled '実行内容' (Action) with the value '拒否' (Deny) selected. There is a green checkmark icon to the left of the dropdown.

ルールビルダで参照できる条件とアクションの詳細については、[93 ページの第 9 章「条件」](#) および [137 ページの第 10 章「アクション」](#) を参照してください。

ヒント






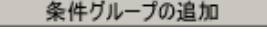

より複雑な条件を作成するには、条件および条件グループと、AND/OR ステートメントを組み合わせることができます。これらの方法は、条件構造を選択し、組み合わせることで変更できます。

図 2-4 条件構造のラジオボタン

条件構造を選択:

- OR 条件, AND グループ
- AND 条件, OR グループ

- ◆ 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。
- ◆ 引数ビルダインター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、 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。

- ◆ ポリシーのトレースを無効にするには、アイコンをクリックします。ポリシーのトレースを再び有効にするには、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] / [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、ボタンを使用します。
- ◆ 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。

2.4 ルール内で引数を作成する

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、[30 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結果を変更します。

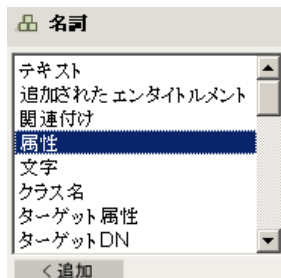
図 2-5 引数ビルダのデフォルトインタフェース



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、引数値に対して引数セットを作成したいとします。

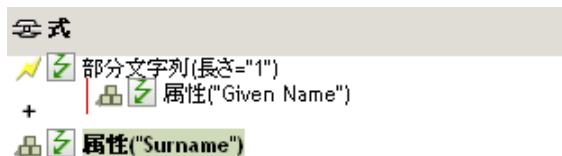
- 1 名詞トークンのリストから「属性」を選択して、「追加」をクリックします。



- 2 エディタで属性名を参照して選択します。



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。この式では、長さが 1 の部分文字列が、[名字] 属性全体と組み合わせられた [名前] 属性を表します。



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。

名詞トークンおよび動詞トークンの詳細リファレンスについては、[211 ページの第 11 章「名詞トークン」](#) および [249 ページの第 12 章「動詞トークン」](#) を参照してください。引数ビルダの詳細については、[30 ページのセクション 3.2「引数ビルダ」](#) を参照してください。

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。

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、[削除] 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。
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。

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。

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。

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。

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。

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。

追加ビルダの使用

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。

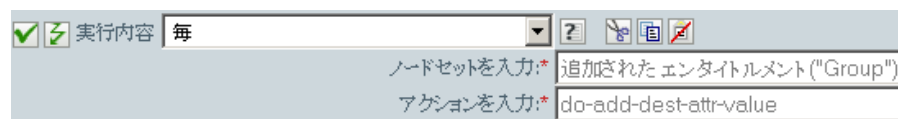
- 29 ページのセクション 3.1 「引数アクションビルダ」
- 30 ページのセクション 3.2 「引数ビルダ」
- 33 ページのセクション 3.3 「一致属性ビルダ」
- 34 ページのセクション 3.4 「アクションの引数コンポーネントビルダ」
- 35 ページのセクション 3.5 「引数値リストビルダ」
- 35 ページのセクション 3.6 「名前付き文字列ビルダ」
- 37 ページのセクション 3.7 「条件の引数コンポーネントビルダ」

3.1 引数アクションビルダ

引数アクションビルダにより、**毎**アクションおよび**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。

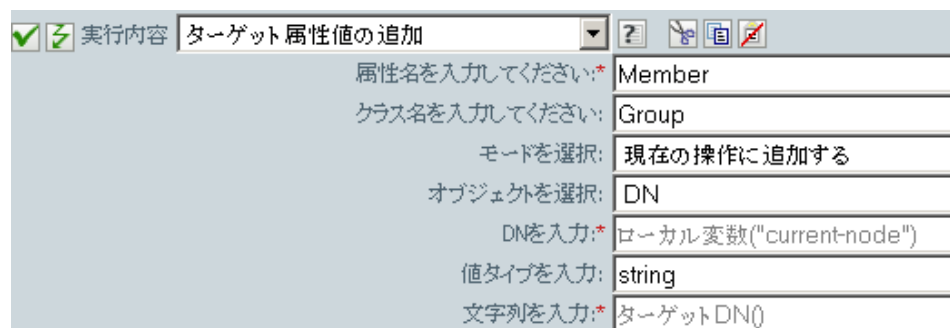
次の例では、ターゲット属性値の追加アクションが **Group** エンタイトルメントごとに実行され、現在の操作に追加されます。

図 3-1 それぞれに対するアクション



ターゲット属性値の追加アクションを定義するには、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 3-2 ターゲット属性値の追加アクション



3.2 引数ビルダ


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。

引数ビルダは5つの別々のセクションで構成されます。

- ◆ **名詞** : 利用可能な名詞トークンすべてのリストを含みます。名詞トークンを選択し、[追加]をクリックしてその名詞トークンを[表現]ペインに追加します。
- ◆ **バープ** : 利用可能な動詞トークンすべてのリストを含みます。動詞トークンを選択し、[追加]をクリックしてその動詞トークンを[表現]ペインに追加します。
- ◆ **説明** : 名詞または動詞トークンの簡単な説明を含みます。[ヘルプ]アイコンをクリックして追加のヘルプを起動します。
- ◆ **Expression** : 作成されている引数を含みます。複数の名詞および動詞トークンを単一の引数に追加することができます。トークンは、[式]ペインで異なる順序に配置することができます。
- ◆ **エディタ** : [エディタ]ペインを使用すると、名詞および動詞に値を指定できます。

図 3-3 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。

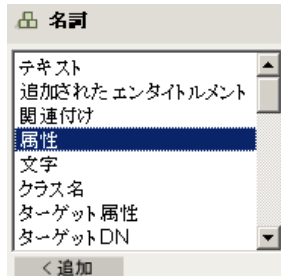
- ◆ 関連付けの追加
- ◆ ターゲット属性値の追加
- ◆ ターゲットオブジェクトの追加
- ◆ ソース属性値の追加
- ◆ XML テキストの追加
- ◆ ターゲット属性値のクリア (選択されたオブジェクトが DN または関連付けである場合)。
- ◆ ソース属性値のクリア (選択されたオブジェクトが DN または関連付けである場合)。
- ◆ ターゲットオブジェクトの削除 (選択されたオブジェクトが DN または関連付けである場合)。
- ◆ ソースオブジェクトの削除 (選択されたオブジェクトが DN または関連付けである場合)。
- ◆ 一致オブジェクトの検索
- ◆ 毎
- ◆ ターゲットオブジェクトの移動
- ◆ ソースオブジェクトの移動
- ◆ 操作属性の再フォーマット
- ◆ 関連付けの削除
- ◆ ターゲット属性値の削除
- ◆ ソース属性値の削除
- ◆ 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 または関連付けで、[文字列を入力] である場合)。
- ◆ 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 または関連付けで、[文字列を入力] である場合)。
- ◆ 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、[値タイプを入力] が指定されていない場合)。
- ◆ ターゲットパスワードの設定
- ◆ ローカル変数の設定
- ◆ 操作関連付けの設定
- ◆ 操作クラス名の設定
- ◆ 操作ターゲット DN
- ◆ 操作プロパティの設定
- ◆ 操作ソース DN の設定
- ◆ 操作テンプレート DN の設定
- ◆ ソース属性値の設定
- ◆ ソースパスワードの設定
- ◆ XML 属性の設定
- ◆ ステータス

◆ **メッセージのトレース**

表現を定義するには、値、オブジェクト、変数などの名詞を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と組み合わせます。

次の例では、名の最初の文字と姓全体からユーザ名の引数を作成します。

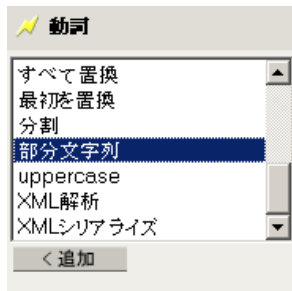
- 1 名詞のリストから [属性] を選択して、[追加] をクリックします。



- 2 名の属性を指定または選択します。



- 3 動詞のリストから部分文字列を選択して、[追加] をクリックします。



- 4 [長さ] フィールドに「1」を入力します。

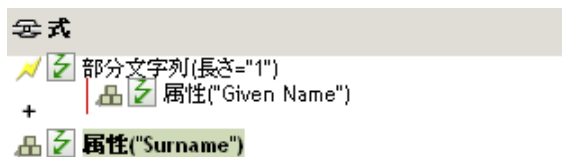


- 5 [名の属性] を選択して、[下へ移動] アイコンをクリックします。



- 6 名詞のリストから属性を選択して、[追加] をクリックします。
- 7 名字属性を指定または参照します。

- 8 名字の属性を選択してから、[下へ移動] アイコンを2回クリックします。



引数に名の属性の最初の文字が渡され、目的の値を構築するために名字の属性に追加されます。

- 9 [OK] をクリックして、引数を保存します。

3.2.1 引数ビルダの **À**f

- 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] / [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。
- 引数内でトークンの位置を変更したり、トークンを削除するには、[上へ移動] / [下へ移動] / [削除] アイコン を使用します。
- 引数ビルダのインタフェースをリフレッシュするには、[式パネルを更新する](#) リンクを使用します。トークンを追加または変更すると、インタフェースは自動的にリフレッシュされます。

3.3 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、**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。

次の例では、ユーザが **Provo** を基にしている、ユニークな **CN** 属性を持っている場合にユーザが一致します。

- 1 ポリシービルダで、[一致オブジェクトの検索] を選択します。
- 2 「サブツリー」としての検索のスコープを選択します。
- 3 検索する場所を参照して選択します。この例では、ユーザコンテナになっています。
- 4 [一致属性を入力] フィールドの横のアイコンをクリックして、一致属性ビルダを起動します。



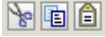
- 5 [新しい一致属性の追加] をクリックして、一致させる属性を追加します。
- 6 [名前] フィールドで **CN** 属性を指定します。
- 7 [現在のオブジェクトから値] を選択し、同じ **CN** 属性を持つ他のユーザがいるかどうかを確認します。
- 8 [新しい一致属性の追加] をクリックして、一致させる別の属性を追加します。

- 9 [名前] フィールドで L 属性を指定します。
- 10 [その他の値] を選択して、「Provo」を値として指定します。

一致する属性:	
名前:* CN	現在のオブジェクトからの値
名前:* L	その他の値
値タイプを入力:	string
文字列を入力:*	"Provo"

- 11 [OK] をクリックします。

3.3.1 一致属性ビルダのヒント

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] / [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。

3.4 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。

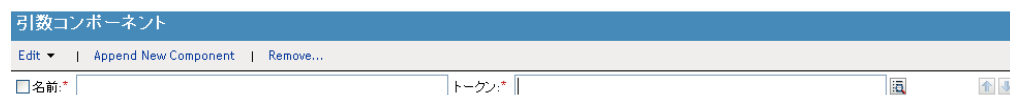
- (140 ページ) ターゲット属性値の追加
- (144 ページ) ソース属性値の追加
- (171 ページ) 操作属性の再フォーマット
- (173 ページ) ターゲット属性値の削除
- (174 ページ) ソース属性値の削除
- (182 ページ) デフォルト属性値の設定
- (193 ページ) ソース属性値の設定

図 3-4 [値のタイプを入力] フィールドを [構造] に設定するアクション

実行内容	ターゲット属性値の追加
属性名を入力してください:*	Given Name
クラス名を入力してください:	User
モードを選択:	ターゲットデータストアに直接書き込む
オブジェクトを選択:	DN
DNを入力:*	"Novell\Users"
値タイプを入力:	structured
コンポーネントを入力:*	user

[値のタイプ] を [構造] に設定したら、[コンポーネントの編集] アイコンをクリックします。

図 3-5 アクションの引数コンポーネントビルダ



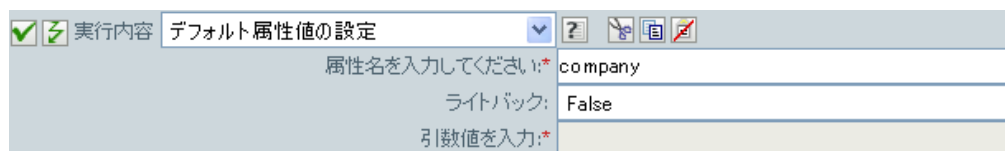
アクション引数コンポーネントビルダが起動し、アクションを作成できるようになります。


3.5 引数値リストビルダ

引数値リストビルダでは、**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、デフォルトの会社名を設定する場合は：

- 1 アクションのリストから [デフォルト属性値の設定] を選択します。
- 2 参照して、会社の属性を選択します。



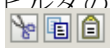
- 3 [値リストを編集する] アイコンをクリックして、会社名を作成します。
- 4 引数値リストビルダで [Append New Value (新しい値の追加)] をクリックします。
- 5 会社の名前を指定します。



この例では、会社名は Digital Airlines です。

- 6 [OK] を 2 回クリックします。

3.5.1 引数値リストビルダのヒント

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] / [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。

3.6 名前付き文字列ビルダ

名前付き文字列ビルダでは、**イベントの生成**、**電子メールの送信**および**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。

- ◆ target

- ◆ target-type
- ◆ subTarget
- ◆ text1
- ◆ text2
- ◆ text3
- ◆ 値
- ◆ value3
- ◆ データ
- ◆ data-type

図 3-6 名前付き文字列ビルダ

文字列		
Edit ▾		Append New String Remove...
<input type="checkbox"/> 名前:*	text1	文字列の値:* 操作属性("Given Name")
<input type="checkbox"/> 名前:*	text2	文字列の値:* 操作()
<input type="checkbox"/> 名前:*	value	文字列の値:* "1000"

[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。

- ◆ >
- ◆ cc
- ◆ bcc
- ◆ 送信者
- ◆ reply-to
- ◆ 件名
- ◆ メッセージ
- ◆ エンコード
- ◆ custom-smpt-header

図 3-7 メール送信アクション

文字列		
Edit ▾		Append New String Remove...
<input type="checkbox"/> 名前:*	to	文字列の値:* "admin@somecompany.com"
<input type="checkbox"/> 名前:*	subject	文字列の値:* "An"+操作()+"Occured in the Directory"
<input type="checkbox"/> 名前:*	message	文字列の値:* "Text of the message"

テンプレートからの [メール送信] アクションの場合、名前付き文字列はテンプレート内のメールの次の要素に対応しています。

- ◆ >
- ◆ cc

- ◆ bcc
- ◆ reply-to
- ◆ エンコード
- ◆ custom-smtp-header

3.7 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。

- ◆ If 属性
- ◆ If ターゲット属性
- ◆ If ソース属性

図 3-8 [構造] オプション

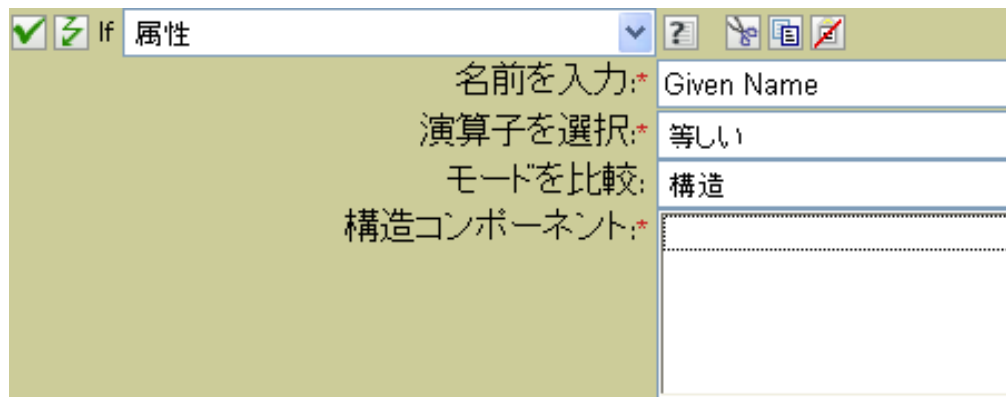
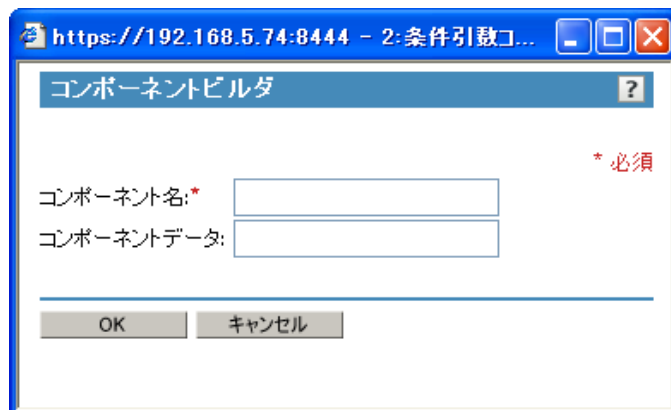


図 3-9 条件の引数コンポーネントビルダ



スキーママッピングポリシーの定義

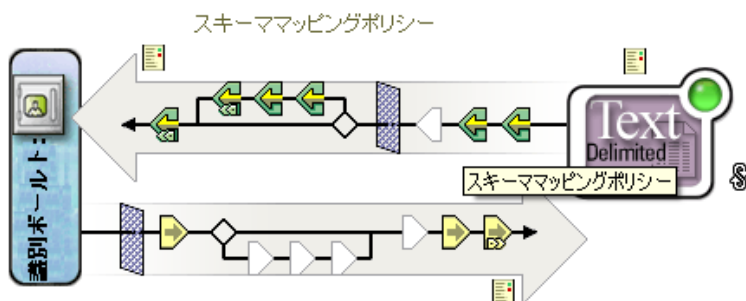
スキーママッピングポリシーは、クラス名および属性名を識別ポールのネームスペースとアプリケーションのネームスペースとの間でマップします。同じスキーママッピングポリシーが両方向に適用されます。メタディレクトリエンジンとアプリケーションシム間で渡されるドキュメントは、いずれのチャンネルでも、またいずれの方向でも、すべてのドキュメントがスキーママッピングポリシーを通過します。

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。

- ◆ 39 ページのセクション 4.1 「スキーママッピングポリシーへのアクセス」
- ◆ 39 ページのセクション 4.2 「スキーママッピングポリシーの編集」

4.1 スキーママッピングポリシーへのアクセス

- 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 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。



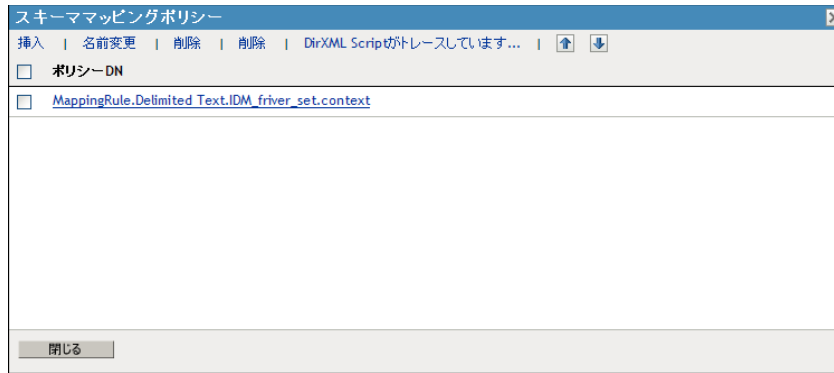
- 4 スキーママッピングポリシーをクリックします。
- 5 [編集] をクリックします。

4.2 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。

4.2.1 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。
削除	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。
リネーム	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。

4.2.2 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。

Identity Managerポリシー: MappingRule.Delimited Text.IDM_frivier_set.context

Identity Manager
Identity Managerポリシー | XMLの編集 | 使用状況

ドライバDN: Delimited Text.IDM_frivier_set.context

eDirectoryクラス	アプリケーションクラス	
User	User	削除
		属性...
[Anything]	<マップされていないクラスなし>	追加
クラス固有でない属性...		
Application Schemaの更新 送信元サーバ: idm.context		
eDirectoryスキーマツール		

OK キャンセル 適用

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。

- ◆ 41 ページの「Identity Manager ポリシー」
- ◆ 42 ページの「XML の編集」
- ◆ 43 ページの「使用率」

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。

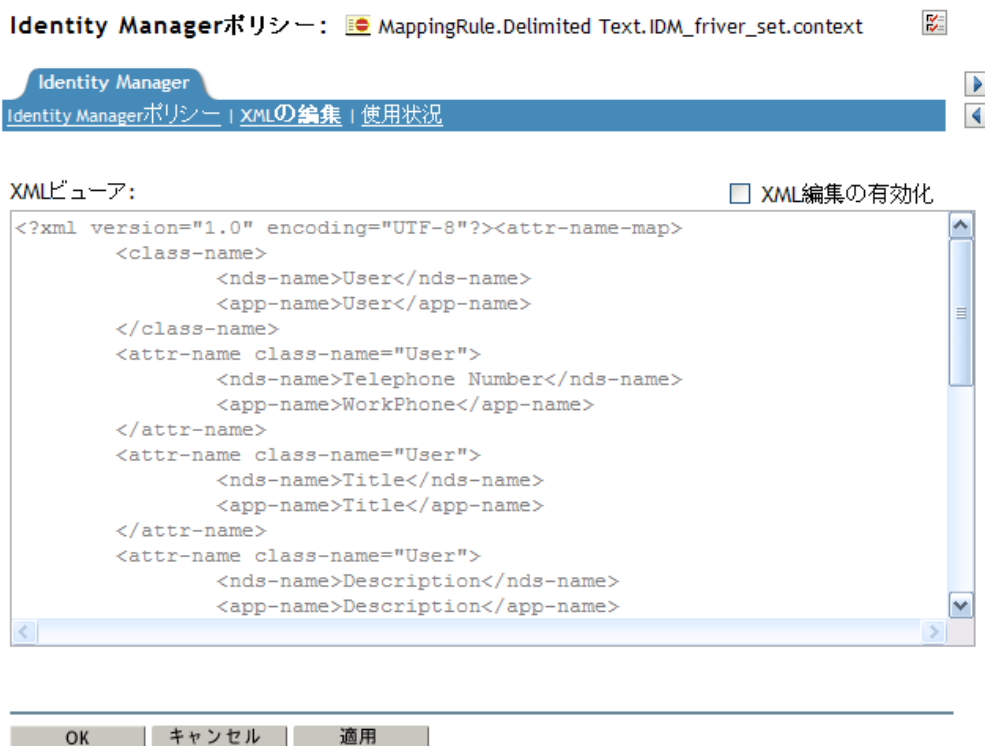
クラスに固有でない属性の一覧表示	クラスに関連付けられていない属性がある場合は、 [Non-specific Class Attributes (クラスに固有でない属性)] アイコンをクリックして、これらの属性をすべて一覧表示します。
アプリケーションスキーマのリフレッシュ	アプリケーションでスキーマが変更されている場合は、 [アプリケーションスキーマのリフレッシュ] アイコンをクリックします。ウィザードが接続システムサーバにアクセスして新しいスキーマを取得します。スキーマは、更新された後にドロップダウンリストに表示されます。
eDirectory スキーマツール	<ul style="list-style-type: none"> ◆ [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 : アイデンティティポータル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。



図 4-1 XML の編集



使用率

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。

図 4-2 使用率

Identity Managerポリシー: MappingRule.Delimited Text.IDM_frivier_set.context  

Identity Manager

Identity Managerポリシー | XMLの編集 | 使用状況  

次の表は、このポリシーを参照するドライバのリストおよびこのポリシーがこれらのドライバによりどのように使用されているかを示しています。

ドライバ	このポリシーを利用したポリシーセット
Delimited Text	スキーママッピングポリシーセット

このポリシーに対する参照は、識別ポータルからのバックリンクを使用して読み取ることができませんでした。このため、このポリシーを含むドライバセットによるこのポリシーへの参照だけが分析されています。このポリシーが他のドライバセットから参照されている場合、それらの参照はここには表示されません。iManagerで使用されるJClientのバージョンをアップグレードすると、この問題を解決できる可能性があります。

OK

キャンセル

適用

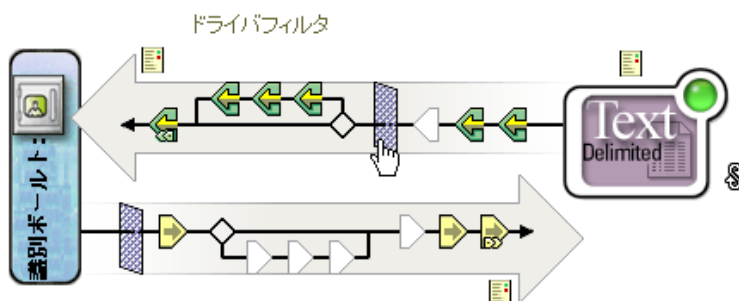
フィルタを使用したオブジェクトのフローの制御

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ンネルおよび購読者チャンネルによって処理される方法を定義します。

- 45 ページのセクション 5.1 「フィルタへのアクセス」
- 45 ページのセクション 5.2 「フィルタの編集」

5.1 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。



- 4 発行者チャンネルまたは購読者チャン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。



5.2 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ド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。

図 5-1 フィルタエディタ



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。

- ◆ 46 ページのセクション 5.2.1 「フィルタからのクラスまたは属性の削除」
- ◆ 47 ページのセクション 5.2.2 「クラスの追加」
- ◆ 47 ページのセクション 5.2.3 「属性の追加」
- ◆ 47 ページのセクション 5.2.4 「フィルタのコピー」
- ◆ 47 ページのセクション 5.2.5 「テンプレートの設定」
- ◆ 47 ページのセクション 5.2.6 「フィルタ設定の変更」

5.2.1 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。

5.2.2 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。

5.2.3 属性の追加

- 1 [Add Attribute] 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。

5.2.4 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。

5.2.5 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。

5.2.6 フィルタ設定の変更

フィルタエディタには、識別ボールド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	<ul style="list-style-type: none">◆ 同期: 接続システムからアイデンティティボールド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ド方向でクラスを同期しません。
購読	<ul style="list-style-type: none">◆ 同期: アイデンティティボールド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ドから接続システム方向でクラスを同期しません。

オプション	定義
ホームディレクトリの作成	<ul style="list-style-type: none"> ◆ 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	<ul style="list-style-type: none"> ◆ はい：発行者チャン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	<ul style="list-style-type: none"> ◆ 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。 ◆ お知らせ君：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、もう一方のチャン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ンネルだけに設定できます。両方には設定できません)。
購読	<ul style="list-style-type: none"> ◆ 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。 ◆ お知らせ君：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、もう一方のチャン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ンネルだけに設定できます。両方には設定できません)。

オプション	定義
マージ権限	<ul style="list-style-type: none"> <li data-bbox="557 258 1286 315">◆ デフォルト：属性がどちらのチャンネルでも同期されていない場合、マージは行われません。 属性が一方のチャンネルだけで同期されている場合は、そのチャンネルのターゲットの既存の値がすべて削除され、そのチャン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。 属性が両方のチャンネルで同期され、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ド値を取得します。ただし、アイデンティティボールド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ドが、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。 属性が両方のチャンネルで同期され、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值側に存在していない場合に限り、複数值側に追加されます。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。 <li data-bbox="557 846 1286 968">◆ 識別ボールド：属性が発行者チャンネルではなく購読者チャン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になります。 これは、購読者チャンネルで同期されている場合に有効な動作です。 <li data-bbox="557 984 1286 1136">◆ アプリケーション：属性が購読者チャンネルではなく発行者チャン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になります。 これは、発行者チャンネルで同期されている場合に有効な動作です。 <li data-bbox="557 1152 1045 1173">◆ なし：同期に関係なく、マージされません。 <p data-bbox="245 1203 505 1260"><i>Identity Manager</i> に対する変更の最適化</p> <ul style="list-style-type: none"> <li data-bbox="557 1203 1286 1260">◆ はい：アイデンティティボールドで行われる変更を最小限にするために、発行者チャンネルでこの属性の変更を調べます。 <li data-bbox="557 1276 878 1297">◆ 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。

事前定義されたルールの使用

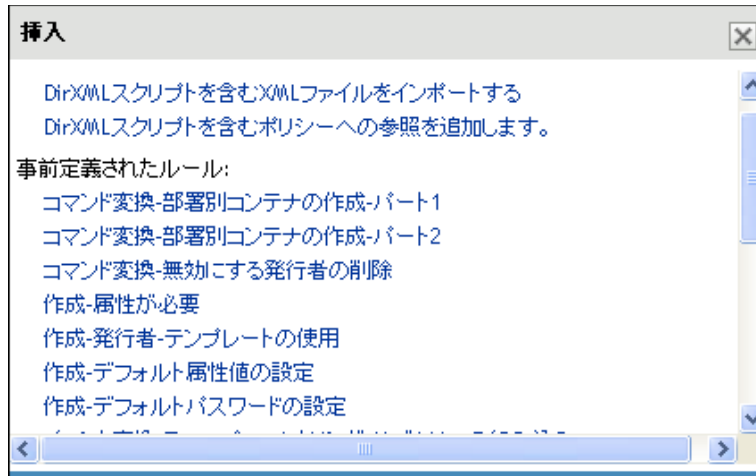
6

iManager には、19 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。

- ◆ 52 ページのセクション 6.1「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」
- ◆ 53 ページのセクション 6.2「コマンド変換-無効にする発行者の削除」
- ◆ 54 ページのセクション 6.3「作成-属性が必要」
- ◆ 56 ページのセクション 6.4「作成-発行者-テンプレートの使用」
- ◆ 57 ページのセクション 6.5「作成-デフォルト属性値の設定」
- ◆ 58 ページのセクション 6.6「作成-デフォルトパスワードの設定」
- ◆ 59 ページのセクション 6.7「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」
- ◆ 60 ページのセクション 6.8「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」
- ◆ 61 ページのセクション 6.9「入出力変換-電話番号の形式を (nnn) nnn-nnnn から nnn-~~nnn~~-~~nnnn~~ に変更」
- ◆ 62 ページのセクション 6.10「入出力変換-電話番号の形式を nnn-~~nnn~~-~~nnnn~~ から (nnn) nnn-~~nnnn~~ に変更」
- ◆ 63 ページのセクション 6.11「一致-発行者(ミラーリング)」
- ◆ 64 ページのセクション 6.12「一致-購読者(ミラーリング)-LDAP 形式」
- ◆ 65 ページのセクション 6.13「一致-属性値別」
- ◆ 67 ページのセクション 6.14「配置-発行者(ミラーリング)」
- ◆ 68 ページのセクション 6.15「配置-購読者(ミラーリング)-LDAP 形式」
- ◆ 69 ページのセクション 6.16「配置-発行者(フラット)」
- ◆ 70 ページのセクション 6.17「配置-購読者(フラット)-LDAP 形式」
- ◆ 71 ページのセクション 6.18「配置-部署別発行者」
- ◆ 72 ページのセクション 6.19「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。



6.1 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

このルールは、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。

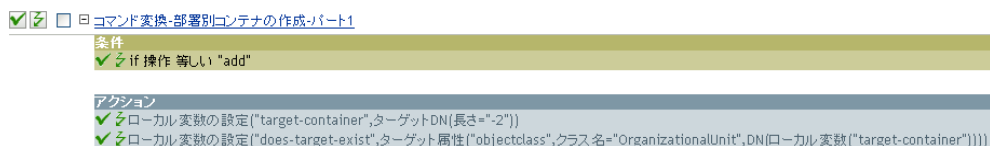
この事前定義されたルールを使用するには、コマンド変換ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの2つの手順を実行します。このルールに追加するコマンド変換ポリシーがすでにある場合は、[52 ページの「事前定義されたルールのインポート」](#)へ進みます。

6.1.1 ポリシーの作成

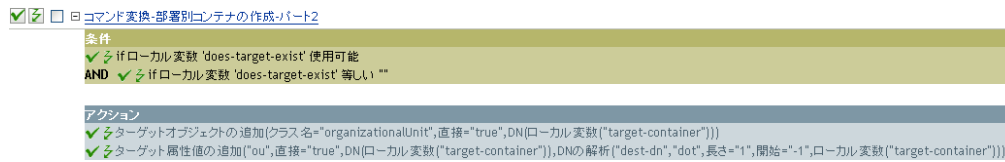
- 1 管理するドライバの [Identity Manager ドライバの概要] を開きます。
- 2 発行者チャンネルまたは購読者チャンネルのコマンド変換ポリシーオブジェクトをクリックします。
- 3 [挿入] をクリックします。
- 4 ポリシーの名前を指定し、このポリシーをポリシービルダで実装することを確認したら、[OK] をクリックします。
ルールビルダが起動します。
- 5 [52 ページのセクション 6.1.2 「事前定義されたルールのインポート」](#) に進みます。

6.1.2 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。

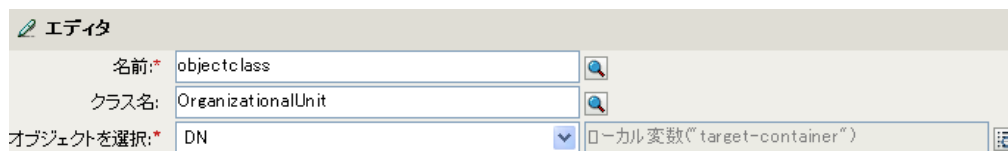
重要: ルールが順序どおりに表示されていることを確認します。パート 1 は 2 よりも先に実行する必要があります。

6.1.3 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 6-1 Create Container



パート 2 では、ローカル変数 `does-target-exist` が使用可能かどうかを確認されます。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。

6.2 コマンド変換 - 無効にする発行者の削除

このルールでは、ユーザオブジェクトの削除操作が、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されます。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、(54 ページ) 事前定義されたルールへのインポートへ進みます。

6.2.1 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 発行者チャン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。
- 5 54 ページのセクション 6.2.2 「事前定義されたルールへのインポート」に進みます。

6.2.2 事前定義されたルールへのインポート

- 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。

6.2.3 ルールの動作

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ド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ド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。

6.3 作成 - 属性が必要

このルールでは、必要な属性が入力されない場合にユーザオブジェクトが作成できないようになります。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へのインポートの 2 つの手順を実行します。このルールに追加

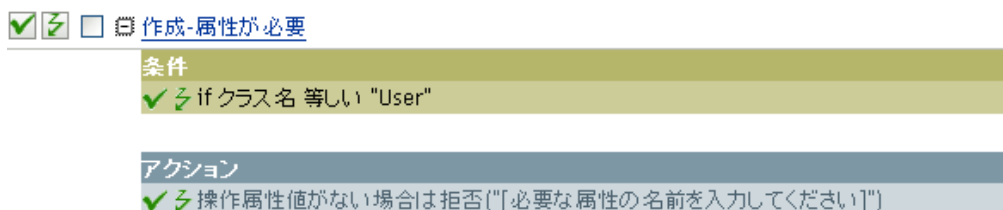
する作成ポリシーがすでにある場合は、55 ページの「事前定義されたルールのインポート」へ進みます。

6.3.1 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 発行者チャンネルまたは購読者チャン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。
- 5 55 ページのセクション 6.3.2 「事前定義されたルールのインポート」に進みます。

6.3.2 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 ルールを編集するには、ルールビルダで [作成 - 属性が必要] をクリックします。
- 4 [必要な属性の名前を入力してください] を [名前を入力] フィールドから削除します。
- 5 [参照] ボタンをクリックして、作成するユーザオブジェクトに必要な属性を参照し、選択します。
- 6 (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。
- 7 [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。
- 8 [OK] を 2 回クリックします。

6.3.3 ルールの動作

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。

6.4 作成 - 発行者 - テンプレートの使用

このルールでは、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なります。このルールは、ドライブ内の発行者作成ポリシーに実装します。

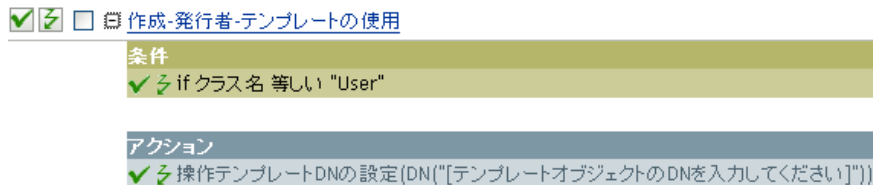
この事前定義されたルールを使用するには、作成ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの2つの手順を実行します。このルールに追加する作成ポリシーがすでにある場合は、[56 ページの「事前定義されたルールのインポート」](#)へ進みます。

6.4.1 ポリシーの作成

- 1 管理するドライブの [Identity Manager ドライバの概要] を開きます。
- 2 発行者チャンネルまたは購読者チャン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。
- 5 [56 ページのセクション 6.4.2 「事前定義されたルールのインポート」](#) に進みます。

6.4.2 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 ルールを編集するには、ルールビルダで [作成 - 発行者 - テンプレートを使用] をクリックします。
- 4 [テンプレートオブジェクトのDNを入力してください] を [DNを入力] フィールドから削除します。
- 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] ボタンをクリックし、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。

6.4.3 ルールの動作

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ド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ます。ユーザオブジェクトの作成時に、Identity Manager 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。

6.5 作成 - デフォルト属性値の設定

このルールでは、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。

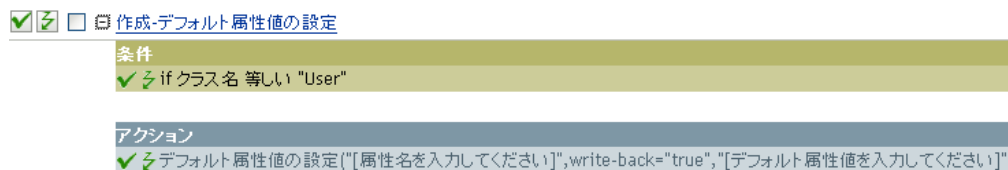
この事前定義されたルールを使用するには、作成ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの2つの手順を実行します。このルールに追加する作成ポリシーがすでにある場合は、[57 ページの「事前定義されたルールのインポート」](#)へ進みます。

6.5.1 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 発行者チャンネルまたは購読者チャン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。
- 5 [57 ページのセクション 6.5.2 「事前定義されたルールのインポート」](#) に進みます。

6.5.2 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 ルールを編集するには、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] を [属性名の入力] フィールドから削除します。

- 5 [参照] ボタンをクリックして、作成する属性を参照して選択します。
- 6 [デフォルト属性値を入力してください] を [引き数値を入力] フィールドから削除します。
- 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。

6.5.3 ルールの動作

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。

6.6 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、このルールで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。

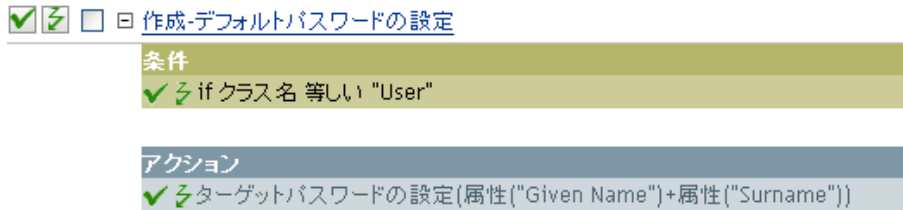
この事前定義されたルールを使用するには、作成ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの2つの手順を実行します。このルールに追加する作成ポリシーがすでにある場合は、[58 ページの「事前定義されたルールのインポート」](#)へ進みます。

6.6.1 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 発行者チャンネルまたは購読者チャン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。
- 5 [58 ページのセクション 6.6.2 「事前定義されたルールのインポート」](#) に進みます。

6.6.2 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。

6.6.3 ルールの動作

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。

6.7 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

このルールでは、特定のサブツリーの外で発生するすべてのイベントが除外されます。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、[59 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。

6.7.1 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 発行者チャンネルまたは購読者チャン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。
- 5 [59 ページのセクション 6.7.2 「事前定義されたルールのインポート」](#) に進みます。

6.7.2 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。

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

条件

if ソースDN サブツリー内になし "[組み込むサブツリーを入力してください]"

アクション

拒否()

- 3 ルールを編集するには、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ポルトを参照し、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。

6.7.3 ルールの動作

このルールは、識別ポ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。イベントが識別ポ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、**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。

6.8 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

このルールは、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、**60 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。

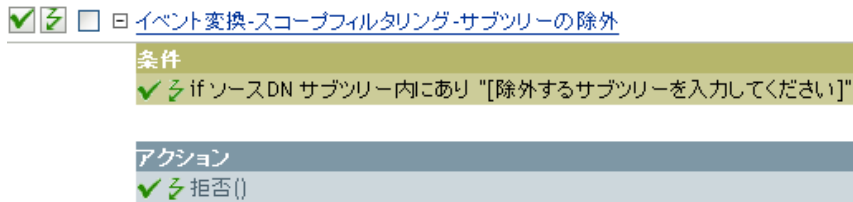
6.8.1 ポリシーの作成

- 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 発行者チャンネルまたは購読者チャン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。
- 5 **60 ページのセクション 6.8.2 「事前定義されたルールのインポート」**に進みます。

6.8.2 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 ルールを編集するには、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ポールドを参照し、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。

6.8.3 ルールの動作

このルールは、識別ポールド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。識別ポールド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せん。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。

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

このルールでは電話番号の形式が変換されます。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。

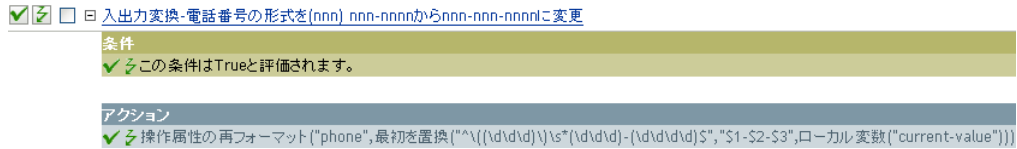
この事前定義されたルールを使用するには、入出力変換ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する入出力変換ポリシーがすでにある場合は、**62 ページの「事前定義されたルールのインポート」**へ進みます。

6.9.1 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 発行者チャンネルまたは購読者チャン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。
- 5 **62 ページのセクション 6.9.2 「事前定義されたルールのインポート」**に進みます。

6.9.2 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 ルールを編集するには、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] をクリックします。
- 4 電話番号の形式変更が実行されるときに条件を定義します。
- 5 [OK] をクリックします。

6.9.3 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。現在の操作での電話番号の属性値で、(nnn) nnn-nnnn に一致するパターンがすべて検索され、nnn-~~nnn~~-nnnn に置き換えられます。

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

このルールでは電話番号の形式が変換されます。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。

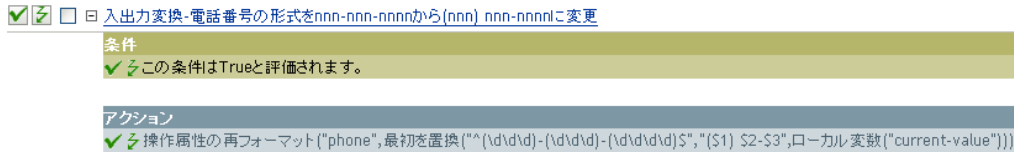
この事前定義されたルールを使用するには、入出力変換ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの2つの手順を実行します。このルールに追加する入出力変換ポリシーがすでにある場合は、63 ページの「事前定義されたルールのインポート」へ進みます。

6.10.1 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。
- 5 63 ページのセクション 6.10.2 「事前定義されたルールのインポート」に進みます。

6.10.2 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 ルールを編集するには、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] をクリックします。
- 4 電話番号の形式変更が実行される時の条件を定義します。
- 5 [OK] をクリックします。

6.10.3 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。現在の操作での電話番号の属性値で、(nnn) nnn-~~nnnn~~ に一致するパターンがすべて検索され、~~nnn-nnn-nnnn~~ に置き換えられます。

6.11 一致 - 発行者 (ミラーリング)

このルールでは、識別ボ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が、名前と場所に基づいて検索されます。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、63 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。

6.11.1 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 発行者チャン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。
- 5 63 ページのセクション 6.11.2 「事前定義されたルールのインポート」に進みます。

6.11.2 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。

一致-発行者(ミラーリング)

条件

✓ if ソースDN サブツリー内にあり "[ソース階層のベースを入力してください]"

アクション

✓ ローカル変数の設定("dest-base","[宛先階層のベースを入力してください]")

✓ 一致オブジェクトの検索(スコープ="entry",DN(ローカル変数("dest-base")+"\")+一致しないソースDN(変換="true"))

- 3 ルールを編集するには、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 [ソース階層のベースを入力してください] を [値] フィールドから削除します。
- 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 [宛先階層のベースを入力してください] を [文字列を入力] フィールドから削除します。
- 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] ボタンをクリックして、ソース構造を一致させるターゲット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。

6.11.3 ルールの動作

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 が識別ポールド内に作成されます。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 (識別ポールド) のサブツリーを指定する必要があります。

6.12 一致 - 購読者 (ミラーリング) - LDAP 形式

このルールでは、LDAP 形式の DN を使用している接続システム内で、識別ポールド内のオブジェクトに一致するものが、名前と場所に基づいて検索されます。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、[65 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。

6.12.1 ポリシーの作成

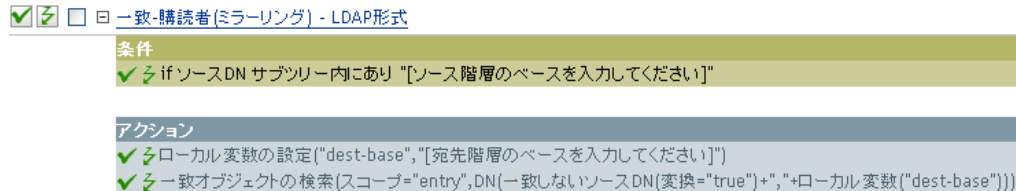
- 1 管理するドライバの [Identity Manager ドライバの概要] を開きます。
- 2 購読者チャンネルの一致ポリシーオブジェクトをクリックします。
- 3 [挿入] をクリックします。
- 4 ポリシーの名前を指定し、このポリシーをポリシービルダで実装することを確認したら、[OK] をクリックします。

ルールビルダが起動します。

- 5 [65 ページのセクション 6.12.2 「事前定義されたルールのインポート」](#)に進みます。

6.12.2 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 ルールを編集するには、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、[ターゲット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] ボタンをクリックして、ソース構造を一致させるターゲット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。

6.12.3 ルールの動作

特定のソースサブツリー内にある識別ボ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。

6.13 一致 - 属性値別

このルールでは、一致するオブジェクトが特定の属性値で検索されます。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。

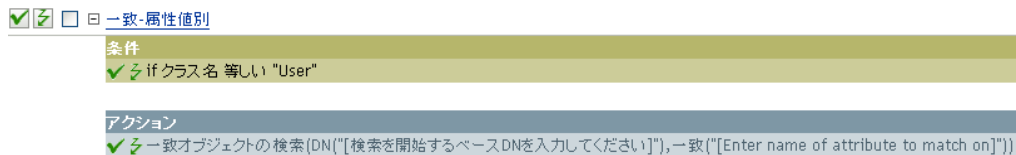
この事前定義されたルールを使用するには、一致ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する一致ポリシーがすでにある場合は、[66 ページの「事前定義されたルールのインポート」](#)へ進みます。

6.13.1 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 発行者チャンネルまたは購読者チャン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。
- 5 66 ページのセクション 6.13.2 「事前定義されたルールのインポート」に進みます。

6.13.2 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 ルールを編集するには、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] ボタンをクリックして、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 参照ボタンをクリックして、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。

6.13.3 ルールの動作

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がターゲットデータストア内で検索されます。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。

6.14 配置 - 発行者 (ミラーリング)

このルールでは、接続システムの名前と場所に基づき、識別ボールド内にオブジェクトが配置されます。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。

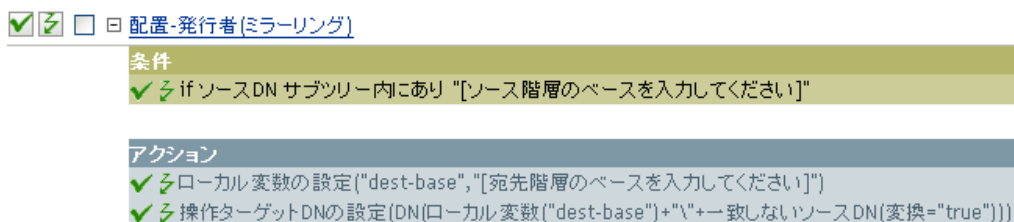
この事前定義されたルールを使用するには、配置ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの2つの手順を実行します。このルールに追加する配置ポリシーがすでにある場合は、67 ページの「事前定義されたルールのインポート」へ進みます。

6.14.1 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 発行者チャン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。
- 5 67 ページのセクション 6.14.2 「事前定義されたルールのインポート」に進みます。

6.14.2 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 ルールを編集するには、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 [ソース階層のベースを入力してください] を [値] フィールドから削除します。
- 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、[ターゲット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] ボタンをクリックして、オブジェクトを配置するターゲット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。

6.14.3 ルールの動作

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ド内に配置されます。ソース (接続システム) の DN およびターゲット (識別ボールド) のサブツリーを指定する必要があります。

6.15 配置 - 購読者 (ミラーリング)-LDAP 形式

このルールは、指定したポイントから識別ボールド内のミラー化された構造を使用して、データストア内にオブジェクトを配置します。このルールは、ドライバ内の配置ポリシーに実装します。このルールは、購読者チャン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、[68 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。

6.15.1 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 購読者チャン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。
- 5 [68 ページのセクション 6.15.2 「事前定義されたルールのインポート」](#)に進みます。

6.15.2 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。

[配置 - 購読者 \(ミラーリング\) - LDAP形式](#)

条件
✓ <input checked="" type="checkbox"/> if ソース DN サブツリー内にあり "[ソース階層のベースを入力してください]"

アクション
✓ <input checked="" type="checkbox"/> ローカル変数の設定 ("dest-base", "[宛先階層のベースを入力してください])"
✓ <input checked="" type="checkbox"/> 操作ターゲット DN の設定 (DN [一致しないソース DN (変換="true")+"; "+ローカル変数 ("dest-base")])

- 3 ルールを編集するには、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 [ソース階層のベースを入力してください] を [値] フィールドから削除します。
- 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、[ターゲット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] ボタンをクリックして、オブジェクトを配置するターゲット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。

6.15.3 ルールの動作

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ド内に配置されます。ソース (識別ボールド) 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。

6.16 配置 - 発行者 (フラット)

このルールは、データストアのオブジェクトを識別ボールドの 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、[69 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。

6.16.1 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 発行者チャン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。
- 5 [69 ページのセクション 6.16.2 「事前定義されたルールのインポート」](#)に進みます。

6.16.2 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 ルールを編集するには、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、[ターゲットコンテナの 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] ボタンをクリックして、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。

6.16.3 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 が `dest-base`(CN 属性)に設定されます。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。

6.17 配置 - 購読者 (フラット)-LDAP 形式

このルールは、識別ポール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、[70 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。

6.17.1 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 購読者チャン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。
- 5 [70 ページのセクション 6.17.2 「事前定義されたルールのインポート」](#)に進みます。

6.17.2 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 ルールを編集するには、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、[ターゲットコンテナの 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。

6.17.3 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、ターゲット DN が `uid=` 一意の名前、`dest-base` に設定されます。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。

6.18 配置 - 部署別発行者

このルールでは、OU 属性の値に基づいて、オブジェクトが、データストア内の 1 つのコンテナから識別ボルト内の複数のコンテナ内に配置されます。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、[71 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。

6.18.1 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 発行者チャン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。
- 5 [71 ページのセクション 6.18.2 「事前定義されたルールのインポート」](#)に進みます。

6.18.2 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 ルールを編集するには、[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、[ターゲット組織の 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、参照ボタンをクリックして、識別ボールド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。

6.18.3 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が正しい部署別コンテナに配置されます。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。

6.19 配置 - 部署別購読者 -LDAP 形式

このルールでは、OU 属性に基づいて、オブジェクトが、識別ボールド内の 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されます。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、購読者チャン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、[72 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。

6.19.1 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 購読者チャン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。
- 5 [72 ページのセクション 6.19.2「事前定義されたルールのインポート」](#)に進みます。

6.19.2 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 ルールを編集するには、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、[ターゲット組織の 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。

6.19.3 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が正しい部署別コンテナに配置されます。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。

リソースオブジェクトに情報を保存する

リソースオブジェクトはドライバが使用する情報を保存します。リソースオブジェクトは任意のデータを任意の形式で保持できます。Novell® Identity Manager 3.5.1 は、異なるタイプのオブジェクトを含みます。

- ◆ 75 ページのセクション 7.1 「ライブラリオブジェクト」
- ◆ 81 ページのセクション 7.2 「マッピングテーブルオブジェクト」
- ◆ 84 ページのセクション 7.3 「ECMA Script」
- ◆ 84 ページのセクション 7.4 「アプリケーションオブジェクト」
- ◆ 84 ページのセクション 7.5 「リポジトリオブジェクト」
- ◆ 84 ページのセクション 7.6 「ソースオブジェクト」

7.1 ライブラリオブジェクト

ライブラリオブジェクトは、複数のポリシーおよび 1 つまたは複数のドライバによって共有されているその他のリソースを保存します。ライブラリオブジェクトは、ドライバセットオブジェクトまたは任意の eDirectory™ コンテナ内で作成できます。eDirectory ツリーには複数のライブラリが存在できます。ドライバは、そのドライバが動作しているサーバがライブラリオブジェクトの読み書き可能レプリカまたはマスタレプリカを保持している限り、ツリー内のどのライブラリでも参照できます。

ライブラリにスタイルシート、ポリシー、ルール、その他のリソースオブジェクトを保存して、これを 1 つ以上のドライバに参照させることができます。

- ◆ 75 ページのセクション 7.1.1 「ライブラリの管理」
- ◆ 77 ページのセクション 7.1.2 「ライブラリにオブジェクトを追加する」
- ◆ 81 ページのセクション 7.1.3 「ライブラリに保存されたポリシーを使用する」

7.1.1 ライブラリの管理

ライブラリの新規作成、および既存のライブラリの削除と検索は、iManager から行えます。

- ◆ 75 ページの 「ライブラリの作成」
- ◆ 76 ページの 「ライブラリの削除」
- ◆ 76 ページの 「既存のライブラリを検索する」

ライブラリの作成

- 1 iManager で、[Identity Manager] > [Identity Manager の概要] の順に選択します。
- 2 [検索] をクリックして、ツリー内のすべてのドライバを検索します。
- 3 [Identity Manager の概要] で、[ライブラリオブジェクト] をクリックします。

Identity Managerの概要

1個のドライブセットが次の場所に見つかりました: ディレクトリ全体
0ライブラリオブジェクト 検索された場所: ディレクトリ全体

- 4 [新規] をクリックします。

ライブラリ管理

ライブラリは見つかりませんでした。[新規]を選択してください。

- 5 ライブラリの名前を指定します。
- 6 ライブラリを作成するコンテナを参照して、選択します。
- 7 [OK] をクリックします。

ライブラリの削除

- 1 iManager で、[Identity Manager] > [Identity Manager の概要] の順に選択します。
- 2 [検索] をクリックして、ツリー内のすべてのドライブを検索します。
- 3 [Identity Manager の概要] で、[ライブラリオブジェクト] をクリックします。

Identity Managerの概要

1個のドライブセットが次の場所に見つかりました: ディレクトリ全体
0ライブラリオブジェクト 検索された場所: ディレクトリ全体

- 4 削除するライブラリを選択して、[削除] をクリックします。
- 5 [はい] をクリックして、削除操作を確定します。

既存のライブラリを検索する

- 1 iManager で、[Identity Manager] > [Identity Manager の概要] の順に選択します。

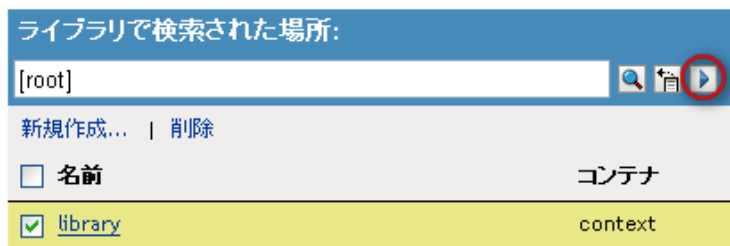
- 2 [検索] をクリックして、ツリー内のすべてのドライバを検索します。
- 3 [Identity Manager の概要] で、[ライブラリオブジェクト] をクリックします。

Identity Managerの概要

1個のドライバセットが次の場所に見つかりました: ディレクトリ全体
 0ライブラリオブジェクト 検索された場所: ディレクトリ全体

- 4 検索を開始するコンテナを参照して、選択します。
- 5 検索を実行するには、矢印アイコンをクリックします。

ライブラリ管理



7.1.2 ライブラリにオブジェクトを追加する

ライブラリへは、ポリシー、マッピングテーブルおよび資格情報プロビジョニングポリシーのリソースオブジェクトを追加できます。

- 77 ページの「ライブラリにポリシーを追加する」
- 79 ページの「マッピングテーブルをライブラリに追加する」
- 80 ページの「ライブラリに資格情報プロビジョニングポリシーのソースオブジェクトを追加する」

ライブラリにポリシーを追加する

- 1 iManager で、[Identity Manager] > [Identity Manager の概要] の順に選択します。
- 2 [検索] をクリックして、ツリー内のすべてのドライバを検索します。
- 3 [Identity Manager の概要] で、[ライブラリオブジェクト] をクリックします。




Identity Managerの概要

1個のドライバセットが次の場所に見つかりました: ディレクトリ全体
 0ライブラリオブジェクト 検索された場所: ディレクトリ全体

- 4 ポリシーを追加するライブラリをクリックします。

ライブラリ管理

ライブラリで検索された場所:

[root]   

新規作成... | 削除

<input type="checkbox"/> 名前	コンテナ
<input checked="" type="checkbox"/> library	context


- 5 [ポリシー] タブをクリックし、プラスアイコンをクリックしてライブラリにポリシーを追加します。

Identity Managerライブラリ

ライブラリ: library.context

ポリシー | マッピングテーブル | 資格情報のプロビジョニング

次のポリシーはこのドライブで見つかりました:
(ポリシーのルールリストを取得するには、ポリシー名の左のイメージをクリックします。)

DirXML-Library 

削除



- 6 ポリシーの名前を入力します。
- 7 ポリシーの実装方法を選択し、[OK] をクリックします。

ポリシーの作成

新しいポリシーで使用される名前を入力します。

作成

ポリシーを作成するコンテナを選択します。

library.context  



このポリシーをどのように実装しますか?

ポリシービルダ

XSLT

ECMAScript

既存のポリシーからコピーを作成する
コピーするポリシーを選択します。

OK | キャンセル

- 7a [ポリシービルダ]、[XSLT]、または [ECMAScript] を選択する場合には、ライブラリにオブジェクトが作成され、表示されます。各オブジェクトを編集して、ポリシー情報をオブジェクトに追加する必要があります。

- 7b [既存のポリシーからコピーを作成する] を選択する場合には、ライブラリに保存するポリシーを参照して選択します。

マッピングテーブルをライブラリに追加する

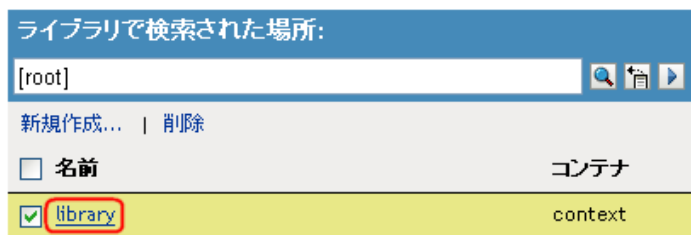
- 1 iManager で、[Identity Manager] > [Identity Manager の概要] の順に選択します。
- 2 [検索] をクリックして、ツリー内のすべてのドライバを検索します。
- 3 [Identity Manager の概要] で、[ライブラリオブジェクト] をクリックします。

Identity Manager の概要

1個のドライバセットが次の場所に見つかりました: ディレクトリ全体
0 ライブラリオブジェクト 検索された場所: ディレクトリ全体

- 4 マッピングテーブルを追加するライブラリをクリックします。

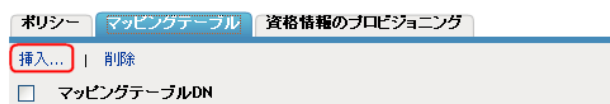
ライブラリ管理



- 5 [マッピングテーブル] タブをクリックし、[挿入] をクリックして、ライブラリにマッピングテーブルを追加します。

Identity Manager ライブラリ

ライブラリ: library.context



マッピングテーブルは見つかりませんでした。[挿入]を選択してください。

- 6 マッピングテーブルの名前を指定します。
- 7 マッピングテーブルを作成するライブラリを参照して選択します。
- 8 [OK] をクリックします。
- 9 列の追加アイコンをクリックします。
- 10 列の値を指定してから、値が「大文字と小文字の区別あり」、「大文字と小文字の区別なし」、または数値なのかを指定します。
- 11 行の追加アイコンをクリックします。
- 12 行の値を指定します。
- 13 [適用] をクリックしてマッピングテーブルを保存し、エディタでの作業を続行します。
または

[OK] をクリックしてマッピングテーブルを保存し、エディタを閉じます。

マッピングテーブルの詳細については、[81 ページのセクション 7.2 「マッピングテーブルオブジェクト」](#)を参照してください。

ライブラリに資格情報プロビジョニングポリシーのソースオブジェクトを追加する

- 1 iManager で、[\[Identity Manager\]](#) > [\[Identity Manager の概要\]](#) の順に選択します。
- 2 [検索] をクリックして、ツリー内のすべてのドライバを検索します。
- 3 [\[Identity Manager の概要\]](#) で、[\[ライブラリオブジェクト\]](#) をクリックします。

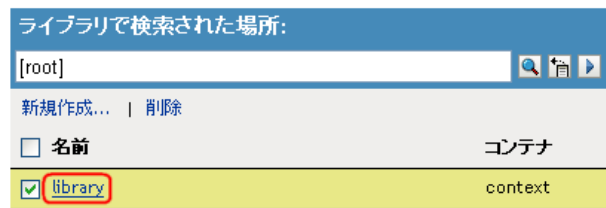
Identity Manager の概要

1個のドライバセットが次の場所に見つかりました: ディレクトリ全体

[0 ライブラリオブジェクト](#) 検索された場所: ディレクトリ全体

- 4 資格情報プロビジョニングポリシーのリソースオブジェクトを追加するライブラリをクリックします。

ライブラリ管理



- 5 [資格情報のプロビジョニング] タブをクリックします。
- 6 [リポジトリ] をクリックしてから [新規] をクリックして、リポジトリオブジェクトをライブラリに追加します。

または

[アプリケーション] をクリックしてから [新規] をクリックして、アプリケーションオブジェクトをライブラリに追加します。

Identity Manager ライブラリ

ライブラリ: library.context



リポジトリは見つかりませんでした - [新規作成] を選択してください

7.1.3 ライブラリに保存されたポリシーを使用する

ライブラリオブジェクトには、何度も使用される情報が保存されています。こうした情報は、複数のドライバによって使用されたり、同じドライバによって何度も使用されたりします。ライブラリ内に保存されたポリシーを使用するには、次の手順に従います。

- 1 iManager で、*[Identity Manager]* > *[Identity Manager の概要]* の順に選択します。
- 2 [検索] をクリックして、ツリー内のすべてのドライバを検索します。
- 3 ライブラリに保存されたポリシーを使用するドライバを参照してクリックします。
- 4 保存された情報を使用するポリシーを選択し、[挿入] をクリックしてポリシーを作成したら、**ステップ 5**に進みます。
または
[編集] をクリックして、既存のポリシーにポリシーを追加したら、**ステップ 8**に進みます。
- 5 [既存のポリシーを使用する] を選択します。
- 6 ライブラリに保存されたポリシーを参照して選択し、[OK] をクリックします。
- 7 [閉じる] をクリックします。
- 8 [挿入] > [DirXML Script を含むポリシーへの参照を追加します] の順にクリックします。
- 9 ライブラリに保存されたポリシーを参照して選択し、[OK] を 2 回クリックします。
- 10 [閉じる] をクリックします。

7.2 マッピングテーブルオブジェクト

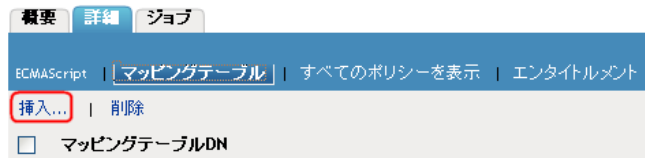
マッピングテーブルオブジェクトは、値セットを別の対応する値セットにマップするために使用されます。マッピングテーブルオブジェクトが作成された後に、**(258 ページ) マップトークン**はマッピングテーブル内に指定した値からの指定したトークンの結果をマップします。

マッピングテーブルオブジェクトを使用するには、次の手順を完了する必要があります。

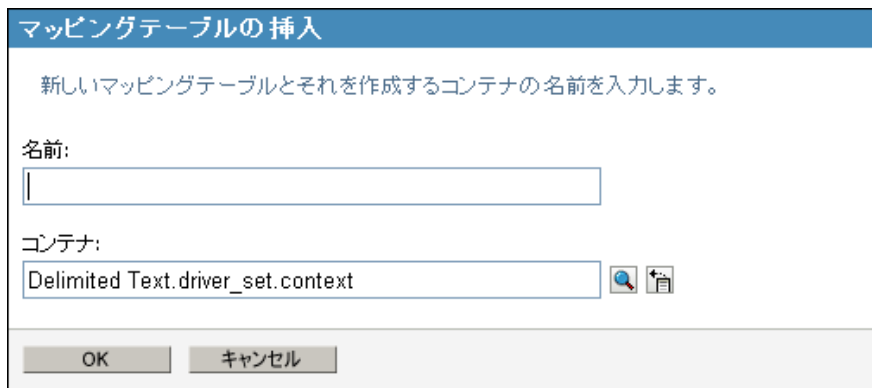
1. **81 ページのセクション 7.2.1 「マッピングテーブルオブジェクトを作成する」**
2. **83 ページのセクション 7.2.2 「マッピングテーブルオブジェクトをポリシーに追加する」**

7.2.1 マッピングテーブルオブジェクトを作成する

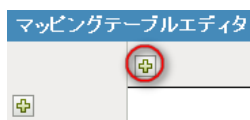
- 1 iManager で、*[Identity Manager]* > *[Identity Manager の概要]* の順に選択します。
- 2 [検索] をクリックして、ツリー内のすべてのドライバを検索します。
- 3 マッピングテーブルを作成するドライバを参照してクリックします。
- 4 [詳細] > [マッピングテーブル] の順に選択し、[挿入] をクリックします。



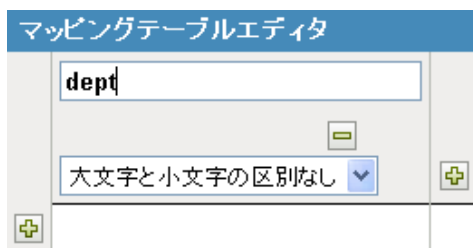
- 5 マッピングテーブルオブジェクトの名前を指定します。
- 6 マッピングテーブルを作成するコンテナを参照して選択し、[OK] をクリックします。



- 7 列の追加アイコンをクリックします。

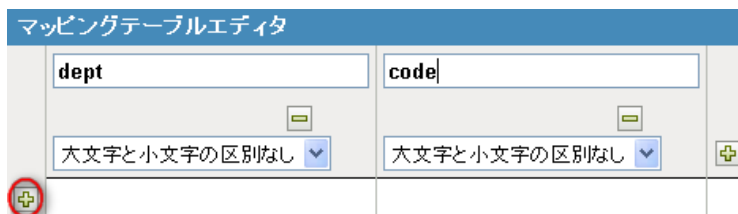


- 8 列の名前を指定し、値のタイプとして [大文字と小文字の区別なし]、[大文字と小文字の区別あり] または [数字] のいずれかを選択します。



列をさらに追加する場合は、列の追加アイコンをクリックします。

- 9 行の追加アイコンをクリックします。



10 行の値を指定します。

マッピングテーブルエディタ		
dept	code	location
大文字と小文字の区別なし	大文字と小文字の区別なし	大文字と小文字の区別なし
Engineering	00001	New York
Sales	00002	London
Accounting	00003	Paris
Marketing	00004	Rome

行をさらに追加する場合は、行の追加アイコンをクリックします。

11 [OK] をクリックしてマッピングテーブルを保存し、マッピングテーブルエディタを閉じます。

7.2.2 マッピングテーブルオブジェクトをポリシーに追加する

- 1 マッピングテーブルを使用するポリシーを作成するか、既存のポリシーを選択して編集します。
- 2 ポリシービルダ内で引数ビルダを起動します。
- 3 [動詞] のリストから [マップ] を選択して、[追加] をクリックします。
- 4 [エディタ] フィールドで、[81 ページのセクション 7.2.1 「マッピングテーブルオブジェクトを作成する」](#) で作成したマッピングテーブルオブジェクトを参照して選択します。
- 5 ソースカラム名を指定します。
- 6 ターゲットカラム名を指定します。

エディタ	
マッピングテーブルDN: *	..¥Departments
ソース列の名前: *	dept
宛先の列の名前: *	code

7 目的の結果にするための名詞を選択したら、[OK] をクリックして引数を保存します。

マッピングテーブルは、この時点でどのような方法でも利用できます。この例では、OU 属性が、マッピングテーブルから派生した値で投入されます。

```
マップ(table="..¥Departments",dest="code",src="dept")
| 品 操作属性("OU")
```

[マップ] トークンは動詞トークンです。動詞トークンが機能するには、従うべき名詞トークンが必要です。

7.3 ECMA Script

ECMAScript オブジェクトは、ECMAScripts を格納するリソースオブジェクトです。ECMAScript は、ポリシーおよびスタイルシートによって使用されます。ECMAScript の詳細については、[85 ページの第 8 章「ポリシーでの ECMAScript の使用」](#)を参照してください。

7.4 アプリケーションオブジェクト

アプリケーションオブジェクトは、Novell 資格情報プロビジョニングポリシーの一部です。アプリケーションには、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。アプリケーションオブジェクトについての詳細は、『[Novell Credential Provisioning Policies for Identity Manager 3.5.1](#)』を参照してください。

7.5 リポジトリオブジェクト

リポジトリオブジェクトは、Novell 資格情報プロビジョニングポリシーの一部です。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。リポジトリオブジェクトについての詳細は、『[Novell Credential Provisioning Policies for Identity Manager 3.5.1](#)』を参照してください。

7.6 ソースオブジェクト

リソースオブジェクトには、ポリシーで使用される情報を保存できます。これはテキストまたは XML フォーマットで保存された任意の情報を使用できます。リソースオブジェクトは、ライブラリまたはドライバオブジェクト内に保存されます。リソースオブジェクトの使用例は、複数のドライバが同じ定数パラメータセットを必要としたときです。リソースオブジェクトは、パラメータを保存し、ドライバはこれらのパラメータをいつでも使用します。

現時点では、リソースオブジェクトの作成は Designer から行います。詳細については、『[“Policies in Designer 2.1”](#)』の「[Storing Information in Resource Objects](#)」を参照してください。

ポリシーでの ECMAScript の使用

8

ECMAScript は、Ecma International によって標準化されたスクリプトプログラミング言語です。しばしば引き合いに出される JavaScript や JScript は、ECMAScript のサブセットです。Identity Manager 3.5.1 は、ECMAScript オブジェクトという新しいオブジェクトタイプをサポートします。ECMAScript オブジェクトは、ECMAScripts を格納するリソースオブジェクトです。ECMAScript は、ポリシーを介して呼び出され、DirXML スクリプトや XSLT スタイルシートでは提供できない高度な機能を実現します。

この節では、ECMAScript エディタの使用法、ECMAScript をポリシーと共に使用する方法、および ECMAScript をカスタムフォームと共に使用する方法について説明します。ECMAScript 言語については説明しません。ECMAScript 言語を使用する方法の詳細については、「[ECMAScript 言語の仕様 \(http://www.ecma-international.org/publications/standards/Ecma-262.htm\)](http://www.ecma-international.org/publications/standards/Ecma-262.htm)」を参照してください。

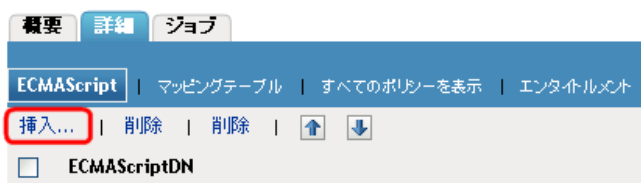
8.1 ECMAScript の作成

ECMAScript は、ドライブ上またはライブラリ内に保存されます。

- 85 ページのセクション 8.1.1 「ドライブ内に ECMAScript を作成する」
- 86 ページのセクション 8.1.2 「ライブラリ内に ECMAScript を作成する」

8.1.1 ドライブ内に ECMAScript を作成する

- 1 iManager で、`[Identity Manager] > [Identity Manager Overview]` の順に選択します。
- 2 [検索] をクリックして、ツリー内のすべてのドライブを検索します。
- 3 ECMAScript を作成するドライブを参照してクリックします。
- 4 [詳細] > [ECMAScript] の順に選択し、[挿入] をクリックします。



- 5 [新しい ECMAScript の作成] を選択します。
- 6 ECMAScript の名前を指定します。
- 7 ECMAScript を保存するドライブを参照して選択し、[OK] をクリックします。



- 8 [ECMAScript の編集を有効にする] をクリックして、ECMAScript を入力します。
ECMAScript が記述されたファイルがすでにあり、これを使用したい場合は、そのファイルをテキストエディタで開き、内容を ECMAScript エディタにコピーします。
- 9 [適用] をクリックして、ECMAScript エディタ内の情報を保存します。
または
[OK] をクリックして変更内容を保存し、ECMAScript エディタを終了します。

8.1.2 ライブラリ内に ECMAScript を作成する

- 1 iManager で、[Identity Manager] > [Identity Manager Overview] の順に選択します。
- 2 [検索] をクリックして、ツリー内のすべてのドライバを検索します。
- 3 [Identity Manager の概要] で、[ライブラリオブジェクト] をクリックします。

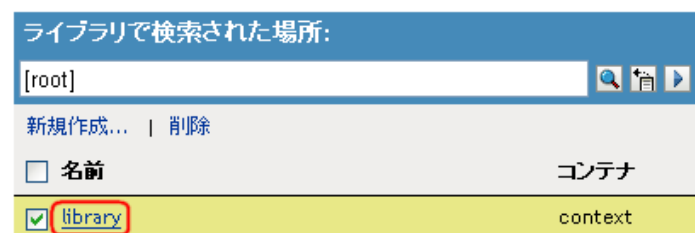
Identity Manager の概要

1個のドライバセットが次の場所に見つかりました: ディレクトリ全体

0 ライブラリオブジェクト 検索された場所: ディレクトリ全体

- 4 ECMAScript を追加するライブラリをクリックします。

ライブラリ管理




- 5 [ポリシー] タブをクリックし、プラスアイコンをクリックします。

Identity Managerライブラリ

ライブラリ: library.context

ポリシー マッピングテーブル 資格情報のプロビジョニング

次のポリシーはこのドライブで見つかりました:
(ポリシーのルールリストを取得するには、ポリシー名の左のイメージをクリックします。)

DirXML-Library 

削除



- 6 ECMAScript の名前を指定します。
- 7 [ECMAScript] を選択し、[OK] をクリックします。

ポリシーの作成

新しいポリシーで使用される名前を入力します。

Join

ポリシーを作成するコンテナを選択します。

library.context  



このポリシーをどのように実装しますか?

ポリシービルダ

XSLT

ECMAScript

既存のポリシーからコピーを作成する
コピーするポリシーを選択します。


 


OK キャンセル


- 8 ライブラリに保存されているポリシーのリストで、ECMAScript をクリックします。


ポリシー マッピングテーブル 資格情報のプロビジョニング


次のポリシーはこのドライブで見つかりました:
(ポリシーのルールリストを取得するには、ポリシー名の左のイメージをクリックします。)

DirXML-Library 

 Join

 copy

 ecma

 xslt

削除

- 9 [ECMAScript の編集を有効にする] を選択して、ECMAScript を入力します。
ECMAScript が記述されたファイルがすでにあり、これを使用したい場合は、そのファイルをテキストエディタで開き、内容を ECMAScript エディタにコピーします。
- 10 [適用] をクリックして、ECMAScript エディタ内の情報を保存します。
または
[OK] をクリックして変更内容を保存し、ECMAScript エディタを終了します。

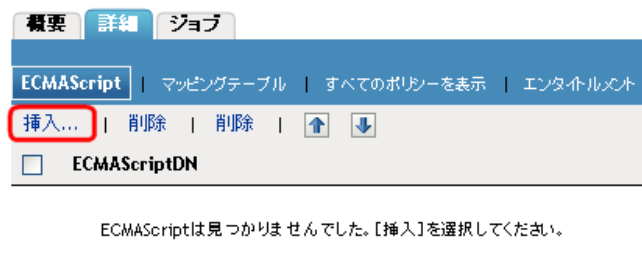
8.2 既存の ECMAScript の使用

Identity Manager 内に既存の ECMAScript がある場合、そのオブジェクトを新しい場所にコピーできます。既存の ECMAScript は、ドライバまたはライブラリにコピーできます。

- ◆ 88 ページのセクション 8.2.1 「既存の ECMAScript をドライバで使用する」
- ◆ 89 ページのセクション 8.2.2 「既存の ECMAScript をライブラリで使用する」

8.2.1 既存の ECMAScript をドライバで使用する

- 1 iManager で、[Identity Manager] > [Identity Manager Overview] の順に選択します。
- 2 [検索] をクリックして、ツリー内のすべてのドライバを検索します。
- 3 ECMAScript をコピーするドライバを参照してクリックします。
- 4 [詳細] > [ECMAScript] の順に選択し、[挿入] をクリックします。



- 5 [既存の ECMAScript の使用] を選択します。
- 6 ECMAScript を参照して選択します。
- 7 [OK] をクリックします。



8.2.2 既存の ECMAScript をライブラリで使用する

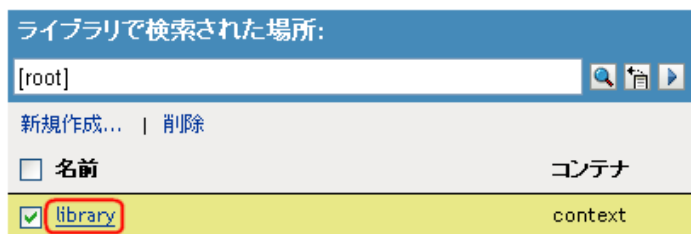
- 1 iManager で、[Identity Manager] > [Identity Manager Overview] の順に選択します。
- 2 [検索] をクリックして、ツリー内のすべてのドライバを検索します。
- 3 [Identity Manager の概要] で、[ライブラリオブジェクト] をクリックします。

Identity Manager の概要

1個のドライバセットが次の場所に見つかりました: ディレクトリ全体
 ライブラリオブジェクト 検索された場所: ディレクトリ全体

- 4 既存の ECMAScript を追加するライブラリをクリックします。

ライブラリ管理



- 5 [ポリシー] タブをクリックし、プラスアイコンをクリックします。

Identity Managerライブラリ

ライブラリ: library.context

ポリシー マッピングテーブル 資格情報のプロビジョニング

次のポリシーはこのドライブで見つかりました:
(ポリシーのルールリストを取得するには、ポリシー名の左のイメージをクリックします。)





- 6 [既存のポリシーからコピーを作成する] を選択します。
- 7 既存の ECMAScript を参照して選択してから、[OK] をクリックします。

ポリシーの作成

新しいポリシーで使用される名前を入力します。


ポリシーを作成するコンテナを選択します。

このポリシーをどのように実装しますか?

- ポリシービルダ
- XSLT
- ECMAScript
- 既存のポリシーからコピーを作成する

コピーするポリシーを選択します。

OK キャンセル

8.3 ポリシーを備えた ECMAScripts の例

次の例は、さまざまなポリシーを備えた ECMAScript ファイル [demo.js \(../samples/demo.js\)](#) を使用しています。demo.js ファイルは、3つの ECMAScript 関数定義を含みます。

8.3.1 ECMAScript 関数を呼び出す DirXML スクリプトポリシー

DirXML スクリプトポリシーは、ECMAScript 関数 `getB64ImageFromURL()` を呼び出すことで、URL 参照である属性を BASE64 でエンコードされた写真データに変換します。このポリシーは、入力変換ポリシーまたは出力変換ポリシーとして使用できます。

この関数は、URL からイメージを読み込んで、その内容を Base64 でエンコードされた文字列として返します。

```
<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE policy PUBLIC "policy-builder-dtd" "C:\Program Files\Novell\Designer\eclipse\plugins\com.novell.designer.idm.policybuilder_1.2.0.200612180606\DTD\dirxmlscript.dtd"><policy><rule>
```

```
<description>Reformat photo from URL to octet</description>
<conditions/><actions>      <do-reformat-op-attr name="photo"><arg-
value type="octet">      <token-xpath
expression="es:getB64ImageFromURL(string($current-value))"/></arg-
value>      </do-reformat-op-attr></actions></rule></policy>
```

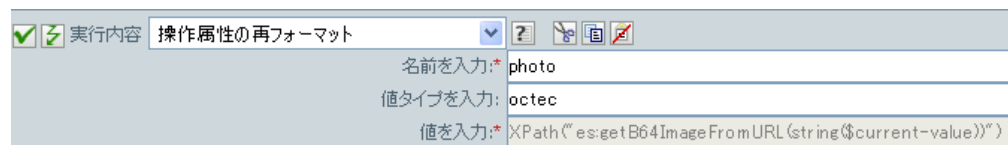
機能 : <static> String getB64ImageFromURL(<String> urlString)

パラメータ : urlString (イメージファイルの URL)

返される値 : イメージの Base64 エンコード化コンテンツ (またはエラーの場合は空の文字列)

ファイル [ReformatPhoto.xml](#) ([../samples/ReformatPhoto.xml](#)) は ECMAScript 関数 getB64ImageFromURL を DirXML Script のポリシーから呼び出します。ファイル [phototest.xml](#) ([../samples/phototest.xml](#)) は、アクション内のポリシーを表示するサンプル入力ドキュメントです。

図 8-1 写真の再フォーマットの例



ECMAScript が getB64ImageFromURL 関数を呼び出すと、現在の値が文字列として返されます。

8.3.2 ECMAScript の関数をドライバレベルで呼び出す XSLT

XSLT ポリシーは、コンマで区切った単一の値を複数の値に分割するか、または複数の値をコンマで区切った単一の値に結合します。XSLT ポリシーはドライバレベルで定義され、入力変換ポリシーまたは出力変換ポリシーとして使用されます。

注 : DirXML スクリプトには分割と結合の機能が組み込まれますが、XSLT には同様の機能はありません。このようなタイプの機能により、XSLT に分割と結合の機能を持たせることができます。

次の 2 つの関数があります。

- ◆ [91 ページの「参加」](#)
- ◆ [92 ページの「分割」](#)

参加

結合機能では、ノードセット内の複数のノードのテキスト値を結合して単一の文字列にします。

```
<!-- template that joins the joinme attribute values into a single
value --><xsl:template match="*[@attr-name='joinme']//*[value] |
*[@attr-name='joinme'] [value]"> <xsl:copy> <xsl:apply-templates
select="*|node()[not(self::value)]"/> <value> <xsl:value-of
select="es:join(value)"/> </value> </xsl:copy></xsl:template>
```

機能: <static> String join(<NodeSet> nodeSet, <string> delimiter)

パラメータ: nodeSet (NodeSet の入力) および delimiter (分割するための区切り記号 (オプション: デフォルト = なし))

返される値: 区切り記号で分割された、nodeSet 内のノードの文字列値の連結。

分割

分割関数は、文字列を NodeSet に分割します。

```
<!-- template that splits the splitme attribute values into multiple
values --><xsl:template match="*[@attr-name='splitme']//value">
<xsl:for-each select="es:split(string(.))"> <value> <xsl:value-
of select="."/> </value> </xsl:for-each></xsl:template>
```

機能: <static> NodeSet split(<String> inputString, <String> delimiter)

パラメータ: inputString (分割するスクリプト) および delimiter ((optional: default = “,”) で分割する区切り記号)

返される値: テキストノードを含む NodeSet。

ファイル [SplitJoin.xml \(../samples/SplitJoin.xml\)](#) は join または split 関数を XSLT スタイルシートで呼び出します。ファイル [splitjointest.xml \(../samples/splitjointest.xml\)](#) は、アクション内のスタイルシートを表示する入力ドキュメントです。

8.3.3 ECMAScript 関数をスタイルシート内で呼び出す XSLT ポリシー

XSLT ポリシーは、XSLT スタイルシートを使用して、埋め込まれた ECMAScript 関数の定義を表現します。この関数は、文字列を大文字に変換します。

```
<!-- define ecmaScript functions --><es:script>function
uppercase(input){return String(input).toUpperCase();}</es:script>
```

ファイル [uppercase.xml \(../samples/uppercase.xml\)](#) は、ECMAScript 関数を XSLT スタイルシートに定義します。ファイル [uppercasetest.xml \(../samples/uppercasetest.xml\)](#) は、アクション内のスタイルシートを表示する入力ドキュメントです。

条件は、アクションが実行されるときに定義されます。条件は、常に [Conjunctive Normal Form \(CNF\)](http://mathworld.wolfram.com/ConjunctiveNormalForm.html) (<http://mathworld.wolfram.com/ConjunctiveNormalForm.html>) または [Disjunctive Normal Form \(DNF\)](http://mathworld.wolfram.com/DisjunctiveNormalForm.html) (<http://mathworld.wolfram.com/DisjunctiveNormalForm.html>) のどちらかで指定されます。これらは論理式形式です。格納されたルールアクションは、CNF または DNF に表される論理式が True と評価されるかまたは条件が指定されていない場合にのみ実行されます。

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。

- ◆ 94 ページの「If 関連付け」
- ◆ 96 ページの「If 属性」
- ◆ 99 ページの「If クラス名」
- ◆ 102 ページの「If ターゲット属性」
- ◆ 105 ページの「If ターゲット DN」
- ◆ 107 ページの「If エンタイトルメント」
- ◆ 110 ページの「If グローバル構成値」
- ◆ 112 ページの「If ローカル変数」
- ◆ 115 ページの「If 名前付きパスワード」
- ◆ 116 ページの「If 操作属性」
- ◆ 119 ページの「If 操作プロパティ」
- ◆ 121 ページの「If 操作」
- ◆ 124 ページの「If パスワード」
- ◆ 127 ページの「If ソース属性」
- ◆ 129 ページの「If ソース DN」
- ◆ 131 ページの「If XML 属性」
- ◆ 133 ページの「If XPath 式」
- ◆ 135 ページの「変数の拡張」

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。実行されるテストのタイプは、操作属性で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。

演算子	次の場合 True が返されます ...
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。
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。
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。
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。
より大きい	指定した比較モードを使用して比較した場合、現在の操作で指定された関連付け値が、条件の内容より大きくなります。
より大きくない	「より大きい」または「等しい」の場合 False が返されます。
より小さい	指定した比較モードを使用して比較した場合、現在の操作で指定された関連付けの値が、条件の内容より小さくなります。
より小さくない	「より小さい」または「等しい」の場合 False が返されます。

値

選択された演算子で定義される値が含まれます。この値は条件によって使用されません。各値で変数の拡張がサポートされています。詳細については、[\(135 ページ\) 変数の拡張](#)を参照してください。値フィールドが含まれる演算子は次のとおりです。

- ◆ 等しい
- ◆ 等しくない
- ◆ より大きくない
- ◆ より小さい
- ◆ より小さくない

比較モード

一部の条件テストには、比較が行われた方法を示すモードパラメータがあります。

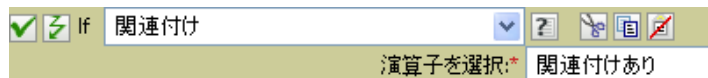
モード	説明
大文字と小文字の 区別あり	1文字ずつ比較する (大文字と小文字の区別あり)。
大文字と小文字の 区別なし	1文字ずつ比較する (大文字と小文字の区別なし)。
正規表現	正規表現が文字列全体に一致します。デフォルトで「大文字と小文字の区別あり」と設定されていても、式の中でエスケープすることで変更できます。 Sun の Web サイト (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html) を参照してください。 [パターン] のオプションには CASE_INSENSITIVE 、 DOTALL 、および 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。
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。
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。
数値	数値で比較します。
バイナリ	バイナリ情報で比較します。

比較モードパラメータが含まれる演算子は次のとおりです。

- ◆ 等しい
- ◆ 等しくない
- ◆ より大きくない
- ◆ より小さい
- ◆ より小さくない

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。



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「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。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。

演算子	次の場合 True が返されます ...
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。
使用不可	「使用可能」の場合 False が返されます。
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。
等しくない	「等しい」の場合 False が返されます。
より大きい	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、条件の内容より大きくなります。
より大きくない	「より大きい」または「等しい」の場合 False が返されます。
より小さい	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、条件の内容より小さくなります。
より小さくない	「より小さい」または「等しい」の場合 False が返されます。

値

選択された演算子で定義される値が含まれます。この値は条件によって使用されます。各値で変数の拡張がサポートされています。詳細については、[\(135 ページ\) 変数の拡張](#)を参照してください。値フィールドが含まれる演算子は次のとおりです。

- ◆ 等しい
- ◆ 等しくない
- ◆ より大きくない
- ◆ より小さい
- ◆ より小さくない

比較モード

一部の条件テストには、比較が行われた方法を示すモードパラメータがあります。

モード	説明
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)。
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	正規表現が文字列全体に一致します。デフォルトで「大文字と小文字の区別あり」と設定されていても、式の中でエスケープすることで変更できます。 Sun の Web サイト (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html) 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。
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。
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。
数値	数値で比較します。
バイナリ	バイナリ情報で比較します。

比較モードパラメータが含まれる演算子は次のとおりです。

- ◆ 等しい
- ◆ 等しくない
- ◆ より大きくない
- ◆ より小さい
- ◆ より小さくない

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 If 属性を使用しています。これは「Policy to Filter Events」というポリシーで、Novell® 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、『[Understanding Policies for Identity Manager 3.5.1](#)』の「[Downloading Identity Manager Policies](#)」を参照してください。ポリシーを XML で表示するには、[001-Event-FilterByContainerDisabledOrTitle.xml \(../samples/001-Event-FilterByContainerDisabledOrTitle.xml\)](#) を参照してください。

Filter events: From Users sub-tree, Users not disabled , no consultants or sales people

条件
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> if ソース DN サブツリー内になし "Users"
OR <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> if 属性 'Login Disabled' 等しい "true"
OR <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> if 属性 'Title' 等しい ".*consultantsales.*"

アクション
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> 拒否()

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。

<input checked="" type="checkbox"/> <input type="checkbox"/> OR If	属性	<input type="button" value="v"/>	<input type="button" value="?"/> <input type="button" value="✂"/> <input type="button" value="📄"/> <input type="button" value="✖"/>
名前を入力:*	Title		
演算子を選択:*	等しい		
モードを比較:	正規表現		
値:	*consultantsales.*		

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。

演算子	次の場合 True が返されます ...
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。
使用不可	「使用可能」の場合 False が返されます。
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。
等しくない	「等しい」の場合 False が返されます。
より大きい	現在の操作に使用可能なオブジェクトクラス名があり、指定された比較モードを使用して比較すると、条件の内容より大きくなります。
より大きくない	「より大きい」または「等しい」の場合 False が返されます。
より小さい	現在の操作に使用可能なオブジェクトクラス名があり、指定された比較モードを使用して比較すると、条件の内容より小さくなります。
より小さくない	「より小さい」または「等しい」の場合 False が返されます。

値

選択された演算子で定義される値が含まれます。この値は条件によって使用されません。各値で変数の拡張がサポートされています。詳細については、[\(135 ページ\) 変数の拡張](#)を参照してください。値フィールドが含まれる演算子は次のとおりです。

- ◆ 等しい
- ◆ 等しくない
- ◆ より大きくない
- ◆ より小さい
- ◆ より小さくない

比較モード

一部の条件テストには、比較が行われた方法を示すモードパラメータがあります。

モード	説明
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)。
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	正規表現が文字列全体に一致します。デフォルトで「大文字と小文字の区別あり」と設定されていても、式の中でエスケープすることで変更できます。 Sun の Web サイト (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html) 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。
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。
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。
数値	数値で比較します。
バイナリ	バイナリ情報で比較します。

比較モードパラメータが含まれる演算子は次のとおりです。

- ◆ 等しい
- ◆ 等しくない
- ◆ より大きくない
- ◆ より小さい
- ◆ より小さくない

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、『*Understanding Policies for Identity Manager 3.5.1*』の「**Downloading Identity Manager Policies**」を参照してください。XML 形式でポリシーを参照するには、[004-Command-GroupChangeOnTitleChange.xml](#) (`./samples/004-Command-GroupChangeOnTitleChange.xml`) を参照してください。

[User changing from Manager to Employee](#)

条件

if クラス名 等しい "User"

AND if ターゲット属性 'Title' 等しい ".*manager.*"

AND if 操作属性 'Title' 等しくない ".*manager.*"

アクション

ターゲット属性値の設定 ("Group Membership", "Users\EmployeesGroup")

操作属性のクローン ("Group Membership", "Security Equals")

[User changing from Employee to Manager](#)

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。

if	クラス名	▼	?	✂	📄	✖
	演算子を選択:	*	等しい			
	モードを比較:		大文字と小文字の区別なし			
	値:		User			

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。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。

演算子	次の場合 True が返されます ...
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。
使用不可	「使用可能」の場合 False が返されます。
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。
等しくない	「等しい」の場合 False が返されます。
より大きい	指定された比較モードを使用して比較した場合に、ターゲットデータストアに、条件のコンテンツより大きい指定した属性の値が使用可能です。モードが「構造」の場合、コンテンツは 1 組の <component> 要素、またはテキストである必要があります。
より大きくない	「より大きい」または「等しい」の場合 False が返されます。
より小さい	指定された比較モードを使用して比較する場合に、ターゲットデータストアに、条件のコンテンツより大きい指定した属性の値が使用可能です。モードが「構造」の場合、コンテンツは 1 組の <component> 要素、またはテキストである必要があります。
より小さくない	「より小さい」または「等しい」の場合 False が返されます。

値

選択された演算子で定義される値が含まれます。この値は条件によって使用されます。各値で変数の拡張がサポートされています。詳細については、[\(135 ページ\) 変数の拡張](#)を参照してください。値フィールドが含まれる演算子は次のとおりです。

- ◆ 等しい
- ◆ 等しくない
- ◆ より大きくない
- ◆ より小さい
- ◆ より小さくない

比較モード

一部の条件テストには、比較が行われた方法を示すモードパラメータがあります。

モード	説明
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)。
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	<p>正規表現が文字列全体に一致します。デフォルトで「大文字と小文字の区別あり」と設定されていても、式の中でエスケープすることで変更できます。</p> <p>Sun の Web サイト (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html) を参照してください。</p> <p>[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。</p>
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。
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。
数値	数値で比較します。
バイナリ	バイナリ情報で比較します。
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。

比較モードパラメータが含まれる演算子は次のとおりです。

- ◆ 等しい
- ◆ 等しくない
- ◆ より大きくない
- ◆ より小さい
- ◆ より小さくない

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、『[Understanding Policies for Identity Manager 3.5.1](#)』の「[Downloading Identity Manager Policies](#)」を参照してください。ポリシーを XML で表示するには、[004-CommandGroupChangeOnTitleChange.xml \(../samples/004-Command-GroupChangeOnTitleChange.xml\)](#) を参照してください。

[User changing from Manager to Employee](#)

条件
 if クラス名 等しい "User"
AND if ターゲット属性 'Title' 等しい ".*manager.*"
AND if 操作属性 'Title' 等しくない ".*manager.*"

アクション
 ターゲット属性値の設定 ("Group Membership", "Users\EmployeesGroup")
 操作属性のクローン ("Group Membership", "Security Equals")

[User changing from Employee to Manager](#)

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。

AND if ターゲット属性

属性名を入力してください:*	Title
演算子を選択:*	等しい
モードを比較:	正規表現
値:	.*manager.*

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。

演算子	次の場合 True が返されます ...
使用可能	使用可能なターゲット DN があります。
使用不可	「使用可能」の場合 False が返されます。
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。
等しくない	「等しい」の場合 False が返されます。
コンテナ内	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。
コンテナにない	「コンテナ内」の場合 False が返されます。
サブツリー内	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。
サブツリーにない	「サブツリー内」の場合 False が返されます。

値

選択された演算子で定義される値が含まれます。この値は条件によって使用されません。各値で変数の拡張がサポートされています。詳細については、[\(135 ページ\) 変数の拡張](#)を参照してください。値フィールドが含まれる演算子は次のとおりです。

- ◆ 等しい
- ◆ 等しくない
- ◆ コンテナ内
- ◆ コンテナにない
- ◆ サブツリー内
- ◆ サブツリーにない

例

<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	If	ターゲットDN	▼	?	✂	📄	✖
				演算子を選択:*	使用可能			
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	AND If	ターゲットDN	▼	?	✂	📄	✖
				演算子を選択:*	コンテナ内にあり			
				値:	Novell® Users			

If エンタイトルメント

現在の操作または識別ポ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。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

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。

演算子	次の場合 True が返されます ...
使用可能	現在の操作または識別ポールトで、指定したエンタイトルメントを使用できます。
使用不可	「使用可能」の場合 False が返されます。
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。
等しくない	「等しい」の場合 False が返されます。
より大きい	指定したエンタイトルメントが、現在の操作または識別ポールトで使用可能で許可されており、指定された比較モードを使用して比較すると、条件の内容より大きくなる値があります。
より大きくない	「より大きい」または「等しい」の場合 False が返されます。
より小さい	指定したエンタイトルメントが、現在の操作または識別ポールトで使用可能で許可されており、指定された比較モードを使用して比較すると、条件の内容より小さくなる値があります。
より小さくない	「より小さい」または「等しい」の場合 False が返されます。
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。
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。
削除指定の変更なし	「変更あり」の場合 False が返されます。
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。
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。

値

選択された演算子で定義される値が含まれます。この値は条件によって使用されます。各値で変数の拡張がサポートされています。詳細については、[\(135 ページ\) 変数の拡張](#)を参照してください。値フィールドが含まれる演算子は次のとおりです。

- ◆ 等しい
- ◆ 等しくない
- ◆ 追加指定の変更あり
- ◆ 削除指定の変更あり
- ◆ 追加指定の変更なし
- ◆ 削除指定の変更なし
- ◆ より大きい
- ◆ より大きくない
- ◆ より小さい
- ◆ より小さくない

比較モード

一部の条件テストには、比較が行われた方法を示すモードパラメータがあります。

モード	説明
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)。
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	正規表現が文字列全体に一致します。デフォルトで「大文字と小文字の区別あり」と設定されていても、式の中でエスケープすることで変更できます。 Sun の Web サイト (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html) を参照してください。 [パターン] のオプションには <code>CASE_INSENSITIVE</code> 、 <code>DOTALL</code> 、および <code>UNICODE_CASE</code> 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。
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。
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。
数値	数値で比較します。
バイナリ	バイナリ情報で比較します。

比較モードパラメータが含まれる演算子は次のとおりです。

- ◆ 等しい
- ◆ 等しくない
- ◆ 追加指定の変更あり
- ◆ 削除指定の変更あり

- ◆ 追加指定の変更なし
- ◆ 削除指定の変更なし
- ◆ より大きい
- ◆ より大きくない
- ◆ より小さい
- ◆ より小さくない

例

✓ ↺ If	エンタイトルメント	▼ ? ✂ 📄 ✖
名前を入力:*	notes-group	
演算子を選択:*	追加指定の変更あり	
モードを比較:	大文字と小文字の区別なし	
値:	Sales	

If グローバル構成値

グローバル構成値をテストします。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。

注

ポリシーで変数を使用する詳細については、「[ポリシーコンポーネントの理解 \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/b6yi6f6.html\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/b6yi6f6.html)」を参照してください。

フィールド

名前

選択した条件をテストするグローバル値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。

演算子	次の場合 True が返されます ...
使用可能	指定した名前のグローバル構成値があります。
使用不可	「使用可能」の場合 False が返されます。
等しい	指定した名前のグローバル設定値があり、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。
等しくない	「等しい」の場合 False が返されます。
より大きい	指定した名前のグローバル構成値があり、その値が、指定された比較モードを使用して比較すると、条件の内容より大きくなります。
より大きくない	「より大きい」または「等しい」の場合 False が返されます。
より小さい	指定した名前のグローバル構成値があり、その値が、指定された比較モードを使用して比較する場合に条件の内容より小さくなります。
より小さくない	「より小さい」または「等しい」の場合 False が返されます。

値

選択された演算子で定義される値が含まれます。この値は条件によって使用されません。各値で変数の拡張がサポートされています。詳細については、[\(135 ページ\) 変数の拡張](#)を参照してください。値フィールドが含まれる演算子は次のとおりです。

- ◆ 等しい
- ◆ 等しくない
- ◆ より大きい
- ◆ より大きくない
- ◆ より小さい
- ◆ より小さくない

比較モード

一部の条件テストには、比較が行われた方法を示すモードパラメータがあります。

モード	説明
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)。
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	正規表現が文字列全体に一致します。デフォルトで「大文字と小文字の区別あり」と設定されていますが、式の中でエスケープすることで変更できます。 Sun の Web サイト (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html) 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。
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。
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。
数値	数値で比較します。
バイナリ	バイナリ情報で比較します。

比較モードパラメータが含まれる演算子は次のとおりです。

- ◆ 等しい
- ◆ 等しくない
- ◆ より大きい
- ◆ より大きくない
- ◆ より小さい
- ◆ より小さくない

例

The screenshot shows a configuration window titled 'グローバル設定値' (Global Settings). It contains two input fields: '名前を入力:' (Name) with the value 'enforce-password-policy' and '演算子を選択:' (Select operator) with the value '使用可能' (Use possible). The interface includes standard icons for help, search, and save.

If ローカル変数

ローカル変数をテストします。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

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。

演算子	次の場合 True が返されます ...
使用可能	ポリシー内にある以前のルール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。
使用不可	「使用可能」の場合 False が返されます。
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。
等しくない	「等しい」の場合 False が返されます。
より大きい	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、条件の内容より大きくなります。
より大きくない	「より大きい」または「等しい」の場合 False が返されます。
より小さい	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、条件の内容より小さくなります。
より小さくない	「より小さい」または「等しい」の場合 False が返されます。

値

選択された演算子で定義される値が含まれます。この値は条件によって使用されません。各値で変数の拡張がサポートされています。詳細については、[\(135 ページ\) 変数の拡張](#)を参照してください。値フィールドが含まれる演算子は次のとおりです。

- ◆ 等しい
- ◆ 等しくない
- ◆ より大きい
- ◆ より大きくない
- ◆ より小さい
- ◆ より小さくない

比較モード

一部の条件テストには、比較が行われた方法を示すモードパラメータがあります。

モード	説明
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)。
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	<p>正規表現が文字列全体に一致します。デフォルトで「大文字と小文字の区別あり」と設定されていても、式の中でエスケープすることで変更できます。</p> <p>Sun の Web サイト (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html) を参照してください。</p> <p>[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。</p>
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。
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。
数値	数値で比較します。
バイナリ	バイナリ情報で比較します。

比較モードパラメータが含まれる演算子は次のとおりです。

- ◆ 等しい
- ◆ 等しくない
- ◆ より大きい
- ◆ より大きくない
- ◆ より小さい
- ◆ より小さくない

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、『**Understanding Policies for Identity Manager 3.5.1**』の「**Downloading Identity Manager Policies**」を参照してください。XML でポリシーを表示するには、**003-Command-AddCreate-Groups.xml** (../samples/003-Command-AddCreateGroups.xml) を参照してください。

[Set local variables to test existence of groups and for placement](#)
 [Create ManagersGroup, if needed](#)

条件
 if ローカル変数 'manager-group-info' 使用可能
AND if ローカル変数 'manager-group-info' 等しくない "group"

アクション
 ターゲットオブジェクトの追加(クラス名="Group",when="before",DN(ローカル変数("manager-group-dn")))

[Create EmployeesGroup, if needed](#)
 [If Title indicates Manager, add to ManagerGroup and set rights](#)
 [If Title does not indicate Manager, add to EmployeeGroup and set rights](#)

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。

[Set local variables to test existence of groups and for placement](#)

条件
 if クラス名 等しい "User"
AND
 if 操作 等しい "add"
OR if 操作 等しい "modify"

アクション
 ローカル変数の設定("manager-group-dn","Users\ManagersGroup")
 ローカル変数の設定("manager-group-info",ターゲット属性("Object Class",DN(ローカル変数("manager-group-dn"))))
 ローカル変数の設定("employee-group-dn","Users\EmployeesGroup")
 ローカル変数の設定("employee-group-info",ターゲット属性("Object Class",DN(ローカル変数("employee-group-dn"))))

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。

<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> If	ローカル変数	名前を入力:*	manager-group-info
		演算子を選択:*	使用可能
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> AND If	ローカル変数	名前を入力:*	manager-group-info
		演算子を選択:*	等しくない
		モードを比較:	大文字と小文字の区別なし
		値:	group

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を確認されます。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。

If 名前付きパスワード

現在の操作にあるドライバからの名前付きパスワードを、指定された名前でテストします。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

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。

演算子	次の場合 True が返されます ...
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

if 名前付きパスワード

名前を入力:* workflow-admin

演算子を選択:* 使用可能

If 操作属性

現在の操作で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。

演算子	次の場合 True が返されます ...
使用可能	値の削除以外の現在の操作に、指定した属性で使用できる値があります。
使用不可	「使用可能」の場合 False が返されます。
等しい	値の削除以外の現在の操作に、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。
等しくない	「等しい」の場合 False が返されます。
より大きい	現在の操作に指定された属性で値の削除以外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場合に、条件の内容より大きくなります。モードが「構造」の場合、コンテンツは 1 組の <component> 要素、またはテキストである必要があります。
より大きくない	「より大きい」または「等しい」の場合 False が返されます。
より小さい	現在の操作に指定された属性で値の削除以外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場合に、条件の内容より小さくなります。モードが「構造」の場合は、内容は 1 組の <component> 要素であるか、テキストである必要があります。
より小さくない	「より小さい」または「等しい」の場合 False が返されます。
変更あり	現在の操作に、指定した属性の値の削除以外の変更があります。
変更なし	「変更あり」の場合 False が返されます。
削除指定の変更あり	現在の操作に、指定した属性の値の削除以外の値を削除する変更があります。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。
削除指定の変更なし	「変更あり」の場合 False が返されます。
追加指定の変更あり	現在の操作に、指定した属性の値を削除する以外に値を追加する変更があります。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。
追加指定の変更なし	「追加指定の変更あり」の場合 False が返されます。

値

選択された演算子で定義される値が含まれます。この値は条件によって使用されます。各値で変数の拡張がサポートされています。詳細については、[\(135 ページ\) 変数の拡張](#)を参照してください。値フィールドが含まれる演算子は次のとおりです。

- ◆ 等しい
- ◆ 等しくない
- ◆ 追加指定の変更あり
- ◆ 削除指定の変更あり
- ◆ 追加指定の変更なし
- ◆ 削除指定の変更なし
- ◆ より大きい
- ◆ より大きくない
- ◆ より小さい
- ◆ より小さくない

比較モード

一部の条件テストには、比較が行われた方法を示すモードパラメータがあります。

モード	説明
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)。
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	正規表現が文字列全体に一致します。デフォルトで「大文字と小文字の区別あり」と設定されていても、式の中でエスケープすることで変更できます。 Sun の Web サイト (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html) を参照してください。 [パターン] のオプションには <code>CASE_INSENSITIVE</code> 、 <code>DOTALL</code> 、および <code>UNICODE_CASE</code> 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。
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。
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。
数値	数値で比較します。
バイナリ	バイナリ情報で比較します。
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。

比較モードパラメータが含まれる演算子は次のとおりです。

- ◆ 等しい
- ◆ 等しくない

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。操作プロパティは、操作内の <operation-data> 要素に属性として格納された名前付きの値です。通常、操作の結果を処理するポリシーによって必要とされること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。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

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。

演算子	次の場合 True が返されます ...
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。
使用不可	「使用可能」の場合 False が返されます。
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。
等しくない	「等しい」の場合 False が返されます。
より大きい	指定した名前の操作プロパティが現在の操作にあり、指定された比較モードを使用して比較すると、その値が条件のコンテンツより大きくなります。
より大きくない	「より大きい」または「等しい」の場合 False が返されます。
より小さい	指定した名前の操作プロパティが現在の操作にあり、指定された比較モードを使用して比較すると、その値が条件のコンテンツより小さくなります。
より小さくない	「より小さい」または「等しい」の場合 False が返されます。

値

選択された演算子で定義される値が含まれます。この値は条件によって使用されません。各値で変数の拡張がサポートされています。詳細については、[\(135 ページ\) 変数の拡張](#)を参照してください。値フィールドが含まれる演算子は次のとおりです。

- ◆ 等しい
- ◆ 等しくない
- ◆ より大きい
- ◆ より大きくない
- ◆ より小さい
- ◆ より小さくない

比較モード

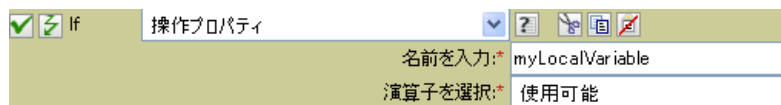
一部の条件テストには、比較が行われた方法を示すモードパラメータがあります。

モード	説明
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)。
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	正規表現が文字列全体に一致します。デフォルトで「大文字と小文字の区別あり」と設定されていますが、式の中でエスケープすることで変更できます。 Sun の Web サイト (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html) 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。
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。
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。
数値	数値で比較します。
バイナリ	バイナリ情報で比較します。

比較モードパラメータが含まれる演算子は次のとおりです。

- ◆ 等しい
- ◆ 等しくない
- ◆ より大きい
- ◆ より大きくない
- ◆ より小さい
- ◆ より小さくない

例



If 操作

現在の操作の名前をテストします。実行されるテストのタイプは、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。

演算子	次の場合に True を返します ...
等しい	現在の操作名は、指定した比較モードを使用して比較した場合、条件のコンテンツと等しくなります。
等しくない	「等しい」の場合 False が返されます。
より大きい	現在の操作名は、指定した比較モードを使用して比較した場合、条件のコンテンツより大きくなります。
より大きくない	「より大きい」の場合 False が返されます。
より小さい	現在の操作名は、指定した比較モードを使用して比較した場合、条件の内容より小さくなります。
より小さくない	「より小さい」の場合 False が返されます。

値

選択された演算子で定義される値が含まれます。この値は条件によって使用されます。各値で変数の拡張がサポートされています。詳細については、[\(135 ページ\) 変数の拡張](#)を参照してください。値フィールドが含まれる演算子は次のとおりです。

- ◆ 等しい
- ◆ 等しくない
- ◆ より大きい
- ◆ より大きくない
- ◆ より小さい
- ◆ より小さくない

値は、メタディレクトリエンジンが検索する操作です。

- ◆ 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
- ◆ 生成されたパスワード
- ◆ 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
- ◆ パスワード
- ◆ クエリー
- ◆ クエリスキーマ
- ◆ 関連付けの削除
- ◆ 名前を変更する
- ◆ スキーマ定義
- ◆ ステータス
- ◆ 同期

このリストは排他的ではありません。カスタム操作はドライバおよび管理者によって実装できます。

比較モード

一部の条件テストには、比較が行われた方法を示すモードパラメータがあります。

モード	説明
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)。
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	<p>正規表現が文字列全体に一致します。デフォルトで「大文字と小文字の区別あり」と設定されていても、式の中でエスケープすることで変更できます。</p> <p>Sun の Web サイト (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html) を参照してください。</p> <p>[パターン] のオプションには <code>CASE_INSENSITIVE</code>、<code>DOTALL</code>、および <code>UNICODE_CASE</code> 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。</p>
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。
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。
数値	数値で比較します。
バイナリ	バイナリ情報で比較します。

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」というポリシー名で、Novell 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、『[Understanding Policies for Identity Manager 3.5.1](#)』の「[Downloading Identity Manager](#)」

Policies”」を参照してください。XML 形式でポリシーを参照するには、[003-Command-AddCreateGroups.xml](#) (../samples/003-Command-AddCreateGroups.xml) を参照してください。

Set local variables to test existence of groups and for placement

条件

- ✓ if クラス名 等しい "User"

AND

- ✓ if 操作 等しい "add"
- OR ✓ if 操作 等しい "modify"

アクション

- ✓ ローカル変数の設定 ("manager-group-dn", "Users\ManagersGroup")
- ✓ ローカル変数の設定 ("manager-group-info", ターゲット属性 ("Object Class", DN(ローカル変数 ("manager-group-dn"))))
- ✓ ローカル変数の設定 ("employee-group-dn", "Users\EmployeesGroup")
- ✓ ローカル変数の設定 ("employee-group-info", ターゲット属性 ("Object Class", DN(ローカル変数 ("employee-group-dn"))))

If 操作

演算子を選択:* 等しい

モードを比較: 大文字と小文字の区別なし

値: add

OR If 操作

演算子を選択:* 等しい

モードを比較: 大文字と小文字の区別なし

値: modify

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。

If パスワード

現在の操作のパスワードをテストします。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。

演算子	次の場合 True が返されます ...
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。
等しい	現在の操作内に使用可能なパスワードがあり、その値が、指定した比較モードを使用して比較すると、条件の内容と等しくなります。
等しくない	「等しい」の場合 False が返されます。
より大きい	現在の操作内に使用可能なパスワードがあり、その値が、指定した比較モードを使用して比較すると、条件の内容より大きくなります。
より大きくない	「より大きい」または「等しい」の場合 False が返されます。
より小さい	現在の操作内に使用可能なパスワードがあり、その値が、指定した比較モードを使用して比較すると、条件の内容より小さくなります。
より小さくない	「より小さい」または「等しい」の場合 False が返されます。

値

選択された演算子で定義される値が含まれます。この値は条件によって使用されません。各値で変数の拡張がサポートされています。詳細については、[\(135 ページ\) 変数の拡張](#)を参照してください。値フィールドが含まれる演算子は次のとおりです。

- ◆ 等しい
- ◆ 等しくない
- ◆ より大きい
- ◆ より大きくない
- ◆ より小さい
- ◆ より小さくない

比較モード

一部の条件テストには、比較が行われた方法を示すモードパラメータがあります。

モード	説明
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)。
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	<p>正規表現が文字列全体に一致します。デフォルトで「大文字と小文字の区別あり」と設定されていても、式の中でエスケープすることで変更できます。</p> <p>Sun の Web サイト (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html) を参照してください。</p> <p>[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。</p>
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。
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。
数値	数値で比較します。
バイナリ	バイナリ情報で比較します。

比較モードパラメータが含まれる演算子は次のとおりです。

- ◆ 等しい
- ◆ 等しくない
- ◆ より大きい
- ◆ より大きくない
- ◆ より小さい
- ◆ より小さくない

例

Novell Credential Provisioning Policies for Identity Manager 3.5.1 を実装している場合は、パスワードの条件を使用する購読者コマンド変換ポリシーのサンプルが用意されています。サンプルファイルは `SampleSubCommandTransform.xml` という名前です。Identity Manager メディアの `DirXML Utilities` フォルダに格納されています。詳細については、「[資格情報のプロビジョニングポリシーのサンプル](#)」(『*Novell Credential Provisioning Policies for Identity Manager 3.5.1*』を参照してください。XML 形式でポリシーを参照するには、`SampleSubCommandTransform.xml` (`./samples/SampleSubCommandTransform.xml`) を参照してください。

購読者コマンド変換ポリシーは、オブジェクトが追加されたときにパスワードが利用可能かどうかを確認します。パスワードが利用可能な場合は、Novell SecureLogin および Novell SecretStore[®] 資格情報はプロビジョニングされます。

- [Add operation-data element to password subscribe operation \(if needed\)](#)
- [Add payload data to modify-password subscribe operations](#)
- [Add payload data to add subscribe operations](#)

条件

- if 操作 等しい "add"
- AND if パスワード: 使用可能

アクション

- XML要素の追加("sso-sync-data","operation-data")
- XML要素の追加("sso-target-user-dn","operation-data/sso-sync-data")
- XMLテキストの追加("operation-data/sso-sync-data/sso-target-user-dn",Source Attribute (DirXML-ADContext))
- XML要素の追加("sso-app-username","operation-data/sso-sync-data")
- XMLテキストの追加("operation-data/sso-sync-data/sso-app-username", Source Attribute ("CN"))
- XML要素の追加("password","operation-data/sso-sync-data")
- XMLテキストの追加("operation-data/sso-sync-data/password", Password())
- XML要素の追加("nsl-set-passphrase-answer","operation-data/sso-sync-data")
- XMLテキストの追加("operation-data/sso-sync-data/nsl-set-passphrase-answer",Source Attribute ("workforceID"))

If

演算子を選択:*

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。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

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。

演算子	次の場合 True が返されます ...
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。 .
使用不可	「使用可能」の場合 False が返されます。
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。
等しくない	「等しい」の場合 False が返されます。
より大きい	指定した属性で使用可能な値がソースデータストアにあります。この値は指定した比較モードを使用して比較すると、条件の内容より大きくなります。モードが「構造」の場合、内容は 1 組のコンポーネントか、テキストである必要があります。
より大きくない	「より大きい」または「等しい」の場合 False が返されます。
より小さい	指定した属性で使用可能な値がソースデータストアにあります。この値は指定した比較モードを使用して比較すると、条件の内容より小さくなります。モードが「構造」の場合、内容は 1 組のコンポーネントか、テキストである必要があります。
より小さくない	「より小さい」または「等しい」の場合 False が返されます。

値

選択された演算子で定義される値が含まれます。この値は条件によって使用されません。各値で変数の拡張がサポートされています。詳細については、[\(135 ページ\) 変数の拡張](#)を参照してください。値フィールドが含まれる演算子は次のとおりです。

- ◆ 等しい
- ◆ 等しくない
- ◆ より大きい
- ◆ より大きくない
- ◆ より小さい
- ◆ より小さくない

比較モード

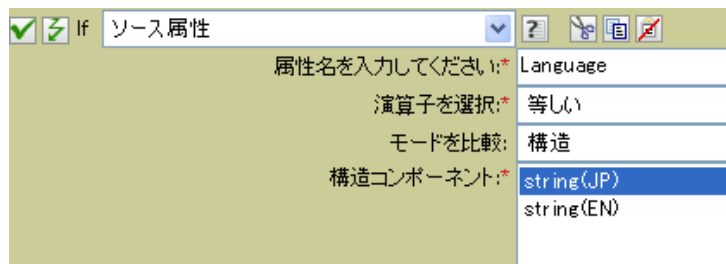
一部の条件テストには、比較が行われた方法を示すモードパラメータがあります。

モード	説明
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)。
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	正規表現が文字列全体に一致します。デフォルトで「大文字と小文字の区別あり」と設定されていても、式の中でエスケープすることで変更できます。 Sun の Web サイト (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html) 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。
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。
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。
数値	数値で比較します。
バイナリ	バイナリ情報で比較します。
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。

比較モードパラメータが含まれる演算子は次のとおりです。

- ◆ 等しい
- ◆ 等しくない
- ◆ より大きい
- ◆ より大きくない
- ◆ より小さい
- ◆ より小さくない

例



If ソース DN

現在の操作でソース 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。

演算子	次の場合 True が返されます ...
使用可能	使用可能なソース DN があります。
使用不可	「使用可能」の場合 False が返されます。
等しい	ソース DN が使用可能で、コンテナ内に指定されている値のコンテンツと等しくなっています。
等しくない	「等しい」の場合 False が返されます。
コンテナ内	使用可能なソース DN があり、ソースデータストアの DN 形式に適したセマンティックを使用して比較する場合に If ソース DN の内容で指定されたコンテナ内のオブジェクトを示す。
コンテナにない	「コンテナ内」の場合 False が返されます。
サブツリー内	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。
サブツリーにない	「サブツリー内」の場合 False が返されます。

値

選択された演算子で定義される値が含まれます。この値は条件によって使用されます。各値で変数の拡張がサポートされています。詳細については、[\(135 ページ\) 変数の拡張](#)を参照してください。値フィールドが含まれる演算子は次のとおりです。

- ◆ 等しい
- ◆ 等しくない
- ◆ コンテナ内
- ◆ コンテナにない
- ◆ サブツリー内
- ◆ サブツリーにない

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。ルールは、Identity Manager とともに提供される事前定義されたルールです。詳細については、[60 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。XML 形式でポリシーを表示するには、[prefdef_transformation_filter_exclude_subtrees.xml \(../samples/prefdef_transformation_filter_exclude_subtrees.xml\)](#) を参照してください。

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

条件
 if ソースDN サブツリー内にあり "[除外するサブツリーを入力してください]"

アクション
 拒否()

If ソースDN

演算子を選択:* サブツリー内にあり
 値: Users

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。

If XML 属性

現在の操作の XML 属性に対してテストを実行します。実行されるテストのタイプは、操作属性で指定された演算子によって異なります。

フィールド

名前

XML 属性の名前を指定します。XML 属性は、XDS ドキュメントの要素に関連付けられている名前 / 値のペアです。

演算子

条件のテストタイプを選択します。

演算子	次の場合 True が返されます ...
使用可能	現在の操作に、指定した名前の XML 属性があります。
使用不可	「使用可能」の場合 False が返されます。
等しい	指定した名前の XML 属性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、条件の内容と等しくなります。
等しくない	「等しい」の場合 False が返されます。
より大きい	指定した名前の XML 属性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、条件の内容より大きくなります。
より大きくない	「より大きい」または「等しい」の場合 False が返されます。
より小さい	指定した比較モードを使用して比較した場合、現在の操作で指定された関連付けの値が、条件の内容より小さくなります。
より小さくない	「より小さい」または「等しい」の場合 False が返されます。

値

選択された演算子で定義される値が含まれます。この値は条件によって使用されません。各値で変数の拡張がサポートされています。詳細については、[\(135 ページ\) 変数の拡張](#)を参照してください。値フィールドが含まれる演算子は次のとおりです。

- ◆ 等しい
- ◆ 等しくない
- ◆ より大きい
- ◆ より大きくない
- ◆ より小さい
- ◆ より小さくない

比較モード

一部の条件テストには、比較が行われた方法を示すモードパラメータがあります。

モード	説明
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)。
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	<p>正規表現が文字列全体に一致します。デフォルトで「大文字と小文字の区別あり」と設定されていても、式の中でエスケープすることで変更できます。</p> <p>Sun の Web サイト (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html) を参照してください。</p> <p>[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。</p>
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。
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。
数値	数値で比較します。
バイナリ	バイナリ情報で比較します。

比較モードパラメータが含まれる演算子は次のとおりです。

- ◆ 等しい
- ◆ 等しくない
- ◆ より大きい
- ◆ より大きくない
- ◆ より小さい
- ◆ より小さくない

例

The screenshot shows a configuration window for an XML attribute. The title bar includes a checkmark icon, a refresh icon, and the text 'If XML属性'. The main area contains two fields: '名前を入力:*' (Name) with the value 'from-merge' and '演算子を選択:*' (Select operator) with the value '使用可能' (Allowed). The window also features a dropdown arrow, a help icon, a search icon, a refresh icon, and a close icon.

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。

演算子	次の場合に True を返します ...
True	XPath 式が True に評価されます。
True ではない	「True」の場合 False が返されます。

備考

ポリシーとともに XPath 式を使用する方法の詳細については、『“ [Understanding Policies for Identity Manager 3.5.1](#) ”』の「[XPath 1.0 Expressions](#)」を参照してください。

例

[Novell Credential Provisioning Policies for Identity Manager 3.5.1](#) を実装している場合は、XPath 式の条件を使用するサンプルの購読者コマンド変換ポリシーが用意されています。サンプルファイルは `SampleSubCommandTransform.xml` という名前です。Identity Manager メディアの `DirXML Utilities` フォルダに格納されています。詳細については、『“ [資格情報のプロビジョニングポリシーのサンプル](#) ”』（『[Novell Credential Provisioning Policies for Identity Manager 3.5.1](#)』）を参照してください。XML 形式でポリシーを参照するには、[SampleSubCommandTransform.xml \(../samples/SampleSubCommandTransform.xml\)](#) を参照してください。

サンプル資格情報プロビジョニングポリシーは、それぞれの追加操作をチェックして、追加に関連付けられている操作データがあるかどうかを確認します。操作データがない場合は、Novell SecureLogin および Novell SecretStore 資格情報がプロビジョニングされます。

[Add operation-data element to password subscribe operation \(if needed\)](#)

条件

- if 操作 等しい "add"
- AND if パスワード: 使用可能
- AND if XPath式 Trueではない "operation-data"

OR

- if 操作 等しい "modify-password"
- AND if XPath式 Trueではない "operation-data"

アクション

- XMLエレメントの追加("operation-data", ".")

[Add payload data to modify-password subscribe operations](#)

[Add payload data to add subscribe operations](#)



変数の拡張

条件での動的変数の使用を許可。

注

多くの条件は、属性またはコンテンツ内での動的変数拡張をサポートしています。サポートされている場合は、`<variable-name>$` に組み込まれた参照が、5つの名前をもつローカル変数またはグローバル変数の値で置き換えられます。「`<variable-name>$`」は有効な変数名である必要があります。有効な XML 名の詳細については、「[W3C Extensible Markup Language \(http://www.w3.org/TR/2006/REC-xml11-20060816/#sec-suggested-names\)](http://www.w3.org/TR/2006/REC-xml11-20060816/#sec-suggested-names)」を参照してください。

アクションは格納したルールの条件が満たされた場合にのみ実行されます。一部のアクションには [モード] フィールドがあります。ポリシーが実行されているコンテキストが選択されたモードに準拠していない場合、モードは実行時に無視されます。

このセクションではポリシービルダのインタフェースで使用可能なすべてのアクションに関する詳細情報を掲載しています。

- ◆ 139 ページの「関連付けの追加」
- ◆ 140 ページの「ターゲット属性値の追加」
- ◆ 142 ページの「ターゲットオブジェクトの追加」
- ◆ 144 ページの「ソース属性値の追加」
- ◆ 145 ページの「ソースオブジェクトの追加」
- ◆ 146 ページの「XML 要素の追加」
- ◆ 148 ページの「XML テキストの追加」
- ◆ 150 ページの「中断」
- ◆ 151 ページの「ターゲット属性値のクリア」
- ◆ 152 ページの「操作プロパティのクリア」
- ◆ 153 ページの「SSO 資格情報のクリア」
- ◆ 154 ページの「ソース属性値のクリア」
- ◆ 155 ページの「XPath 式によるクローン」
- ◆ 156 ページの「操作属性のクローン」
- ◆ 157 ページの「ターゲットオブジェクトの削除」
- ◆ 158 ページの「ソースオブジェクトの削除」
- ◆ 159 ページの「一致オブジェクトの検索」
- ◆ 161 ページの「毎」
- ◆ 162 ページの「イベントの生成」
- ◆ 165 ページの「If」
- ◆ 167 ページの「エンタイトルメントの実装」
- ◆ 168 ページの「ターゲットオブジェクトの移動」
- ◆ 170 ページの「ソースオブジェクトの移動」
- ◆ 171 ページの「操作属性の再フォーマット」
- ◆ 172 ページの「関連付けの削除」
- ◆ 173 ページの「ターゲット属性値の削除」
- ◆ 174 ページの「ソース属性値の削除」
- ◆ 175 ページの「ターゲットオブジェクトの名前変更」
- ◆ 176 ページの「操作属性の名前変更」
- ◆ 177 ページの「ソースオブジェクトの名前変更」

- ◆ 178 ページの「電子メールの送信」
- ◆ 180 ページの「テンプレートから電子メールを送信」
- ◆ 182 ページの「デフォルト属性値の設定」
- ◆ 183 ページの「ターゲット属性値の設定」
- ◆ 185 ページの「ターゲットパスワードの設定」
- ◆ 186 ページの「ローカル変数の設定」
- ◆ 187 ページの「操作関連付けの設定」
- ◆ 188 ページの「操作クラス名の設定」
- ◆ 189 ページの「操作ターゲット DN」
- ◆ 190 ページの「操作プロパティの設定」
- ◆ 191 ページの「操作ソース DN の設定」
- ◆ 192 ページの「操作テンプレート DN の設定」
- ◆ 193 ページの「ソース属性値の設定」
- ◆ 195 ページの「ソースパスワードの設定」
- ◆ 196 ページの「SSO 資格情報の設定」
- ◆ 197 ページの「SSO パスフレーズの設定」
- ◆ 198 ページの「XML 属性の設定」
- ◆ 199 ページの「ステータス」
- ◆ 200 ページの「ワークフローの開始」
- ◆ 202 ページの「操作属性のストリップ」
- ◆ 203 ページの「XPath のストリップ」
- ◆ 204 ページの「メッセージのトレース」
- ◆ 206 ページの「拒否」
- ◆ 207 ページの「操作属性値がない場合は拒否」
- ◆ 209 ページの「While」
- ◆ 210 ページの「変数の拡張」

関連付けの追加

識別ポートに特定の関連付けを指定した関連付けコマンドを送信します。

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。

DN

ターゲットオブジェクトの DN を指定するか、またはフィールドを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

✓ 実行内容	関連付けの追加	?	✖	📄	✍
モードを選択:	現在の操作に追加する				
オブジェクトを選択:	DN				
DNを入力:*	ソースDN0				
関連付けを入力:*	ソース名0				

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。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

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、フィールドを空白のままにします。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

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。

値タイプ

追加する属性値の構文を選択します。オプションは、文字列、カウンタ、DN、INT、間隔、オクテット、状態、構造、電話番号、または時間です。

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 に付属する事前定義されたルールです。詳細については、[52 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。XML 形式でポリシーを表示するには、[predef_command_create_dept_container1.xml \(../samples/predef_command_create_dept_container1.xml\)](#) および [predef_command_create_dept_container2.xml \(../samples/predef_command_create_dept_container2.xml\)](#) を参照してください。

コマンド変換-部署別コンテナの作成-パート1

条件

if 操作 等しい "add"

アクション

ローカル変数の設定 ("target-container", ターゲット DN (長さ="2"))

ローカル変数の設定 ("does-target-exist", ターゲット属性 [{"objectclass", "クラス名="OrganizationalUnit", "DN(ローカル変数["target-container"])}])

☑ ☑ ☐ コマンド変換-部署別コンテナの作成-パート2

条件

- ☑ ☑ ifローカル変数 'does-target-exist' 使用可能
- AND** ☑ ☑ ifローカル変数 'does-target-exist' 等しい ""

アクション

- ☑ ☑ ターゲットオブジェクトの追加(クラス名="organizationalUnit",直接="true",DN(ローカル変数("target-container")))
- ☑ ☑ ターゲット属性値の追加("ou",直接="true",DN(ローカル変数("target-container")),DNの解析("dest-dn","dot",長さ="1",開始="-1",ローカル変数("target-container")))

☑ ☑ 実行内容 ターゲット属性値の追加

属性名を入力してください:

クラス名を入力してください:

モードを選択:

オブジェクトを選択:

DNを入力:

値タイプを入力:

文字列を入力:

ターゲットオブジェクトの追加

指定したタイプのオブジェクトを、`[DNを入力]` フィールドで指定された名前と場所を使用してターゲットデータストアに作成します。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、この後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。

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。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

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[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 に付属する事前定義されたルールです。[52 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」](#) XML 形式でポリシーを表示するには、[predef_command_create_dept_container1.xml \(../samples/predef_command_create_dept_container1.xml\)](#) および [predef_command_create_dept_container2.xml \(../samples/predef_command_create_dept_container2.xml\)](#) を参照してください。

コマンド変換-部署別コンテナの作成-パート1

条件

- ✓ if 操作 等しい "add"

アクション

- ✓ ローカル変数の設定 ("target-container", ターゲット DN (長さ="2"))
- ✓ ローカル変数の設定 ("does-target-exist", ターゲット属性 ("objectclass", クラス名="OrganizationalUnit", DN (ローカル変数 ("target-container"))))

コマンド変換-部署別コンテナの作成-パート2

条件

- ✓ if ローカル変数 'does-target-exist' 使用可能
- AND ✓ if ローカル変数 'does-target-exist' 等しい ""

アクション

- ✓ ターゲットオブジェクトの追加 (クラス名="organizationalUnit", 直接="true", DN (ローカル変数 ("target-container")))
- ✓ ターゲット属性値の追加 ("ou", 直接="true", DN (ローカル変数 ("target-container")), DNの解析 ("dest-dn", "dot", 長さ="1", 開始="-1", ローカル変数 ("target-container")))

✓	実行内容	ターゲットオブジェクトの追加	?	✖	📄	✍
		クラス名を入力してください。*	Organizational Unit			
		モードを選択:	ターゲットデータストアに直接書き込む			
		DNを入力:*	<input type="checkbox"/> ローカル変数 ("target-container")			

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。

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。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

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、フィールドを空白のままにします。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

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。

値タイプ

追加する属性値の構文を選択します。オプションは、文字列、カウンタ、DN、INT、間隔、オクテット、状態、構造、電話番号、または時間です。

文字列

追加する属性値を指定します。

例

実行内容	ソース属性値の追加
属性名を入力してください*	Member
クラス名を入力してください:	
オブジェクトを選択:	DN
DNを入力:	"Novell#Users#ManagerGroup"
値タイプを入力:	string
文字列を入力:	ターゲットDN0

ソースオブジェクトの追加

指定したタイプのオブジェクトを、[DN] フィールドで指定された名前と場所を使用してソースデータストアに作成します。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 (144 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。変数の拡張がサポートされます。詳細については、(210 ページ) 変数の拡張を参照してください。

DN

追加するオブジェクトの DN を指定します。

例

✓ 実行内容	ソースオブジェクトの追加	?	✖	🔍	✓
	クラス名を入力してください:*	Group			
	DNを入力:*	"Novell#Users"			
✓ 実行内容	ソース属性値の追加	?	✖	🔍	✓
	属性名を入力してください:*	Member			
	クラス名を入力してください:*				
	オブジェクトを選択:	DN			
	DNを入力:*	"Novell#Users#ManagerGroup"			
	値タイプを入力:	string			
	文字列を入力:*	ターゲットDN0			

XML 要素の追加

XPath 式で選択された要素のセットに、*[名前]* フィールドで指定された名前を使用してカスタム要素を追加します。*[XPath 式の前]* が指定されていない場合は、選択した要素の既存の子の後に、新しい要素が追加されます。*[XPath 式の前]* が指定されている場合は、式で選択された各要素と比較して評価され、手前に挿入する子が決めます。*[XPath 式の前]* により空のノードセット、または選択された要素の子を含まないノードセットが評価される場合、新しい要素は既存の子の後に追加されます。それ以外の場合は、新しい要素は、「before」で選択されたノードセット内にあり、選択されたノードの子である各ノードの前に挿入されます。

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。変数の拡張がサポートされます。詳細については、(210 ページ) *変数の拡張* を参照してください。

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。

XPath 式の前

XPath 1.0 式を指定します。この式は、新しい要素が手前に挿入される子ノードを含むノードセットを返す式で選択された各ノードと比較して評価されます。

備考

ポリシーとともに XPath 式を使用する方法の詳細については、『*Understanding Policies for Identity Manager 3.5.1*』の「*XPath 1.0 Expressions*」を参照してください。

例

『*Novell Credential Provisioning Policies for Identity Manager 3.5.1*』を実装している場合は、XPath 式の条件を使用する購読者コマンド変換ポリシーのサンプルが用意されています。サンプルファイルは `SampleSubCommandTransform.xml` という名前です。Identity Manager メディアの `DirXML® Utilities` フォルダに格納されています。詳細については、『*Novell Credential Provisioning Policies for Identity Manager 3.5.1*』の「*Example Credential Provisioning Policies*」を参照してください。XML 形式でポリシーを参照するには、`SampleSubCommandTransform.xml (./samples/SampleSubCommandTransform.xml)` を参照してください。

サンプルファイルはユーザオブジェクトがプロビジョニングされたとき、*[XML エレメントの追加]* アクションを使用して、Novell® SecureLogin または Novell SecretStore® 資格情報をユーザオブジェクトに追加します。

[Add operation-data element to password subscribe operation \(if needed\)](#)

[Add payload data to modify-password subscribe operations](#)

[Add payload data to add subscribe operations](#)

条件

if 操作 等しい "add"

AND if パスワード: 使用可能

アクション

XML要素の追加("sso-sync-data","operation-data")

XML要素の追加("sso-target-user-dn","operation-data/sso-sync-data")

XMLテキストの追加("operation-data/sso-sync-data/sso-target-user-dn",Source Attribute (DirXML-ADContext))

XML要素の追加("sso-app-username","operation-data/sso-sync-data")

XMLテキストの追加("operation-data/sso-sync-data/sso-app-username", Source Attribute ("CN"))

XML要素の追加("password","operation-data/sso-sync-data")

XMLテキストの追加("operation-data/sso-sync-data/password", Password())

XML要素の追加("nsl-set-passphrase-answer","operation-data/sso-sync-data")

XMLテキストの追加("operation-data/sso-sync-data/nsl-set-passphrase-answer",Source Attribute ("workforceID"))

実行内容 XML要素の追加

名前を入力:* operation-data

XPath式を入力:* .

XPATH式の前に入力:

XML テキストの追加

XPath 式で選択された要素のセットに、指定したテキストを追加します。[XPath 式の前] が指定されていない場合は、選択した要素の既存の子の後にテキストが追加されます。[XPath 式の前] が指定されている場合は、式で選択された各要素と比較して評価され、手前に挿入する子が決めます。[XPath 式の前] により、空のノードセットまたは選択した要素の子を含まないノードセットが評価される場合は、テキストは既存の子の後に追加されます。それ以外の場合は、テキストは、「before」で選択されたノードセット内にあり、選択されたノードの子である各ノードの前に挿入されます。

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 の式を指定します。

XPath 式の前

XPath 1.0 式を指定します。この式は、テキストが手前に挿入される子ノードを含むノードセットを返す式で選択された各ノードと比較して評価されます。

文字列

追加するテキストを指定します。

備考

ポリシーとともに XPath 式を使用する方法の詳細については、『[Understanding Policies for Identity Manager 3.5.1](#)』の「[XPath 1.0 Expressions](#)」を参照してください。

例

『[Novell Credential Provisioning Policies for Identity Manager 3.5.1](#)』を実装している場合は、XPath 式の条件を使用する購読者コマンド変換ポリシーのサンプルが用意されています。サンプルファイルは `SampleSubCommandTransform.xml` という名前です。Identity Manager メディアの `DirXML Utilities` フォルダに格納されています。詳細については、『[Novell Credential Provisioning Policies for Identity Manager 3.5.1](#)』の「[Example Credential Provisioning Policies](#)」を参照してください。XML 形式でポリシーを参照するには、`SampleSubCommandTransform.xml (./samples/SampleSubCommandTransform.xml)` を参照してください。

この例では Novell SecureLogin または Novell SecretStore アプリケーションユーザ名を検索するために、[XML テキストの追加] アクションを使用します。アプリケーション名を取得することで、ユーザオブジェクトがプロビジョニングされる時、ユーザオブジェクトに対して資格情報を設定することができます。

[Add operation-data element to password subscribe operation \(if needed\)](#)

[Add payload data to modify-password subscribe operations](#)

[Add payload data to add subscribe operations](#)

条件

if 操作 等しい "add"

AND if パスワード: 使用可能

アクション

XML要素の追加("sso-sync-data","operation-data")

XML要素の追加("sso-target-user-dn","operation-data/sso-sync-data")

XMLテキストの追加("operation-data/sso-sync-data/sso-target-user-dn",Source Attribute (DirXML-ADContext))

XML要素の追加("sso-app-username","operation-data/sso-sync-data")

XMLテキストの追加("operation-data/sso-sync-data/sso-app-username", Source Attribute ("CN"))

XML要素の追加("password","operation-data/sso-sync-data")

XMLテキストの追加("operation-data/sso-sync-data/password", Password())

XML要素の追加("nsl-set-passphrase-answer","operation-data/sso-sync-data")

XMLテキストの追加("operation-data/sso-sync-data/nsl-set-passphrase-answer",Source Attribute ("workforceID"))

実行内容 XMLテキストの追加

XPath式を入力:*

XPATH式の前に入力:

文字列を入力:*

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



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、指定した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。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

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、フィールドを空白のままにします。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

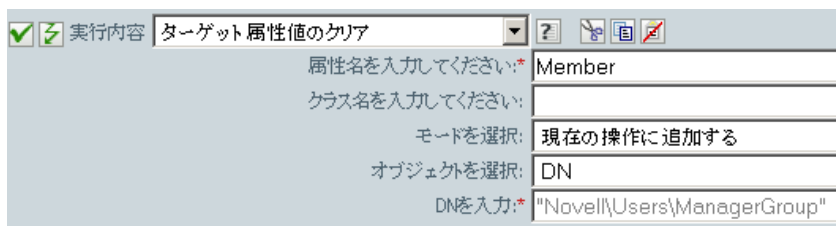
モード

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



The screenshot shows a configuration window for the 'Clear Target Attribute Values' action. The title bar includes a green checkmark icon, a green refresh icon, and the text '実行内容 ターゲット属性値のクリア'. Below the title bar are several input fields:

属性名を入力してください*	Member
クラス名を入力してください:	
モードを選択:	現在の操作に追加する
オブジェクトを選択:	DN
DNを入力:*	"Novell\Users\ManagerGroup"

操作プロパティのクリア

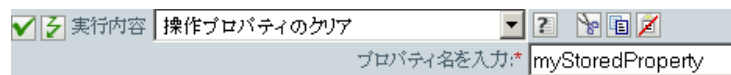
現在の操作から、指定された名前の操作プロパティをクリアします。操作プロパティは、ポリシーによって <operation-data> 要素に添付されている XML 属性です。XML 属性は、XDS ドキュメント内の要素と関連付けられている名前の値のペアです。

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。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

例



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。クリアする資格情報に関する追加の情報は、[ログインパラメータの文字列を入力] フィールドに入力できます。使用される文字列と名前の数は、資格情報リポジトリと資格情報のターゲットアプリケーションによって異なります。詳細については、『[Novell Credential Provisioning Policies for Identity Manager 3.5.1](#)』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。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

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。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

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

実行内容	SSO資格情報のクリア
資格情報リポジトリオブジェクトDNを入力:	Novell\Driver Set\GroupWise\GroupWise_Repository
	<input checked="" type="checkbox"/> ポリシーに関連する参照したDNのレンダリング
ターゲットユーザのDNを入力:	ターゲット属性("DirXML-ADContext",クラス名="User")
	アプリケーションオブジェクトから次の内容を使用
アプリケーションのアクティベーションキーIDを入力:	GroupWise_Credential
ログインパラメータの文字列を入力:	Username,Password

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。変数の拡張がサポートされます。変数の拡張がサポートされません。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

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、フィールドを空白のままにします。この値は、オブジェクトが現在のオブジェクトでない場合に、スキーママッピングを行うのに必要です。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

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

実行内容	ソース属性値のクリア	?	🔍	🏠	✖
属性名を入力してください*	Member				
クラス名を入力してください:					
オブジェクトを選択:	DN				
DNを入力*	"Novell\Users\ManagerGroup"				

XPath 式によるクローン

ソースフィールドで指定されたノードのディープコピーを、ターゲットフィールドで指定された要素のセットに追加します。[XPath 式の前] が指定されていない場合は、選択した要素の既存の子の後に非属性の複製ノードが追加されます。[XPath 式の前] が指定されている場合は、式で選択された各要素と比較して評価され、手前に挿入する子が決められます。[XPath 式の前] により、空のノードセットまたは選択した要素の子を含まないノードセットが評価される場合は、非属性の複製ノードは既存の子の後に追加されます。それ以外の場合は、非属性の複製ノードは、以前選択されたノードセット内にあり、選択されたノードの子である各ノードの前に挿入されます。

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。

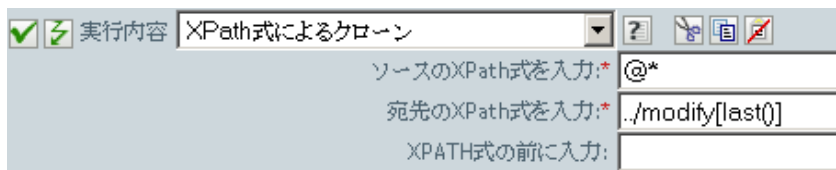
XPath 式の前

XPath 1.0 式を指定します。この式は、非属性の複製ノードを手前に挿入する子ノードを含むノードセットを返すターゲット XPath 式で選択された各ノードと比較して評価されます。

備考

ポリシーとともに XPath 式を使用する方法の詳細については、『[Understanding Policies for Identity Manager 3.5.1](#)』の「[XPath 1.0 Expressions](#)」を参照してください。

例



The screenshot shows a configuration window titled "XPath式によるクローン". It has a toolbar with icons for help, copy, paste, and delete. Below the toolbar are three input fields:

- ソースのXPath式を入力:* @*
- 宛先のXPath式を入力:* ../modify[last()]
- XPATH式の前に入力:

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。変数の拡張がサポートされます。詳細については、(210 ページ) [変数の拡張](#)を参照してください。

ターゲット名

コピー先の属性の名前を指定します。変数の拡張がサポートされます。詳細については、(210 ページ) [変数の拡張](#)を参照してください。

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、『[Understanding Policies for Identity Manager 3.5.1](#)』の「[Downloading Identity Manager Policies](#)」を参照してください。XML 形式でポリシーを表示するには、[003-Command-AddCreateGroups.xml](#) (`./samples/003-Command-AddCreateGroups.xml`) を参照してください。

<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Set local variables to test existence of groups and for placement
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Create ManagersGroup, if needed
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Create EmployeesGroup, if needed
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	If Title indicates Manager, add to ManagerGroup and set rights
条件	
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	if クラス名 等しい "User"
AND <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	if 操作属性 'Title' 等しい ".*manager.*"
アクション	
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	ターゲット属性値の設定("Group Membership",ローカル変数("manager-group-dn"))
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	操作属性のクローン("Group Membership","Security Equals")
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	If Title does not indicate Manager, add to EmployeeGroup and set rights

<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	実行内容	操作属性のクローン	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		ソース名を入力:	*	Group Membership		
		ターゲット名を入力:		Security Equals		

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。

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

クラス名

(オプション) ターゲットデータストアで削除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。

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

例

実行内容	ターゲットオブジェクトの削除	?			
クラス名を入力してください:	User				
モードを選択:	現在の操作に追加する				
オブジェクトを選択:	DN				
DNを入力:*	"novell\users\jdoe"				

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

クラス名

(オプション) ソースデータストアで削除するオブジェクトのクラス名を指定します。

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タイプを選択します。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。

DN

ターゲットオブジェクトとしての DN、関連付け、現在のオブジェクトを選択します。

例

実行内容	ソースオブジェクトの削除
クラス名を入力してください:	User
オブジェクトを選択:	DN
DNを入力:*	"novell\users\jdoe"

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ため、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ドである場合は、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ため、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 つだけが返され、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 `�` に設定されます。

例

この例では、属性 CN と L を持つユーザオブジェクトを照合します。ルールが検索する場所は Users 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 に付属する事前定義されたルールです。詳細については、[65 ページのセクション 6.13 「一致 - 属性値別」](#) を参照してください。XML 形式でポリシーを表示するには、[predef_match_by_attribute.xml \(../samples/predef_match_by_attribute.xml\)](#) を参照してください。

一致-属性値別

条件

if クラス名 等しい "User"

アクション

一致オブジェクトの検索(DN("[検索を開始するベースDNを入力してください]"),一致("[Enter name of attribute to match on"])

<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> 実行内容	一致オブジェクトの検索	<input type="button" value="?"/> <input type="button" value="X"/> <input type="button" value="Home"/> <input type="button" value="Refresh"/>
スコープを選択:	サブツリー	
DNを入力:	"Novell"	
一致属性を入力:	CN,L	

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。

一致する属性:	
Edit ▼	Append New Matching Attribute Remove...
<input type="checkbox"/> 名前:*	CN <input type="button" value="🔍"/> 現在のオブジェクトからの値
<input type="checkbox"/> 名前:*	L <input type="button" value="🔍"/> 現在のオブジェクトからの値

左側のフィールドには照合する属性が格納されます。右側のフィールドでは、照合する現在のオブジェクトの値を使用するか、別の値を使用して指定できます。[その他の値]を選択すると、複数の値タイプを指定できます。

- ◆ カウンタ
- ◆ dn
- ◆ int
- ◆ 間隔
- ◆ octet (オクテット)
- ◆ 都道府県
- ◆ 文字列
- ◆ 構造
- ◆ teleNumber
- ◆ タイム

毎

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。

ノードセットの現在のノードがエンタイトルメント要素である場合、アクションは**エンタイトルメントの実装**アクションにも格納されているようにマークされます。現在のノードがクエリから返されるクエリ要素の場合、そのトークンはクエリ結果の次のバッチを自動的に取得し、処理するために使用されます。

例

実行内容	毎	?	🗑️	📄	✖️
ノードセットを入力:*	追加されたエンタイトルメント("Group")				
アクションを入力:*	do-add-dest-attr-value				

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。

実行内容	ターゲット属性値の追加	?	🗑️	📄	✖️
属性名を入力してください:*	Member				
クラス名を入力してください:	Group				
モードを選択:	現在の操作に追加する				
オブジェクトを選択:	DN				
DNを入力:*	ローカル変数("current-node")				
値タイプを入力:	string				
文字列を入力:*	ターゲットDN()				

イベントの生成

ユーザ定義イベントを Novell Audit または Sentinel に送信します。

フィールド

ID

イベントの ID です。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。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

レベル

イベントのレベル。

レベル	説明
log-emergency	Metadirectory 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	Metadirectory 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	Metadirectory エンジンまたはドライバによって処理できるエラーを説明するイベント。
log-warning	問題を表さ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	いずれかの重要度を持つポジティブイベントです。
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です。現在定義済みの target-type の値を示します。 <ul style="list-style-type: none">◆ 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。

タグ	説明
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。
値	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せん。
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。
データ	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 または Sentinel のイベント構造には、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

例には名字属性の最初の 1 文字を基にユーザオブジェクトの配置ポリシーを実装する 4 つのルールが含まれます。これはメッセージのトレースとカスタム Novell Audit または Sentinel 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 または Sentinel 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、『“[Understanding Policies for Identity Manager 3.5.1](#)”』の「[Downloading Identity Manager Policies](#)」を参照してください。XML 形式でポリシーを参照するには、[001-Placement-BySurname.xml](#) ([../samples/001-Placement-BySurname.xml](#)) を参照してください。

The screenshot shows a configuration window with several policy rules and an action configuration dialog.

Policy Rules:

- Setup Local Variables**
- Surname A-I: place in Users1**
 - 条件**
 - if クラス名 等しい "User"
 - AND** if 操作属性 'Surname' 等しい "[a-i].*"
 - アクション**
 - 操作ターゲットDNの設定(DN("Training\Users\Active\Users1"+"\"+操作属性("CN")))
 - メッセージのトレース(color="yellow",ローカル変数("LVUsers1"))
 - イベントの生成(id="1000",text1=ローカル変数("LVUsers1"))
- Surname J-R:place in Users2**
- Surname S-Z: place in Users3**

Action Configuration Dialog:

- 実行内容:** イベントの生成
- IDを入力してください:*** 1000
- レベルを選択:** 情報
- 文字列を入力:** text1

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。

文字列

名前:* text1

文字列の値:* ローカル変数("LVUsers1")

イベントの生成により、ID 1000 のイベントが作成され、LVUser1 のローカル変数により生成されたテキストが表示されます。ローカル変数 LVUser1 はユーザ: 操作属性 "cn" + "+"Training\Users\Active\Users1"+" container" に追加された文字列です。このイベントは、Training\Users\Active\Users1 コンテナに追加されたユーザ :jsmith を読み込みます。

If

条件に従って一連のアクションを実行します。

フィールド

条件

目的の条件を指定します。

If アクション

条件が **True** の場合の目的のアクションを指定します。

Else アクション

(オプション) 条件が **False** の場合の目的のアクションを指定します。

例

操作の追加または変更中に、役職の属性がマネージャに等しい場合は、そのユーザオブジェクトが **ManagerGroup** グループに追加されます。役職がマネージャに等しくない場合は、そのユーザオブジェクトは **UsersGroup** グループに追加されます。XML 形式でポリシーを参照するには、[if.xml \(../samples/if.xml\)](#) を参照してください。

条件

✓ if 操作 等しい "add"
OR ✓ if 操作 等しい "modify"

Actions

✓ if(conditions(and(if operation attribute "Title" equal "manager")),actions(set destination attribute value("Group Membership",class name="User","Novel\Users\ManagerGroup")),actions(set default attribute value("Group Membership","Novel\Users\UsersGroup")))

実行内容 if

条件を入力:* and(if-op-attr)
ifアクションを入力:* do-set-default-attr-value
elseアクションを入力: do-set-dest-attr-value

if アクションを作成するときは、条件とアクションを 1 つずつ追加する必要があります。この例には 2 つの異なるアクションが含まれます。「ユーザオブジェクトの役職がマネージャである場合」が条件です。

if 操作属性

名前を入力:* Title
演算子を選択:* 等しい
モードを比較: 大文字と小文字の区別なし
値: manager

アクションはユーザオブジェクトを **ManagerGroup** グループに追加することです。

実行内容	デフォルト 属性値の設定	?	🔍	📄	✎
属性名を入力してください*	Group Membership				
ライトバック:	False				
引数値を入力:*	"Novell\Users\ManagerGroup"				

役職がマネージャに等しくない場合は、そのユーザオブジェクトは UsersGroup グループに追加されます。

実行内容	デフォルト 属性値の設定	?	🔍	📄	✎
属性名を入力してください*	Group Membership				
ライトバック:	False				
引数値を入力:*	"Novell\Users\UsersGroup"				

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できるようにします。

フィールド

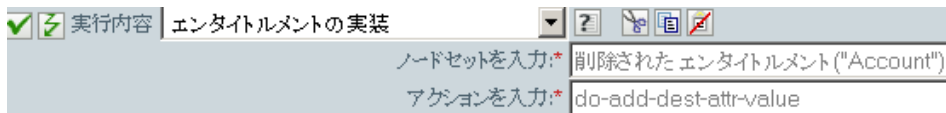
ノードセット

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。

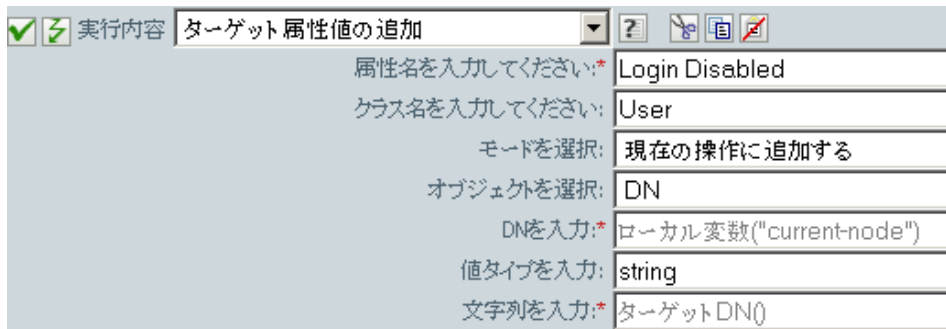
アクション

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。



ターゲットオブジェクトの移動

オブジェクトをターゲットデータストアに移動します。

フィールド

クラス名

(オプション) ターゲットデータストアに移動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。

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。

移動先

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。

DN または関連付け

コンテナの DN または関連付けが使用されているかどうかを指定します。

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、説明属性がそ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、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、『[Understanding Policies for Identity Manager 3.5.1](#)』の「[Downloading Identity Manager Policies](#)」を参照してください。XML 形式でポリシーを表示するには、[005-Command-DisableMoveOnTermination \(../samples/005-Command-DisableMoveOnTermination.xml\)](#) を参照してください。

On Termination , disable user and move to Disabled container

条件

✓ ↻ if 操作 等しい "modify"
AND ✓ ↻ if クラス名 等しい "User"
AND ✓ ↻ if 操作属性 'Description' 等しい "^terminated.*"

アクション

✓ ↻ ターゲット属性値の設定("Login Disabled","True")
✓ ↻ ターゲットオブジェクトの移動(when="after",DN("Users\Disabled"))

実行内容		ターゲットオブジェクトの移動
クラス名を入力してください:		
モードを選択:	現在の操作に追加する	
移動するオブジェクトを選択:	現在のオブジェクト:	
移動するコンテナを選択:	DN	
DNを入力:*	"Users\Disabled"	

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。

ソースオブジェクトの移動

オブジェクトをソースデータストアに移動します。

フィールド

クラス名

(オプション) ソースデータストアに移動するオブジェクトのクラス名を指定します。

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。このオブジェクトには現在のオブジェクトを選択することも、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

実行内容	ソースオブジェクトの移動	?	🗑️	📄	✍️
クラス名を入力してください:	User				
移動するオブジェクトを選択:	DN				
DNを入力:*	"Users\Active\jdoe"				
移動するコンテナを選択:	DN				
DNを入力:*	"Users\Inactive"				

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。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

値タイプ

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-xxxx から nnn-xxx-xxxx に変更します。このルールは、Identity Manager に付属する事前定義されたルールです。詳細については、[61 ページの「入出力変換 - 電話番号の形式を \(nnn\) nnn-xxxx から nnn-xxx-xxxx に変更」](#)を参照してください。XML 形式でポリシーを参照するには、[prefdef_transformation_reformat_telephone1.xml \(./samples/prefdef_transformation_reformat_telephone1.xml\)](#) を参照してください。

☑ ☑ ☐ ☐ 入出力変換 - 電話番号の形式を(nnn) nnn-xxxxからnnn-xxx-xxxxに変更

条件
☑ ☑ この条件はTrueと評価されます。

アクション
☑ ☑ 操作属性の再フォーマット("phone",最初を置換("^\\(\\d\\d\\d\\d\\)\\s*(\\d\\d\\d\\d)-(\\d\\d\\d\\d\\d\\d)\$", "\$1-\$2-\$3",ローカル変数("current-value"))

☑ ☑ 実行内容 操作属性の再フォーマット

名前を入力:* phone

値タイプを入力: string

文字列を入力:* 最初を置換("^\\(\\d\\d\\d\\d\\)\\s*(\\d\\d\\d\\d)-(\\d\\d\\d\\d\\d\\d)\$", "\$1-\$2-\$3")

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。

☑ ☑ 最初を置換("^\\(\\d\\d\\d\\d\\)\\s*(\\d\\d\\d\\d)-(\\d\\d\\d\\d\\d\\d)\$", "\$1-\$2-\$3")

☑ ☑ ローカル変数("current-value")

関連付けの削除

関連付けを削除するコマンドを識別ボールドに送信します。

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。イベントを変換します。このルールは、Identity Manager に付属する事前定義されたルールです。詳細については、53 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。XML 形式でポリシーを参照するには、[predef_command_delete_to_disable.xml \(../samples/predef_command_delete_to_disable.xml\)](#) を参照してください。

The screenshot shows a configuration window for a rule titled "コマンド変換-無効にする発行者の削除". It is divided into three sections: "条件" (Conditions), "アクション" (Actions), and "実行内容" (Execution Content).

- 条件 (Conditions):**
 - if 操作 等しい "delete"
 - OR if クラス名 等しい "User"
- アクション (Actions):**
 - ターゲット属性値の設定 ("Login Disabled", "true")
 - 関連付けを削除 (関連付け (関連付け ()))
- 実行内容 (Execution Content):**
 - 実行内容: 関連付けを削除
 - モードを選択: 現在の操作に追加する
 - 関連付けを入力*: 関連付け()

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」の属性が 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。

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。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

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、フィールドを空白のままにします。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

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。

値タイプ

新しい属性値の構文を指定します。

文字列

新しい属性の値を指定します。

例

実行内容	ターゲット属性値の削除
属性名を入力してください*	Member
クラス名を入力してください:	
モードを選択:	現在の操作に追加する
オブジェクトを選択:	DN
DNを入力:	"Novell\Users\ManagerGroup"
値タイプを入力:	string
文字列を入力:	ターゲットDN()

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。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

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、フィールドを空白のままにします。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

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。

値タイプ

削除する属性値の構文を選択します。

文字列

削除する属性値を指定します。

例

ラベル	値
属性名を入力してください*	Member
クラス名を入力してください:	
オブジェクトを選択:	DN
DNを入力*	"Novell\Users\ManagerGroup"
値タイプを入力:	string
文字列を入力*	ソースDN()

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

クラス名

(オプション) ターゲットデータストアで名前を変更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。

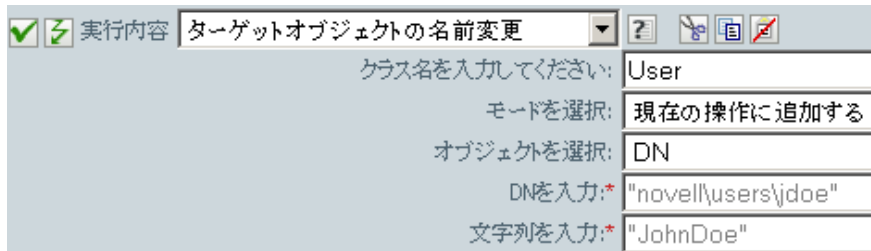
オブジェクト

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



The screenshot shows a configuration window for the action 'ターゲットオブジェクトの名前変更' (Rename Target Object). The window has a title bar with a green checkmark icon and a '実行内容' (Execution Content) label. Below the title bar, there are several input fields and dropdown menus:

クラス名を入力してください:	User
モードを選択:	現在の操作に追加する
オブジェクトを選択:	DN
DNを入力:*	"novell\users\jdoe"
文字列を入力:*	"JohnDoe"

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

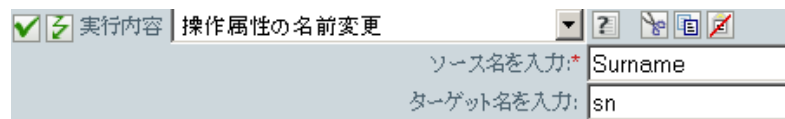
ソース名

変更前の属性名を指定します。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

ターゲット名

新しい属性名を指定します。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

例



実行内容 操作属性の名前変更	
ソース名を入力:	Surname
ターゲット名を入力:	sn

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

クラス名

(オプション) ソースデータストアのオブジェクトのクラス名を指定します。

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。このオブジェクトには現在のオブジェクトを選択することも、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

実行内容	ソースオブジェクトの名前変更
クラス名を入力してください:	User
オブジェクトを選択:	DN
DNを入力:*	"novell\users\jdoe"
文字列を入力:*	"JohnDoe"

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。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

サーバ

SMTP サーバ名を指定します。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

メッセージタイプ

電子メールメッセージのタイプを選択します。

[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 : SMTP サーバのアカウントパスワードをドライバオブジェクトに名前付きパスワードとして格納できます。これにより、パスワードを暗号化できます。そうしない場合は、入力したパスワードはクリアテキストで格納されます。名前付きパスワードの詳細については、『*Novell Identity Manager 3.5.1 Administration Guide*』の「**Using Named Passwords**」を参照してください。

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。受信者をコンマで区切ったリストを含めることができます。
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。受信者をコンマで区切ったリストを含めることができます。
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。受信者をコンマで区切ったリストを含めることができます。
from	電子メールの発信アドレスとして使用す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す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。
メッセージ	電子メールメッセージの内容を指定します。

文字列名	説明
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。
custom-smtp-header	電子メールメッセージに追加するカスタム SMTP ヘッダを指定します。

例

実行内容 電子メールの送信

IDを入力してください: ssmith

サーバを入力:* smtp.digitalairlines.com

メッセージタイプを選択: Text

パスワードの入力: 名前付きパスワード("smtp-admin")

文字列を入力: to,subject,message

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。

文字列			
<input type="checkbox"/> 名前:*	to	文字列の値:*	"ManagerGroup@digitalairlines.com"
<input type="checkbox"/> 名前:*	subject	文字列の値:*	"This is the e-mail subject"
<input type="checkbox"/> 名前:*	message	文字列の値:*	"This is the e-mail message"

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。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

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。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

[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 : SMTP サーバのアカウントパスワードをドライバオブジェクトに名前付きパスワードとして格納できます。これにより、パスワードを暗号化できます。そうしない場合は、入力したパスワードはクリアテキストで格納されます。名前付きパスワードの詳細については、『*Novell Identity Manager 3.5.1 Administration Guide*』の「[Using Named Passwords](#)」を参照してください。

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。受信者をコンマで区切ったリストを含めることができます。
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。受信者をコンマで区切ったリストを含めることができます。
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。受信者をコンマで区切ったリストを含めることができます。
reply-to	電子メールメッセージの返信アドレスとして使用す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。
custom-smtp-header	電子メールメッセージに追加するカスタム SMTP ヘッダを指定します。

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る場合があります。

例

実行内容	テンプレートから電子メールを送信
通知DNを入力:*	Security\Default Notification Collection
テンプレートDNを入力:*	Security\Default Notification Collection\Forgot Password
パスワードの入力	名前付きパスワード("smtp-admin")
文字列を入力:	to,cc

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。

文字列			
<input type="checkbox"/> 名前:*	to	文字列の値:*	"to_user@comany.com"
<input type="checkbox"/> 名前:*	cc	文字列の値:*	"cc_user@comany.com"

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。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

ライトバック

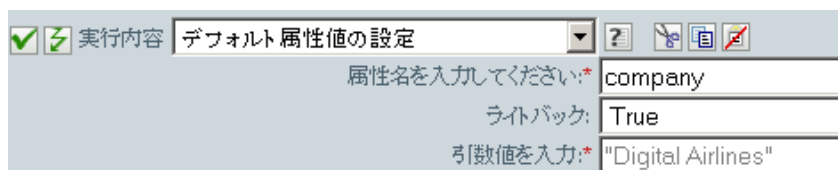
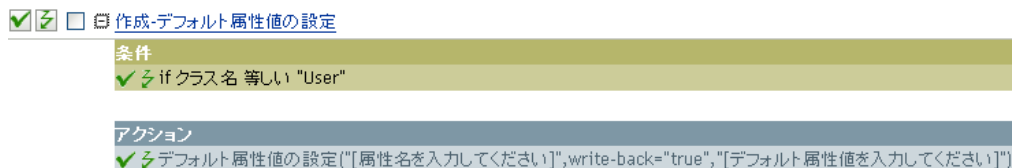
デフォルト値をソースデータストアにもライトバックするかどうかを選択します。

値

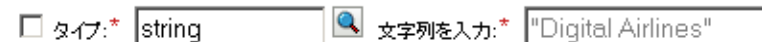
属性のデフォルト値を指定します。

例

この例では、`company` 属性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せん。このルールは、Identity Manager に付属する事前定義されたルールです。詳細については、[57 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。XML 形式でポリシーを表示するには、[predef_creation_set_default_attribute_value.xml \(../samples/predef_creation_set_default_attribute_value.xml\)](#)を参照してください。



引数値



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、[35 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。

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。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

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、フィールドを空白のままにします。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

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。

値タイプ

設定する属性値の構文を選択します。

文字列

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 に付属する事前定義されたルールです。詳細については、[53 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。XML 形式でポリシーを参照するには、[predef_command_delete_to_disable.xml \(../samples/predef_command_delete_to_disable.xml\)](#)を参照してください。

[コマンド変換-無効にする発行者の削除](#)

条件

✓ ↻ if 操作 等しい "delete"

OR ✓ ↻ if クラス名 等しい "User"

アクション

✓ ↻ ターゲット属性値の設定("Login Disabled","true")

✓ ↻ 関連付けを削除(関連付け(関連付け {}))

実行内容		ターゲット属性値の設定
属性名を入力してください:*		Login Disabled
クラス名を入力してください:		
モードを選択:		現在の操作に追加する
オブジェクトを選択:		現在のオブジェクト:
値タイプを入力:		string
文字列を入力:*		"true"

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、[30 ページの「引数ビルダ」](#)を参照してください。

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

クラス名

(オプション) ターゲットデータストアでパスワードを設定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。

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 に付属する事前定義されたルールです。詳細については、58 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。

作成-デフォルトパスワードの設定	
条件	ifクラス名 等しい "User"
アクション	ターゲットパスワードの設定(属性("Given Name")+属性("Surname"))

実行内容	ターゲットパスワードの設定
クラス名を入力してください:	User
モードを選択:	現在の操作に追加する
オブジェクトを選択:	現在のオブジェクト:
文字列を入力:*	属性("Given Name")+属性("Surname")

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。

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。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

変数のスコープ

ローカル変数のスコープを選択します。これはドライバまたはポリシーに設定できません。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

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の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、『[Understanding Policies for Identity Manager 3.5.1](#)』の「[Downloading Identity Manager Policies](#)」を参照してください。XML 形式でポリシーを参照するには、[003-AddCreateGroups.xml](#) ([../samples/003-Command-AddCreateGroups.xml](#)) を参照してください。

The screenshot shows a configuration window for a policy. At the top, there is a checkbox labeled "Set local variables to test existence of groups and for placement" which is checked. Below this, there are two sections: "条件" (Conditions) and "アクション" (Actions).

条件 (Conditions):

- if クラス名 等しい "User"
- if 操作 等しい "add"
- if 操作 等しい "modify"

The conditions are connected by an "AND" operator.

アクション (Actions):

- ローカル変数の設定 ("manager-group-dn", "Users\ManagersGroup")
- ローカル変数の設定 ("manager-group-info", ターゲット属性 ("Object Class", DN(ローカル変数 ("manager-group-dn"))))
- ローカル変数の設定 ("employee-group-dn", "Users\EmployeesGroup")
- ローカル変数の設定 ("employee-group-info", ターゲット属性 ("Object Class", DN(ローカル変数 ("employee-group-dn"))))

Below the actions, there is a summary section titled "実行内容" (Execution Content) for the policy "ローカル変数の設定" (Local Variable Setting). It shows the following configuration:

- 変数名を入力してください:
- ローカル変数のスコープを選択:
- 変数タイプの選択:
- 文字列を入力:

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、[30 ページの「引数ビルダ」](#)を参照してください。

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



実行内容 操作関連付けの設定

関連付けを入力: * ソース名()

操作クラス名の設定

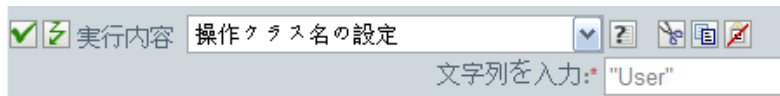
現在の操作のオブジェクトクラス名を設定します。

フィールド

文字列

新しいクラス名を指定します。

例



操作ターゲット DN

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、識別ボールド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 に付属する事前定義されたルールです。詳細については、67 ページの「[配置 - 発行者 \(ミラーリング\)](#)」を参照してください。XML 形式でポリシーを参照するには、[predef_place_pub_mirrored.xml](#) (`./samples/predef_place_pub_mirrored.xml`) を参照してください。

[配置-発行者\(ミラーリング\)](#)

条件

if ソースDN サブツリー内にあり "[ソース階層のベースを入力してください]"

アクション

ローカル変数の設定 ("dest-base", "[宛先階層のベースを入力してください])

操作ターゲットDNの設定 (DN(ローカル変数("dest-base")+\"\\\"+一致しないソースDN(変換=true)))

実行内容 **操作ターゲットDNの設定**

DNを入力:*

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。

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

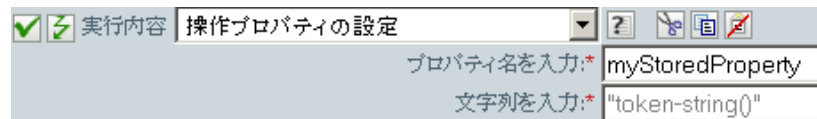
プロパティ名

操作プロパティの名前を指定します。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

文字列

文字列の名前を指定します。

例



実行内容	操作プロパティの設定	?	🔍	📄	✖
プロパティ名を入力: *	myStoredProperty				
文字列を入力: *	"token-string()				

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



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、『[Understanding Policies for Identity Manager 3.5.1](#)』の「[Downloading Identity Manager Policies](#)」を参照してください。XML 形式でポリシーを参照するには、[003-Create-AssignTemplateByTitle.xml](#) ([../samples/003-Create-AssignTemplateByTitle.xml](#)) を参照してください。

The screenshot shows a configuration window for a policy. At the top, the policy name is "Assign Manager template if Title contains 'Manager'". Below this, there are two sections: "条件" (Conditions) and "アクション" (Actions). The "条件" section lists three conditions: "if クラス名 等しい 'User'", "AND if 操作属性 'Title' 使用可能", and "AND if 操作属性 'Title' 等しい '*manager*'". The "アクション" section lists one action: "操作テンプレートDNの設定 (DN('Users\ManagerTemplate'))". Below the policy name, there is a section for "Assign Employee template if Title does not contain 'Manager'". At the bottom, there is a toolbar with icons for "実行内容" (Execution Content), "操作テンプレートDNの設定" (Assign Template DN), and a text input field for "DNを入力:" (Enter DN) with the value "Users\ManagerTemplate".

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。

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。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

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、フィールドを空白のままにします。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

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、『[Understanding Policies for Identity Manager 3.5.1](#)』の「[Downloading Identity Manager Policies](#)」を参照してください。XML 形式でポリシーを参照するには、[001-Input_PushBackOnEmail \(../samples/001-Input-PushBackOnEmail.xml\)](#) を参照してください。

[Push back on email changing](#)

条件

- if クラス名 等しい "User"
- AND if 操作属性 'Email' 変更あり

アクション

- ソース属性値の設定("Email", ターゲット属性("Internet EMail Address"))
- 操作属性のストリップ("Email")

実行内容 ソース属性値の設定	
属性名を入力してください*	Email
クラス名を入力してください:	
オブジェクトを選択:	現在のオブジェクト:
値タイプを入力:	string
文字列を入力*	ターゲット属性("Internet Email Address")

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。

ソースパスワードの設定

ソースデータストアにあるオブジェクトのパスワードを設定します。

フィールド

クラス名

(オプション) ソースデータストアでパスワードを設定するオブジェクトのクラス名を指定します。

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

✓ 実行内容	ソースパスワードの設定	?	✖	🏠	✍
クラス名を入力してください:	User				
オブジェクトを選択:	現在のオブジェクト:				
文字列を入力:*	属性("Given Name")+属性("Surname")				

SSO 資格情報の設定

ユーザオブジェクトが作成またはパスワードが変更されるときの、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、『*Novell Credential Provisioning Policies for Identity Manager 3.5.1*』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。変数の拡張がサポートされます。詳細については、(210 ページ) [変数の拡張](#) を参照してください。

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。変数の拡張がサポートされます。詳細については、(210 ページ) [変数の拡張](#) を参照してください。

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

実行内容	SSO資格情報の設定
資格情報リポジトリオブジェクトDNを入力:	Novell\Driver Set\GroupWise\GroupWise_Repository
	<input checked="" type="checkbox"/> ポリシーに関連する参照したDNのレンダリング
ターゲットユーザのDNを入力:	ターゲット属性("DirXML-ADContext"クラス名="User")
	アプリケーションオブジェクトから次の内容を使用
アプリケーションのアクティベーションキーIDを入力:	GroupWise_Credential
ログインパラメータの文字列を入力:	Username,Password

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時の 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、『[Novell Credential Provisioning Policies for Identity Manager 3.5.1](#)』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。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

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

資格情報リポジトリオブジェクトDNを入力:	Novell\Driver Set\GroupWise\GroupWise_Repository
ターゲットユーザのDNを入力:	ターゲット 属性("DirXML-ADContext",クラス名="User")
質問と回答の文字列を入力:	"Employee Code?",属性("workforceID")

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、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。

文字列

質問:	"Employee Code?"
応答:	属性("workforceID")

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。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

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

The screenshot displays two instances of the 'XML属性の設定' (XML Attribute Configuration) dialog box. Each instance has a title bar with a green checkmark, a dropdown menu set to '実行内容 XML属性の設定', and icons for help, copy, paste, and delete. The first instance shows the following fields: '名前を入力:*' (Name) with 'cert-id', 'XPath式を入力:*' (XPath expression) with '.', and '文字列を入力:*' (String value) with '"c:\lotus\domino\data\eng.id"'. The second instance shows: '名前を入力:*' with 'cert=pwd', 'XPath式を入力:*' with '.', and '文字列を入力:*' with '""certify2eng""'.

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。レベルはエラー、致命的、再試行、成功、および警告です。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

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

実行内容	ステータス	?			
レベルを入力*	warning				
メッセージ*	ソースDN{+}: operation vetoed on out-of-scope object"				

ワークフローの開始

URL および ID とパスワードで指定された資格情報を使用して指定された、ユーザアプリケーションサーバの受信者 DN のワークフロー ID で指定されたワークフローを開始します。受信者は、ユーザアプリケーションサーバによって提供されたディレクトリ内のオブジェクトの LDAP フォーマット DN である必要があります。ワークフローの追加の引数は、名前付き文字列で指定できます。使用される文字列と名前の数は、資格情報ストアと資格情報のターゲットとなるアプリケーションによって異なります。

リマーク

特別な意味を持ち、ワークフローが開始しているかどうかに関わらず、使用可能な名前があります。

- ◆ **:InitiatorOverrideDN:** 認証にユーザ以外を使用する場合の、ワークフローのイニシエータの LDAP 形式 DN。
- ◆ **:CorrelationID:** 関連するワークフローを関連させるために使用される ID。

ワークフローの開始中に何らかのエラーが発生した場合、ローカル変数名 `error.do-start-workflow` に格納されているポリシーに対して使用できるエラー文字列。それ以外の場合、ローカル変数は使用できません。

フィールド

プロビジョニング要求 DN

開始するワークフローの DN を LDAP 形式で指定します。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

ユーザアプリケーション URL

ワークフローが実行されるユーザアプリケーションサーバ URL を指定します。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

認定ユーザ DN

ユーザアプリケーションサーバで、ワークフローの開始を承認されたユーザの DN を LDAP 形式で指定します。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

認定ユーザパスワード

ユーザアプリケーションサーバで、ワークフローの開始を承認されたユーザのパスワードを指定します。ドライバオブジェクトに名前付きパスワードとしてパスワードを格納します。これにより、パスワードの保存時、パスワードが暗号化されます。

受信者 DN

ワークフローの受信者の DN を LDAP 形式で指定します。

追加の引数

ワークフローの引数を指定します。引数は各ワークフローで異なります。

例

次の例では、追加操作があるたびに、ワークフロープロセスを開始します。ワークフローは携帯電話の要求です。XML形式でポリシーを参照するには、[start_workflow.xml](#) ([../samples/start_workflow.xml](#)) を参照してください。

The screenshot displays a configuration window for a workflow named "Start WorkFlow". It is divided into two main sections: "条件" (Conditions) and "アクション" (Actions).

条件 (Conditions):

- if 操作 等しい "add"

アクション (Actions):

- ワークフローの開始 (id="CN=ApproveCellPhone,CN=RequestDefs,CN=AppConfig,CN=UserApplication,CN=Driverset,O=novell", url="http://localhost:8080/IDMProv", workflow-id="cn=WorkflowAdmin,o=People", arg-password{名前付きパスワード("workflow-admin")}, DN(DNの解析("qualified-slash","ldap",XPath("@qualified-src-dn"))), provider="ACMEWireless", reason="new hire")

Below the action list, there is a detailed view of the "ワークフローの開始" (Start Workflow) action. It includes a table of input parameters:

実行内容	ワークフローの開始
プロビジョニングリクエストDNを入力:	CN=ApproveCellPhone,CN=RequestDefs,CN=AppConfig,CN=UserApplication
ユーザアプリケーションのURLを入力:	http://localhost:8080/IDMProv
認可ユーザのDNを入力:	cn=WorkflowAdmin,o=People
認定ユーザパスワードを入力:	Named Password("workflow-admin")
受信者のDNを入力:	Parse DN("qualified-slash","ldap",XPath("@qualified-src-dn"))
追加の引数を入力:	provider,reason

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。変数の拡張がサポートされます。詳細については、[\(210 ページ\) 変数の拡張](#)を参照してください。

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、『[Understanding Policies for Identity Manager 3.5.1](#)』の「[Downloading Identity Manager Policies](#)」を参照してください。XML 形式でポリシーを参照するには、[001-Input-PushBackOnEmail.xml \(./samples/001-Input-PushBackOnEmail.xml\)](#) を参照してください。

The screenshot shows the configuration for a policy named "Push back on email changing". It includes a "条件" (Conditions) section with two rules: "if クラス名 等しい 'User'" and "AND if 操作属性 'Email' 変更あり". The "アクション" (Actions) section contains two actions: "ソース属性値の設定 ('Email', ターゲット属性 ('Internet EMail Address'))" and "操作属性のストリップ ('Email')". At the bottom, the "実行内容" (Execution Content) is set to "操作属性のストリップ" with the field name "Email" entered in the "名前を入力*" (Enter name*) field.

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。

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

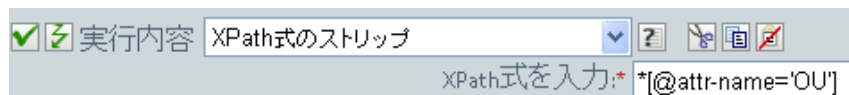
XPath 式

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。

備考

ポリシーとともに XPath 式を使用する方法の詳細については、『[Understanding Policies for Identity Manager 3.5.1](#)』の「[XPath 1.0 Expressions](#)」を参照してください。

例



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 3.5.1 Administration Guide](#)』の「[Viewing Version Information](#)」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

例には名字属性の最初の1文字を基にユーザオブジェクトの配置ポリシーを実装する4つのルールが含まれます。これはメッセージのトレースとカスタム Novell Audit または Sentinel 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、『[Understanding Policies for Identity Manager 3.5.1](#)』の「[Downloading Identity Manager Policies](#)」を参照してください。XML 形式でポリシーを参照するには、[001-Placement-BySurname.xml](#) (../samples/001-Placement-BySurname.xml) を参照してください。

The screenshot shows a configuration window with several policy rules. The rule 'Surname A-I: place in Users1' is expanded to show its conditions and actions.

[Setup Local Variables](#)

[Surname A-I: place in Users1](#)

条件

- if クラス名 等しい "User"
- AND** if 操作属性 'Surname' 等しい "[a-i].*"

アクション

- 操作ターゲットDNの設定(DN("Training\Users\Active\Users1"+"\"+操作属性("CN")))
- メッセージのトレース(color="yellow",ローカル変数("LVUsers1"))
- イベントの生成(id="1000",text1=ローカル変数("LVUsers1"))

[Surname J-R:place in Users2](#)

[Surname S-Z: place in Users3](#)

実行内容 メッセージのトレース

レベルを入力:

色を選択: 黄色

文字列を入力:* ローカル変数("LVUsers1")

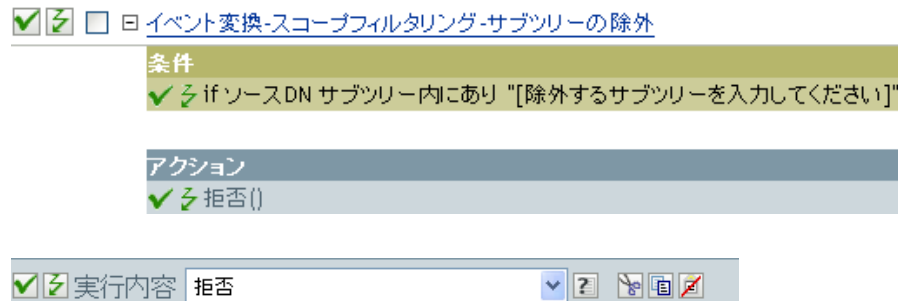
DSTRAC へトレースメッセージを送信します。ローカル変数の内容は LVUsers1 で、DSTRACE では黄色で表示されます。

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 に付属する事前定義されたルールです。60 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」XML 形式でポリシーを参照するには、[predef_transformation_filter_exclude_subtrees.xml \(../samples/predef_transformation_filter_exclude_subtrees.xml\)](#) 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。

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。変数の拡張がサポートされます。詳細については、(210 ページ) [変数の拡張](#) を参照してください。

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、『[Understanding Policies for Identity Manager 3.5.1](#)』の「[Downloading Identity Manager Policies](#)」を参照してください。XML 形式でポリシーを参照するには、[001-Create-RequiredAttrs.xml \(../samples/001-Create-RequiredAttrs.xml\)](#) を参照してください。

The screenshot shows a policy configuration window titled "User required attributes: First/Last Name , Title,Description,Email". It has a "条件" (Conditions) section with one condition: "if クラス名 等しい "User"". Below it is an "アクション" (Actions) section with five actions, each with a "操作属性値がない場合は拒否" (Deny if attribute value is missing) action type:

- 操作属性値がない場合は拒否 ("Given Name")
- 操作属性値がない場合は拒否 ("Surname")
- 操作属性値がない場合は拒否 ("Title")
- 操作属性値がない場合は拒否 ("Description")
- 操作属性値がない場合は拒否 ("Internet EMail Address")

The screenshot shows the "実行内容" (Execution Content) section of the policy configuration. It lists five attributes, each with a "操作属性値がない場合は拒否" (Deny if attribute value is missing) action type and a corresponding input field:

- 名前を入力:* Given Name
- 名前を入力:* Surname
- 名前を入力:* Title
- 名前を入力:* Description
- 名前を入力:* Internet EMail Address

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。

While

指定した条件が **True** と評価されている間、指定したアクションが繰り返されます。

フィールド

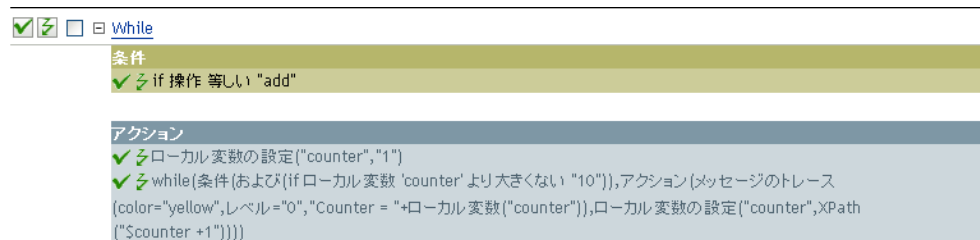
条件

評価される条件を指定します。

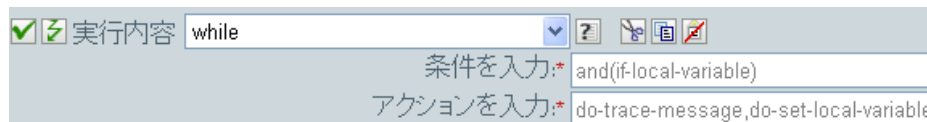
アクション

条件が **True** と評価された場合に繰り返されるアクションを指定します。

例



The screenshot shows a configuration window titled "While". It has two main sections: "条件" (Condition) and "アクション" (Action).
Under "条件", there is a checked checkbox and the text "if 操作 等しい 'add'".
Under "アクション", there are three checked checkboxes and the following text:
"ローカル変数の設定('counter','1')"
"while(条件(および{if ローカル変数 'counter' より大きくない '10'}),アクション(メッセージのトレース {color='yellow',レベル='0','Counter = '+ローカル変数('counter')},ローカル変数の設定('counter',XPath ['\$counter +1'])))"



The screenshot shows the "実行内容" (Execution Content) field. The dropdown menu is set to "while". Below the dropdown, there are two input fields:
"条件を入力:*" with the value "and(if-local-variable)"
"アクションを入力:*" with the value "do-trace-message,do-set-local-variable"

変数の拡張

アクションで動的変数を使用できるようにします。

リマーク

多くのアクションでは、属性またはコンテンツで動的変数の拡張機能をサポートしています。サポートされている場合は、`<variable-name>$` に組み込まれた参照が、5つの名前をもつローカル変数またはグローバル変数の値で置き換えられます。「`<variable-name>$`」は有効な変数名である必要があります。有効な XML 名の詳細については、「[W3C Extensible Markup Language \(http://www.w3.org/TR/2006/REC-xml11-20060816/#sec-suggested-names\)](http://www.w3.org/TR/2006/REC-xml11-20060816/#sec-suggested-names)」を参照してください。

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。

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。

- ◆ 212 ページの「追加されたエンタイトルメント」
- ◆ 213 ページの「関連付け」
- ◆ 214 ページの「属性」
- ◆ 215 ページの「文字」
- ◆ 216 ページの「クラス名」
- ◆ 217 ページの「ターゲット属性」
- ◆ 219 ページの「ターゲット DN」
- ◆ 221 ページの「ターゲット名」
- ◆ 222 ページの「ドキュメント」
- ◆ 223 ページの「エンタイトルメント」
- ◆ 224 ページの「パスワードの生成」
- ◆ 225 ページの「グローバル構成値」
- ◆ 226 ページの「ローカル変数」
- ◆ 228 ページの「名前付きパスワード」
- ◆ 230 ページの「説明」
- ◆ 231 ページの「操作属性」
- ◆ 232 ページの「操作プロパティ」
- ◆ 233 ページの「[パスワード]」
- ◆ 234 ページの「クエリ」
- ◆ 235 ページの「削除された属性」
- ◆ 236 ページの「削除されたエンタイトルメント」
- ◆ 237 ページの「解決」
- ◆ 238 ページの「ソース属性」
- ◆ 239 ページの「ソース DN」
- ◆ 240 ページの「ソース名」
- ◆ 241 ページの「[時刻]」
- ◆ 242 ページの「テキスト」
- ◆ 243 ページの「一意の名前」
- ◆ 246 ページの「一致しないソース DN」
- ◆ 247 ページの「XPath」
- ◆ 248 ページの「変数の拡張」

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です。変数の拡張がサポートされます。詳細については、[\(248 ページ\) 変数の拡張](#)を参照してください。

備考

トークンがノードセットが期待されているコンテキスト内で使用されている場合、トークンはエンタイトルメントのすべての値を含むノードセットに展開します。文字列が期待されているコンテキスト内で使用されている場合は、トークンは検出された文字列値に展開します。

例

 `追加されたエンタイトルメント("manager")`

関連付け

現在の操作から関連付けの値に展開します。



例



例では、Identity Manager とともに提供される事前定義されたルールです。eMBox クライアントの詳細については、「53 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。XML 形式でポリシーを参照するには、[predef_command_delete_to_disable.xml \(../samples/predef_command_delete_to_disable.xml\)](#) を参照してください。

   コマンド変換-無効にする発行者の削除

条件



  if 操作 等しい "delete"

OR   if クラス名 等しい "User"

アクション

  ターゲット属性値の設定("Login Disabled","true")

  関連付けを削除(関連付け(関連付け {}))

  **関連付け()**

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。変数の拡張がサポートされます。詳細については、[\(248 ページ\) 変数の拡張](#)を参照してください。

備考

トークンがノードセットが期待されているコンテキストで使用されている場合は、トークンはその属性のすべての値を含むノードセットに展開します。文字列が期待されているコンテキスト内で使用されている場合は、トークンは検出された文字列値に展開します。

例

例では、Identity Manager とともに提供される事前定義されたルールです。詳細については、[58 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。ポリシーを XML で表示するには、[predef_creation_set_default_password.xml \(../samples/predef_creation_set_default_password.xml\)](#)を参照してください。

[作成-デフォルトパスワードの設定](#)

条件


ifクラス名 等しい "User"

アクション

ターゲットパスワードの設定(属性("Given Name")+属性("Surname"))

属性("Given Name")

属性("Surname")

 **エディタ**

名前:* Given Name



文字

ユニコード*コードポイントで指定される文字に展開します。

備考

Unicode 値と文字の一覧については、[Unicode Code Charts \(http://www.unicode.org/charts/\)](http://www.unicode.org/charts/) を参照してください。



フィールド

文字の値

文字の Unicode コードポイントです。変数の拡張がサポートされます。詳細については、[\(248 ページ\) 変数の拡張](#)を参照してください。

16 進値は、C ベースのプログラミング言語のように 0x というプリフィックスが付いている場合に指定できます。

例

  文字(value="10")

 エディタ

文字の値:*

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



ターゲット属性

指定した属性値オブジェクトに展開します。

フィールド

名前

属性の名前です。変数の拡張がサポートされます。詳細については、[\(248 ページ\) 変数の拡張](#)を参照してください。

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、フィールドを空白のままにします。変数の拡張がサポートされます。詳細については、[\(248 ページ\) 変数の拡張](#)を参照してください。

オブジェクトの選択

現在のオブジェクト、DN、または関連付けを選択します。

備考

トークンがノードセットが期待されているコンテキストで使用されている場合は、トークンはその属性のすべての値を含むノードセットに展開します。文字列が期待されているコンテキスト内で使用されている場合は、トークンは検出された文字列値に展開します。

例


この例は「Govern Groups for User Based on Title」ポリシーからのもので、Novell 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、『[Understanding Policies for Identity Manager 3.5.1](#)』の「[Downloading Identity Manager Policies](#)」を参照してください。ポリシーを XML で表示するには、[003-Command-AddCreateGroups.xml](#) ([../samples/003-Command-AddCreateGroups.xml](#)) 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。

[Set local variables to test existence of groups and for placement](#)

条件	
✓ if クラス名 等しい "User"	AND
✓ if 操作 等しい "add"	
OR ✓ if 操作 等しい "modify"	

アクション	
✓ ローカル変数の設定 ("manager-group-dn", "Users\ManagersGroup")	
✓ ローカル変数の設定 ("manager-group-info", ターゲット属性 ("Object Class", DN(ローカル変数 ("manager-group-dn"))))	
✓ ローカル変数の設定 ("employee-group-dn", "Users\EmployeesGroup")	
✓ ローカル変数の設定 ("employee-group-info", ターゲット属性 ("Object Class", DN(ローカル変数 ("employee-group-dn"))))	

 **ターゲット属性("Object", DN())**

エディタ

名前:	<input type="text" value="Object"/>	
クラス名:	<input type="text"/>	
オブジェクトを選択:	<input type="text" value="DN"/>	<input type="text" value="ローカル変数('manager-group-dn')"/>

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。

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 セグメントの数を指定します。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。ポリシーは、Identity Manager とともに提供される事前定義されたルールです。詳細については、52 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。ポリシーを XML で表示するには、[predef_command_create_dept_container1.xml](#) (./samples/predef_command_create_dept_container1.xml) を参照してください。

☑ ☑ ☐ コマンド変換-部署別コンテナの作成-パート1


条件


- ☑ if 操作 等しい "add"

アクション

- ☑ ローカル変数の設定 ("target-container", ターゲット DN (長さ="-2"))
- ☑ ローカル変数の設定 ("does-target-exist", ターゲット属性 ("objectclass", クラス名="OrganizationalUnit", DN (ローカル変数 ("target-container"))))

📦 ターゲット DN (長さ="-2")

 **エディタ**

開始:	<input type="text" value="0"/>
長さ:	<input type="text" value="-2"/>
ソースDN形式に変換:	<input type="text" value="False"/> 

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



ドキュメント


URI でポイントされた XML ドキュメントを読み込み、ドキュメントノードをノードセットで返します。URI は含まれるポリシーの URI に対して相対的な場合があります。エラーが発生すると、結果は空のノードセットになります。

フィールド

XML ドキュメント URI

XML ドキュメント URI を指定します。

例

 ドキュメント("Novell\South\Driver Set\Delimited Text")

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です。変数の拡張がサポートされます。詳細については、[\(248 ページ\) 変数の拡張](#)を参照してください。

備考

トークンがノードセットが期待されているコンテキスト内で使用されている場合、トークンはエンタイトルメントのすべての値を含むノードセットに展開します。文字列が期待されているコンテキスト内で使用されている場合は、トークンは検出された文字列値に展開します。

例

品  エンタイトルメント("manager")

パスワードの生成

指定したパスワードポリシーに一致するランダムパスワードを生成します。

フィールド

パスワードポリシー

ランダムに生成されたパスワードを受け取るパスワードポリシーの DN です。変数の拡張がサポートされます。詳細については、[\(248 ページ\) 変数の拡張](#)を参照してください。

ポリシーに関連する参照した DN のレンダリング

パスワードポリシーの DN を、作成するポリシーに対して相対的にするかどうかを選択します。

例

 `パスワードの生成(policy-dn="Security\Password\Policies\Sample Password Policy")`

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。変数の拡張がサポートされます。詳細については、[\(248 ページ\) 変数の拡張](#)を参照してください。

例

 **グローバル設定値("ConnectedSystemName")**

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。変数の拡張がサポートされます。詳細については、[\(248 ページ\) 変数の拡張](#)を参照してください。

例

この例は「Govern Groups for User Based on Title」ポリシーからのもので、Novell 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、『[Understanding Policies for Identity Manager 3.5.1](#)』の「[Downloading Identity Manager Policies](#)」を参照してください。ポリシーを XML で表示するには、[003-Command-AddCreateGroups.xml](#) ([../samples/003-Command-AddCreateGroups.xml](#)) 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。

このスクリーンショットは、ポリシーのアクションと条件を設定するためのインターフェースを示しています。各項目にはチェックボックスと展開アイコンがあります。

- [Set local variables to test existence of groups and for placement](#)
- [Create ManagersGroup, if needed](#)
- 条件**
 - if ローカル変数 'manager-group-info' 使用可能
 - AND** if ローカル変数 'manager-group-info' 等しくない "group"
- アクション**
 - ターゲットオブジェクトの追加 (クラス名="Group", when="before", DN(ローカル変数("manager-group-dn")))
- [Create EmployeesGroup, if needed](#)
- [If Title indicates Manager, add to ManagerGroup and set rights](#)
- [If Title does not indicate Manager, add to EmployeeGroup and set rights](#)

ローカル変数("manager_group_dn")

このスクリーンショットは、「エディタ」ダイアログボックスを示しています。変数名として「manager_group_dn」が入力されています。下部にはローカル変数の候補リストと「閉じる」ボタンがあります。

エディタ

変数名: *

ローカル変数

- [current-node](#)
- [current-op](#)
- [current-value](#)
- [fromNds](#)

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。例では、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。

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。変数の拡張がサポートされます。詳細については、[\(248 ページ\) 変数の拡張](#)を参照してください。

例

名前付きパスワード名詞トークンは、名前付きパスワードがドライバオブジェクトに設定されている場合のみ使用可能です。名前付きパスワードは、パスワードを暗号化形式で保存するために使用されます。ときには、アクションを機能させるのを許可するためにパスワードを指定することが必要な場合があります。パスワードをクリアテキストとして入力した場合、セキュリティに危険が生じます。

例では、[ワークフローの開始](#)アクションを使用します。ワークフロー管理者用のパスワードの入力が要求されます。XML 形式でポリシーを参照するには、[start_workflow.xml \(../samples/start_workflow.xml\)](#) を参照してください。

Start WorkFlow

条件

if 操作 等しい "add"

アクション

ワークフローの開始 (id="CN=ApproveCellPhone,CN=RequestDefs,CN=AppConfig,CN=UserApplication,CN=Driverset,O=novell",url="http://localhost:8080/IDMProv",workflow-id="cn=WorkflowAdmin,o=People",arg-password{名前付きパスワード("workflow-admin")},DN(DNの解析("qualified-slash","ldap",XPath("@qualified-src-dn"))),provider="ACMEWireless",reason="new hire")

実行内容

ワークフローの開始	
プロビジョニングリクエストDNを入力:	CN=ApproveCellPhone,CN=RequestDefs,CN=AppConfig,CN=UserApplication
ユーザアプリケーションのURLを入力:	http://localhost:8080/IDMProv
認可ユーザのDNを入力:	cn=WorkflowAdmin,o=People
認定ユーザパスワードを入力:	Named Password("workflow-admin")
受信者のDNを入力:	Parse DN("qualified-slash","ldap",XPath("@qualified-src-dn"))
追加の引数を入力:	provider,reason

名前付きパスワード("workflow-admin")

エディタ

パスワード名: * workflow-admin

名前付きパスワード

[smtp-admin](#)

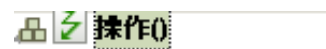
[workflow-admin](#)

閉じる

説明

現在の操作の名前に展開します。

例



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。変数の拡張がサポートされます。詳細については、[\(248 ページ\) 変数の拡張](#)を参照してください。

例

例には名字属性の最初の 1 文字を基にユーザオブジェクトの配置ポリシーを実装する 4 つのルールが含まれます。これはメッセージのトレースとカスタム Novell Audit または Sentinel イベントの両方を生成します。これは、「Policy to Place by Surname」という名前のポリシーで、Novell 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、『[Understanding Policies for Identity Manager 3.5.1](#)』の「[Downloading Identity Manager Policies](#)」を参照してください。XML 形式でポリシーを参照するには、[001-Placement-BySurname.xml](#) ([../samples/001-Placement-BySurname.xml](#)) を参照してください。

The screenshot displays a configuration window with a list of policies and a detailed view of one policy.

Policy List:

- Setup Local Variables
- Surname A-I: place in Users1
- Surname J-R: place in Users2
- Surname S-Z: place in Users3

Policy Details for 'Surname A-I: place in Users1':

条件 (Conditions):

- if クラス名 等しい "User"
- AND if 操作属性 'Surname' 等しい "[a-i].*"

アクション (Actions):

- 操作ターゲットDNの設定 (DN("Training\Users\Active\Users1"+"\"+操作属性("CN")))
- メッセージのトレース (color="yellow",ローカル変数("LVUsers1"))
- イベントの生成 (id="1000",text1=ローカル変数("LVUsers1"))

Target List:

- "Training\Users\Active\Users1"
- "\"
- 操作属性("CN")

エディタ (Editor):

名前: * CN

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。

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。変数の拡張がサポートされます。詳細については、[\(248 ページ\) 変数の拡張](#)を参照してください。


例

  ターゲット属性("myStoredproperty")

[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

品  パスワード:0

クエリ

クエリがソースまたはターゲットデータストアで実行され、結果のインスタンスが返されます。

フィールド

データストア

クエリするデータストアを指定します。

スコープ

クエリのスコープを選択します。オプションは、エントリー、従属、またはサブツリーです。

結果の最大件数

クエリから返される結果の最大値を指定します。

クラス名

クエリのクラス名を指定します。クラス名を指定しないと、すべてのクラスが検索されます。変数の拡張がサポートされます。詳細については、[\(248 ページ\) 変数の拡張](#)を参照してください。

オブジェクトの選択

クエリのベースを指定します。現在のオブジェクト、DN、または関連付けを指定できます。


一致属性

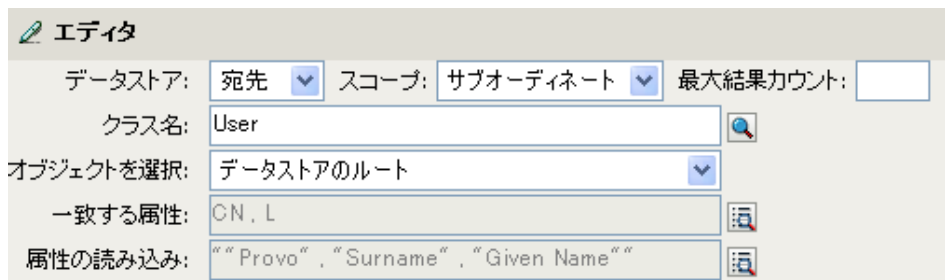
検索する属性を選択します。

文字列

返される属性のセットを指定します。何も指定しないと、属性は読み取られません。すべての属性を読み取るにはアスタリスク (*) を使用します。

例

 **検索(スコープ="subordinates",クラス名="User",一致("CN"),一致("L"),"Provo", "Surname", "Given Name")**



エディタ

データストア:	宛先	スコープ:	サブオーディネート	最大結果カウント:	
クラス名:	User				
オブジェクトを選択:	データストアのルート				
一致する属性:	CN, L				
属性の読み込み:	"Provo", "Surname", "Given Name"				

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。変更操作にのみ適用します。

フィールド



名前

属性の名前を指定します。変数の拡張がサポートされます。詳細については、[\(248 ページ\) 変数の拡張](#)を参照してください。

備考

トークンがノードセットが期待されているコンテキストで使用されている場合は、トークンはその属性のすべての値を含むノードセットに展開します。文字列が期待されているコンテキスト内で使用されている場合は、トークンは検出された文字列値に展開します。

例

  削除された属性("Member")

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。変数の拡張がサポートされません。詳細については、[\(248 ページ\) 変数の拡張](#)を参照してください。

備考

トークンがノードセットが期待されているコンテキスト内で使用されている場合、トークンはエンタイトルメントのすべての値を含むノードセットに展開します。文字列が期待されているコンテキスト内で使用されている場合は、トークンは検出された文字列値に展開します。

例

  削除されたエンタイトルメント("manager")

解決

DN を関連付けキーに、または関連付けキーを指定したデータストア内の DN に解決します。

フィールド

データストア


クエリするターゲットまたはソースデータストアを選択します。

選択した解決タイプ

関連付けキーを DN に解決するか、DN を関連付けキーに解決するかを選択します。

例

 **解決(datastore="src",DN())**

 **エディタ**

データストア:

解決タイプの選択:

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、フィールドを空白のままにします。変数の拡張がサポートされます。詳細については、[\(248 ページ\) 変数の拡張](#)を参照してください。

名前

属性の名前です。変数の拡張がサポートされます。詳細については、[\(248 ページ\) 変数の拡張](#)を参照してください。



オブジェクト

ソース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。

備考

トークンがノードセットが期待されているコンテキストで使用されている場合は、トークンはその属性のすべての値を含むノードセットに展開します。文字列が期待されているコンテキスト内で使用されている場合は、トークンは検出された文字列値に展開します。

例

  ソース属性("Member",クラス名="Group")

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

  ソースDN(長さ="-2")

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



[時刻]

現在の日付 / 時刻を指定された書式、言語、および時間帯に展開します。

フィールド

書式

日付 / 時刻の書式を指定します。指定した時刻フォーマットを選択するか、またはカスタムフォーマットパターンを指定します。変数の拡張がサポートされます。詳細については、[\(248 ページ\) 変数の拡張](#)を参照してください。



Language

言語を指定します。(デフォルトはシステムの現在の言語になります) 変数の拡張がサポートされます。詳細については、[\(248 ページ\) 変数の拡張](#)を参照してください。

時間帯

タイムゾーンを指定します。(デフォルトはシステムの現在のタイムゾーンになります) 変数の拡張がサポートされます。詳細については、[\(248 ページ\) 変数の拡張](#)を参照してください。

例

  時間(format="!CTIME",tz="UTC")

エディタ	
書式:*	<input type="text" value="!CTIME"/>
使用言語:	<input type="text"/>
タイムゾーン:	<input type="text" value="UTC"/>

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。変数の拡張がサポートされます。詳細については、[\(248 ページ\) 変数の拡張](#)を参照してください。

例

この例は「Govern Groups for User Based on Title」ポリシーからのもので、Novell 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、『[“ Understanding Policies for Identity Manager 3.5.1 ”](#)』の「[/Downloading Identity Manager Policies](#)」を参照してください。ポリシーを XML 形式で参照するには、[003-Command-AddCreateGroups.xml](#) ([../samples/003-Command-AddCreateGroups.xml](#)) 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。

Set local variables to test existence of groups and for placement

条件

- ✓ if クラス名 等しい "User"
- AND
- ✓ if 操作 等しい "add"
- OR ✓ if 操作 等しい "modify"

アクション

- ✓ ローカル変数の設定 ("manager-group-dn", "Users\ManagersGroup")
- ✓ ローカル変数の設定 ("manager-group-info", ターゲット属性 ("Object Class", DN(ローカル変数 ("manager-group-dn"))))
- ✓ ローカル変数の設定 ("employee-group-dn", "Users\EmployeesGroup")
- ✓ ローカル変数の設定 ("employee-group-info", ターゲット属性 ("Object Class", DN(ローカル変数 ("employee-group-dn"))))

"Users\ManagersGroup"

エディタ

テキスト:

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。

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

属性名

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。オブジェクトはサブツリーまたは従属です。

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。

カウンタの使用

カウンタをいつ使うかを選択します。オプションは次のとおりです。

- ◆ 常にカウンタを使用する
- ◆ カウンタを使用しない
- ◆ なしですべてのパターンの失敗後

カウンタパターン

カウンタとともに使用するパターンを選択します。オプションは次のとおりです。

- ◆ 最初のパターンのみ
- ◆ 最後のパターンのみ
- ◆ すべてのパターンで使用

開始

カウンタの開始値です。

桁

カウンタの桁の幅を指定します。デフォルトは1です。[カウンタに先行ゼロを埋め込む] オプションは、桁長を一致するためにプリペンドされます。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。

名前を構成できなかった場合

一意な名前を構成できなかった場合に実行するアクションを選択します。オプションは次のとおりです。

- ◆ 無視、空を返す
- ◆ 警告を生成、空の名前を返す
- ◆ エラーを生成、現在のトランザクションを中断

- ◆ 致命的なエラーを生成、ドライバをシャットダウン

備考

各 <arg-string> 要素は、指定された名前を生成するために使用されるパターンを提供します。

指定された名前は、<arg-dn> 要素または <arg-association> 要素を消えるオンベースとしておよびスコープをクエリのスコープとして使用してターゲットデータストアに対して名前属性内の値に対するクエリを実行することによってテストされます。ターゲットデータストアが識別ポールドであり、名前が省略されている場合は、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アプリケーションの場合、name は必須です。

パターンは、counter-use と counter-pattern で示されるカウンタを使用して、または使用しないでテストできます。パターンをカウンタを使用してテストする場合、パターンは名前が見つかるまで付加されたカウンタで繰り返しテストされます。インスタンスが返されない場合、カウンタを使い果たします。カウンタの開始値は、counter-start で指定され、カウンタの最大値は、counter-digits で指定される最大桁数として指定されます。桁数が指定された値より小さい場合、counter-pad 属性が「False」に設定されている場合を除き、カウンタの右側にゼロが埋め込まれます。カウンタを指定した桁数で表すことができない場合、カウンタが最大の値までカウントされたと見なされます。

指定された名前が一意なものだと判断されると、名前のテストは終了し、一意な名前が返されます。

指定した名前は次の順番でテストされます。

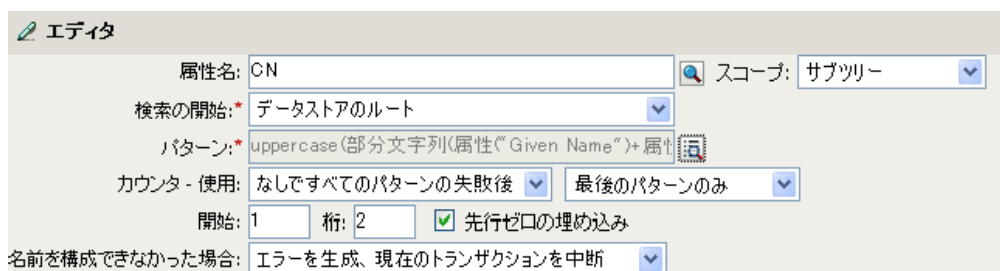
- ◆ 各パターンは指定した順にテストされます。counter-use="always" で、パターンが counter-pattern に示されているいずれかのパターンの場合、パターンはカウンタを使用してテストされます。それ以外はカウンタなしでテストされます。
- ◆ counter-use="fallback" で、すべてのパターンで一意な名前が見つからなかった場合は、counter-pattern に示されているパターンがカウンタ付きで再試行されます。

パターンとカウンタのすべての組み合わせで見つからなかった場合は、on-unavailable のアクションが実行されます。

例

 一意の名前("CN",スコープ="subtree",uppercase()+uppercase()+uppercase())

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。



エディタ

属性名: CN スコープ: サブツリー

検索の開始: データストアのルート

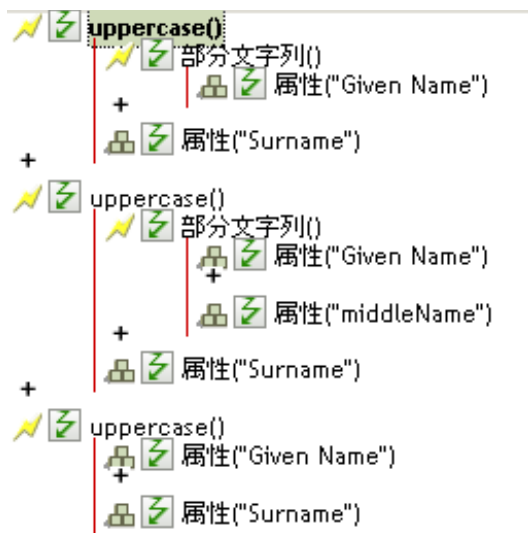
パターン: uppercase(部分文字列(属性("Given Name")+属)

カウンタ - 使用: なしですべてのパターンの失敗後 最後のパターンのみ

開始: 1 桁: 2 先行ゼロの埋め込み

名前を構成できなかった場合: エラーを生成、現在のトランザクションを中断

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99つ生成されます(パターン1からパターン9)。

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

一致するものがない場合は、DN 全体が使用されます。

例

例では、Identity Manager とともに提供される事前定義されたルールです。詳細については、64 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。ポリシーを XML で表示するには、[predef_match_sub_mirrored.xml \(../samples/predef_match_sub_mirrored.xml\)](#) 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。

The screenshot shows a configuration window for a policy named "一致-購読者(ミラーリング) - LDAP形式". It includes sections for "条件" (Condition) and "アクション" (Action). Below these are several tokens with expand/collapse icons. At the bottom, there is an "エディタ" (Editor) section with a dropdown menu for "ターゲットDN形式に変換:" (Convert to target DN format) set to "True".

一致-購読者(ミラーリング) - LDAP形式

条件

- if ソースDN サブツリー内にあり "[ソース階層のベースを入力してください]"

アクション

- ローカル変数の設定("dest-base", "[宛先階層のベースを入力してください]")
- 一致オブジェクトの検索(スコープ="entry", DN{一致しないソースDN(変換="true")+","+ローカル変数("dest-base")})

一致しないソースDN(変換="true")

","

ローカル変数("dest-base")

エディタ

ターゲットDN形式に変換: True

XPath

XPath 1.0 の式の評価結果に展開します。

フィールド



Expression

評価する XPath 1.0 の式。

備考

ポリシーとともに XPath 式を使用する方法の詳細については、『*Understanding Policies for Identity Manager 3.5.1*』の「**XPath 1.0 Expressions**」を参照してください。

例

  XPath("[@attr-name='OU']/value[starts-with(string(.),xxx)])

変数の拡張

名詞トークンでダイナミック変数を使用できるようにします。

注

多くの名詞トークンは、属性またはコンテンツ内での動的変数拡張をサポートしていません。サポートされている場合は、`<variable-name>$` に組み込まれた参照が、5つの名前をもつローカル変数またはグローバル変数の値で置き換えられます。「`<variable-name>$`」は有効な変数名である必要があります。有効な XML 名の詳細については、「[W3C Extensible Markup Language \(http://www.w3.org/TR/2006/REC-xml11-20060816/#sec-suggested-names\)](http://www.w3.org/TR/2006/REC-xml11-20060816/#sec-suggested-names)」を参照してください。

動詞トークン

12

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。

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。

- ◆ 250 ページの「Base64 デコード」
- ◆ 251 ページの「Base64 エンコード」
- ◆ 252 ページの「変換時間」
- ◆ 254 ページの「ターゲット DN のエスケープ」
- ◆ 255 ページの「ソース DN のエスケープ」
- ◆ 256 ページの「参加」
- ◆ 257 ページの「小文字」
- ◆ 258 ページの「マップ」
- ◆ 259 ページの「DN の解析」
- ◆ 261 ページの「すべて置換」
- ◆ 262 ページの「最初を置換」
- ◆ 264 ページの「分割」
- ◆ 265 ページの「部分文字列」
- ◆ 267 ページの「大文字」
- ◆ 268 ページの「XML 解析」
- ◆ 269 ページの「XML シリアライズ」
- ◆ 270 ページの「変数の拡張」

Base64 デコード

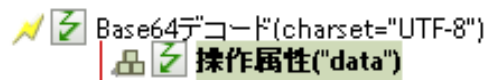
Base64 エンコードデータからの格納されたトークンの結果をバイト単位にデコードして、指定された文字セットを使用して、それらのバイトを文字列に変換します。

フィールド

文字セット

デコードされたバイトを文字列に変換する文字のセットを指定します。任意の Java サポート文字セットを使用できます。フィールドを空白のままにすると、デフォルトで `file.encoding` システムプロパティで指定されたシステムエンコードが文字セットとして使用されます。変数の拡張がサポートされます。詳細については、[\(270 ページ\)](#) **変数の拡張**を参照してください。

例



The image shows a configuration field with a lightning bolt icon on the left. The text of the field is "Base64デコード(charset='UTF-8')". Below the text, there is a vertical line and a small icon of a box with a checkmark, followed by the text "操作属性('data')".

Base64 エンコード

格納されたトークンの結果を指定された文字セットを使用してバイト単位に変換してから、Base64 エンコードします。

フィールド

文字セット

文字列をバイト単位に変換する文字セットを指定します。任意の Java サポート文字セットを使用できます。フィールドを空白のままにすると、デフォルトで `file.encoding` システムプロパティで指定されたシステムエンコードが文字セットとして使用されます。変数の拡張がサポートされます。詳細については、[\(270 ページ\) 変数の拡張](#)を参照してください。

例

```
Base64エンコード(charset="UTF-8")
| 操作属性("Surname")
```

変換時間

格納されたトークンの結果で表わされた日付および時刻を、ソースのフォーマット、言語、およびタイムゾーンから、ターゲットのフォーマット、言語、およびタイムゾーンに変換します。

フィールド

ソースフォーマット

ソースの日付/時刻のフォーマットを指定します。指定した時刻フォーマットを選択するか、またはカスタムフォーマットパターンを指定します。変数の拡張がサポートされます。詳細については、[\(270 ページ\) 変数の拡張](#)を参照してください。

ソース言語

ソース言語を指定します(デフォルトはシステムの現在の言語になります)。変数の拡張がサポートされます。詳細については、[\(270 ページ\) 変数の拡張](#)を参照してください。

ソースタイムゾーン

ソースタイムゾーンを指定します(デフォルトはシステムの現在の時間帯になります)。変数の拡張がサポートされます。詳細については、[\(270 ページ\) 変数の拡張](#)を参照してください。

ターゲットフォーマット

ターゲットの日付/時刻のフォーマットを指定します。指定した時刻フォーマットを選択するか、またはカスタムフォーマットパターンを指定します。変数の拡張がサポートされます。詳細については、[\(270 ページ\) 変数の拡張](#)を参照してください。

ターゲット言語


ターゲット言語を指定します(デフォルトはシステムの現在の言語になります)。変数の拡張がサポートされます。詳細については、[\(270 ページ\) 変数の拡張](#)を参照してください。









ターゲットタイムゾーン

ターゲットタイムゾーンを指定します(デフォルトはシステムの現在の時間帯になります)。変数の拡張がサポートされます。詳細については、[\(270 ページ\) 変数の拡張](#)を参照してください。

例

```
変換時間:(src-format="yyyy-MM-dd",src-lang="ja-JP",src-tz="Japan",dest-format="yyyy-dd-MM"  
操作属性("birthdate")
```


 **エディタ**

ソースの形式:*	yyyy-MM-dd	 
ソースの言語:	ja-JP	
ソースのタイムゾーン:	Japan	
宛先の形式:*	yyyy-dd-MM	 
宛先の言語	ja-JP	
宛先のタイムゾーン:	Japan	

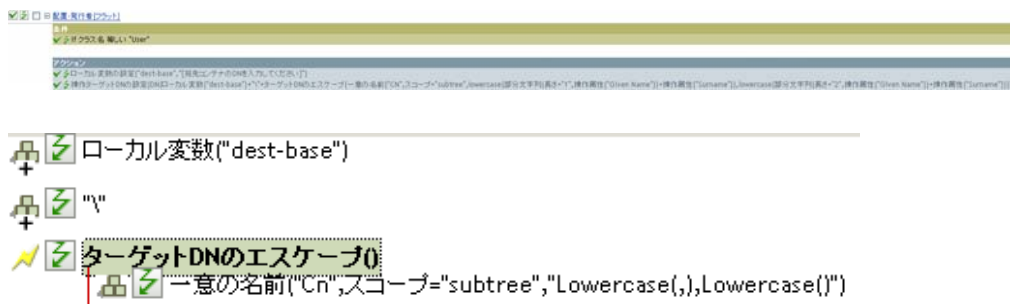
ターゲット DN のエスケープ

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格納されたトークンをエスケープします。

例

例では、Identity Manager とともに提供される事前定義されたルールです。詳細については、[69 ページのセクション 6.16 「配置 - 発行者 \(フラット\)」](#) を参照してください。ポリシーを XML で表示するには、[predef_place_pub_flat.xml \(../samples/predef_place_pub_flat.xml\)](#) 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。

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格納されたトークンをエスケープします。

例

✎  ターゲット DN のエスケープ()
|   属性("Surname")

参加

格納されたトークンのノードセット結果内のノードの値を結合し、区切り記号で指定された文字で値を区切ります。カンマ区切り値 (CSV) が True の場合、CSV 引用ルールが値に適用されます。

フィールド

区切り記号



(オプション) 結合された値を区切るために使用される文字列を指定します。変数の拡張がサポートされます。詳細については、[\(270 ページ\) 変数の拡張](#)を参照してください。


CSV 引用ルールの適用

CSV 引用値を適用します。

例

例では、グループのすべてのメンバを CSV レコードに組み込みます。

  **結合 (delimiter=";", csv="true")**
  操作属性 ("Member")

 **エディタ**

区切り記号: CSV 引用ルールの適用

小文字

格納されたトークン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。このポリシーの名前は「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)」で、Novell 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、『[Understanding Policies for Identity Manager 3.5.1](#)』の「[Downloading Identity Manager Policies](#)」を参照してください。ポリシーを XML で表示するには、[001-Command-SetEmailByGivenNameAndSurname.xml](#) (../samples/001-Command-SetEmailByGivenNameAndSurname.xml) を参照してください。

Set email address: name@slartybartfast.com; name = (1 char of Given Name + Surname) <= 8 chars

条件

- if クラス名 等しい "User"
- AND if 操作属性 'Given Name' 使用可能
- AND if 操作属性 'Surname' 使用可能

アクション

- 操作属性のストリップ("Internet EMail Address")
- ターゲット属性値の設定("Internet Email Address", Lowercase(Substring(length="8", Substring(length="1", Operation Attribute("FirstName")) + Operation Attribute("LastName")) + "@slartybartfast.com"))

lowercase()

部分文字列(長さ="8")

+ 操作属性("FirstName")

+ 操作属性("LastName")

+ "@slartybartfast.com"

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。

マップ

指定したマッピングテーブル内のソース列からターゲット列によって指定された値から格納されたトークンの結果をマップします。

備考

このトークンが、ノードセットの結果が予測され、複数の行がマップされている値と一致するコンテキスト内で評価される場合、各一致する行のターゲット列からの値を含むノードセットが返されます。それ以外の場合は、最初の一致する行の値だけが返されます。

テーブル属性は、使用されるマッピングテーブルを含むリソースオブジェクトのスラッシュ形式の DN である必要があります。DN は含まれるポリシーと関連していることがあります。

フィールド

マッピングテーブルの DN

マッピングテーブルを含むリソースオブジェクトのスラッシュ形式の DN を指定します。変数の拡張がサポートされます。詳細については、[\(270 ページ\) 変数の拡張](#)を参照してください。

ポリシーに関連する参照された DN

これが有効になっている場合、ポリシーに関連するマッピングテーブル DN が表示されます。これがデフォルトの設定です。

ソース列名

ソース列の名前を指定します。変数の拡張がサポートされます。詳細については、[\(270 ページ\) 変数の拡張](#)を参照してください。

ターゲット列名

ターゲット列の名前を指定します。変数の拡張がサポートされます。詳細については、[\(270 ページ\) 変数の拡張](#)を参照してください。

例

```
マップ(table="/Department Table",dest="code",src="dept")
| 品 操作属性("OU")
```

エディタ		*必須
マッピングテーブルDN:	<input type="text" value="/Department Table"/>	<input checked="" type="checkbox"/> ポリシーに関連する参照したDNのリンク
ソース列の名前:	<input type="text" value="dept"/>	
宛先の列の名前:	<input type="text" value="code"/>	

DN の解析

格納されたトークンの DN を別のフォーマットに変換します。

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 $-2 = (5 + (-2)) + 1 = 4$ 。

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。

- ◆ タイプされた名前のブールフラグ: 0 は名前が入力されていない、1 は入力されていることを意味します。
- ◆ Unicode* マップなし文字ブールフラグ: 0 はマップできない Unicode 文字 (\\FFFF などのエスケープ文字付きの 16 進数文字列) を出力または解釈しないことを意味します。eDirectory では、Unicode 文字の 0xfeff、0xffff、0xfffd、および 0xffff は使用できません。

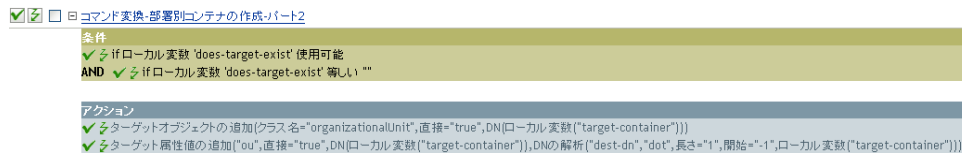
- ◆ 相対 RDN 区切り記号
- ◆ RDN 区切り記号
- ◆ 名前ディバイダ
- ◆ 名前の値の区切り記号
- ◆ ワイルドカード文字
- ◆ エスケープ文字

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。例では、Identity Manager とともに提供される事前定義されたルールです。詳細については、52 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。ポリシーを XML で表示するには、[predef_command_create_dept_container2.xml \(../samples/predef_command_create_dept_container2.xml\)](#) を参照してください。



DNの解析("dest-dn","dot",開始="-1",長さ="1")
 ローカル変数("target-container")

エディタ	
開始:	<input type="text" value="-1"/>
長さ:	<input type="text" value="-1"/>
ソースDNのフォーマット:	<input type="text" value="ターゲットDN"/>
ターゲットDNのフォーマット:	<input type="text" value="ドット"/>

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。

すべて置換

格納されたトークン内のすべての正規表現を置換します。

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。変数の拡張がサポートされます。詳細については、[\(270 ページ\) 変数の拡張](#)を参照してください。

置換文字列

置換する文字列を指定します。変数の拡張がサポートされます。詳細については、[\(270 ページ\) 変数の拡張](#)を参照してください。

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\(java.lang.String\)](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



エディタ	
正規表現:*	()
置換文字列:	\$1

最初を置換

格納されたトークン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。変数の拡張がサポートされます。詳細については、[\(270 ページ\) 変数の拡張](#)を参照してください。

置換文字列

置換する文字列を指定します。変数の拡張がサポートされます。詳細については、[\(270 ページ\) 変数の拡張](#)を参照してください。

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文字列で置き換えられます。

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)) (<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。ルールは、Identity Manager とともに提供される事前定義されたルールです。詳細については、[61 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。ポリシーを XML で表示するには、[prefdef_transformation_reformat_telephone1](#) (`./samples/prefdef_transformation_reformat_telephone1.xml`) 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。

The screenshot shows a configuration window for a policy rule. At the top, there is a title bar: "入出力変換-電話番号の形式を(nnn) nnn-nnnnからnnn-nnn-nnnnに変更". Below the title bar, there are two sections: "条件" (Condition) and "アクション" (Action). The "条件" section contains a green checkmark icon and the text "この条件はTrueと評価されます。". The "アクション" section contains a green checkmark icon and the text "操作属性の再フォーマット('phone',最初を置換('^\\(\\d\\d\\d\\)\\s*(\\d\\d\\d)-(\\d\\d\\d\\d)\$','\$1-\$2-\$3',ローカル変数('current-value'))". Below these sections, there is a summary line: "最初を置換('^\\(\\d\\d\\d\\)\\s*(\\d\\d\\d)-(\\d\\d\\d\\d)\$','\$1-\$2-\$3')". To the right of this line, there is a small icon of a box with a checkmark and the text "ローカル変数('current-value')".

✎ エディタ

正規表現:	<code>^(?<d1>[0-9]{3})\s*(?<d2>[0-9]{3})-(?<d3>[0-9]{4})\$</d3></d2></d1></code>
置換文字列:	<code>1\$-2\$-3\$</code>

正規表現 `^(?[0-9]{3})\s*(?[0-9]{3})-(?[0-9]{4})$` は、`(nnn) nnn-nnnn` を、正規表現 `1-2-3$` 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。

分割

格納されたトークンの結果を、区切り記号で指定されたパターンに基づくテキストノードで構成されているノードセットに分割します。カンマ区切り値 (CSV) が **True** の場合、CSV 引用ルールが文字列の解析中に使用されます。

フィールド

区切り記号

区切り記号の文字と一致する正規表現です。変数の拡張がサポートされます。詳細については、[\(270 ページ\) 変数の拡張](#)を参照してください。

CSV 引用ルールの適用

CSV 引用値を適用します。

例



分割(delimiter=',',csv='true')
"Doe,John,"Doe,John"



エディタ
区切り記号: CSV引用ルールの適用

部分文字列

格納されたトークン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です。負の数值は (文字の合計数 + 長さ) + 1 として解釈されます。たとえば、-1 は全長または元の文字列を表します。-2 が指定されている場合は、長さは全てで -1 です。5 文字の文字列の場合、長さが -1 の場合は、 $(5 + (-1)) + 1 = 5$ 、長さが -2 の場合は、 $(5 + (-2)) + 1 = 4$ 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。このポリシーの名前は「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)」で、Novell 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、『[Understanding Policies for Identity Manager 3.5.1](#)』の「[Downloading Identity Manager Policies](#)」を参照してください。ポリシーを XML で表示するには、[001-Command-SetEmailByGivenNameAndSurname.xml \(../samples/001-Command-SetEmailByGivenNameAndSurname.xml\)](#) を参照してください。

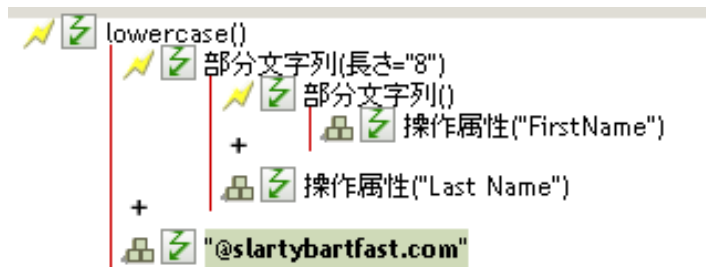
Set email address: name@slartybartfast.com; name = (1 char of Given Name + Surname) <= 8 chars

条件

- ✓ if クラス名 等しい "User"
- AND ✓ if 操作属性 'Given Name' 使用可能
- AND ✓ if 操作属性 'Surname' 使用可能

アクション

- ✓ 操作属性のストリップ ("Internet EMail Address")
- ✓ ターゲット属性値の設定 ("Internet Email Address", Lowercase(Substring (length="8", Substring (length="1", Operation Attribute ("FirstName")) + Operation Attribute ("LastName")) + "@slartybartfast.com"))



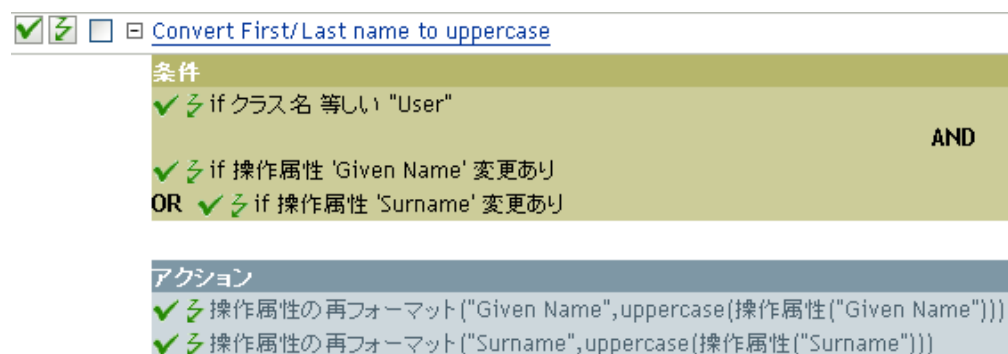
部分文字列トークンは、ターゲット属性値の設定アクションで2度使用されます。名前属性の最初の文字列を取得し、名字属性の8文字を追加して、1つの部分文字列を作成します。

大文字

格納されたトークン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 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、『[Understanding Policies for Identity Manager 3.5.1](#)』の「[Downloading Identity Manager Policies](#)」を参照してください。ポリシーを XML で表示するには、[002-Command-UppercaseNames.xml \(../samples/002-Command-UppercaseNames.xml\)](#) を参照してください。



Convert First/Last name to uppercase

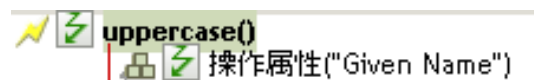
条件

- if クラス名 等しい "User"
- if 操作属性 'Given Name' 変更あり
- OR if 操作属性 'Surname' 変更あり

AND

アクション

- 操作属性の再フォーマット("Given Name",uppercase(操作属性("Given Name")))
- 操作属性の再フォーマット("Surname",uppercase(操作属性("Surname")))



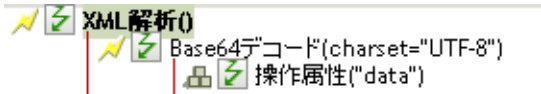
uppercase()

- 操作属性("Given Name")

XML 解析

XML として格納されたトークンの結果を解析し、ノードセット内のドキュメントノードを返します。内容のトークンの結果が正しい形式の XML ではない場合、または何らかの理由で解析できない場合は、空のノードセットが返されます。

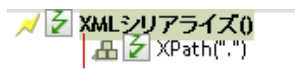
例



XML シリアライズ

格納されたトークンのノードセット結果を XML としてシリアライズします。ノードセットの内容に応じて、結果の文字列は、適格な XML ドキュメントか解析済みの適格な標準エンティティになります。

例



変数の拡張

動詞トークンでの動的変数の使用を許可。

注

多くの動詞トークンは、属性またはコンテンツ内での動的変数拡張をサポートしています。サポートされている場合は、`<variable-name>$` に組み込まれた参照が、5つの名前をもつローカル変数またはグローバル変数の値で置き換えられます。「`<variable-name>$`」は有効な変数名である必要があります。有効な XML 名の詳細については、「[W3C Extensible Markup Language \(http://www.w3.org/TR/2006/REC-xml11-20060816/#sec-suggested-names\)](http://www.w3.org/TR/2006/REC-xml11-20060816/#sec-suggested-names)」を参照してください。