

Novell Designer 2.1 のポリシー

2.1

September 28, 2007

DESIGNER 2.1 のポリシー

www.novell.com

N

Novell®

保証と著作権

米国 Novell, Inc. およびノベル株式会社は、この文書の内容または使用について、いかなる保証、表明または約束も行っておりません。また文書の商品性、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。また、本書の内容は予告なく変更されることがあります。

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おりません。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。

本契約の下で提供される製品または技術情報はすべて、米国の輸出規制および他国の商法の制限を受けます。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. 本ドキュメントの一部または全体を無断で複写・転載することは、その形態を問わず禁じます。

米国 Novell, Inc. は、本文書に記載されている製品に統合されている技術に関する知的所有権を保有します。具体的には、これらの知的所有権には、[Novell Legal Web サイト \(http://www.novell.com/company/legal/patents/\)](http://www.novell.com/company/legal/patents/) に記載されている 1 つまたは複数の米国特許、米国および他の国における 1 つまたは複数のその他の特許、または申請中の特許が含まれますが、これらに限定されるものではありません。

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	19
1 概要	21
1.1 ポリシー	21
2 プレ Identity Manager 3.5 ポリシービルダ	23
3 ポリシービルダを使用したポリシーの管理	25
3.1 ポリシービルダへのアクセス	25
3.1.1 [Model Outline (モデルアウトライン)] ビュー	26
3.1.2 [Policy Flow (ポリシーフロー)] ビュー	26
3.1.3 ポリシーセット	27
3.2 Policy Builder の使用	29
3.2.1 ヒント	29
3.3 ポリシーの作成	30
3.3.1 ポリシーセットへのアクセス	30
3.3.2 ポリシーセットの使用	31
3.3.3 Add Policy Wizard (ポリシー追加ウィザード) の使用	33
3.4 ルールの作成	37
3.4.1 新しいルールの作成	37
3.4.2 事前定義されたルールの使用	42
3.4.3 既存のルールの包含	42
3.4.4 XML ファイルからのポリシーのインポート	43
3.5 引数の作成	44
3.6 ポリシーの編集	46
3.6.1 ポリシービルダのアクションおよびメニュー項目	47
3.6.2 キーボード操作	48
3.6.3 ポリシーの名前変更	48
3.6.4 自分の作業の保存	49
3.6.5 ポリシーの説明	49
3.7 XML でポリシーを表示	50
4 追加ビルダの使用	51
4.1 アクションビルダ	51
4.1.1 アクションの作成	51
4.1.2 アクションビルダの追加オプション	52
4.2 アクションビルダ	52
4.3 引数ビルダ	53
4.3.1 引数ビルダの起動	54
4.3.2 引数ビルダの例	55
4.4 条件ビルダ	56
4.4.1 条件の作成	57
4.4.2 条件ビルダの追加オプション	57
4.5 条件ビルダ	58
4.6 一致属性ビルダ	59
4.7 アクションの引数コンポーネントビルダ	61
4.8 引数値リストビルダ	62

4.9	名前付き文字列ビルダ	62
4.10	条件の引数コンポーネントビルダ	63
4.11	パターン文字列ビルダ	64
4.12	文字列ビルダ	65
4.13	XPath ビルダ	66
4.14	ネームスペースエディタ	66
4.14.1	ネームスペースを使用した Java クラスへのアクセス	67
5	XPath ビルダの使用	69
6	スキーママッピングポリシーの定義	75
6.1	スキーママップエディタへのアクセス	75
6.1.1	[Outline (アウトライン)] ビュー	75
6.1.2	[Policy Flow (ポリシーフロー)] ビュー	76
6.1.3	ポリシーセットビュー	77
6.1.4	キーボード操作	78
6.2	スキーママッピングポリシーの編集	79
6.2.1	クラスと属性の削除または追加	79
6.2.2	アプリケーションスキーマのリフレッシュ	81
6.2.3	項目の編集	81
6.2.4	項目のソート	81
6.2.5	スキーマの管理	81
6.3	スキーママッピングポリシーのテスト	82
6.4	XML のスキーママッピングポリシーへのアクセス	84
6.5	追加のスキーママップポリシーオプション	84
6.5.1	[Outline (アウトライン)] ビューの追加オプション	84
6.5.2	[Policy Flow (ポリシーフロー)] ビューの追加オプション	85
6.5.3	[Policy Set (ポリシーセット)] ビューの追加オプション	87
7	フィルタを使用したオブジェクトのフローの制御	89
7.1	フィルタエディタへのアクセス方法	90
7.1.1	[Model Outline (モデルアウトライン)] ビュー	90
7.1.2	[Policy Flow (ポリシーフロー)] ビュー	90
7.1.3	ポリシーセットビュー	91
7.1.4	キーボード操作	92
7.2	フィルタの編集	93
7.2.1	クラスと属性の削除または追加	93
7.2.2	複数の属性の変更	94
7.2.3	既存のフィルタのコピー	94
7.2.4	属性のデフォルト値の設定	94
7.2.5	フィルタ設定の変更	95
7.3	フィルタのテスト	99
7.4	XML のフィルタを表示	101
7.5	追加のフィルタオプション	101
7.5.1	[Outline (アウトライン)] ビューの追加オプション	101
7.5.2	[Policy Flow (ポリシーフロー)] ビューの追加オプション	102
7.5.3	[Policy Set (ポリシーセット)] ビューの追加オプション	102
8	事前定義されたルールの使用	105
8.1	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	106
8.1.1	ポリシーの作成	107
8.1.2	事前定義されたルールのインポート	107

8.1.3	ルールの動作	108
8.2	コマンド変換 - 無効にする発行者の削除	109
8.2.1	ポリシーの作成	109
8.2.2	事前定義されたルールのインポート	110
8.2.3	ルールの動作	110
8.3	作成 - 属性が必要	110
8.3.1	ポリシーの作成	110
8.3.2	事前定義されたルールのインポート	111
8.3.3	ルールの動作	112
8.4	作成 - 発行者 - テンプレートの使用	112
8.4.1	ポリシーの作成	112
8.4.2	事前定義されたルールのインポート	113
8.4.3	ルールの動作	113
8.5	作成 - デフォルト属性値の設定	114
8.5.1	ポリシーの作成	114
8.5.2	事前定義されたルールのインポート	115
8.5.3	ルールの動作	115
8.6	作成 - デフォルトパスワードの設定	115
8.6.1	ポリシーの作成	116
8.6.2	事前定義されたルールのインポート	116
8.6.3	ルールの動作	117
8.7	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	117
8.7.1	ポリシーの作成	117
8.7.2	事前定義されたルールのインポート	118
8.7.3	ルールの動作	119
8.8	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	119
8.8.1	ポリシーの作成	119
8.8.2	事前定義されたルールのインポート	120
8.8.3	ルールの動作	120
8.9	入出力変換 - 電話番号の形式を (nnn) nnn-nnnn から nnn-nnn-nnnn に変更	120
8.9.1	ポリシーの作成	121
8.9.2	事前定義されたルールのインポート	121
8.9.3	ルールの動作	122
8.10	入出力変換 - 電話番号の形式を nnn-nnn-nnnn から (nnn) nnn-nnnn に変更	122
8.10.1	ポリシーの作成	122
8.10.2	事前定義されたルールのインポート	123
8.10.3	ルールの動作	124
8.11	一致 - 発行者 (ミラーリング)	124
8.11.1	ポリシーの作成	124
8.11.2	事前定義されたルールのインポート	125
8.11.3	ルールの動作	125
8.12	一致 - 購読者 (ミラーリング) - LDAP 形式	126
8.12.1	ポリシーの作成	126
8.12.2	事前定義されたルールのインポート	127
8.12.3	ルールの動作	127
8.13	一致 - 属性値別	128
8.13.1	ポリシーの作成	128
8.13.2	事前定義されたルールのインポート	129
8.13.3	ルールの動作	129
8.14	配置 - 発行者 (ミラーリング)	129
8.14.1	ポリシーの作成	130
8.14.2	事前定義されたルールのインポート	130
8.14.3	ルールの動作	131
8.15	配置 - 購読者 (ミラーリング) - LDAP 形式	131
8.15.1	ポリシーの作成	132
8.15.2	事前定義されたルールのインポート	132
8.15.3	ルールの動作	133

8.16	配置 - 発行者 (フラット)	133
8.16.1	ポリシーの作成	134
8.16.2	事前定義されたルールのインポート	134
8.16.3	ルールの動作	135
8.17	配置 - 購読者 (フラット)-LDAP 形式	135
8.17.1	ポリシーの作成	135
8.17.2	事前定義されたルールのインポート	136
8.17.3	ルールの動作	137
8.18	配置 - 部署別発行者	137
8.18.1	ポリシーの作成	137
8.18.2	事前定義されたルールのインポート	138
8.18.3	ルールの動作	139
8.19	配置 - 部署別購読者 -LDAP 形式	139
8.19.1	ポリシーの作成	139
8.19.2	事前定義されたルールのインポート	140
8.19.3	ルールの動作	141
9	ポリシーシミュレータを使用したポリシーのテスト	143
9.1	ポリシーシミュレータへのアクセス	143
9.1.1	[Outline (アウトライン)] ビュー	143
9.1.2	[Policy Flow (ポリシーフロー)] ビュー	144
9.1.3	エディタ	144
9.2	ポリシーシミュレータの使用	145
9.3	Java 拡張を使用したポリシーのシミュレート	149
10	リソースオブジェクトに情報を保存する	151
10.1	汎用リソースオブジェクト	151
10.1.1	リソースオブジェクトの作成	151
10.1.2	汎用リソースオブジェクトの使用	153
10.2	マッピングテーブルオブジェクト	153
10.2.1	マッピングテーブルオブジェクトの作成	154
10.2.2	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	155
10.2.3	マッピングテーブルオブジェクトの編集	156
10.2.4	マッピングテーブルオブジェクトのテスト	157
10.3	ECMAScript オブジェクト	157
10.4	アプリケーションオブジェクト	158
10.5	リポジトリオブジェクト	158
10.6	ライブラリオブジェクト	158
10.6.1	ライブラリオブジェクトの作成	158
10.6.2	ライブラリオブジェクトへのポリシーの追加	159
10.6.3	ライブラリオブジェクト内でのポリシーでの使用	160
11	ポリシーでの ECMAScript の使用	161
11.1	ECMAScript オブジェクトの作成	161
11.2	ECMAScript エディタの使用	162
11.2.1	メインスクリプト記述エリア	163
11.2.2	式ビルダ	165
11.2.3	関数と変数	167
11.2.4	エラー表示	168
11.2.5	シェルエリア	169
11.3	ポリシーを備えた ECMAScripts の例	170
11.3.1	ECMAScript 関数を呼び出す DirXML スクリプトポリシー	170
11.3.2	ECMAScript 関数をドライバレベルで呼び出す XSLT ポリシー	171

11.3.3	ECMAScript 関数をスタイルシート内で呼び出す XSLT ポリシー	172
11.4	JavaScript ファイル初期設定の変更	172
11.4.1	JavaScript ファイルの初期設定	173
12	条件	177
	If 関連付け	178
	フィールド 178	
	例 179	
	If 属性	180
	フィールド 180	
	例 181	
	If クラス名	183
	フィールド 183	
	例 184	
	If ターゲット属性	186
	フィールド 186	
	例 187	
	If ターゲット DN	189
	フィールド 189	
	例 190	
	If エンタイトルメント	191
	フィールド 191	
	例 193	
	If グローバル構成値	194
	注 194	
	フィールド 194	
	例 195	
	If ローカル変数	196
	注 196	
	フィールド 196	
	例 197	
	If 名前付きパスワード	199
	フィールド 199	
	例 199	
	If 操作属性	200
	フィールド 200	
	例 202	
	If 操作プロパティ	204
	フィールド 204	
	例 205	
	If 操作	206
	フィールド 206	
	例 208	
	If パスワード	209
	フィールド 209	
	例 210	
	If ソース属性	212

フィールド 212	
例 213	
If ソース DN	214
フィールド 214	
例 214	
If XML 属性	216
フィールド 216	
例 217	
If XPath 式	218
フィールド 218	
備考 218	
例 218	
変数の拡張	220
注 220	
13 [アクション]	221
関連付けの追加	223
フィールド 223	
例 223	
ターゲット属性値の追加	224
フィールド 224	
例 224	
ターゲットオブジェクトの追加	226
フィールド 226	
備考 226	
例 226	
ソース属性値の追加	228
フィールド 228	
例 228	
ソースオブジェクトの追加	229
フィールド 229	
例 229	
XML 要素の追加	230
フィールド 230	
備考 230	
例 230	
XML テキストの追加	232
フィールド 232	
備考 232	
例 232	
中断	234
例 234	
ターゲット属性値のクリア	235
フィールド 235	
例 235	
操作プロパティのクリア	236
フィールド 236	

例 236	
ソース属性値のクリア	237
フィールド 237	
例 237	
SSO 資格情報のクリア	238
フィールド 238	
例 238	
XPath 式によるクローン	239
フィールド 239	
備考 239	
例 239	
操作属性のクローン	240
フィールド 240	
例 240	
ターゲットオブジェクトの削除	241
フィールド 241	
例 241	
ソースオブジェクトの削除	242
フィールド 242	
例 242	
一致オブジェクトの検索	243
フィールド 243	
備考 243	
例 243	
毎	246
フィールド 246	
備考 246	
例 246	
イベントの生成	247
フィールド 247	
備考 248	
例 248	
if	250
フィールド 250	
例 250	
エンタイトルメントの実装	252
フィールド 252	
例 252	
ターゲットオブジェクトの移動	253
フィールド 253	
例 253	
ソースオブジェクトの移動	255
フィールド 255	
例 255	
操作属性値の再フォーマット	256
フィールド 256	
例 256	
関連付けの削除	257

フィールド 257	
例 257	
ターゲット属性値の削除	258
フィールド 258	
例 258	
ソース属性値の削除	259
フィールド 259	
例 259	
ターゲットオブジェクトの名前変更	260
フィールド 260	
例 260	
操作属性の名前変更	261
フィールド 261	
例 261	
ソースオブジェクトの名前変更	262
フィールド 262	
例 262	
電子メールの送信	263
フィールド 263	
例 264	
テンプレートから電子メールを送信	265
フィールド 265	
例 266	
デフォルト属性値の設定	267
フィールド 267	
例 267	
ターゲット属性値の設定	269
フィールド 269	
例 269	
ターゲットパスワードの設定	271
フィールド 271	
例 271	
ローカル変数の設定	272
フィールド 272	
例 272	
操作関連付けの設定	274
フィールド 274	
例 274	
操作クラス名の設定	275
フィールド 275	
例 275	
操作ターゲット DN	276
フィールド 276	
例 276	
操作プロパティの設定	277
フィールド 277	
例 277	
操作ソース DN の設定	278

フィールド 278	
例 278	
操作テンプレート DN の設定	279
フィールド 279	
例 279	
ソース属性値の設定	280
フィールド 280	
例 280	
ソースパスワードの設定	282
フィールド 282	
例 282	
SSO 資格情報の設定	283
フィールド 283	
例 283	
SSO パスフレーズの設定	284
フィールド 284	
例 284	
XML 属性の設定	285
フィールド 285	
例 285	
ステータス	286
フィールド 286	
備考 286	
例 286	
ワークフローの開始	287
注 287	
フィールド 287	
例 288	
操作属性のストリップ	289
フィールド 289	
例 289	
XPath のストリップ	290
フィールド 290	
備考 290	
例 290	
メッセージのトレース	291
フィールド 291	
例 291	
拒否	293
例 293	
操作属性値がない場合は拒否	294
フィールド 294	
例 294	
While	295
フィールド 295	
例 295	
変数の拡張	296
注 296	

14 名詞トークン	297
追加されたエンタイトルメント	298
フィールド 298	
備考 298	
例 298	
関連付け	299
例 299	
属性	300
フィールド 300	
備考 300	
例 300	
文字	301
備考 301	
フィールド 301	
例 301	
クラス名	302
例 302	
ターゲット属性	303
フィールド 303	
備考 303	
例 303	
ターゲット DN	305
フィールド 305	
備考 305	
例 305	
ターゲット名	307
例 307	
ドキュメント	308
フィールド 308	
例 308	
エンタイトルメント	309
フィールド 309	
備考 309	
例 309	
パスワードの生成	310
フィールド 310	
例 310	
グローバル構成値	311
フィールド 311	
例 311	
ローカル変数	312
フィールド 312	
例 312	
名前付きパスワード	314
フィールド 314	
例 314	
説明	316

例 316	
操作属性	317
フィールド 317	
例 317	
操作プロパティ	319
フィールド 319	
例 319	
[パスワード]	320
例 320	
クエリ	321
フィールド 321	
例 321	
削除された属性	322
フィールド 322	
備考 322	
例 322	
削除されたエンタイトルメント	323
フィールド 323	
備考 323	
例 323	
解決	324
フィールド 324	
例 324	
ソース属性	325
フィールド 325	
備考 325	
例 325	
ソース DN	326
フィールド 326	
備考 326	
例 326	
ソース名	327
例 327	
[時刻]	328
フィールド 328	
例 328	
テキスト	329
フィールド 329	
例 329	
一意の名前	331
フィールド 331	
備考 332	
例 332	
一致しないソース DN	334
フィールド 334	
備考 334	
例 334	
XPath	335

フィールド	335
備考	335
例	335
変数の拡張	336
注	336
15 動詞トークン	337
Base64 デコード	338
フィールド	338
例	338
Base64 エンコード	339
フィールド	339
例	339
変換時間	340
フィールド	340
例	340
ターゲット DN のエスケープ	342
例	342
ソース DN のエスケープ	343
例	343
参加	344
フィールド	344
例	344
小文字	345
例	345
マップ	346
備考	346
フィールド	346
例	346
DN の解析	347
フィールド	347
備考	347
例	348
すべて置換	350
フィールド	350
備考	350
例	350
最初を置換	351
フィールド	351
備考	351
例	351
分割	353
フィールド	353
例	353
部分文字列	354
フィールド	354
例	354

大文字	356
例 356	
XML 解析	357
例 357	
XML シリアライズ	358
例 358	
変数の拡張	359
注 359	

16 プレ Identity Manager 3.5 ビルダ 361

16.1 アクションビルダ	361
16.1.1 アクションの作成	361
16.1.2 アクションビルダの追加オプション	361
16.2 アクションビルダ	362
16.3 引数ビルダ	363
16.3.1 引数ビルダの起動	364
16.3.2 引数ビルダの例	365
16.4 アクションの引数コンポーネントビルダ	367
16.5 条件ビルダ	367
16.5.1 条件の作成	368
16.5.2 条件ビルダの追加オプション	368
16.6 条件の引数コンポーネントビルダ	369
16.7 一致属性ビルダ	369
16.8 名前付き文字列ビルダ	371
16.9 パターン文字列ビルダ	372
16.10 引数値リストビルダ	373
16.11 ネームスペースエディタ	374
16.11.1 ネームスペースを使用した Java クラスへのアクセス	375

17 プレ Identity Manager 3.5 の条件 377

If 関連付け	378
フィールド 378	
If 属性	379
フィールド 379	
If クラス名	381
フィールド 381	
If ターゲット属性	383
フィールド 383	
If ターゲット DN	385
フィールド 385	
If エンタイトルメント	386
フィールド 386	
If グローバル構成値	388
注 388	
フィールド 388	
If ローカル変数	390
注 390	
フィールド 390	
If 名前付きパスワード	392

フィールド 392	
If 操作属性	393
フィールド 393	
f 操作プロパティ	395
フィールド 395	
If 操作	397
フィールド 397	
If パスワード	399
フィールド 399	
If ソース属性	400
フィールド 400	
If ソース DN	402
フィールド 402	
If XPath 式	403
フィールド 403	
備考 403	
18 プレ Identity Manager 3.5 のアクション	405
関連付けの追加	407
フィールド 407	
ターゲット属性値の追加	408
フィールド 408	
ターゲットオブジェクトの追加	409
フィールド 409	
備考 409	
ソース属性値の追加	410
フィールド 410	
ソースオブジェクトの追加	411
フィールド 411	
XML 要素の追加	412
フィールド 412	
備考 412	
XML テキストの追加	413
フィールド 413	
備考 413	
中断	414
フィールド 414	
ターゲット属性値のクリア	415
フィールド 415	
操作プロパティのクリア	416
フィールド 416	
ソース属性値のクリア	417
フィールド 417	
SSO 資格情報のクリア	418
フィールド 418	
XPath 式によるクローン	419
フィールド 419	
備考 419	

操作属性のクローン	420
フィールド 420	
ターゲットオブジェクトの削除	421
フィールド 421	
ソースオブジェクトの削除	422
フィールド 422	
一致オブジェクトの検索	423
フィールド 423	
備考 423	
毎	424
フィールド 424	
備考 424	
イベントの生成	425
フィールド 425	
備考 426	
エンタイトルメントの実装	427
フィールド 427	
ターゲットオブジェクトの移動	428
フィールド 428	
ソースオブジェクトの移動	429
フィールド 429	
操作属性の再フォーマット値	430
フィールド 430	
関連付けの削除	431
フィールド 431	
ターゲット属性値の削除	432
フィールド 432	
ソース属性値の削除	433
フィールド 433	
ターゲットオブジェクトの名前変更	434
フィールド 434	
操作属性の名前変更	435
フィールド 435	
ソースオブジェクトの名前変更	436
フィールド 436	
電子メールの送信	437
フィールド 437	
テンプレートから電子メールを送信	438
フィールド 438	
デフォルト属性値の設定	439
フィールド 439	
ターゲット属性値の設定	440
フィールド 440	
ターゲットパスワードの設定	441
フィールド 441	
ローカル変数の設定	442
フィールド 442	
操作関連付けの設定	443

フィールド 443	
操作クラス名の設定	444
フィールド 444	
操作ターゲット DN	445
フィールド 445	
操作プロパティの設定	446
フィールド 446	
操作ソース DN の設定	447
フィールド 447	
操作テンプレート DN の設定	448
フィールド 448	
ソース属性値の設定	449
フィールド 449	
ソースパスワードの設定	450
フィールド 450	
SSO 資格情報の設定	451
フィールド 451	
SSO パスフレーズの設定	452
フィールド 452	
XML 属性の設定	453
フィールド 453	
備考 453	
ステータス	454
フィールド 454	
備考 454	
操作属性のストリップ	455
フィールド 455	
XPath のストリップ	456
フィールド 456	
備考 456	
メッセージのトレース	457
フィールド 457	
拒否	458
フィールド 458	
操作属性値がない場合は拒否	459
フィールド 459	
19 プレ Identity Manager 3.5 名詞トークン	461
追加されたエンタイトルメント	462
フィールド 462	
備考 462	
関連付け	463
フィールド 463	
属性	464
フィールド 464	
備考 464	
クラス名	465
フィールド 465	

ターゲット属性	466
フィールド 466	
備考 466	
ターゲット DN	467
フィールド 467	
備考 467	
ターゲット名	468
フィールド 468	
エンタイトルメント	469
フィールド 469	
備考 469	
グローバル構成値	470
フィールド 470	
ローカル変数	471
フィールド 471	
名前付きパスワード	472
フィールド 472	
説明	473
フィールド 473	
操作属性	474
フィールド 474	
操作プロパティ	475
フィールド 475	
[パスワード]	476
フィールド 476	
削除された属性	477
フィールド 477	
備考 477	
削除されたエンタイトルメント	478
フィールド 478	
備考 478	
ソース属性	479
フィールド 479	
備考 479	
ソース DN	480
フィールド 480	
備考 480	
ソース名	481
フィールド 481	
テキスト	482
フィールド 482	
一意の名前	483
フィールド 483	
備考 483	
一致しないソース DN	485
フィールド 485	
備考 485	
XPath	486

フィールド 486	
備考 486	
20 プレ Identity Manager 3.5 動詞トークン	487
ターゲット DN のエスケープ	488
フィールド 488	
ソース DN のエスケープ	489
フィールド 489	
小文字	490
フィールド 490	
DN の解析	491
フィールド 491	
備考 491	
すべて置換	493
フィールド 493	
備考 493	
最初を置換	494
フィールド 494	
備考 494	
部分文字列	495
フィールド 495	
大文字	496
フィールド 496	

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、Designer for Identity Manager 3.5.1 のポリシービルダおよびドライバ環境設定について詳しく説明します。

- ◆ 21 ページの第 1 章「概要」
- ◆ 25 ページの第 3 章「ポリシービルダを使用したポリシーの管理」
- ◆ 51 ページの第 4 章「追加ビルダの使用」
- ◆ 69 ページの第 5 章「XPath ビルダの使用」
- ◆ 75 ページの第 6 章「スキーママッピングポリシーの定義」
- ◆ 89 ページの第 7 章「フィルタを使用したオブジェクトのフローの制御」
- ◆ 105 ページの第 8 章「事前定義されたルールの使用」
- ◆ 143 ページの第 9 章「ポリシーシミュレータを使用したポリシーのテスト」
- ◆ 151 ページの第 10 章「リソースオブジェクトに情報を保存する」
- ◆ 161 ページの第 11 章「ポリシーでの ECMAScript の使用」
- ◆ 177 ページの第 12 章「条件」
- ◆ 221 ページの第 13 章「[アクション]」
- ◆ 297 ページの第 14 章「名詞トークン」
- ◆ 337 ページの第 15 章「動詞トークン」

プレ Identity Manager 3.5 ポリシービルダ用のリファレンス章もあります。

- ◆ 23 ページの第 2 章「プレ Identity Manager 3.5 ポリシービルダ」
- ◆ 361 ページの第 16 章「プレ Identity Manager 3.5 ビルダ」
- ◆ 377 ページの第 17 章「プレ Identity Manager 3.5 の条件」
- ◆ 405 ページの第 18 章「プレ Identity Manager 3.5 のアクション」
- ◆ 461 ページの第 19 章「プレ Identity Manager 3.5 名詞トークン」
- ◆ 487 ページの第 20 章「プレ Identity Manager 3.5 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

最新バージョンの *Designer* のポリシーについては、[Identity Manager](http://www.novell.com/documentation/idm35) のマニュアルの [Web サイト](http://www.novell.com/documentation/idm35) (<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 ドライバの使用に関するマニュアルについては、[Identity Manager Driver](http://www.novell.com/documentation/idm35drivers/index.html) のマニュアルの [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。

Designer の使用に関するマニュアルについては、[Designer 2.1 for Identity Manager 3.5.1](http://www.novell.com/documentation/designer21/) のマニュアルの [Web サイト](http://www.novell.com/documentation/designer21/) (<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。

概要

1

ポリシーは、識別ボールドとリモートデータストア間で同期してい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。「[Identity Manager 3.5.1 ポリシーの理解](#)」を参照してください。Designer は、1つのシステムから別のシステムへの情報のフロー方法と条件を制御するポリシーを定義およびデバッグするためのさまざまなツールセットを提供します。以下の節では、ポリシーの管理を支援するために提供されているツールを使用する方法を説明します。

- ◆ 25 ページの第 3 章「ポリシービルダを使用したポリシーの管理」
- ◆ 51 ページの第 4 章「追加ビルダの使用」
- ◆ 69 ページの第 5 章「XPath ビルダの使用」
- ◆ 75 ページの第 6 章「スキーママッピングポリシーの定義」
- ◆ 89 ページの第 7 章「フィルタを使用したオブジェクトのフローの制御」
- ◆ 105 ページの第 8 章「事前定義されたルールの使用」
- ◆ 143 ページの第 9 章「ポリシーシミュレータを使用したポリシーのテスト」
- ◆ 151 ページの第 10 章「リソースオブジェクトに情報を保存する」
- ◆ 161 ページの第 11 章「ポリシーでの ECMAScript の使用」

この節では、DirXML[®] Script 内のすべての要素への詳細リファレンスセクションも含まれています。DirXML Script の詳細については、『[Identity Manager 3.5.1 ポリシーの理解](#)』の「[DirXML スクリプト](#)」を参照してください。

- ◆ 177 ページの第 12 章「条件」
- ◆ 221 ページの第 13 章「[アクション]」
- ◆ 297 ページの第 14 章「名詞トークン」
- ◆ 337 ページの第 15 章「動詞トークン」

Identity Manager ポリシービルダ用の追加のリファレンス章もあります。

- ◆ 23 ページの第 2 章「プレ Identity Manager 3.5 ポリシービルダ」
- ◆ 361 ページの第 16 章「プレ Identity Manager 3.5 ビルダ」
- ◆ 377 ページの第 17 章「プレ Identity Manager 3.5 の条件」
- ◆ 405 ページの第 18 章「プレ Identity Manager 3.5 のアクション」
- ◆ 461 ページの第 19 章「プレ Identity Manager 3.5 名詞トークン」
- ◆ 487 ページの第 20 章「プレ Identity Manager 3.5 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 (221 ページの第 13 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 (177 ページの第 12 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 (297 ページの第 14 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 (337 ページの第 15 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 (、『*Identity Manager 3.5.1 ポリシーの理解*』の「“正規表現”」を参照) および XPath 1.0 式 (、『*Identity Manager 3.5.1 ポリシーの理解*』の「“XPath 1.0 式”」を参照) は、一般的にポリシーに対する目的の結果を作成するためにルールで使用されます。
- ◆ ポリシーとは XDS ドキュメント上で操作を実行するもので、その主な目的はドキュメントを調べて変更を加えることです。
- ◆ 操作とは XDS ドキュメント内の要素のことで、入力要素と出力要素の子になります。要素は、Novell® nds.dtd の一部です。詳細については、『*Identity Manager 3.5.1 DTD リファレンス*』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。

プレ Identity Manager 3.5 ポリシービルダ

2

Designer には、プレ Identity Manager 3.5 ポリシービルダと Identity Manager 3.5 以降ポリシービルダの 2 つのポリシービルダがあります。ポリシービルダは、以下の点を除いて同様です。

- ◆ プレ Identity Manager 3.5 ポリシービルダでは、トレースをドライバレベルでのみ有効または無効にすることができます。
- ◆ DirXML スクリプト要素は、2 つのビルダで異なります。

これらの差異により、2 つのポリシービルダが必要です。両方のポリシービルダの使用方法については、[25 ページの第 3 章「ポリシービルダを使用したポリシーの管理」](#)を参照してください。ここでは、Identity Manager 3.5 以降ポリシービルダについて説明しています。唯一の異なる点は、ルール、アクション、条件、およびトークンのトレースを有効  および無効  にする追加アイコンです。

プレ Identity Manager 3.5 ポリシービルダ用の DirXML スクリプト要素のリスト：

- ◆ [377 ページの第 17 章「プレ Identity Manager 3.5 の条件」](#)
- ◆ [405 ページの第 18 章「プレ Identity Manager 3.5 のアクション」](#)
- ◆ [461 ページの第 19 章「プレ Identity Manager 3.5 名詞トークン」](#)
- ◆ [487 ページの第 20 章「プレ Identity Manager 3.5 動詞トークン」](#)

Identity Manager 3.5 以降ポリシービルダ用の DirXML スクリプト要素：

- ◆ [177 ページの第 12 章「条件」](#)
- ◆ [221 ページの第 13 章「\[アクション\]」](#)
- ◆ [297 ページの第 14 章「名詞トークン」](#)
- ◆ [337 ページの第 15 章「動詞トークン」](#)

ポリシービルダを使用したポリシーの管理

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。

- ◆ 25 ページのセクション 3.1 「ポリシービルダへのアクセス」
- ◆ 29 ページのセクション 3.2 「Policy Builder の使用」
- ◆ 30 ページのセクション 3.3 「ポリシーの作成」
- ◆ 37 ページのセクション 3.4 「ルールの作成」
- ◆ 44 ページのセクション 3.5 「引数の作成」
- ◆ 46 ページのセクション 3.6 「ポリシーの編集」
- ◆ 50 ページのセクション 3.7 「XML でポリシーを表示」

3.1 ポリシービルダへのアクセス

Designer には、プレ Identity Manager 3.5 ポリシービルダと Identity Manager 3.5 以降ポリシービルダの 2 つのポリシービルダがあります。ポリシービルダのバージョンは、Identity Manager のバージョンによって決定します。Identity Manager のバージョンをテストする

- 1 Designer でプロジェクトを開きます。
- 2 [Outline (アウトライン)] タブをクリックし、[Show Model Outline (モデルアウトラインの表示)] アイコンを選択します。>
- 3 サーバオブジェクトを右クリックし、[プロパティ] をクリックします。
- 4 [Identity Manager のバージョン] を選択します。

オプションは次のとおりです。

- ◆ 2.0
- ◆ 2.0.1
- ◆ 2.0.2
- ◆ 3.0
- ◆ 3.0.1
- ◆ 3.5
- ◆ 3.5.1

Identity Manager のバージョンが 3.5 以上に設定されている場合は、新しいポリシービルダが利用できます。バージョンが 3.5 よりも下位の場合は、古いポリシービルダが利用できません。

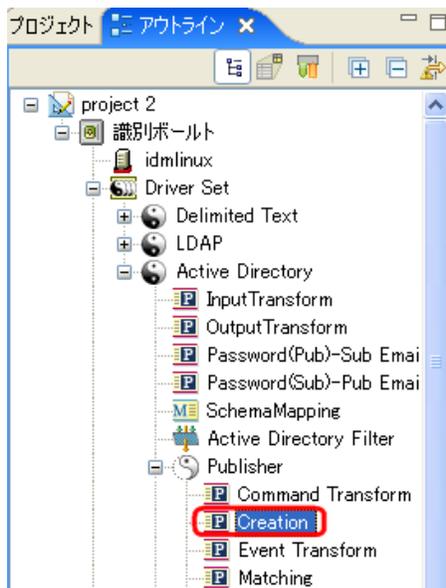
ポリシービルダは [モデルアウトライン] ビュー、[ポリシーフロー] ビュー、またはポリシーセットからアクセスできます。

- ◆ 26 ページのセクション 3.1.1 「[Model Outline (モデルアウトライン)] ビュー」

- ◆ 26 ページのセクション 3.1.2 「[Policy Flow (ポリシーフロー)] ビュー」
- ◆ 27 ページのセクション 3.1.3 「ポリシーセット」

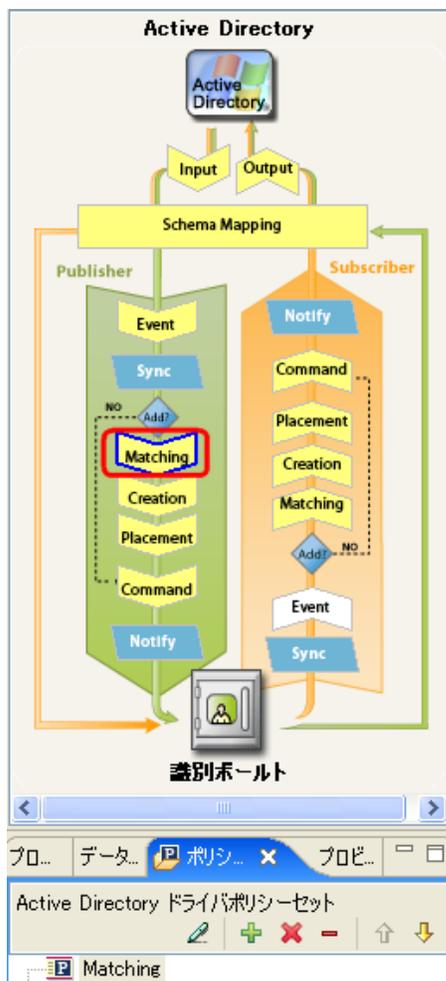
3.1.1 [Model Outline (モデルアウトライン)] ビュー

- 1 Designer でプロジェクトを開きます。
- 2 [アウトライン] タブをクリックし、[モデルアウトラインの表示] アイコンを選択します。
- 3 [モデルアウトライン] ビューにリストされているポリシーをダブルクリックするか、または右クリックして [編集] を選択します。



3.1.2 [Policy Flow (ポリシーフロー)] ビュー

- 1 Designer でプロジェクトを開きます。
- 2 [アウトライン] タブを選択し、[ポリシーフローの表示] アイコンを選択します。
- 3 [Policy Flow (ポリシーフロー)] ビューでポリシー (たとえば、一致ポリシー) を右クリックし、[Edit Policy (ポリシーの編集)] を選択します。
または
[ポリシーフロー] で一致ポリシーをダブルクリックします



4 ポリシーを選択してから、[編集] をクリックします。

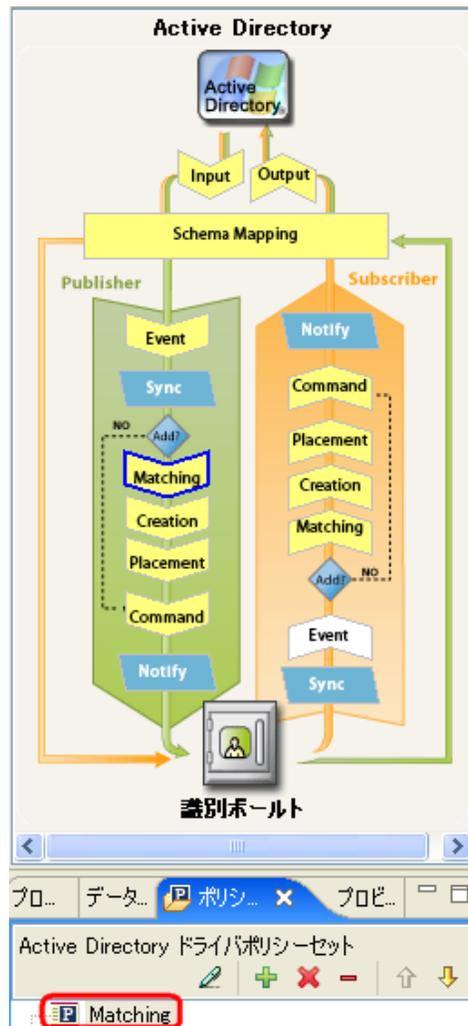


3.1.3 ポリシーセット

1 ポリシーセットでポリシーを右クリックし、[編集] をクリックします。

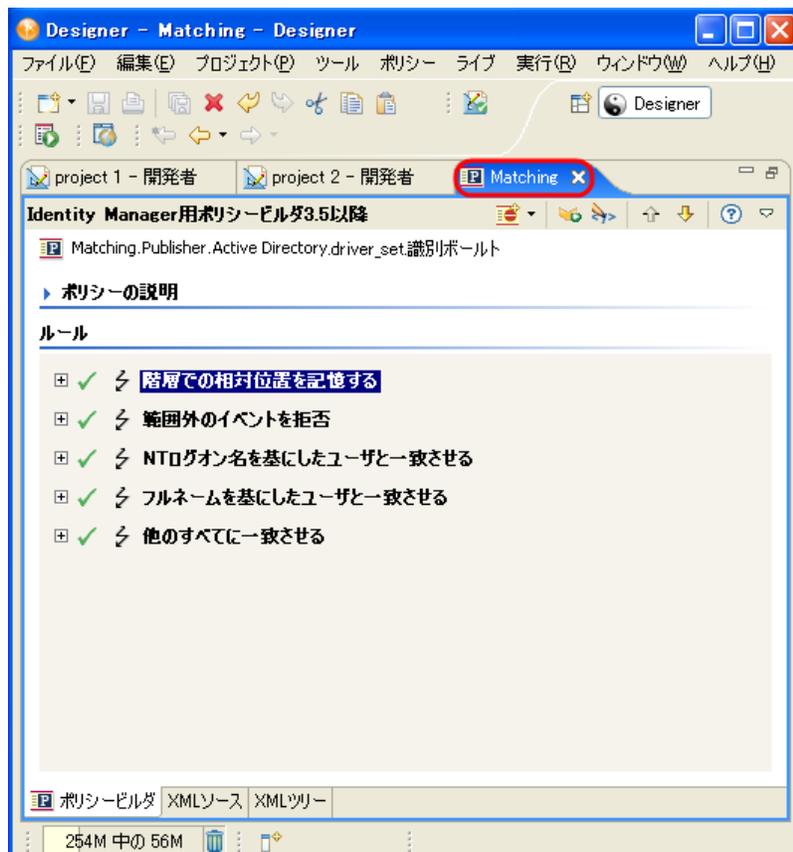
または

ポリシーセットでポリシーを選択して、[ポリシーを編集します] アイコンをクリックします。



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

図 3-1 ポリシービルダのフルスクリーン表示



ポリシービルダの使用の詳細については、29 ページのセクション 3.2 「Policy Builder の使用」を参照してください。

3.2 Policy Builder の使用

ポリシービルダでは、ポリシーを構成するルールを追加、表示、および削除できます。また、ポリシービルダを使用して、ポリシーおよびルールのインポートと保存、および XML ネームスペースの管理もできます。ポリシービルダは 51 ページの「アクションビルダ」および 56 ページの「条件ビルダ」を含んでいます。

3.2.1 ヒント

- ◆ [] アイコンをクリックして、ポリシー、ルール、条件、またはアクションを無効にします。[] アイコンをクリックして、再び有効にします。
- ◆ [トレースの無効化] アイコンをクリックして、ポリシー、ルール、アクション、条件、またはトークンに対するトレースを無効にします。[トレースの有効化] アイコンをクリックして、トレースを再び有効にします。
- ◆ [編集] アイコン をクリックして、ルールの名前またはルールの説明を編集します。
- ◆ [] アイコンをクリックして、フィールドの値のリストを表示します。
- ◆ [削除] アイコン をクリックして、ルールまたはポリシーを削除します。

- ◆ [ルール] アイコンをクリックして、新しいルールを追加します。右にある下向き矢印をクリックした場合、ルールを追加、事前定義ルールを使用、ポリシーを含める、または新しい条件グループを作成することができます。
- ◆ [ファイルからポリシーをインポート] アイコンをクリックして、ポリシーまたはルールをインポートします。[ファイルに保存] アイコンをクリックして、ポリシーまたはルールをエクスポートします。
- ◆ [ネームスペースの編集] アイコンをクリックして、複数の XML ネームスペースをルールまたはポリシーに追加します。
- ◆ [XPath ビルダの起動] アイコンをクリックして、XPath ビルダを起動します。これにより、XPath 式を作成できます。
- ◆ をクリックして、ルールすべてを展開するか、または をクリックしてルールすべてを縮小します。
- ◆ アイコンをクリックして、ルールを上へ移動します。アイコンをクリックして、ルールを下へ移動します。
- ◆ 作業をメインメニューから保存するには、[ファイル]> [保存] をクリックするか、または Ctrl+S を押します。
- ◆ ポリシーまたはルールにコメントを追加するには、[ポリシーの説明] フィールドに入力します。コメントは、ポリシーまたはルールに直接保存され、必要なだけ保持できます。

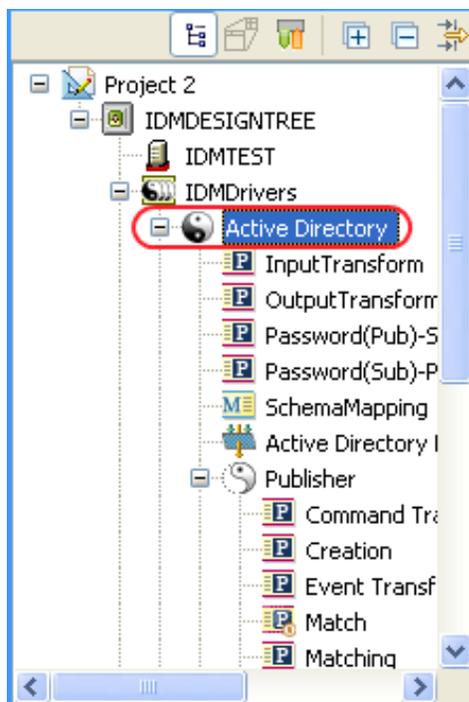
3.3 ポリシーの作成

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

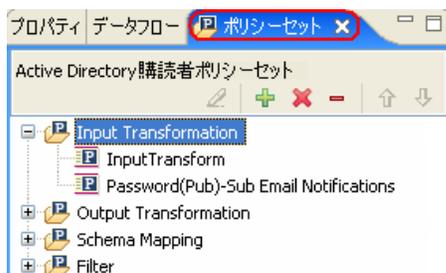
- ◆ 30 ページのセクション 3.3.1 「ポリシーセットへのアクセス」
- ◆ 31 ページのセクション 3.3.2 「ポリシーセットの使用」
- ◆ 33 ページのセクション 3.3.3 「Add Policy Wizard (ポリシー追加ウィザード) の使用」

3.3.1 ポリシーセットへのアクセス

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



2 [Policy Set (ポリシーセット)] タブを選択します。

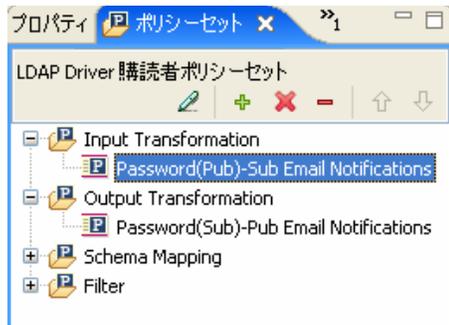


3.3.2 ポリシーセットの使用

ポリシーセットには、ツールバーおよびポリシーリストが含まれます。

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

図 3-2 ポリシーセット



ポリシーセットのツールバー

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

表 3-1 ポリシーセットのツールバー

説明	説明
ポリシーの編集 	ポリシービルダを起動します。
ポリシーセットに新しいポリシーを作成または追加します 	Add Policy Wizard (ポリシー追加ウィザード) を起動します。
選択したポリシーを除去し、削除します 	プロジェクトからポリシーを削除します。
選択したポリシーをポリシーセットから除外するが削除しない 	選択したポリシーセットオブジェクトからポリシーを除外しますが、そのポリシー自体は削除しません。
ポリシーチェーンのポリシーを上に移動します 	ポリシーの処理順序を上に移動します。
ポリシーチェーンのポリシーを下に移動します 	ポリシーの処理順序を下に移動します。

キーボード操作

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

表 3-2 キーボード操作

ショートカットキー	説明
< 上矢印 >	選択したポリシーの処理順序を上に移動します。
< 下矢印 >	選択したポリシーの処理順序を下に移動します。
削除	プロジェクトからポリシーを削除します。
< マイナス記号 >	選択したポリシーセットからポリシーを除外しますが、そのポリシー自体は削除されません。

ショートカットキー	説明
<プラス記号>	Add Policy Wizard (ポリシー追加ウィザード) を起動します。
<Ctrl>+<Z>	最後の操作を元に戻します。
<Ctrl>+<Y>	最後の操作をやり直します。

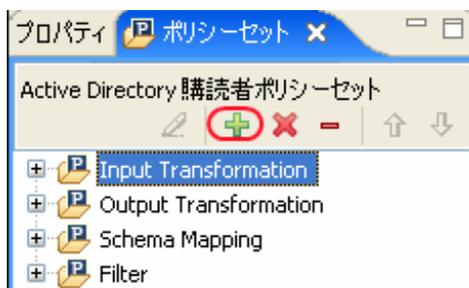
3.3.3 Add Policy Wizard (ポリシー追加ウィザード) の使用

ポリシーの追加ウィザードは、ツールバーの [ポリシーセットに新しいポリシーを作成または追加します] アイコンをクリックすると起動します。Add Policy Wizard (ポリシー追加ウィザード) で実行できる作業は、次のとおりです。

- ◆ 33 ページの「ポリシーの作成」
- ◆ 35 ページの「ポリシーのコピー」
- ◆ 36 ページの「ポリシーへのリンク」

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

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



ポリシーの作成

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

方法を選択します

ポリシーをポリシーセットに追加する方法を選択します。



- 2 ポリシー名を入力します。

作成ポリシー

新しいポリシーおよびポリシーを作成するコンテナの名前を指定します。



ポリシー名:

ポリシーコンテナ:

オブジェクトの作成後にエディタを開く(O)

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

ポリシーが複数のドライバで再利用されない場合は、通常、そのポリシーをドライバ、またはそれを使用しているチャンネルの下に作成します。

作成先は、ポリシーを整理する方法に応じて決定します。デフォルトでは、ポリシーは、Add Policy Wizard (ポリシー追加ウィザード) が起動されたときに [Outline (アウトライン)] タブで選択されていたコンテナオブジェクトに配置されます。

たとえば、[Outline (アウトライン)] タブで発行者オブジェクトに移動し、その後でポリシーをポリシーセットに追加する場合、ポリシーはデフォルトで発行者コンテナに配置されます。

別のコンテナにポリシーを作成する場合は、この設定を変更できます。たとえば、ポリシーライブラリを設定し、共通ポリシーをすべてこのドライバの下に配置して、他のドライバからポリシーを簡単に参照できるようにできます。これでポリシーを共有できます。ポリシーの変更が必要な場合は、1回だけ変更すれば済みます。

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

タイプの選択

このポリシーを実装する方法を選択します。



DirXMLスクリプト
 XSLT

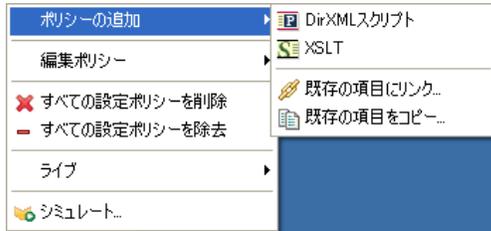
- 5 [完了] をクリックします。

スキーママッピングポリシーセットを作成する場合は、[スキーママッピング] で追加オプションが利用できます。拡張されたポリシーセットに新しいポリシーが表示されます。

[ポリシーフロー] ビュー内でポリシーセットを右クリックすることによってポリシーを追加することもできます。

- 1 ポリシーセット (たとえば、入力変換セット) を右クリックします。
- 2 [Add Policy (ポリシーの追加)] を選択します。
- 3 ポリシーの実装方法を選択します。
 - ◆ DirXML スクリプト
 - ◆ XSLT
 - ◆ 既存の項目にリンク

- ◆ 既存の項目をコピー
- ◆ スキーママッピング(スキーママッピングポリシーセットが選択されている場合にのみ表示されます。)



4 ポリシーに名前を付けます。

ポリシー名の設定

新しいポリシーの名前を入力します。



名前(N):

オブジェクトの作成後にエディタを開く(O)

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

ポリシーのコピー

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

方法を選択します

ポリシーをポリシーセットに追加する方法を選択します。



新しいポリシーを作成

ポリシーをコピー

ポリシーをリンク

- 2 ポリシーに名前を付けます。
- 3 デフォルトコンテナをそのまま使用するか、ポリシーの作成先にするドライバオブジェクト、発行者オブジェクト、または購読者オブジェクトを参照し、選択します。
- 4 コピーするポリシーを参照して選択し、[OK] をクリックします。

コピーポリシー

新しいポリシーの名前、ポリシーを作成するコンテナ、およびコピーするポリシーを指定します。



ポリシー名:

ポリシーコンテナ:

コピーするポリシー:

オブジェクトの作成後にエディタを開く(O)

- 5 [終了] をクリックして、選択したポリシーをコピーします。

ポリシーへのリンク

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

方法を選択します

ポリシーをポリシーセットに追加する方法を選択します。



新しいポリシーを作成

ポリシーをコピー

ポリシーをリンク

- 2 [参照] をクリックし、モデルブラウザを起動します。

リンクポリシー

ポリシーセットにリンクする既存のポリシーを指定します。



リンクするポリシー:

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

オブジェクトを選択:

- [-] 識別レポート
 - [-] driver_set
 - [-] Delimited Text
 - [-] LDAP
 - [-] Active Directory
 - [-] InputTransform
 - [-] Password(Pub)-Sub Email Notifications
 - [-] OutputTransform
 - [-] Password(Sub)-Pub Email Notifications
 - [-] SchemaMapping
 - [-] Publisher
 - [-] Subscriber

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

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

- 4 [終了] をクリックして、選択したポリシーにリンクします。

3.4 ルールの作成

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

ルールは、4つの方法で作成できます。

- ◆ 37 ページのセクション 3.4.1 「新しいルールの作成」
- ◆ 42 ページのセクション 3.4.2 「事前定義されたルールの使用」
- ◆ 42 ページのセクション 3.4.3 「既存のルールの包含」
- ◆ 43 ページのセクション 3.4.4 「XML ファイルからのポリシーのインポート」

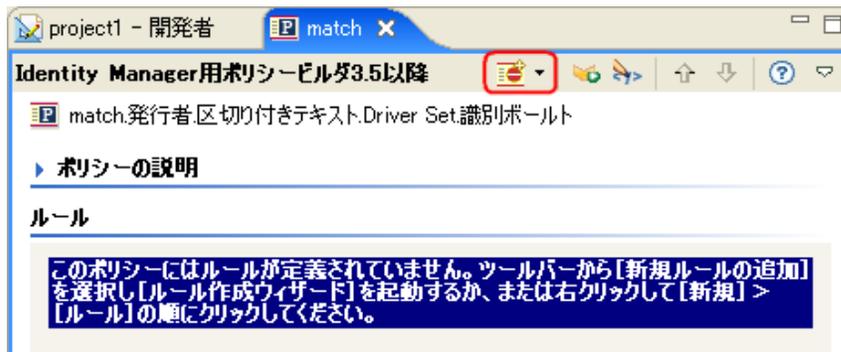
3.4.1 新しいルールの作成

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

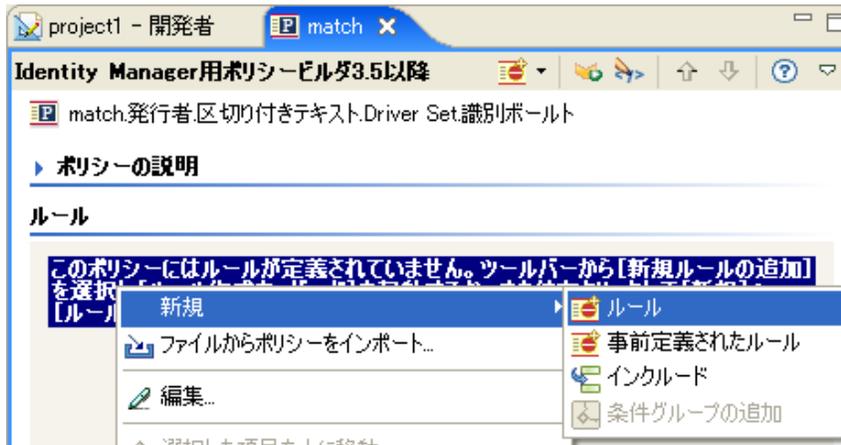
- ◆ 37 ページの 「ルールの作成」
- ◆ 40 ページの 「条件グループの作成」
- ◆ 41 ページの 「条件の作成」
- ◆ 41 ページの 「アクションの作成」

ルールの作成

- 1 ポリシービルダのツールバーから、[Rule (ルール)] を選択します。



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



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

- 2 ルールの名前を指定して、[次へ] をクリックします。

名前および説明ルール

ルールと説明はルールビルダエディタのルール上に表示されます。
いずれもルールビルダ内でルール名をダブルクリックすることで、編集できます。

名前

説明

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

条件構造の選択

条件構造は条件グループのロジックを定義します。

条件構造

OR条件、ANDグループ

AND条件、ORグループ

- 4 必要な条件を選択し、適切な情報を指定し、[次へ] をクリックします。

条件の定義

条件の構文を完成させる値を選択します。*付きの値は有効である必要があります。最初の条件は新しい条件グループに自動的に挿入されます。



グループ 1 の条件 1

条件

名前 *

オペレータ *

作成できる個々の条件については、[ヘルプ] アイコン をクリックします。

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

条件の定義を続行しますか?

条件の定義を続行するか、またはルールアクション定義に進むかを選択します。



エントリを選択してください。

続行(ルールアクションを定義)

同じ条件グループで別の条件を定義

新規条件グループで新しい条件を定義

- 6 必要なアクションを選択し、[次へ] をクリックします。

アクションの定義

アクションの構文に必要な値を選択します。*のついた値は必須です。



アクション 1

実行内容

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

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

アクションの定義を続行しますか?

新しいアクションを定義するか、[完了] をクリックしてルールを完成します。



エントリを選択してください。

続行(概要ページへ移動)

別のアクションを定義

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

概要

次は、作成される新規ルールの概要です。



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



条件グループの作成

- 1 [条件] タブを右クリックするか、または [条件グループ] の名前を右クリックして、[新規作成] > [条件グループを前に挿入] または [条件グループを後に挿入] をクリックします。

ルール

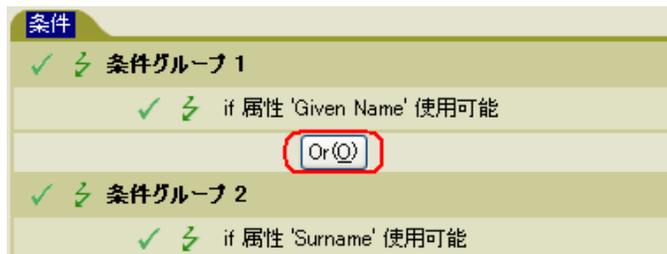


[条件グループ] の名前を右クリックして、[新規作成] > [条件グループを前に挿入] または [条件グループを後に挿入] をクリックすることもできます。

ルール



条件グループの条件は、[および/または] アイコンをクリックすることによって変更できます。

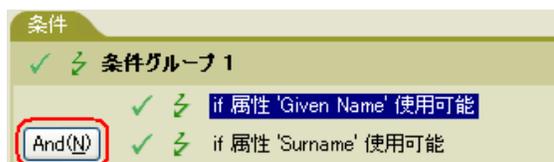


条件の作成

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



条件は、[および/または] アイコンをクリックすることによって変更できます。



アクションの作成

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

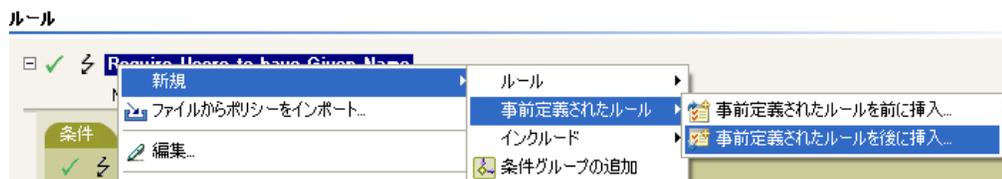


3.4.2 事前定義されたルールの使用

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

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

詳細については、105 ページの第 8 章「事前定義されたルールの使用」を参照してください。



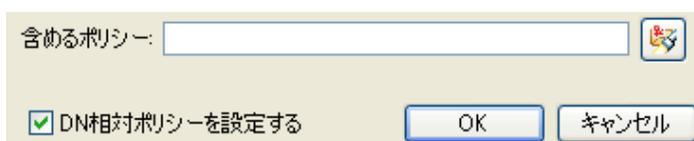
3.4.3 既存のルールの包含

Designer では、別のポリシーのルールを含めることができます。

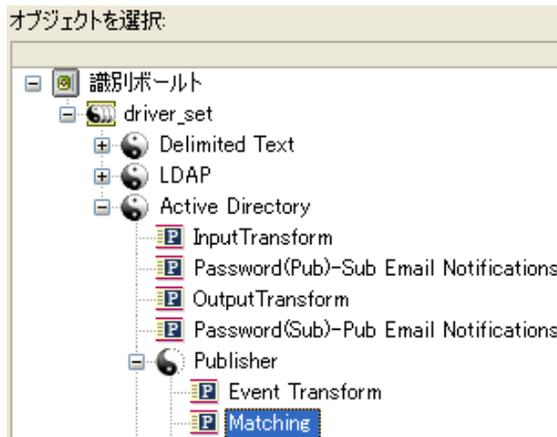
- 1 ポリシービルダ内を右クリックし、[New (新規作成)] > [Include (包含)] > [Insert Include Before (包含を前に挿入)] または [Insert Include After (包含を後に挿入)] の順にクリックします。



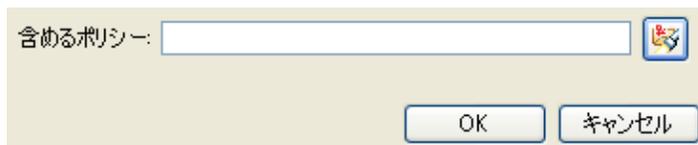
- 2 参照アイコンをクリックします。



- 3 含めるポリシーを参照して選択し、[OK] をクリックします。



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



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

ルール

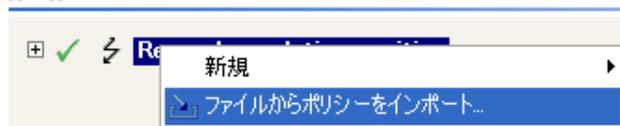


3.4.4 XML ファイルからのポリシーのインポート

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

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

ルール



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



- 3 参照アイコンをクリックして、ポリシーを含むファイルを選択し、[開く] をクリックします。
- 4 [OK] をクリックします。

3.5 引数の作成

引数ビルダでは、動的なグラフィカルインタフェースによって、ポリシービルダで使用する複雑な引数の式を作成できます。引数ビルダにアクセスするには、53 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。

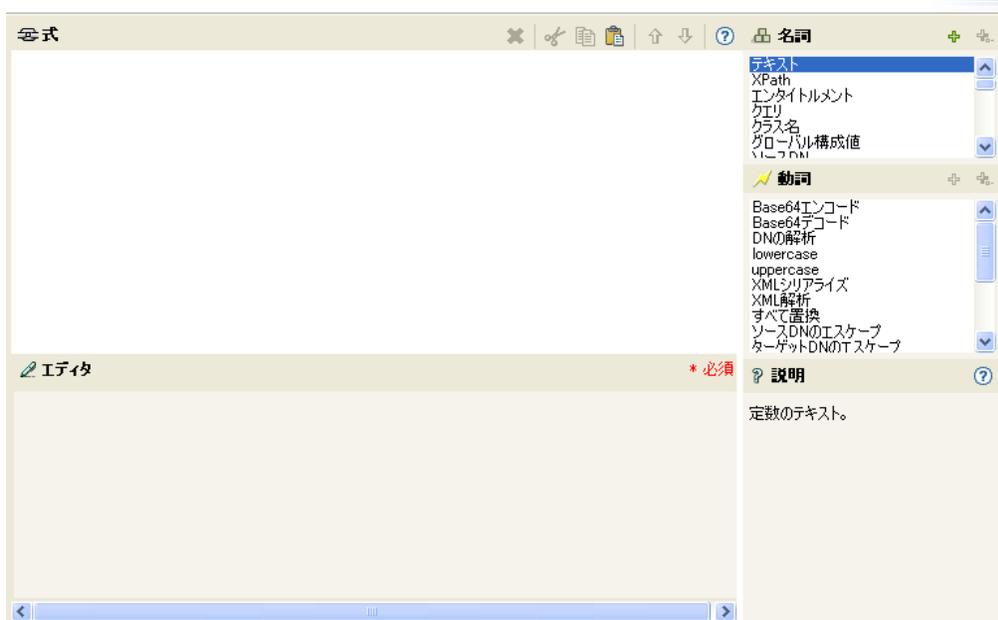
トークンは、名詞と動詞の 2 つに分類できます。名詞トークンは、現在の操作、ソースやターゲットのデータストア、または外部ソースなどから派生する値を展開します。動詞トークンは、そのトークンのサブオーディネイトにある他のトークンの結果を変更します。

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。

図 3-3 引数ビルダ

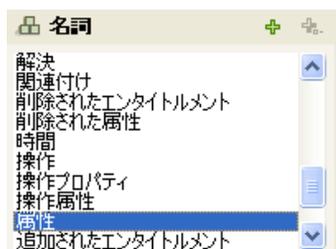
引数の作成および編集

引数を構成するために式領域にコンポーネントを追加または式領域からコンポーネントを削除します。エディタでコンポーネント値を指定します。

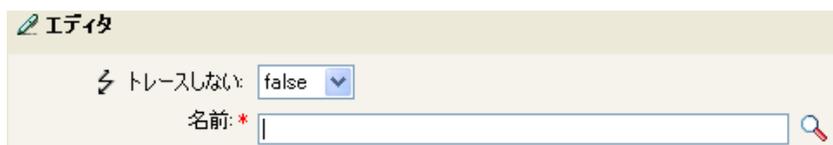


たとえば、引数を属性値に設定する場合は、属性の名詞を選択してから、属性名を選択します。

- 1 名詞トークンのリストから [属性] をダブルクリックして、選択します。

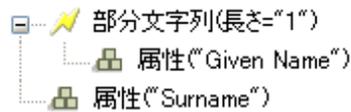


- 2 [エディタ] フィールドで属性名を参照して選択します。



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。式は、名字属性全体と組み合わせた名前属性に対して部分文字列の長さ 1 を表示します。

公式



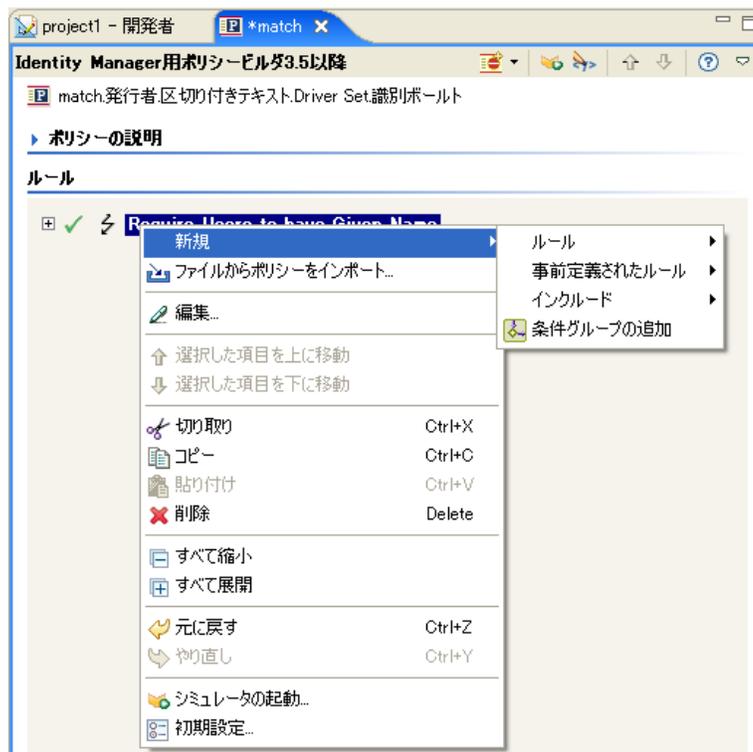
名詞または動詞を追加した後は、エディタを使用して値を入力してから、次の名詞または動詞を追加できます。変更を適用するために [式] ペインを更新する必要はなく、次の操作を実行すれば変更は反映されます。

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、[297 ページの「名詞トークン」](#) および [337 ページの「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、[53 ページの「引数ビルダ」](#) を参照してください。

3.6 ポリシーの編集

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

図 3-4 ポリシービルダのコンテキストメニューおよびツールバー



3.6.1 ポリシービルダのアクションおよびメニュー項目

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

表 3-3 ポリシービルダのアクションおよびメニュー項目

説明	説明
すべて縮小 	展開されているルールをすべて縮小します。
コピー 	選択された項目をクリップボードにコピーします。
Copy and drop (コピーしてドロップ)	項目を選択し、キーを押しながらドラッグします。
切り取り 	選択された項目を切り取り、クリップボードにコピーします。
削除 	選択された項目を削除します。
無効 	ルール、条件、またはアクションを無効として表示します。
トレースの無効化 	ルールに対するトレースを無効にします。
Drag and drop (ドラッグアンドドロップ)	項目を選択し、再配置できます。項目を選択し、それを新しい場所にドラッグします。
編集 	選択された項目を編集できます。ルールビルダを開くには、ルールを選択し、[編集] をクリックします。
有効 	ルール、条件、またはアクションを有効として表示します。
トレースの有効化 	ルールに対するトレースを有効にします。
すべて展開 	各ルールの条件およびアクションを表示できるように、ルールをすべて展開します。
ファイルからポリシーをインポート 	ファイルシステムからポリシーをインポートし、それをポリシーに追加するか、ポリシーのルールをすべて置き換えます。
シミュレータの起動 	ポリシーシミュレータを起動します。
Move and drop (移動してドロップ)	項目を選択し、移動できます。項目を選択して、ドラッグします。
選択した項目を下へ移動します 	項目をポリシーのリスト内で下に移動します。
選択した項目を上へ移動します 	項目をポリシーのリスト内で上に移動します。
[新規] > [条件グループの追加]	選択された項目の後に、新しい条件グループを作成します。
[新規] > [インクルード] > [インクルードを前に挿入] または [インクルードを後に挿入]	選択された項目の前または後に、新しいインクルードを作成します。
[新規] > [事前定義されたルール] > [事前定義されたルールを前に挿入] または [事前定義されたルールを後に挿入]	事前定義されたルールを選択した項目の前または後に挿入します。

説明	説明
[新規]> [ルール]> [ルールを前に挿入] または [ルールを後に挿入]	選択された項目の前または後に、新しいルールを作成します。
貼り付け 	選択された項目の後に、クリップボードの内容を貼り付けます。
初期設定 	情報の表示方法を変更できます。
やり直し 	直前のアクションをやり直します。
選択	項目をクリックして選択します。
元に戻す 	直前のアクションを元に戻します。

3.6.2 キーボード操作

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

表 3-4 ポリシービルダでのキーボード操作

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

3.6.3 ポリシーの名前変更

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

1. 一般

ポリシー名(P):

4 [OK] をクリックします。

3.6.4 自分の作業の保存

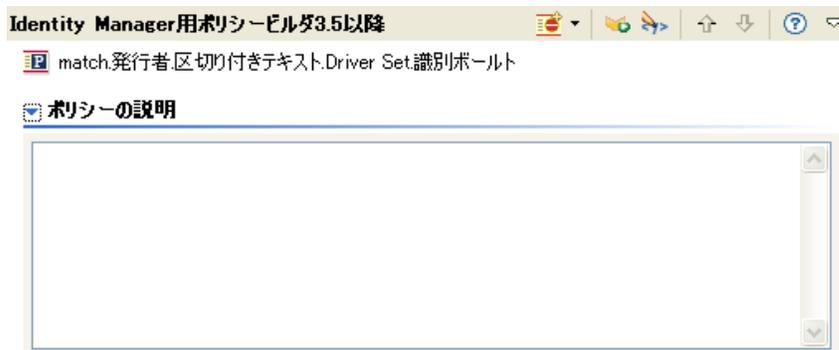
次のいずれかの操作を行います。

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

3.6.5 ポリシーの説明

[説明] フィールドは、ポリシーの機能についてのメモを追加する場所です。ポリシーの説明およびルールの説明を追加できます。

1 ポリシービルダで、[ポリシーの説明] をダブルクリックします。

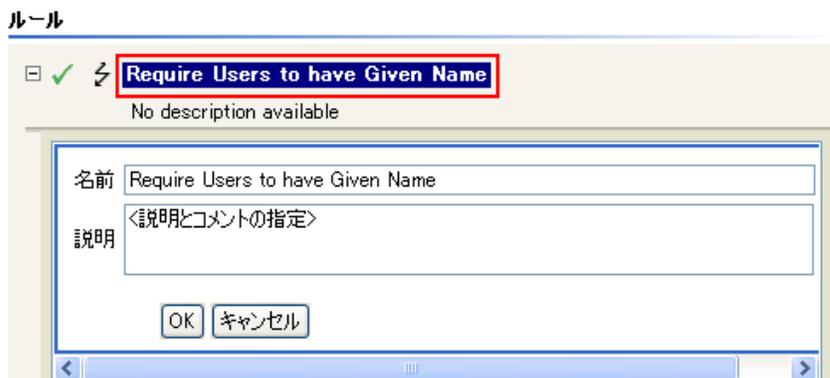


2 ポリシーの説明を入力します。

3 <Ctrl>+<S> キーを押してポリシーを保存します。

ルールに説明を追加する

1 ルールの名前をダブルクリックします。



- 2 [説明] フィールドに、ルールの説明を指定します。
- 3 <Ctrl>+<S> キーを押してルールを保存します。

3.7 XML でポリシーを表示

Designer では、XML エディタを使用して、XML を表示、編集、および検証できます。
[XML ソース] または [XML ツリー] タブをクリックして、XML エディタにアクセスします。XML エディタの詳細については、『*Designer 2.1 for Identity Manager 3.5.1*』の「**Novell XML エディタ**」を参照してください。

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。

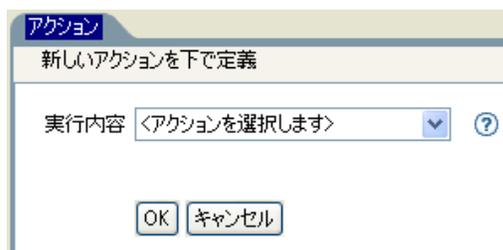
- ◆ 51 ページのセクション 4.1 「アクションビルダ」
- ◆ 52 ページのセクション 4.2 「アクションビルダ」
- ◆ 53 ページのセクション 4.3 「引数ビルダ」
- ◆ 56 ページのセクション 4.4 「条件ビルダ」
- ◆ 58 ページのセクション 4.5 「条件ビルダ」
- ◆ 59 ページのセクション 4.6 「一致属性ビルダ」
- ◆ 61 ページのセクション 4.7 「アクションの引数コンポーネントビルダ」
- ◆ 62 ページのセクション 4.8 「引数値リストビルダ」
- ◆ 62 ページのセクション 4.9 「名前付き文字列ビルダ」
- ◆ 63 ページのセクション 4.10 「条件の引数コンポーネントビルダ」
- ◆ 64 ページのセクション 4.11 「パターン文字列ビルダ」
- ◆ 65 ページのセクション 4.12 「文字列ビルダ」
- ◆ 66 ページのセクション 4.13 「XPath ビルダ」
- ◆ 66 ページのセクション 4.14 「ネームスペースエディタ」

4.1 アクションビルダ

アクションビルダでは、ルールを構成するアクションを追加、表示、および削除できます。アクションは他のアクションを含むことができます。

4.1.1 アクションの作成

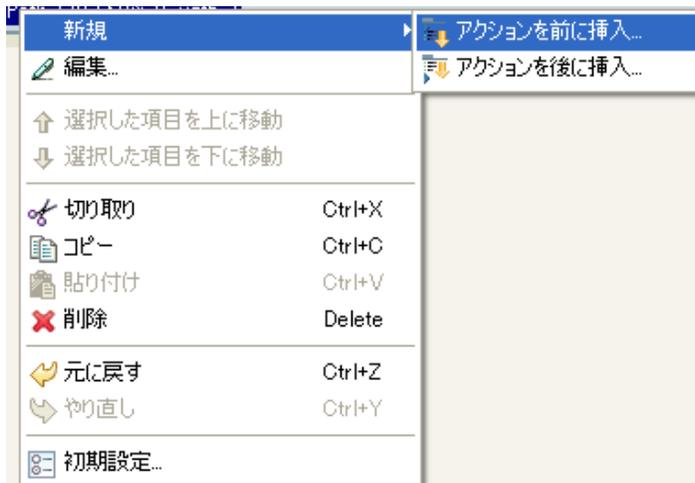
- 1 ポリシービルダでは、新しいルールを作成するかまたは既存のルールを編集します。
- 2 [アクション] タブをダブルクリックして、アクションビルダを起動します。



- 3 ドロップダウンリストから目的のアクションを選択し、[OK] をクリックします。

4.1.2 アクションビルダの追加オプション

- 1 アクションを右クリックして追加オプションを表示します。



- ◆ [新規] > [アクションを前に挿入] : 現在のアクションの前に新しいアクションを追加します。
- ◆ [新規] > [アクションを後に挿入] : 現在のアクションの後に新しいアクションを追加します。
- ◆ 編集 : アクションビルダを起動します。
- ◆ 選択した項目を上に移動 : 選択したアクションを実行順序内の上に移動します。
- ◆ 選択した項目を下に移動 : 選択したアクションを実行順序内の下に移動します。
- ◆ アクションの切り取り、コピー、貼り付け、または削除 : アクションを切り取り、コピー、貼り付け、または削除します。
- ◆ 元に戻す、または、やり直し : 最後のアクションを元に戻す、またはやり直します。
- ◆ 初期設定 : ポリシービルダ内でデフォルト機能を設定できます。
- ◆ ヘルプ : アクションを選択して、[ヘルプ] アイコンをクリックし、そのアクションに固有の情報を参照します。

4.2 アクションビルダ

アクションビルダでは、別のアクション内にアクションを作成できます。アクションビルダを起動するには、以下のいずれかのアクションを選択して、[引数を編集する] アイコンをクリックします。

- ◆ (246 ページ) 毎
- ◆ (252 ページ) エンタイトルメントの実装
- ◆ (250 ページ) If
- ◆ (295 ページ) While

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 4-1 For Each アクション

実行内容 毎

ノードセットを指定してください: * "Added Entitlement("Group")"

アクションを指定してください: * do-add-dest-attr-value

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

図 4-2 アクションビルダ

実行内容 ターゲット属性値の追加

属性名を指定してください: * Member

クラス名を指定してください: 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の指定: * ローカル変数("current-node")

値のタイプを指定してください: string

文字列を入力: * ターゲットDN0

4.3 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。

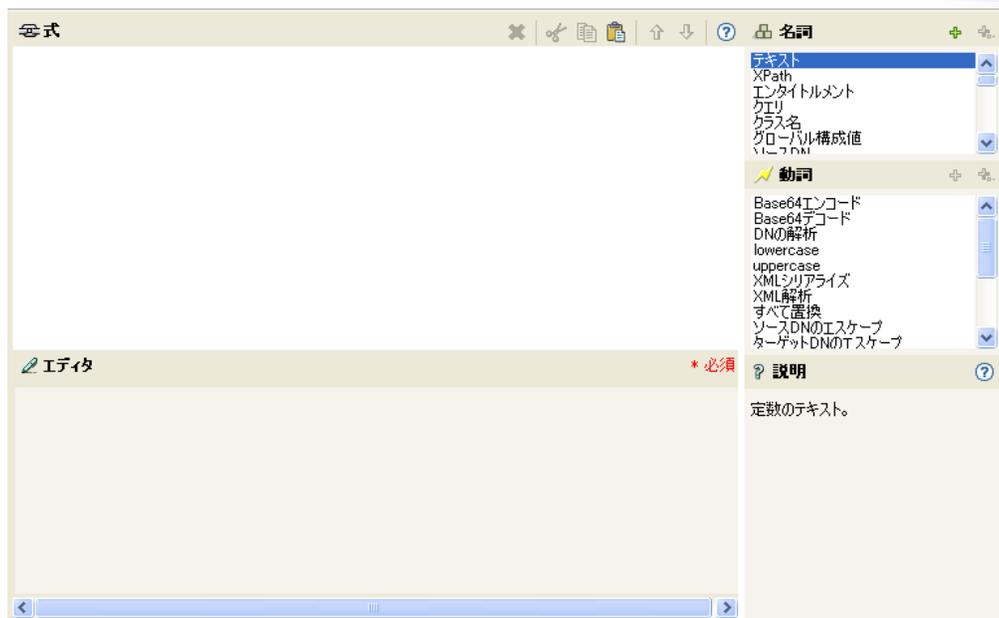
引数ビルダは 5 つの別々のセクションで構成されます。

- ◆ **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] を選択して名詞トークンを [式] ペインに追加します。詳細については、297 ページの「名詞トークン」を参照してください。
- ◆ **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] を選択して動詞トークンを [式] ペインに追加します。詳細については、337 ページの「動詞トークン」を参照してください。
- ◆ **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] ペイン内で名詞および動詞に値を提供します。

図 4-3 引数ビルダ

引数の作成および編集

引数を構成するために式領域にコンポーネントを追加または式領域からコンポーネントを削除します。エディタでコンポーネント値を指定します。



- ◆ 54 ページのセクション 4.3.1 「引数ビルダの起動」
- ◆ 55 ページのセクション 4.3.2 「引数ビルダの例」

4.3.1 引数ビルダの起動

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

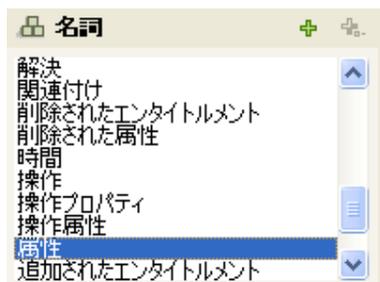
- ◆ (223 ページ) 関連付けの追加
- ◆ (224 ページ) ターゲット属性値の追加
- ◆ (226 ページ) ターゲットオブジェクトの追加
- ◆ (228 ページ) ソース属性値の追加
- ◆ (232 ページ) XML テキストの追加
- ◆ (235 ページ) ターゲット属性値のクリア (選択されたオブジェクトが [DN] または [関連付け] である場合)。
- ◆ (237 ページ) ソース属性値のクリア (選択されたオブジェクトが [DN] または [関連付け] である場合)。
- ◆ (241 ページ) ターゲットオブジェクトの削除 (選択されたオブジェクトが [DN] または [関連付け] である場合)。
- ◆ (242 ページ) ソースオブジェクトの削除 (選択されたオブジェクトが [DN] または [関連付け] である場合)。
- ◆ (243 ページ) 一致オブジェクトの検索
- ◆ (246 ページ) 毎

- ◆ (253 ページ) ターゲットオブジェクトの移動
- ◆ (255 ページ) ソースオブジェクトの移動
- ◆ (256 ページ) 操作属性値の再フォーマット
- ◆ (257 ページ) 関連付けの削除
- ◆ (258 ページ) ターゲット属性値の削除
- ◆ (259 ページ) ソース属性値の削除
- ◆ (260 ページ) ターゲットオブジェクトの名前変更 (選択されたオブジェクトが [DN] または [関連付け] および [文字列を入力] である場合)。
- ◆ (262 ページ) ソースオブジェクトの名前変更 (選択されたオブジェクトが [DN] または [関連付け] および [文字列を入力] である場合)。
- ◆ (269 ページ) ターゲット属性値の設定 (選択されたオブジェクトが [DN] または [関連付け] であり、[値タイプを入力] が指定されていない場合)。
- ◆ (271 ページ) ターゲットパスワードの設定
- ◆ (272 ページ) ローカル変数の設定
- ◆ (274 ページ) 操作関連付けの設定
- ◆ (275 ページ) 操作クラス名の設定
- ◆ (276 ページ) 操作ターゲット DN
- ◆ (277 ページ) 操作プロパティの設定
- ◆ (278 ページ) 操作ソース DN の設定
- ◆ (279 ページ) 操作テンプレート DN の設定
- ◆ (280 ページ) ソース属性値の設定
- ◆ (282 ページ) ソースパスワードの設定
- ◆ (285 ページ) XML 属性の設定
- ◆ (286 ページ) ステータス
- ◆ (291 ページ) メッセージのトレース

4.3.2 引数ビルダの例

次の例では、名と姓全体の最初の文字からユーザ名の引数を作成します。

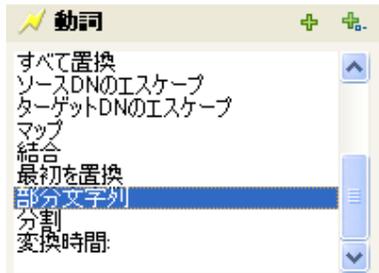
- 1 名詞のリストから [属性] をダブルクリックします。



- 2 名の属性を指定または選択します。



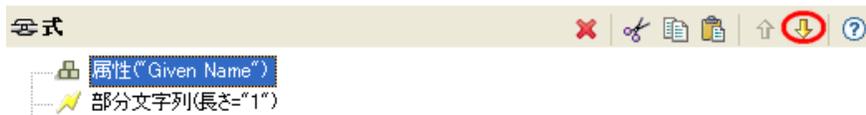
- 3 動詞のリストから [部分文字列] をダブルクリックします。



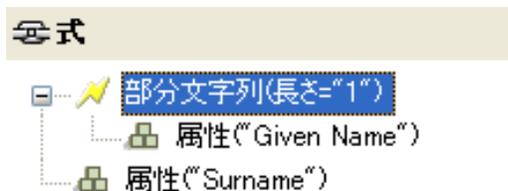
- 4 [長さ] フィールドに「1」を入力します。



- 5 [名前] 属性を選択して、[下へ移動] アイコンをクリックします。



- 6 名詞リストから [属性] をダブルクリックします。
7 [名字] 属性を指定または参照します。



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。

- 8 [OK] をクリックして、引数を保存します。

4.4 条件ビルダ

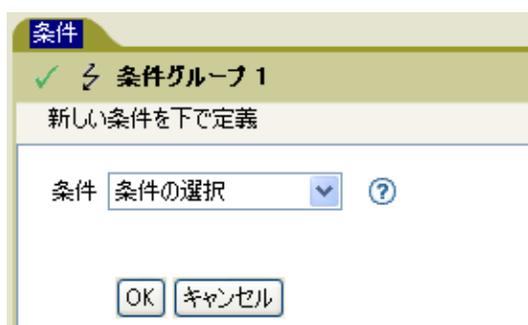
条件ビルダでは、ルールを構成する条件を追加、表示、および削除できます。条件には、1つ以上の条件と1つ以上の条件グループが含まれます。条件グループには2つの異なる

条件構造が含まれます。条件構造は、条件グループの論理を定義します。2つの条件構造は次のとおりです。

- ◆ OR 条件、AND グループ
- ◆ AND 条件、OR グループ
- ◆ 57 ページのセクション 4.4.1 「条件の作成」
- ◆ 57 ページのセクション 4.4.2 「条件ビルダの追加オプション」

4.4.1 条件の作成

- 1 ポリシービルダでは、新しいルールを作成するかまたは既存のルールを編集します。
- 2 [条件] タブをダブルクリックして、条件ビルダを起動します。



- 3 ドロップダウンリストから目的の条件を選択し、[OK] をクリックします。

4.4.2 条件ビルダの追加オプション

- 1 条件を右クリックして追加オプションを表示します。



- ◆ [新規] > [条件を前に挿入] : 現在の条件の前に新しい条件を追加します。
- ◆ [新規] > [条件を後に挿入] : 現在の条件の後に新しい条件を追加します。

- ◆ **編集** : 条件ビルダを起動します。
- ◆ **選択した項目を上に移動します** : 選択した条件を実行順序内の上に移動します。
- ◆ **選択した項目を下に移動します** : 選択した条件を実行順序内の下に移動します。
- ◆ **切り取り、コピー、貼り付け、または削除** : 条件を切り取り、コピー、貼り付け、または削除します。
- ◆ **元に戻す、または、やり直し** : 最後のアクションを元に戻す、またはやり直します。
- ◆ **初期設定** : ポリシービルダ内でデフォルト機能を設定できます。
- ◆ **ヘルプ** : 条件を選択して、[ヘルプ] アイコンをクリックし、その条件に固有の情報を参照します。

条件ビルダとルールの詳細については、[37 ページのセクション 3.4 「ルールの作成」](#) を参照してください。

4.5 条件ビルダ

条件ビルダでは、アクション内に条件を作成できます。条件ビルダを起動するには、次のいずれかのアクションを選択し、[引数を編集する] アイコンをクリックします。

- ◆ [\(250 ページ\) If](#)
- ◆ [\(295 ページ\) While](#)

1 条件ビルダで、目的の条件を参照して選択します。

条件のリストを作成する
条件のリストを作成、削除、または再配置します。



2 条件を定義して、[OK] をクリックします。

条件ビルダでは、ツールバーを使用して多くの異なるアクションを実行できます。

- ◆ [編集] アイコンを使用すると、条件が編集用に開きます。
- ◆ [条件グループの追加] アイコンを使用すると、新しい条件を追加できます。複数の条件グループを持つことができます。
- ◆ [新規アイテムの追加] アイコンを使用すると、新しい条件が追加されます。複数の条件を持つことができます。
- ◆ [削除] アイコンを使用すると、選択した条件または条件グループが削除されます。
- ◆ [切り取り]、[コピー]、および [貼り付け] アイコンを使用すると、クリップボードを使用して条件および条件グループをコピーできます。

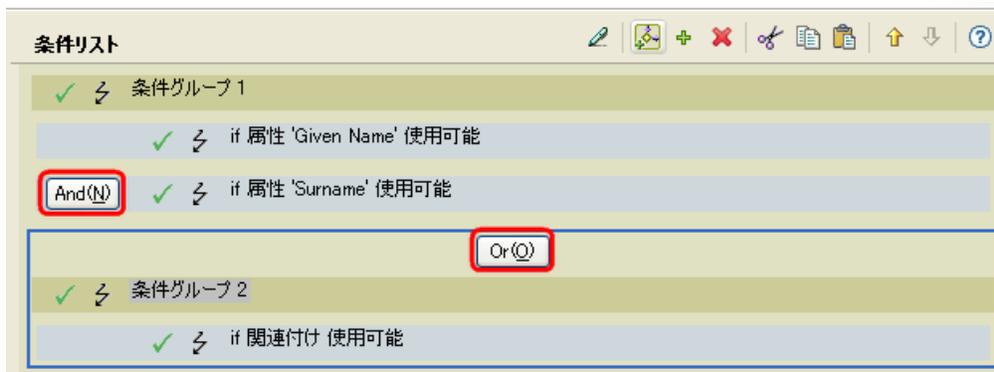
- ◆ [上へ移動] および [下へ移動] アイコンを使用すると、条件および条件グループを移動できます。

複数の条件および条件グループを持っている場合は、[および/または] アイコンで組み合わせることができます。条件グループの [および/または] アイコンを変更すると、条件に対しても変更されます。

図 4-4 条件ビルダ内の [および/または] アイコン

条件のリストを作成する

条件のリストを作成、削除、または再配置します。



4.6 一致属性ビルダ

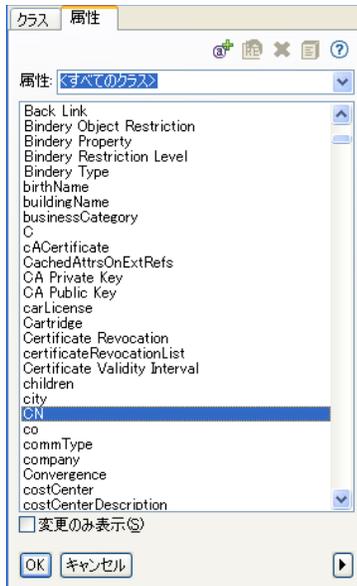
一致属性ビルダでは、データストアに一致するオブジェクトが存在するかどうかを判断するために、(243 ページ) **一致オブジェクトの検索** によって使用される属性および値を選択できます。

たとえば、共通名と場所に基づいてユーザを一致させるには

- 1 [一致オブジェクトの検索] のアクションを選択します。
- 2 一致オブジェクトの検索の範囲を選択します。[エントリー]、[サブオーディネート]、または [サブツリー] から選択します。>
- 3 検索の開始点となる DN を指定します。
- 4 [一致属性の編集] アイコン をクリックして、一致属性ビルダを起動します。

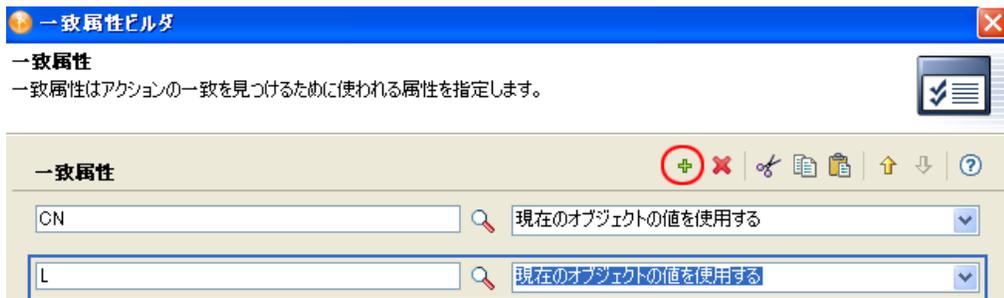


- 5 [属性の参照] アイコンをクリックして、スキーマブラウザを起動します。
- 6 [属性] タブをクリックして、属性を参照して選択します。



7 [OK] をクリックします。

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



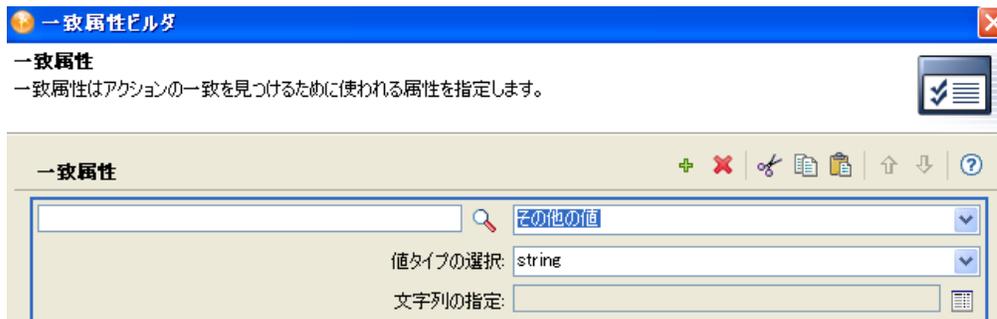
8 [完了] をクリックします。

一致属性ビルダでは、現在のオブジェクトからの値を使用する代わりに別の値を指定することができます。別の値を使用するには、[現在のオブジェクトの値を使用する] の代わりに [その他の値] を選択します。指定できる値タイプは複数あります。

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

その他の値を使用する

- 1 一致属性ビルダを起動して、[その他の値] を選択します。



- 2 目的の値タイプを選択します。
- 3 値を指定して [OK] をクリックします。

4.7 アクションの引数コンポーネントビルダ

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

- ◆ (224 ページ) ターゲット属性値の追加
- ◆ (228 ページ) ソース属性値の追加
- ◆ (256 ページ) 操作属性値の再フォーマット
- ◆ (258 ページ) ターゲット属性値の削除
- ◆ (259 ページ) ソース属性値の削除
- ◆ (269 ページ) ターゲット属性値の設定
- ◆ (280 ページ) ソース属性値の設定

図 4-5 ターゲット属性値の追加アクション



- 1 値タイプが「構造」に設定されているときに [コンポーネントを編集する] アイコンをクリックします。
- 2 アクションコンポーネントの値を作成します。
値を入力するか、または [引数を編集する]  アイコンをクリックして、引数ビルダ内で値を作成します。

引数コンポーネント
引数コンポーネントは構造化された引数値です。



名前	値
value	user

3 [完了] をクリックします。

4.8 引数値リストビルダ

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

- ◆ (267 ページ) デフォルト属性値の設定

図 4-6 デフォルト属性値の設定

実行内容 **デフォルト属性値の設定** 

属性名を指定してください: * company 

ライトバック: false 

引数値を指定してください: * 0 

- 1 値のタイプ ([counter (カウンタ)], [DN], [int (整数)], [間隔], [octet (オクテット)], [state (状態)], [文字列], [構造], [teleNumber (電話番号)], [時間]) を選択します。> > >
- 2 [値リストを編集する] アイコンをクリックします。

引数値
引数値は属性に使用される値を指定します。



タイプ	引数値
structured	

- 3 [引数を編集する] アイコンをクリックします。
- 4 アクションコンポーネントの値を作成します。
値を入力するか、または [引数を編集する]  アイコンをクリックして引数ビルダ内で値を作成します。
- 5 [完了] をクリックします。

4.9 名前付き文字列ビルダ

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

- ◆ (247 ページ) イベントの生成
- ◆ (263 ページ) 電子メールの送信

◆ (265 ページ) テンプレートから電子メールを送信

- 1 ドロップダウンリストから、文字列の名前を選択します。
- 2 [引数を編集する] アイコン  をクリックして引数ビルダを起動し、文字列の値を作成します。

名前付き文字列ビルダ

文字列要素は引数の値を指定します。



- 3 [完了] をクリックします。

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

名前付き文字列ビルダ

文字列要素は引数の値を指定します。



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

4.10 条件の引数コンポーネントビルダ

条件の引数コンポーネントビルダを起動するには、次のいずれかの条件を選択し、[モード] で [構造] を選択して [ArgComponent ビルダの起動] アイコン  を表示します。

- ◆ (180 ページ) If 属性
- ◆ (186 ページ) If ターゲット属性
- ◆ (178 ページ) If 関連付け

図 4-7 If 属性モード

- 1 条件コンポーネントの名前および値を指定します。

引数コンポーネント

条件引数コンポーネントは名前/値のペアです。



名前	値
<input type="text"/>	<input type="text"/>

2 [完了] をクリックします。

4.11 パターン文字列ビルダ

パターン文字列ビルダは、(331 ページ) 一意の名前トークンが選択されているときに引数ビルダのエディタから起動できます。引数ビルダのエディタペインに [パターン] フィールドが表示されるので、このフィールドをクリックして、パターン文字列ビルダを起動します。

図 4-8 引数ビルダ内の一意の名前トークン

引数の作成および編集

引数を構成するために式領域にコンポーネントを追加または式領域からコンポーネントを削除します。エディタでコンポーネント値を指定します。



式

一意の名前(counter-pattern="last", counter-use="fallback", on-unavailable="error", "")

品名詞

- ターゲットDN
- ターゲット属性
- ターゲット名
- ドキュメント
- パスワード
- パスワードの生成
- ローカル変数
- 一意の名前
- 一致しないソースDN
- 秘密

動詞

- Base64エンコード
- Base64デコード
- DNの解析
- lowercase
- uppercase
- XMLシリアライズ
- XML解析
- すべて置換
- ソースDNのエスケープ
- ターゲットDNのTスケープ

エディタ

トレースしない: false

属性名:

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: *

カウンタの使用タイミング: fallback

カウンタとともに使用するパターン: last

* 必須

説明

生成された一意の名前。

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

パターンビルダ パターンリストの定義



パターン値

パターン:

3 [完了] をクリックします。

4.12 文字列ビルダ

文字列ビルダでは、**ワークフローの開始**などの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。

- 1 [名前] フィールドに文字列の名前を指定します。
- 2 [引数を編集する] アイコンをクリックして、文字列の値を構築します。
- 3 [OK] をクリックし、[Finish (終了)] をクリックします。

ワークフローの開始アクションには、[追加の引数] フィールドがあり、特定のワークフローに必要な値を作成できます。これらのフィールドは、それぞれのワークフローで一意です。文字列ビルダでは、文字列を作成し、文字列用の値を設定できます。

以下の例では、入力および追加操作が行われるたびにワークフロープロセスが開始されます。このワークフローは携帯電話の要求です。[追加の引数] フィールド内の文字列は、携帯電話のプロバイダおよび要求の理由です。

図 4-9 ワークフローの開始の例

実行内容 ?

プロビジョニング要求DNを指定してください: * 🔍

ユーザアプリケーションURLを指定してください: *

承認されたユーザDNを指定してください: * 🔍

承認されたユーザパスワードを指定してください: * 📄

受信者DNを指定してください: * 📄

追加の引数を指定してください: 📄

[追加の引数] フィールドの右にあるアイコンをクリックすると、文字列ビルダが起動します。この例では、プロバイダの値が「ACMEWireless」に設定され、理由の値が「new hire」になっています。例をXMLで表示するには、[start_workflow.xml \(../samples/start_workflow.xml\)](#) を参照してください。

図 4-10 文字列ビルダの例

文字列ビルダ

文字列要素は引数の値を指定します。



名前	文字列の値:	+	×	✂	📄	📁	↑	↓	?
provider	"ACMEWireless"								
reason	"new hire"								

4.13 XPath ビルダ

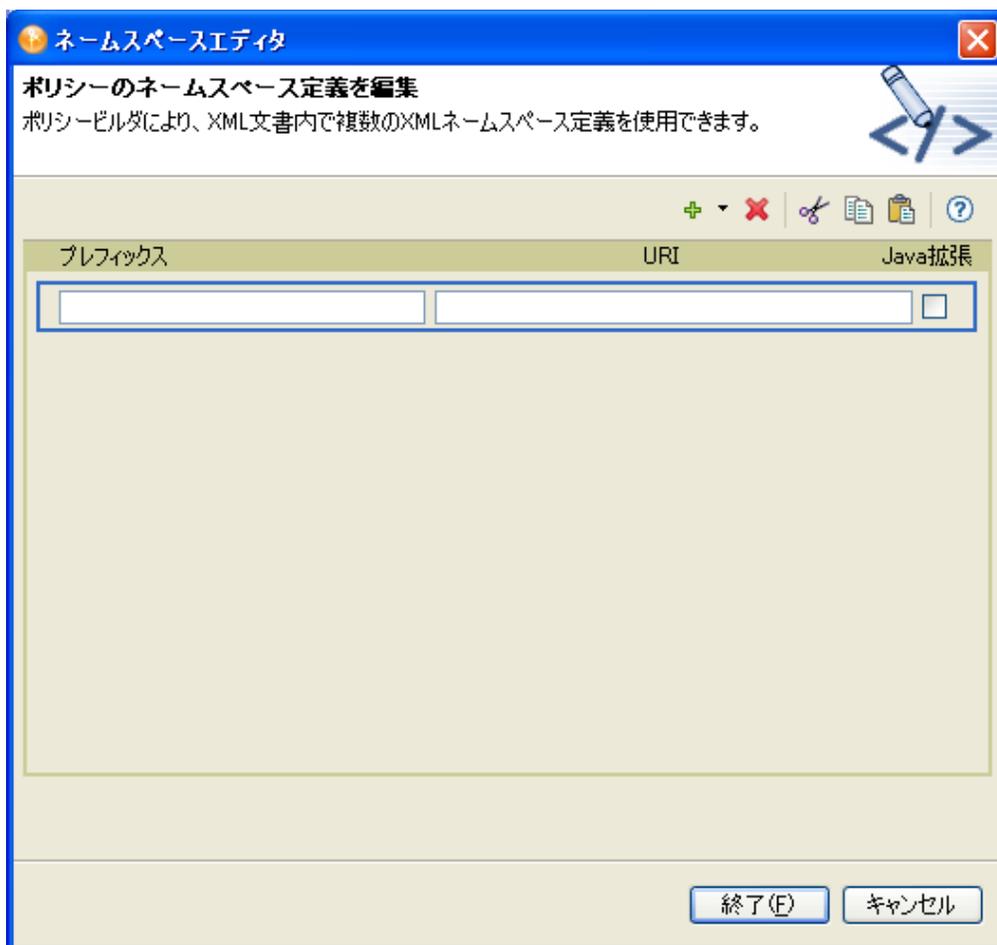
XPath ビルダは、任意の XML ドキュメントに対して XPath 式を作成してテストできる強力なツールです。詳細については、69 ページの「XPath ビルダの使用」を参照してください。

4.14 ネームスペースエディタ

ポリシービルダでは、XML ドキュメント内で、複数の XML ネームスペースを使用できます。ネームスペースを定義するには、[名前] フィールドでネームスペースのプレフィックスを指定して、[URI] フィールドで URI を指定します。[Java 拡張] チェックボックスはオフのままにします。

XML ネームスペースを使用し、XPath を介して Java* クラスにアクセスすることもできます。Java クラスのネームスペースを作成するには、ネームスペースプレフィックスを [名前] フィールドに指定し、クラス名を [URI] フィールドに指定して、[Java 拡張] チェックボックスを選択します。

図 4-11 ネームスペースエディタ



4.14.1 ネームスペースを使用した Java クラスへのアクセス

Novell では、ポリシービルダから XPath 式を使用して呼び出すことができる Identity Manager Java* クラスが複数用意されています。次のリンクから、これらの Java クラスの Javadoc リファレンスを開くことができます。

- ◆ [com.novell.nds.dirxml.driver.XdsQueryProcessor](http://developer.novell.com/documentation/dirxml/dirxmlbk/api/com/novell/nds/dirxml/driver/XdsQueryProcessor.html) (<http://developer.novell.com/documentation/dirxml/dirxmlbk/api/com/novell/nds/dirxml/driver/XdsQueryProcessor.html>)
- ◆ [com.novell.nds.dirxml.driver.XdsCommandProcessor](http://developer.novell.com/documentation/dirxml/dirxmlbk/api/com/novell/nds/dirxml/driver/XdsCommandProcessor.html) (<http://developer.novell.com/documentation/dirxml/dirxmlbk/api/com/novell/nds/dirxml/driver/XdsCommandProcessor.html>)
- ◆ [com.novell.nds.dirxml.driver.DNConverter](http://developer.novell.com/documentation/dirxml/dirxmlbk/api/com/novell/nds/dirxml/driver/DNConverter.html) (<http://developer.novell.com/documentation/dirxml/dirxmlbk/api/com/novell/nds/dirxml/driver/DNConverter.html>)

Java Developer Kit (JDK*) にも、`java.lang.String` や `java.lang.System` など、役に立つクラスがいくつかあります。これらのクラスのリファレンスは JDK で使用できます。

XPath の使用およびリストされている Novell Java クラスの詳細については、[DirXML Driver Developer Kit](http://developer.novell.com/documentation/dirxml/dirxmlbk/ref/dirxmlfaq.html) (<http://developer.novell.com/documentation/dirxml/dirxmlbk/ref/dirxmlfaq.html>) を参照してください。

XPath ビルダの使用

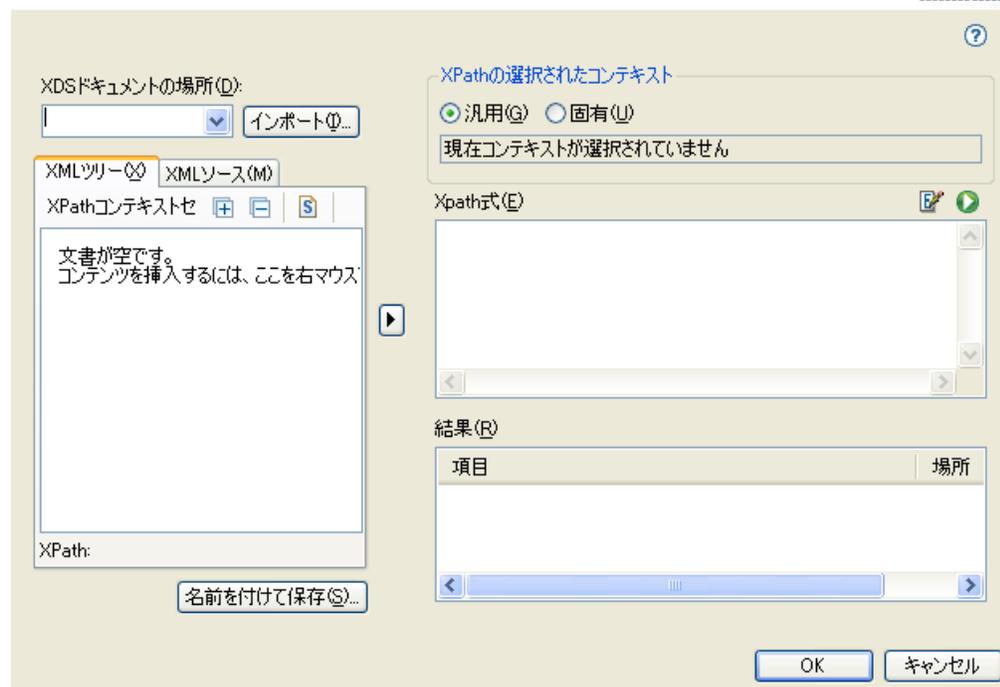
5

XPath ビルダは、任意の XML ドキュメントに対して XPath 式を作成してテストできる強力なツールです。XDS ドキュメントに対して異なる式をテストして、式をテストしている間に XDS ドキュメントを変更することができます。XPath 式の詳細については、『*Identity Manager 3.5.1 ポリシーの理解*』の「**XPath 1.0 式**」を参照してください。

図 5-1 XPath ビルダ

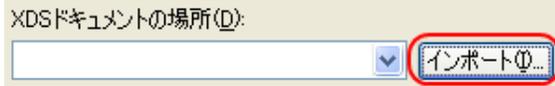
XPath 式の作成

XDS ドキュメントをインポートし、XPath コンテキストセクタで現在のコンテキストを選択して、XPath 式のテスト領域で XPath 式を構築し、テストします。

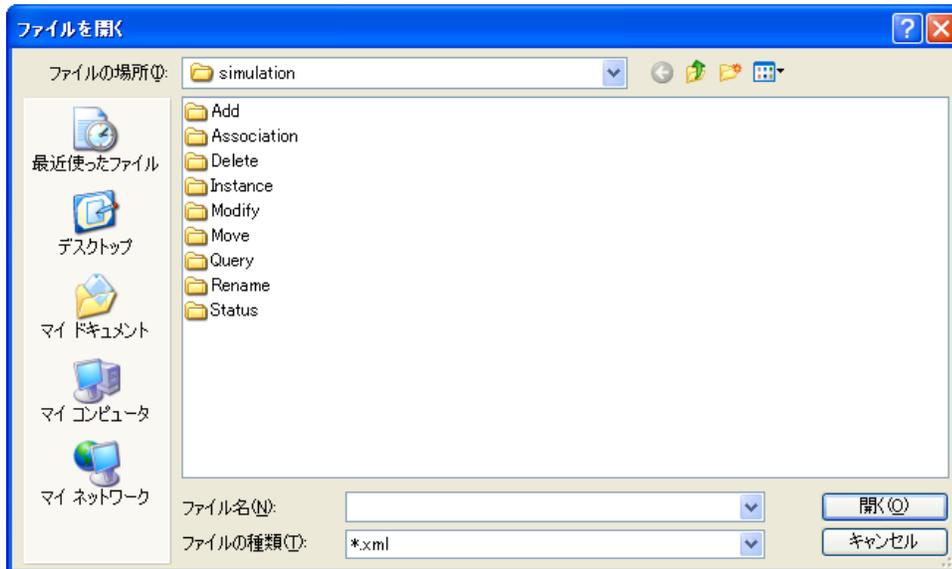


XPath ビルダを使用する

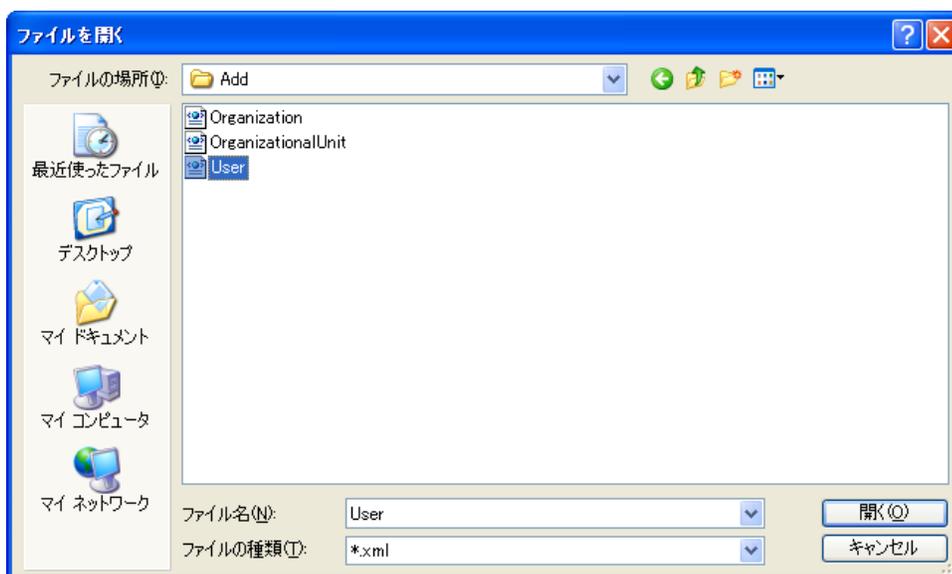
- 1 ポリシービルダで、以下の条件またはアクションのいずれかを選択して、[XPath ビルダの起動] アイコンをクリックします。
 - ◆ (216 ページ) If XML 属性
 - ◆ (218 ページ) If XPath 式
 - ◆ (230 ページ) XML 要素の追加
 - ◆ (232 ページ) XML テキストの追加
 - ◆ (239 ページ) XPath 式によるクローン
 - ◆ (285 ページ) XML 属性の設定
 - ◆ (290 ページ) XPath のストリップ
- 2 [インポート] を選択し、テストする XDS ドキュメントを参照して選択します。



Designer には、XPath 式をテストするためのサンプルのイベントファイルが付属します。ファイルは、プラグイン `com.novell.designer.idm.policy_version\simulation` にあります。version は、Designer の現在のバージョンです。イベントは、追加、関連付け、削除、インスタンス、変更、移動、クエリ、名前変更、およびステータスです。

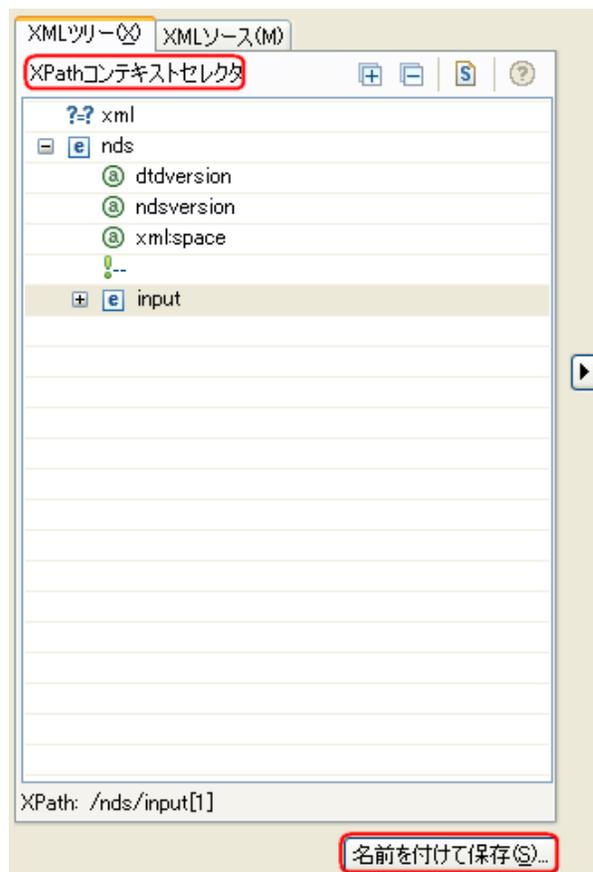


- 3 フォルダをダブルクリックして、使用可能なイベントを表示します。選択できるファイルは、イベントごとに異なります。たとえば、[追加] を選択した場合は、`Organization.xml`、`OrganizationalUnit.xml`、および `User.xml` の 3 つのオプションがあります。ファイルはイベントを示します。User.xml を選択した場合、これはユーザオブジェクトの追加のイベントになります。

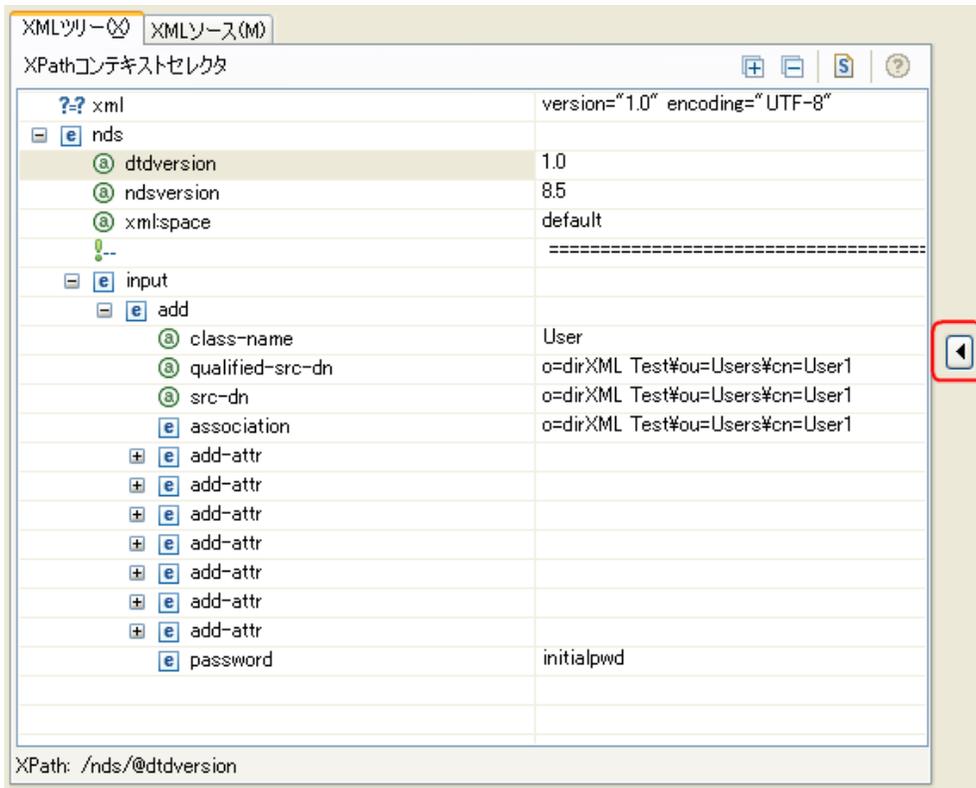


4 ファイルを選択して、[開く] をクリックします。

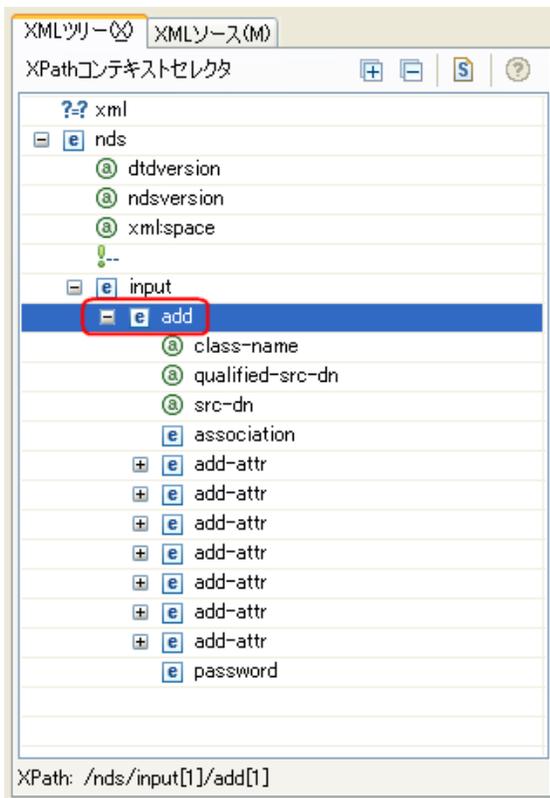
入力ドキュメントが [XPath コンテキストセクタ] ビューに表示されます。[XML ソース] タブを使用すると、XML ソースエディタを使用してインポートされたドキュメントを編集したり、別のエディタから XML ドキュメントをコピーしてソースビューに貼り付けることができます。ドキュメントを変更した場合は、[名前を付けて保存] をクリックして、変更したドキュメントを保存します。



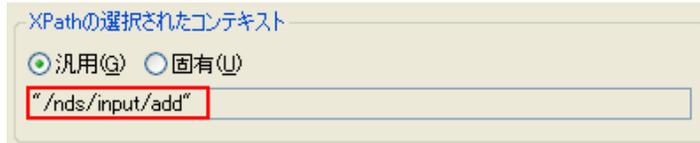
XDS ドキュメントをスクロールせずに参照したい場合は、[XPath の詳細の非表示] アイコン■をクリックします。[XPath 式] および [結果] ウィンドウを表示するには、[XPath の詳細の表示] アイコン■をクリックします。



- 5 XPath 式の作成を開始するドキュメント内の現在の位置を選択します。



選択した XPath コンテキストは、示されているように [XPath の選択されたコンテキスト] に表示されます。

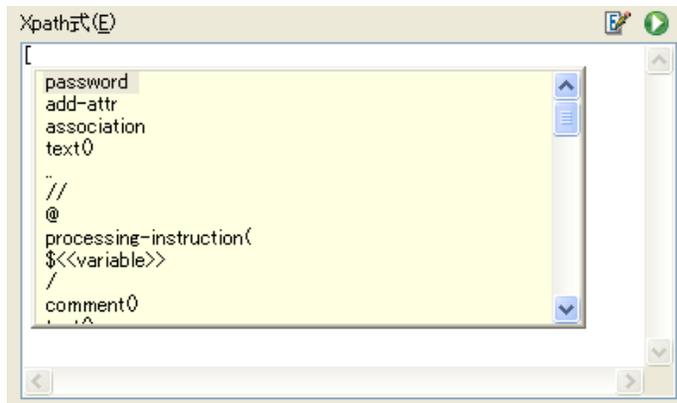


6 [汎用] または [固有] を選択します。

[汎用] は、XML ドキュメント全体を検索して指定した XPath 式と一致させます。XPath 式の各インスタンスに対して結果を返します。この例では、XPath 式は “/nds/input/add” です。追加されたそれぞれのインスタンスに対して XML ドキュメント全体を検索します。

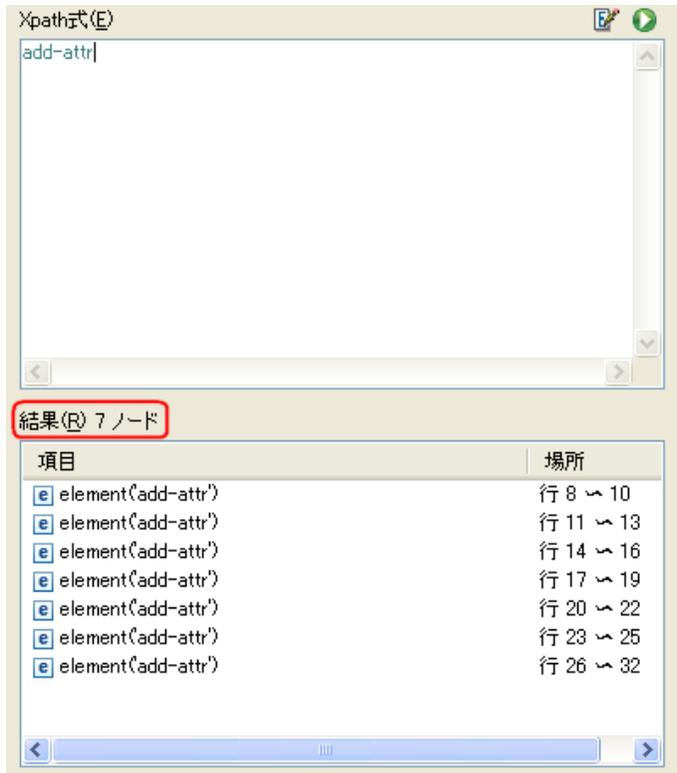
[固有] は、一致を検出するまで XML ドキュメントを検索して停止します。固有の XPath 式は、“/nds/input[1]/add[1]” です。追加の最初のインスタンスを検索して停止します。どのインスタンスを使用するかは、[XML コンテキストセクタ] 内で XPath 要素の次のインスタンスを選択することによって指定することができます。

7 XPath 式を [XPath 式] フィールドに指定します。



注：Ctrl Space 3、/、[、または (のキー操作の組み合わせを使用して、コードの完了をトリガします。式は、カーソルの位置まで評価され、挿入可能な要素がドロップダウンリストに表示されます。

XPath 式の結果は、下の [結果] テキストエリアに表示されます。



XPath エディタが指揮を評価しない場合は、[Evaluate XPath expression (XPath 式の評価)] アイコン  をクリックして、XPath ビルダが式を評価するようにします。

- 8 XPath 式の作成およびテストが完了したら、[OK] をクリックして XPath ビルダを終了します。[XPath 式] に表示されているテキストは、編集しているポリシー内に配置されます。

スキーママッピングポリシーの定義

6

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ボールド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。

- ◆ 75 ページのセクション 6.1 「スキーママップエディタへのアクセス」
- ◆ 79 ページのセクション 6.2 「スキーママッピングポリシーの編集」
- ◆ 82 ページのセクション 6.3 「スキーママッピングポリシーのテスト」
- ◆ 84 ページのセクション 6.4 「XML のスキーママッピングポリシーへのアクセス」
- ◆ 84 ページのセクション 6.5 「追加のスキーママップポリシーオプション」

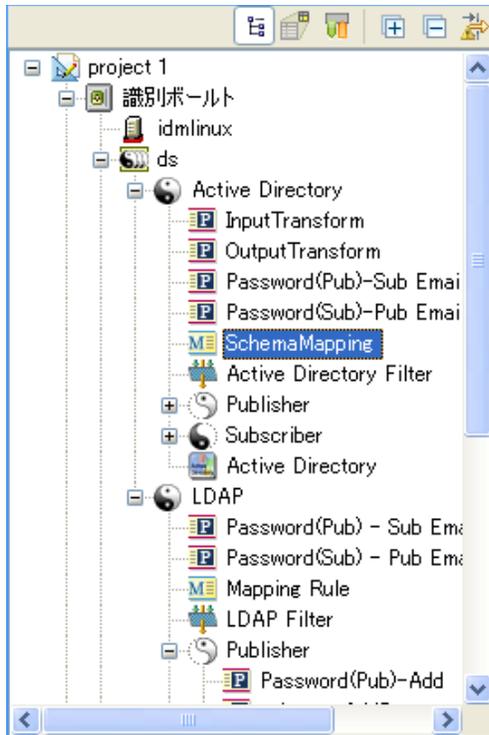
6.1 スキーママップエディタへのアクセス

スキーママップエディタでは、スキーママッピングポリシーを編集できます。Designer でスキーママップエディタにアクセスする方法には、[Outline (アウトライン)] ビュー、[Policy Flow (ポリシーフロー)] ビュー、または [Policy Set (ポリシーセット)] ビューのそれぞれを使用する3つの方法があります。

- ◆ 75 ページのセクション 6.1.1 「[Outline (アウトライン)] ビュー」
- ◆ 76 ページのセクション 6.1.2 「[Policy Flow (ポリシーフロー)] ビュー」
- ◆ 77 ページのセクション 6.1.3 「ポリシーセットビュー」
- ◆ 78 ページのセクション 6.1.4 「キーボード操作」

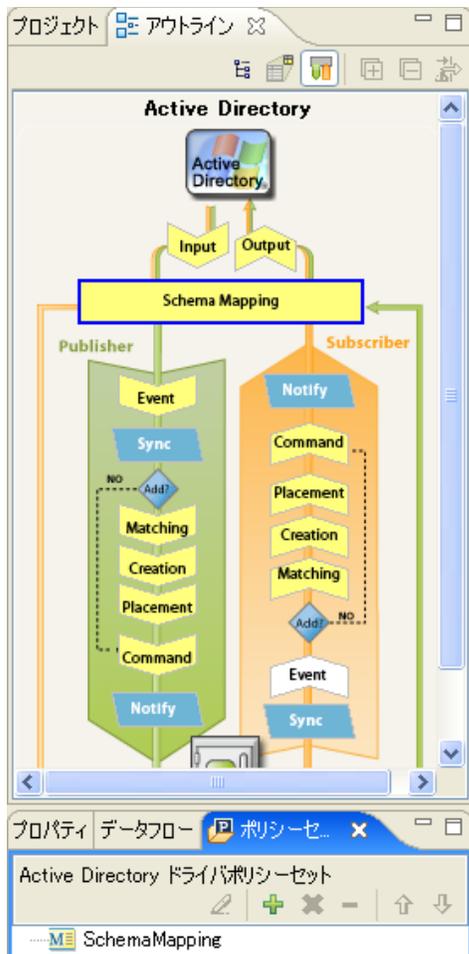
6.1.1 [Outline (アウトライン)] ビュー

- 1 開いているプロジェクトで、[Outline (アウトライン)] タブをクリックします。
- 2 [モデルアウトラインの表示] アイコンをクリックします。
- 3 スキーママッピングポリシーを管理するドライバを選択し、右側のプラス記号をクリックします。
- 4 [Schema Map (スキーママップ)] アイコンをダブルクリックして、スキーママップエディタを起動します。
または
右クリックして、[編集] を選択します。



6.1.2 [Policy Flow (ポリシーフロー)] ビュー

- 1 開いているプロジェクトで、[Outline (アウトライン)] タブをクリックします。
- 2 [ポリシーフローの表示] アイコンをクリックします。
- 3 スキーママッピングポリシーをダブルクリックして、スキーママップエディタを起動します。
または
右クリックして [Edit Policy (ポリシーの編集)] を選択して、スキーママップエディタを起動します。

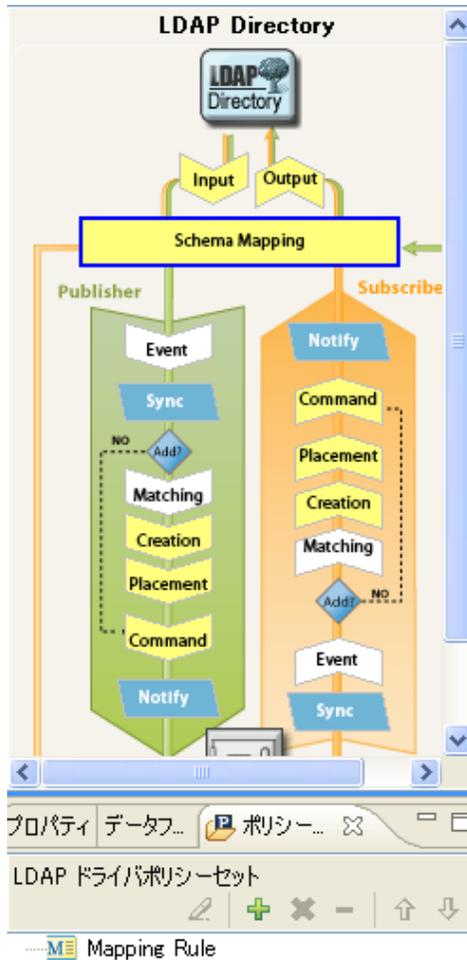


6.1.3 ポリシーセットビュー

- 1 スキーママップポリシーをダブルクリックして、[Policy Set (ポリシーセット)] ビューを起動します。

または

スキーママップポリシーを右クリックして、[編集] を選択します。



6.1.4 キーボード操作

表 6-1 スキーママップエディタでのキーボード操作

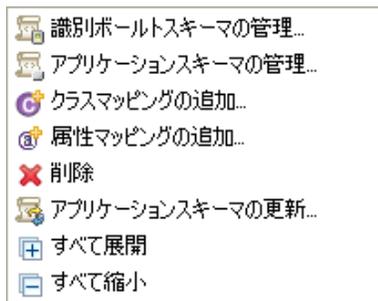
アクション	説明
< 上矢印 >	スキーママップエディタ内でカーソルを上方向に移動します。
< 下矢印 >	スキーママップエディタ内でカーソルを下方向に移動します。
< 左矢印 >	表示されている情報を縮小表示します
< 右矢印 >	表示されている情報を展開します。
挿入	クラスを追加します。
<Ctrl+Insert>	属性を追加します。
削除	選択された項目を削除します。

アクション	説明
<Enter>	編集モードに切り替えます。変更を保存するには、<Enter> を 2 回押します。
Esc	編集モードを終了します。

6.2 スキーママッピングポリシーの編集

スキーママップエディタでは、スキーママッピングポリシーを作成および編集できます。コンテキストに応じたメニューを表示するには、項目を右クリックします。

図 6-1 スキーママップエディタのコンテキストに応じたメニュー



- ◆ 79 ページのセクション 6.2.1 「クラスと属性の削除または追加」
- ◆ 81 ページのセクション 6.2.2 「アプリケーションスキーマのリフレッシュ」
- ◆ 81 ページのセクション 6.2.3 「項目の編集」
- ◆ 81 ページのセクション 6.2.4 「項目のソート」
- ◆ 81 ページのセクション 6.2.5 「スキーマの管理」

6.2.1 クラスと属性の削除または追加

- ◆ 79 ページの 「クラスまたは属性の削除」
- ◆ 80 ページの 「クラスの追加」
- ◆ 80 ページの 「属性の追加」

クラスまたは属性の削除

クラスまたは属性を接続システム上のクラスまたは属性にマップしないようにする場合、クラスまたは属性をスキーママッピングポリシーから完全に削除するのが最善の方法です。スキーママッピングポリシーで属性とクラスを追加または削除するには、次の 3 つの方法があります。

- ◆ 削除するクラスまたは属性を選択して右クリックし、[削除] をクリックします。
- ◆ 削除するクラスまたは属性を選択し、右上隅の [削除] アイコン  をクリックします。
- ◆ 削除するクラスまたは属性を選択し、キーを押します。

複数のクラスまたは属性を選択して、一度に削除することができます。

- 1 キーを押しながら、各項目をマウスで選択します。
- 2 キーを押して項目を削除します。



クラスの追加

- 1 スキーママップエディタ内で右クリックし、[Add Class Mapping (クラスマッピングの追加)] をクリックします。

または

右上隅の [クラスマッピングの追加] アイコンを選択します。

- 2 アイデンティティポールのドロップダウンリストで、追加するクラスを選択します。
- 3 接続システムのドロップダウンリストで、マップするクラスを選択します。
- 4 変更を保存するため、[ファイル] > [保存] の順にクリックします。

属性の追加

- 1 スキーママップエディタ内で右クリックし、[Add Attribute Mapping (属性マッピングの追加)] をクリックします。

または

右上隅の [属性マッピングの追加] アイコンを選択します。

- 2 アイデンティティポールのドロップダウンリストで、追加する属性を選択します。
- 3 接続システムのドロップダウンリストで、マップする属性を選択します。
- 4 変更を保存するため、[ファイル] > [保存] の順にクリックします。

6.2.2 アプリケーションスキーマのリフレッシュ

アプリケーションでスキーマを変更している場合は、それらの変更をスキーママッピングポリシーに反映させる必要があります。新しいスキーマを使用できるようにするには、ツールバーの「アプリケーションスキーマのリフレッシュ」アイコンをクリックします。

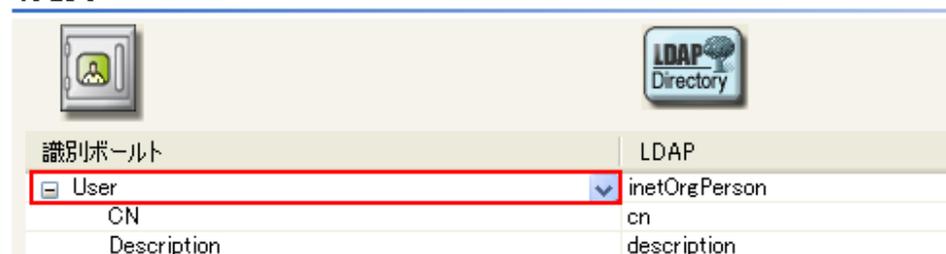
新しいクラスまたは属性のマッピングを作成する場合は、接続されているアプリケーションのドロップダウンリストに新しいスキーマを表示できます。

6.2.3 項目の編集

マッピングを編集するには、選択した行をダブルクリックします。インプレースエディタが表示されるので、マッピングを編集できます。

図 6-2 スキーママップエディタ

マッピング

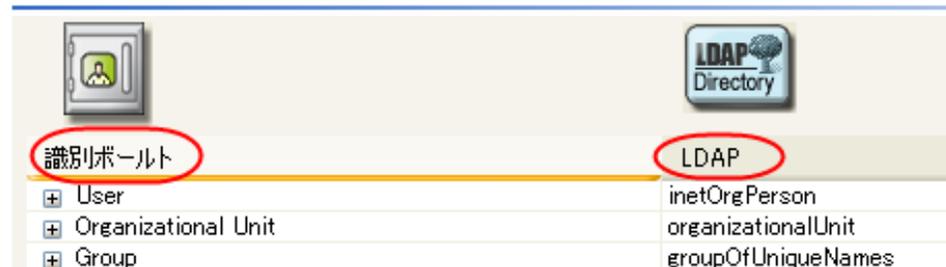


6.2.4 項目のソート

スキーマエディタでは、Identity Manager または接続システムに基づいて、項目を昇順にソートできます。ソートするには、いずれかの列見出しをクリックします。

図 6-3 スキーママップエディタでの項目のソート

マッピング



6.2.5 スキーマの管理

Designer では、アイデンティティボールドスキーマおよび任意の接続システムのスキーマを管理できます。スキーマをインポートして変更し、それを識別ボールドまたは接続システムに戻すことができます。識別ボールドスキーマを管理するには、スキーママップエディタ内を右クリックし、「[識別ボールドスキーマの管理]」をクリックします。接続システムのスキーマを管理するには、スキーママップエディタ内を右クリックし、「アプリ

「[セッションスキーマの管理](#)」をクリックします。スキーマの管理方法については、「[“スキーマの管理”](#)」、『[Designer 2.1 for Identity Manager 3.5.1](#)』を参照してください。

6.3 スキーママッピングポリシーのテスト

Designer には、ポリシーシミュレータと呼ばれるツールが付属しています。生産環境に実装せずにポリシーをテストできます。スキーママッピングエディタからポリシーシミュレータを起動して、変更後のポリシーをテストできます。

ポリシーシミュレータにアクセスしてスキーママッピングポリシーをテストするには

- 1 ツールバーの [ポリシーシミュレータの起動] アイコンをクリックします。
- 2 [識別ポールドへ] または [識別ポールドから] をスキーママップポリシーのシミュレーションポリシーとして選択します。
- 3 [Import (インポート)] を選択し、イベントをシミュレートするファイルを参照して選択します。
- 4 ファイルを選択して、[開く] をクリックします。

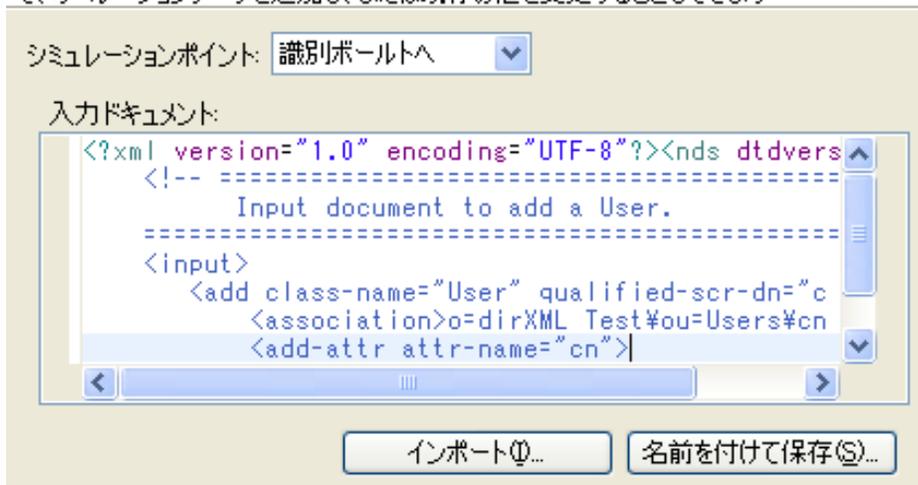
この例では、com.novell.designer.idm.policy\simulation\add\user.xml ファイルを使用して、ユーザオブジェクトの「追加」イベントをシミュレートします。

ポリシーシミュレータに、ユーザの「追加」イベントの入力ドキュメントが表示されます。

- 5 [次へ] をクリックして、シミュレーションを開始します。

入力ドキュメントのカスタマイズ

インデントポールドへのポリシー、またはインデントポールドからのポリシーをシミュレートするかどうかを選択します。入力ドキュメントをカスタマイズして、オペレーションデータを追加し、または既存の値を変更することもできます

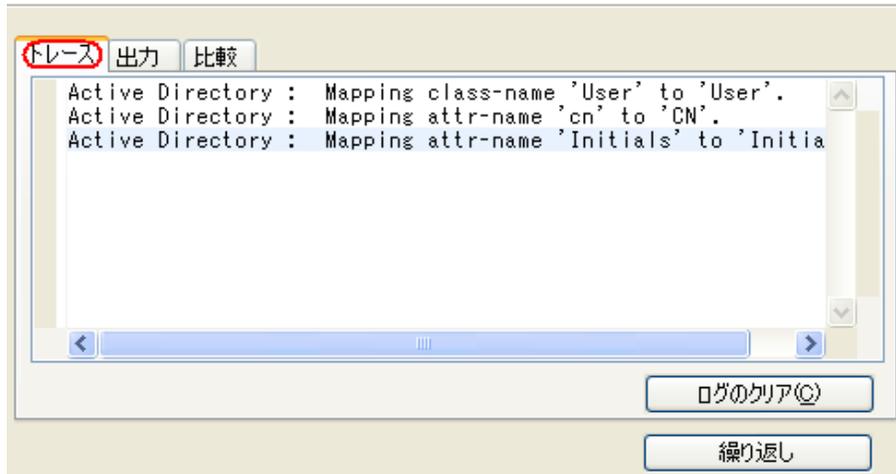


ポリシーシミュレータに、追加イベントのログ、出力ドキュメント、および入力ドキュメントと生成された出力ドキュメントの比較が表示されます。

- 6 [トレース] タブを選択して、DSTRACE に表示されるとおりの追加イベントの結果を表示します。

変換結果の表示

シミュレーションの詳細を確認するには[トレース]を、変換されたドキュメントを確認するには[出力]を、入力ドキュメントと出力ドキュメントの差異を確認するには[比較]を選択します。

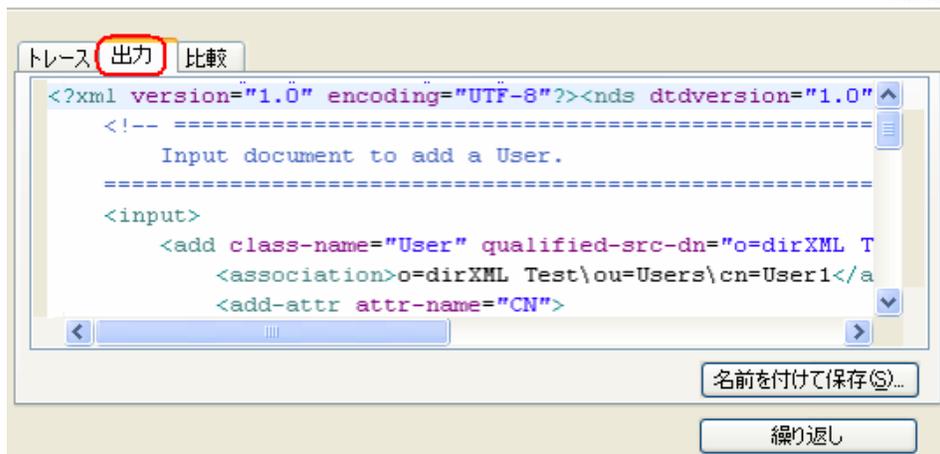


[ログのクリア] をクリックし、[繰り返す] をクリックして、新しいトレースログでシミュレーションを再び実行します。

- 7 [出力] タブを選択し、入力ドキュメントに対して実行されるスキーママップポリシーから生成される出力ドキュメントを表示します。この例では、ユーザの追加イベントです。

変換結果の表示

シミュレーションの詳細を確認するには[トレース]を、変換されたドキュメントを確認するには[出力]を、入力ドキュメントと出力ドキュメントの差異を確認するには[比較]を選択します。

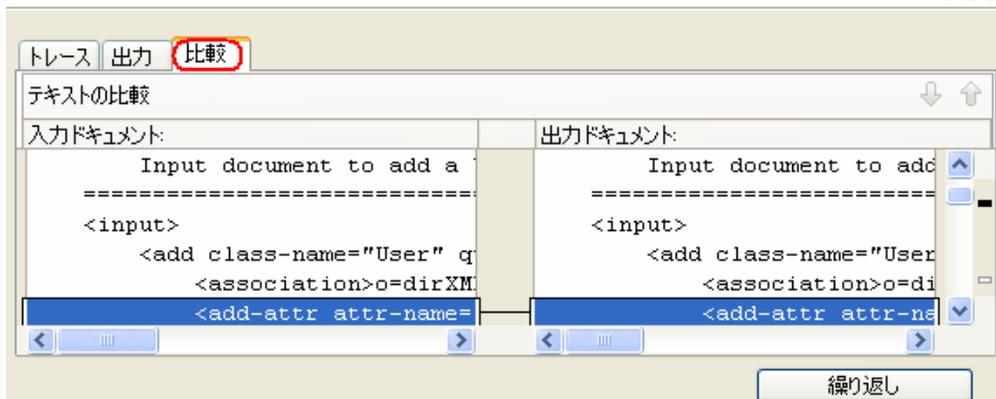


入力および出力ドキュメントを編集した後で、[名前を付けて保存] をクリックして出力を XML ファイルで保存します。

- 8 [Compare (比較)] タブを選択して、入力ドキュメントのテキストと、生成されたドキュメント (出力ドキュメント) を比較します。

変換結果の表示

シミュレーションの詳細を確認するには[トレース]を、変換されたドキュメントを確認するには[出力]を、入力ドキュメントと出力ドキュメントの差異を確認するには[比較]を選択します。



- 9 [Repeat (繰り返し)] をクリックして、別の入力ドキュメントを選択し、イベントの結果を表示します。
- 10 スキーママッピングポリシーのテストが済んだら、[終了] をクリックして、ポリシーシミュレータを閉じます。

6.4 XML のスキーママッピングポリシーへのアクセス

Designer では、XML エディタを使用して、XML を表示、編集、および検証できます。[XML ソース] タブまたは [XML ツリー] タブをクリックして XML エディタにアクセスします。XML エディタの詳細については、『*Designer 2.1 for Identity Manager 3.5.1*』の「**Novell XML エディタ**」を参照してください。

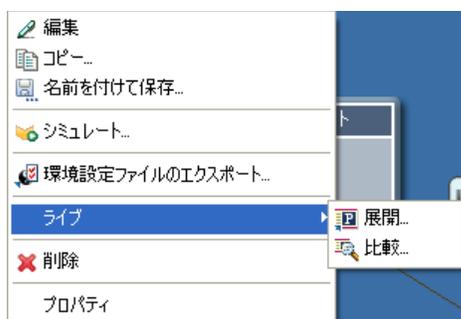
6.5 追加のスキーママップポリシーオプション

スキーママップポリシーを右クリックすると、[アウトライン] ビュー、[ポリシーフロー] ビュー、および [ポリシーセット] ビューで、複数のオプションが表示されます。

- 84 ページのセクション 6.5.1 「[Outline (アウトライン)] ビューの追加オプション」
- 85 ページのセクション 6.5.2 「[Policy Flow (ポリシーフロー)] ビューの追加オプション」
- 87 ページのセクション 6.5.3 「[Policy Set (ポリシーセット)] ビューの追加オプション」

6.5.1 [Outline (アウトライン)] ビューの追加オプション

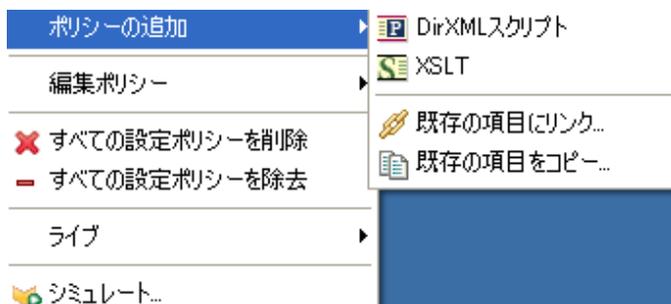
- 1 [Outline (アウトライン)] ビューでスキーママップポリシーを右クリックします。



- **編集** : スキーママップエディタを起動します。詳細については、[75 ページの第 6 章「スキーママッピングポリシーの定義」](#)を参照してください。
- **コピー** : スキーママップポリシーのコピーを作成します。
- **名前を付けて保存** : スキーママップポリシーを .xml ファイルとして保存します。
- **新規** : Modeler 内でドメインまたは識別ポータルを作成します。
- **データフロービューを表示** : データフロービューを起動します。
- **[Simulate (シミュレート)]** : スキーママップポリシーをテストします。詳細については、[82 ページのセクション 6.3「スキーママッピングポリシーのテスト」](#)を参照してください。
- **ファイルにエクスポート** : スキーママップポリシーを .xml ファイルとして保存します。
- **[ライブ] > [展開]** : アイデンティティポータルにスキーママップポリシーを展開します。
- **[ライブ] > [比較]** : Designer のスキーママップポリシーとアイデンティティポータルのスキーママップポリシーを比較します。
- **削除** : スキーママップポリシーを削除します。
- **プロパティ** : スキーママップポリシーの名前を変更できます。

6.5.2 [Policy Flow (ポリシーフロー)] ビューの追加オプション

- 1 [Policy Flow (ポリシーフロー)] ビューでスキーママップポリシーを右クリックします。



- **[Add Policy (ポリシーの追加)] > [DirXML スクリプト]** : DirXML[®] スクリプトを使用して、新しいスキーママップポリシーを追加します。

- ◆ [Add Policy (ポリシーの追加)] > [XSLT] : XSLT を使用して、新しいスキーママップポリシーを追加します。
- ◆ [Add Policy (ポリシーの追加)] > [スキーマのマッピング] : 情報を含まない新しいスキーママップポリシーを追加します。
- ◆ [Add Policy (ポリシーの追加)] > [Link to Existing (既存へのリンク)] : 既存のスキーママップポリシーを参照して選択し、現在のスキーママップポリシーにリンクできます。
- ◆ [Add Policy (ポリシーの追加)] > [Copy Existing (既存のコピー)] : 既存のスキーママップポリシーを参照して選択し、現在のスキーママップポリシーにコピーできます。
- ◆ [Edit Policy (ポリシーの編集)] > [スキーマのマッピング] : スキーママップエディタを起動します。詳細については、79 ページのセクション 6.2 「スキーママッピングポリシーの編集」を参照してください。
- ◆ [Delete All Set Policies (セット内のすべてのポリシーの削除)] : 選択したポリシーセット内のすべてのポリシーを削除します。
- ◆ [Remove All Set Policies (セット内のすべてのポリシーの除外)] : 選択したポリシーセットからすべてのポリシーを除外しますが、既存のポリシーの削除はしません。



- ◆ [ライブ] > [ドライバのインポート] : アイデンティティボールドから既存のドライバをインポートします。
- ◆ [ライブ] > [ドライバの展開] : アイデンティティボールドに既存のドライバを展開します。
- ◆ [ライブ操作] > [ドライバ環境設定] > [属性のインポート] : アイデンティティボールドから属性をインポートし、それを Designer 内の属性と比較できます。
- ◆ [ライブ] > [ドライバ環境設定] > [属性の展開] : Designer からアイデンティティボールドに属性を展開し、Designer 内の属性とアイデンティティボールド内の属性を比較します。
- ◆ [ライブ] > [ドライバステータス] : ドライバのステータスを表示します。
- ◆ [ライブ] > [ドライバの起動] : ドライバを起動します。
- ◆ [ライブ] > [ドライバの停止] : ドライバを停止します。

- ◆ [ライブ] > [ドライバの再起動] : ドライバを再起動します。
- ◆ [Simulate (シミュレート)] : スキーママップポリシーをテストします。詳細については、[82 ページのセクション 6.3 「スキーママッピングポリシーのテスト」](#)を参照してください。

6.5.3 [Policy Set (ポリシーセット)] ビューの追加オプション

- 1 [Policy Set (ポリシーセット)] ビューでスキーママップポリシーを右クリックします。



- ◆ **編集** : スキーママップエディタを起動します。詳細については、[79 ページのセクション 6.2 「スキーママッピングポリシーの編集」](#)を参照してください。
- ◆ **コピー** : スキーママップポリシーのコピーを作成します。
- ◆ **名前を付けて保存** : スキーママップポリシーを .xml ファイルとして保存します。
- ◆ **削除** : ポリシーセットからスキーママップポリシーを除外しますが、識別ボールドからそのポリシー自体は削除しません。
- ◆ **[Link to Existing Policy (既存のポリシーへのリンク)]** : 別のスキーママップポリシーを参照して選択し、既存のポリシーにリンクできます。
- ◆ **[上へ移動]** : ポリシーの実行順で、スキーママップポリシーを上に移動します。
- ◆ **[下に移動]** : ポリシーの実行順で、スキーママップポリシーを下に移動します。
- ◆ **ファイルにエクスポート** : スキーママップポリシーを .xml ファイルとして保存します。
- ◆ **[ライブ] > [展開]** : アイデンティティボールドにスキーママップポリシーを展開します。
- ◆ **[ライブ] > [比較]** : Designer のスキーママップポリシーとアイデンティティボールドのスキーママップポリシーを比較します。
- ◆ **削除** : スキーママップポリシーを削除します。
- ◆ **プロパティ** : スキーママップポリシーの名前を変更できます。
- ◆ **[Simulate (シミュレート)]** : スキーママップポリシーをテストします。詳細については、[82 ページのセクション 6.3 「スキーママッピングポリシーのテスト」](#)を参照してください。
- ◆ **データフロービューを表示** : データフロービューを起動します。

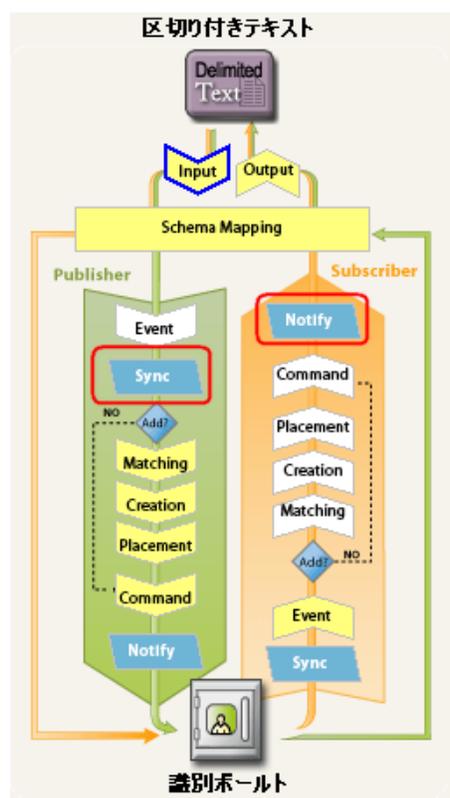
フィルタを使用したオブジェクトのフローの制御

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。

- ◆ 90 ページのセクション 7.1 「フィルタエディタへのアクセス方法」
- ◆ 93 ページのセクション 7.2 「フィルタの編集」
- ◆ 99 ページのセクション 7.3 「フィルタのテスト」
- ◆ 101 ページのセクション 7.4 「XML のフィルタを表示」
- ◆ 101 ページのセクション 7.5 「追加のフィルタオプション」

接続されたシステム間で情報が同期されると、接続されたシステムは変更を受け入れるか、または変更が行われたことを通知されます。Designer は、この情報を [ポリシーフロー] ビュー内に [同期] および [通知] フィルタとして表示します。

図 7-1 ポリシーフロービュー内のフィルタ



フィルタが [同期] に設定されている場合は、オブジェクト変更は自動的に接続されたシステムに同期されます。フィルタが [通知] に設定されている場合は、オブジェクト変更はメタディレクトリエンジンにレポートされますが、オブジェクトは自動的に同期されま

せん。詳細については、95 ページのセクション 7.2.5 「フィルタ設定の変更」を参照してください。

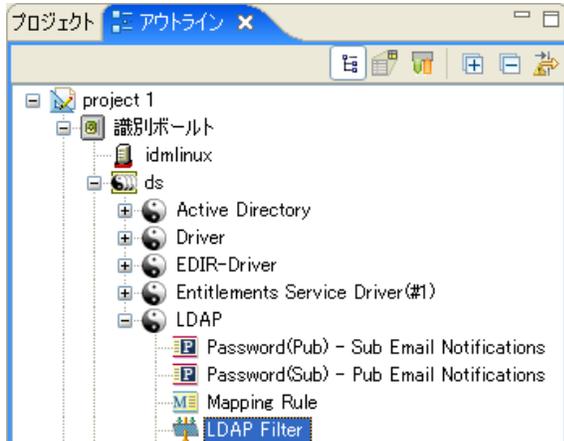
7.1 フィルタエディタへのアクセス方法

フィルタエディタにより、フィルタの編集ができます。フィルタエディタへのアクセス方法には、モデルアウトライン、ポリシーフローおよびポリシーセットビューのそれぞれを使用する3つの方法があります。

- ◆ 90 ページのセクション 7.1.1 「[Model Outline (モデルアウトライン)] ビュー」
- ◆ 90 ページのセクション 7.1.2 「[Policy Flow (ポリシーフロー)] ビュー」
- ◆ 91 ページのセクション 7.1.3 「ポリシーセットビュー」
- ◆ 92 ページのセクション 7.1.4 「キーボード操作」

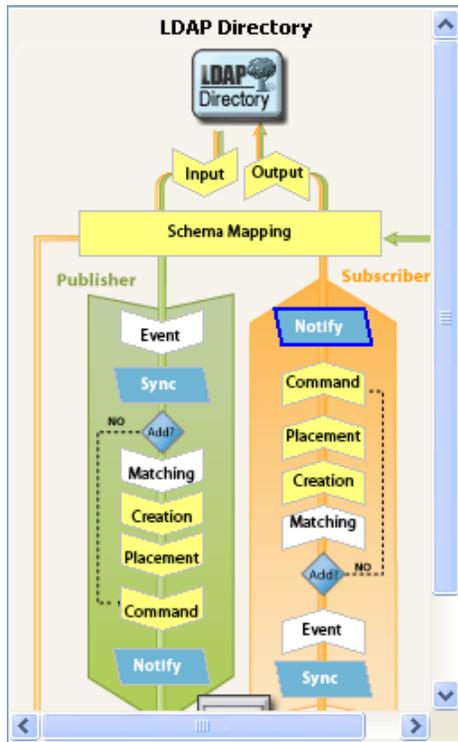
7.1.1 [Model Outline (モデルアウトライン)] ビュー

- 1 開いているプロジェクトで、[Outline (アウトライン)] タブをクリックします。
- 2 [モデルアウトラインの表示] アイコンをクリックします。
- 3 フィルタを管理するドライバを選択し、右側のプラス記号をクリックします。
- 4 [フィルタ] アイコンをダブルクリックして、フィルタエディタを起動します。
または
右クリックして、[編集] を選択します。



7.1.2 [Policy Flow (ポリシーフロー)] ビュー

- 1 開いているプロジェクトで、[Outline (アウトライン)] タブをクリックします。
- 2 [ポリシーフローの表示] アイコンを選択します。
- 3 [同期] アイコンまたは [通知] アイコンをダブルクリックして、フィルタエディタを起動します。

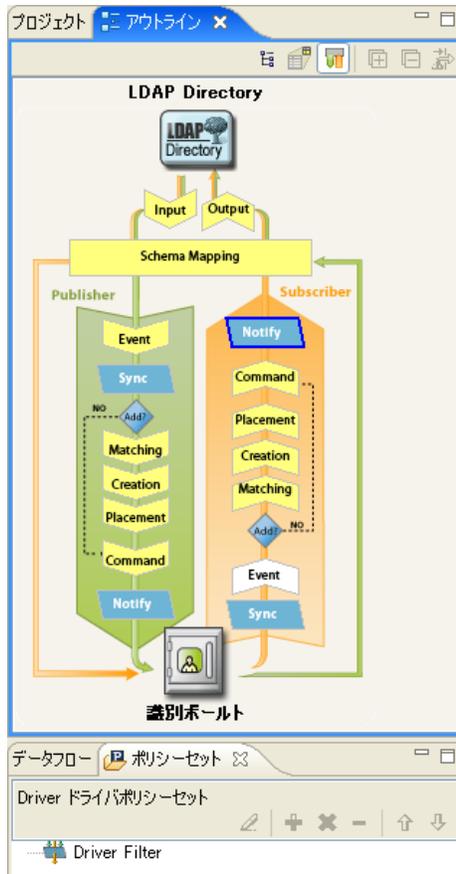


または

右クリックして、*[Edit Policy (ポリシーの編集)]* > *[フィルタ]* の順に選択します。

7.1.3 ポリシーセットビュー

- 1 [ポリシーセット] ビューでフィルタポリシーをダブルクリックします。



7.1.4 キーボード操作

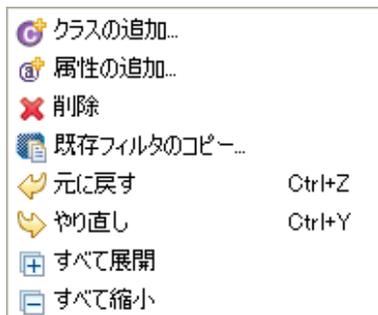
表 7-1 フィルタエディタでのキーボード操作

アクション	説明
< 上矢印 >	フィルタエディタ内でカーソルを上方向に移動します。
< 下矢印 >	フィルタエディタ内でカーソルを下方向に移動します。
< 左矢印 >	表示されている情報を縮小表示します
< 右矢印 >	表示されている情報を展開します。
挿入	クラスを追加します。
<Ctrl+Insert>	属性を追加します。
削除	選択された項目を削除します。
<Enter>	編集モードに切り替えます。変更を保存するには、<Enter> を 2 回押します。
Esc	編集モードを終了します。

7.2 フィルタの編集

フィルタエディタにより、フィルタの作成と編集ができます。コンテキストに応じたメニューを表示するには、項目を右クリックします。

図7-2 フィルタオプション



- ◆ 93 ページのセクション 7.2.1 「クラスと属性の削除または追加」
- ◆ 94 ページのセクション 7.2.2 「複数の属性の変更」
- ◆ 94 ページのセクション 7.2.3 「既存のフィルタのコピー」
- ◆ 94 ページのセクション 7.2.4 「属性のデフォルト値の設定」
- ◆ 95 ページのセクション 7.2.5 「フィルタ設定の変更」

7.2.1 クラスと属性の削除または追加

クラスと属性を削除または追加することで、接続されたデータストアとアイデンティティボールドとの間で同期するオブジェクトを指定できます。

クラスまたは属性の削除

クラスまたは属性を同期しないようにする場合は、クラスまたは属性をフィルタから完全に削除するのが最善の方法です。フィルタから属性とクラスを追加または削除するには、次の2つの方法があります。

- ◆ 削除するクラスまたは属性を右クリックし、[削除] を選択します。
- ◆ 削除するクラスまたは属性を選択し、右上隅の [削除] アイコン  をクリックします。

クラスの追加

- 1 フィルタエディタ内で右クリックし、[Add Classes (クラスの追加)] をクリックします。

または

右上隅の [クラスの追加] アイコン  をクリックします。

- 2 追加するクラスを参照して選択し、[OK] をクリックします。
- 3 情報を同期するオプションを変更します。
- 4 変更を保存するには、[ファイル] > [保存] の順にクリックします。

属性の追加

- 1 フィルタエディタ内で右クリックし、[属性の追加] をクリックします。
または
右上隅の [属性] アイコンをクリックします。
- 2 追加する属性を参照して選択し、[OK] をクリックします。
- 3 情報を同期するオプションを変更します。
- 4 変更を保存するには、[ファイル] > [保存] の順にクリックします。

7.2.2 複数の属性の変更

フィルタエディタでは、複数の属性を一度に変更できます。キーを押しながら複数の属性を選択します。オプションを変更すると、選択したすべての属性でそれが変更されます。

7.2.3 既存のフィルタのコピー

既存のフィルタを別のドライバからコピーし、現在作業しているドライバで使用できます。

- 1 [既存フィルタのコピー] アイコンをクリックするか、
または
フィルタエディタ内で右クリックし、[Copy an Existing Filter (既存のフィルタのコピー)] をクリックします。
- 2 コピーするフィルタオブジェクトを参照して選択し、[OK] をクリックします。
プロジェクトに複数の識別ボールドがある場合は、他の識別ボールドからフィルタをコピーできます。他のオブジェクトを参照して選択する場合は、他のアイデンティティボールドを参照し、そこに保存されているフィルタを使用できます。



7.2.4 属性のデフォルト値の設定

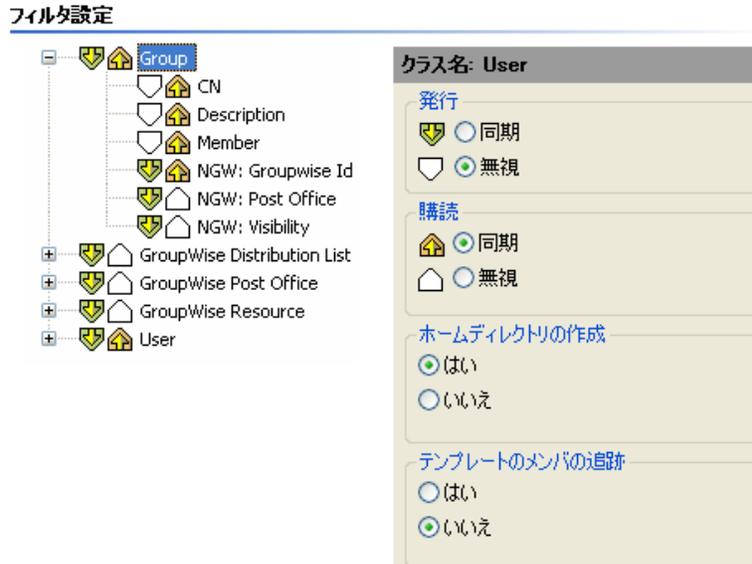
新しい属性をフィルタに追加する場合は、そのデフォルト値を定義できます。

- 1 右上隅の [新しい属性のデフォルト値の設定] アイコンをクリックします。
- 2 新しい属性に指定するオプションを選択し、[OK] をクリックします。

7.2.5 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。
ホームディレクトリの作成	<p>[ホームディレクトリの作成] を使用すると、eDirectory 内にユーザオブジェクト用のホームディレクトリを作成できます。このオプションは、eDirectory でのみ有効です。</p> <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の追跡	<ul style="list-style-type: none"> ◆ はい: テンプレートのメンバを追跡します。 ◆ いいえ: テンプレートのメンバを追跡しません。

オプション	定義
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。 <p>eDirectory Template オブジェクトを使用してユーザオブジェクトを作成すると、[テンプレートのメンバの追跡] オプションが選択されている場合、eDirectory ドライバは、Member of Template 属性を保持します。このオプションは、eDirectory でのみ有効です。</p>

3 属性を選択します。

フィルタ設定

The screenshot shows the 'Filter Settings' dialog box. On the left, a tree view lists attributes for the 'Group' class, with 'CN' selected. On the right, the configuration options for the selected attribute are displayed:

- クラス名:** Group
- 属性名:** CN
- 発行 (Issuance):**
 - 同期 (Synchronous)
 - 無視 (Ignore)
 - お知らせ君 (Notify Me)
 - リセット (Reset)
- 購読 (Subscription):**
 - 同期 (Synchronous)
 - 無視 (Ignore)
 - お知らせ君 (Notify Me)
 - リセット (Reset)
- マージ権限 (Merge Permissions):**
 - デフォルト (Default)
 - 識別ポールド (Identification Poloid)
 - アプリケーション (Application)
 - なし (None)
- 識別ポールドへの変更を最適化する (Optimize changes to the identification poloid):**
 - はい (Yes)
 - いいえ (No)

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)。 [リセット] オプションは、データストアを情報の信頼できるソースにします。たとえば、従業員アドレスが HR データベース内でのみ変更される場合は、この属性用のフィルタ内で [リセット] を設定します。電子メールシステム内でアドレスが変更されて HR データベースに送信されると、フィルタは情報を HR データベースから電子メールシステムに戻して従業員のアドレスは変更されません。
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)。 [リセット] オプションは、データストアを情報の信頼できるソースにします。たとえば、従業員アドレスが HR データベース内でのみ変更される場合は、この属性用のフィルタ内で [リセット] を設定します。電子メールシステム内でアドレスが変更されて HR データベースに送信されると、フィルタは情報を HR データベースから電子メールシステムに戻して従業員のアドレスは変更されません。

オプション	定義
マージ権限	<ul style="list-style-type: none"> ◆ 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。 ◆ 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。 ◆ 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。 ◆ なし：同期に関係なく、マージされません。
<i>Identity Manager</i> に対する変更の最適化	<ul style="list-style-type: none"> ◆ 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。 ◆ いいえ：変更を調べません。 操作が発行者チャンネルでの変更の場合は、メタディレクトリエンジンは識別ボールド内のオブジェクトの現在の状態を検証して、変更されている値にもを更新するように変更を変更します。たとえば、オブジェクトが a、b、c、の属性を持ち、発行者チャンネルがすべての既存の値を削除して a、b、d、および e を追加する変更イベントを受け取った場合、最適化プロセスは最小の変更として d を削除して e を追加します。 このオプションを使用すると、1,000 よりも多い値を持つ属性でのイベント処理に長い時間がかかります。

5 [保存] アイコン  をクリックして、変更内容を保存します。

7.3 フィルタのテスト

Designer には、ポリシーシミュレータと呼ばれるツールが付属しています。このツールを使用すると、運用環境に実装しなくてもポリシーをテストできます。ポリシーシミュレータをフィルタエディタから起動して、変更後のポリシーをテストできます。

- 1 ツールバーの [ポリシーシミュレータの起動] アイコンをクリックします。
- 2 [識別ボールドへ] または [識別ボールドから] をフィルタのシミュレーションポリシーとして選択します。
- 3 [Import (インポート)] を選択し、イベントをシミュレートするファイルを参照して選択します。
- 4 ファイルを選択して、[開く] をクリックします。この例では、`com.novell.designer.idm.policy.simulation\add\User.xml` ファイルを使用して、ユーザオブジェクトの追加イベントをシミュレートします。
ポリシーシミュレータに、ユーザの「追加」イベントの入力ドキュメントが表示されます。
- 5 [次へ] をクリックして、シミュレーションを開始します。

入力ドキュメントのカスタマイズ

インデントポリシーへのポリシー、またはインデントポリシーからのポリシーをシミュレートするかどうかを選択します。入力ドキュメントをカスタマイズして、オペレーションデータを追加し、または既存の値を変更することも



シミュレーションポイント: 識別ボールドへ

入力ドキュメント:

```
<?xml version="1.0" encoding="UTF-8"?><nds d
<!-- =====
      Input document to add a User.
      =====
<input>
  <add class-name="User" qualified-src
    <association>a=dirXML Test#ou=U
    <add-attr attr-name="cn">
      <value>User1</value>
```

インポート(I)... 名前を付けて保存(S)...

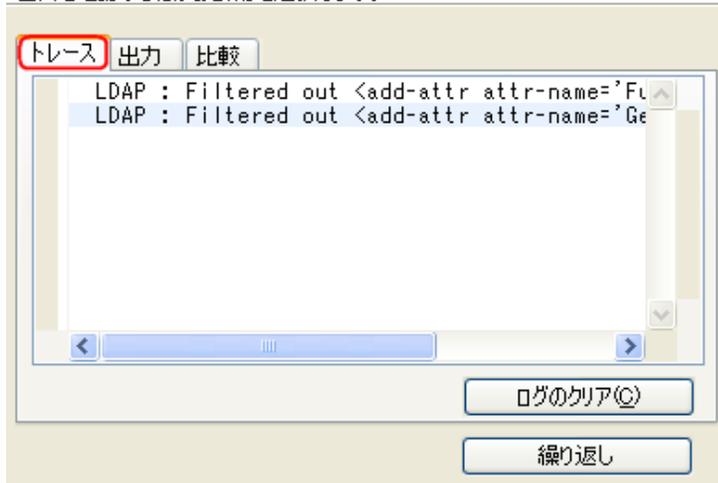
? <戻る(B) 次へ(N) > 終了(F) キャンセル

ポリシーシミュレータに、追加イベントのログ、出力ドキュメント、および入力ドキュメントと生成された出力ドキュメントの比較が表示されます。

- 6 [トレース] タブを選択して、DSTRACE に表示されるとおりの追加イベントの結果を表意 j します。

変換結果の表示

シミュレーションの詳細を確認するには[トレース]を、変換されたドキュメントを確認するには[出力]を、入力ドキュメントと出力ドキュメントの差異を確認するには[比較]を選択します。



[ログのクリア] をクリックし、[繰り返し] をクリックして、新しいトレースログでシミュレーションを再び実行します。

- 7 [出力] タブを選択し、入力ドキュメントに対してフィルタが実行されたときに生成される出力ドキュメントを表示します。入力ドキュメントはユーザの「追加」イベントです。

変換結果の表示

シミュレーションの詳細を確認するには[トレース]を、変換されたドキュメントを確認するには[出力]を、入力ドキュメントと出力ドキュメントの差異を確認するには[比較]を選択します。

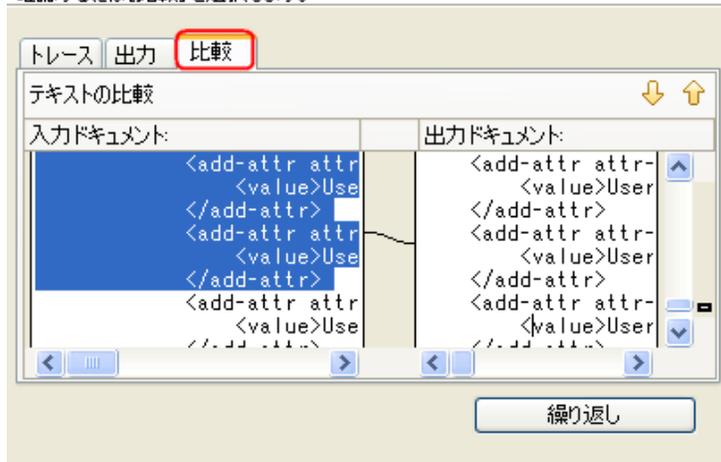


入力ドキュメントおよび出力ドキュメントを編集できます。変更を保持する場合は、[名前を付けて保存] をクリックします。

- 8 [Compare (比較)] タブを選択して、入力ドキュメントのテキストと、生成された出力ドキュメントを比較します。

変換結果の表示

シミュレーションの詳細を確認するには[トレース]を、変換されたドキュメントを確認するには[出力]を、入力ドキュメントと出力ドキュメントの差異を確認するには[比較]を選択します。



- 9 [Repeat (繰り返し)] をクリックして、別の入力ドキュメントを選択し、イベントの結果を表示します。
- 10 フィルタのテストが済んだら、[終了] をクリックして、ポリシーシミュレータを閉じます。

7.4 XML のフィルタを表示

Designer では、XML エディタを使用して、XML を表示、編集、および検証できます。[XML ソース] タブまたは [XML ツリー] タブをクリックして XML エディタにアクセスします。XML エディタの詳細については、『*Designer 2.1 for Identity Manager 3.5.1*』の「**Novell XML エディタ**」を参照してください。

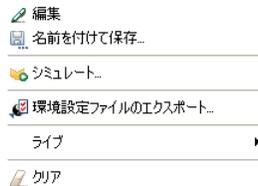
7.5 追加のフィルタオプション

フィルタオブジェクトを右クリックすると、[アウトライン] ビュー、[ポリシーフロー] ビュー、および [ポリシーセット] ビューで、複数のオプションが表示されます。

- ◆ 101 ページのセクション 7.5.1 「[Outline (アウトライン)] ビューの追加オプション」
- ◆ 102 ページのセクション 7.5.2 「[Policy Flow (ポリシーフロー)] ビューの追加オプション」
- ◆ 102 ページのセクション 7.5.3 「[Policy Set (ポリシーセット)] ビューの追加オプション」

7.5.1 [Outline (アウトライン)] ビューの追加オプション

- 1 [Outline (アウトライン)] ビューで、フィルタオブジェクトを右クリックします。



- ◆ **編集** : フィルタエディタを起動します。詳細については、[93 ページのセクション 7.2 「フィルタの編集」](#)を参照してください。
- ◆ **名前を付けて保存** : フィルタを .xml ファイルとして保存します。
- ◆ **新規** : Modeler 内でドメインまたは識別ポールドを作成します。
- ◆ **データフロービューを表示** : データフロービューを起動します。
- ◆ **[Simulate (シミュレート)]** : ポリシーシミュレータを起動します。詳細については、[99 ページのセクション 7.3 「フィルタのテスト」](#)を参照してください。
- ◆ **[ライブ] > [展開]** : アイデンティティポールドにフィルタを展開します。
- ◆ **クリア** : フィルタポリシーからすべての内容を削除します。ただしオブジェクトは残します。

7.5.2 [Policy Flow (ポリシーフロー)] ビューの追加オプション

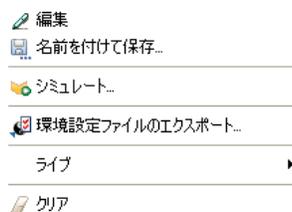
- 1 [Policy Flow (ポリシーフロー)] ビューで、フィルタオブジェクトを右クリックします。



- ◆ **[Edit Policy (ポリシーの編集)] > [フィルタ]** : フィルタエディタを起動します。詳細については、[93 ページのセクション 7.2 「フィルタの編集」](#)を参照してください。
- ◆ **[Simulate (シミュレート)]** : ポリシーシミュレータを起動します。詳細については、[99 ページのセクション 7.3 「フィルタのテスト」](#)を参照してください。

7.5.3 [Policy Set (ポリシーセット)] ビューの追加オプション

- 1 [Policy Set (ポリシーセット)] ビューで、フィルタオブジェクトを右クリックします。



- ◆ **編集** : フィルタエディタを起動します。詳細については、[93 ページのセクション 7.2 「フィルタの編集」](#)を参照してください。

- ◆ **名前を付けて保存** : フィルタを .Xml ファイルとして保存します。
- ◆ **[ライブ] > [展開]** : アイデンティティボールドにフィルタを展開できます。
- ◆ **クリア** : フィルタポリシーからすべての内容を削除します。ただしオブジェクトは残します。
- ◆ **[Simulate (シミュレート)]** : ポリシーシミュレータを起動します。詳細については、[99 ページのセクション 7.3 「フィルタのテスト」](#)を参照してください。
- ◆ **データフロービューを表示** : データフロービューを起動します。

事前定義されたルールの使用

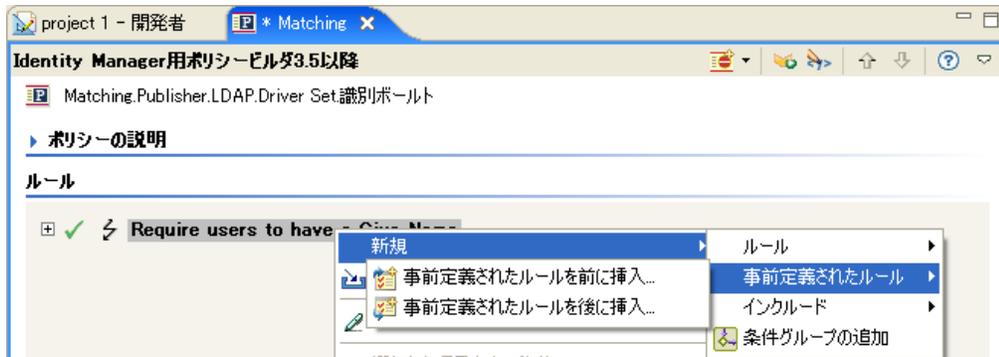
8

Designer 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。

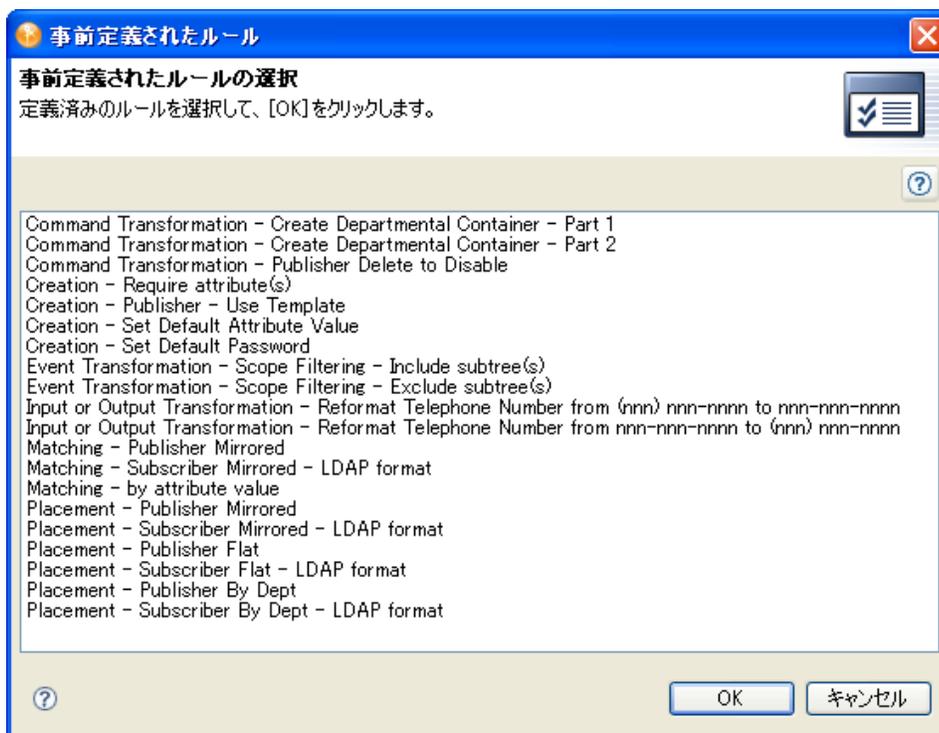
- ◆ 106 ページのセクション 8.1 「コマンド変換 - 部署別コンテナの作成 - パート 1 とパート 2」
- ◆ 109 ページのセクション 8.2 「コマンド変換 - 無効にする発行者の削除」
- ◆ 110 ページのセクション 8.3 「作成 - 属性が必要」
- ◆ 112 ページのセクション 8.4 「作成 - 発行者 - テンプレートの使用」
- ◆ 114 ページのセクション 8.5 「作成 - デフォルト属性値の設定」
- ◆ 115 ページのセクション 8.6 「作成 - デフォルトパスワードの設定」
- ◆ 117 ページのセクション 8.7 「イベント変換 - スコープフィルタリング - サブツリーの組み込み」
- ◆ 119 ページのセクション 8.8 「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」
- ◆ 120 ページのセクション 8.9 「入出力変換 - 電話番号の形式を (nnn) nnn-nnnn から nnn-~~nnn~~-nnnn に変更」
- ◆ 122 ページのセクション 8.10 「入出力変換 - 電話番号の形式を nnn-~~nnn~~-nnnn から (nnn) nnn-~~nnn~~ に変更」
- ◆ 124 ページのセクション 8.11 「一致 - 発行者 (ミラーリング)」
- ◆ 126 ページのセクション 8.12 「一致 - 購読者 (ミラーリング) - LDAP 形式」
- ◆ 128 ページのセクション 8.13 「一致 - 属性値別」
- ◆ 129 ページのセクション 8.14 「配置 - 発行者 (ミラーリング)」
- ◆ 131 ページのセクション 8.15 「配置 - 購読者 (ミラーリング) - LDAP 形式」
- ◆ 133 ページのセクション 8.16 「配置 - 発行者 (フラット)」
- ◆ 135 ページのセクション 8.17 「配置 - 購読者 (フラット) - LDAP 形式」
- ◆ 137 ページのセクション 8.18 「配置 - 部署別発行者」
- ◆ 139 ページのセクション 8.19 「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

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



[事前定義されたルール] ダイアログボックスに、使用可能なルールの一覧が表示されます。



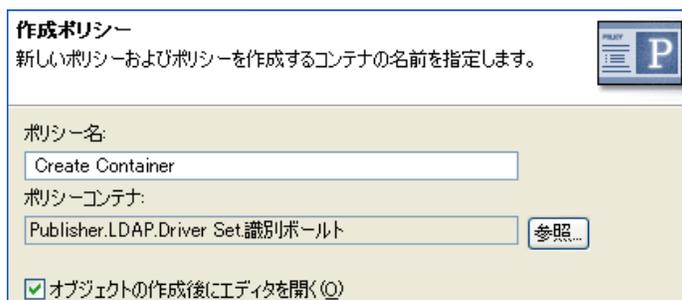
8.1 コマンド変換 - 部署別コンテナの作成 - パート 1 とパート 2

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。ルールは、発行者と購読者のどちらのチャンネルにも、また両方のチャンネルにも設定できます。

この事前定義されたルールを使用する手順には、コマンド変換ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの2つがあります。このルールに追加するコマンド変換ポリシーがすでにある場合は、107 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。

8.1.1 ポリシーの作成

- 1 [Outline (アウトライン)] ビューまたは [Policy Flow (ポリシーフロー)] ビューから発行者チャンネルまたは購読者チャンネルを選択します。
- 2 [ポリシーセット] ビューでコマンド変換ポリシーセットを選択し、[ポリシーセットに新しいポリシーを作成または追加します] アイコン  をクリックして、新しいポリシーを作成します。
- 3 [新しいポリシーの作成] をクリックし、[次へ] をクリックします。
- 4 ポリシーに名前を付けます。
- 5 デフォルトの場所を使用するか、または別の場所を参照して使用して、ドライバにポリシーを配置します。



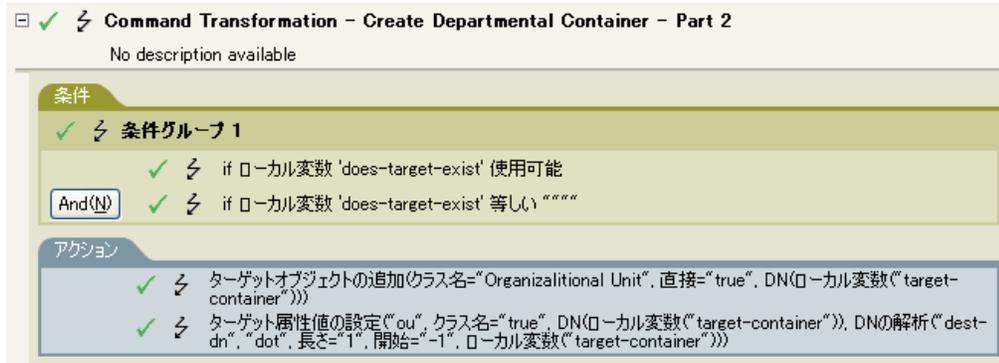
- 6 [Open Editor after creating policy (ポリシーの作成後にエディタを開く)] を選択し、[次へ] をクリックします。
- 7 ポリシーのタイプとして、[DirXML スクリプト] を選択し、[終了] をクリックします。
- 8 ファイルの矛盾ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue?」というメッセージが表示されたら、[はい] をクリックします。ポリシービルダが起動され、新しいコマンド変換ポリシーが保存されます。
- 9 [107 ページのセクション 8.1.2 「事前定義されたルールのインポート」](#) に進みます。

8.1.2 事前定義されたルールのインポート

- 1 ポリシービルダ内で、右クリックして [新規作成] > [事前定義されたルール] > [事前定義されたルールを前に挿入] または [事前定義されたルールを後に挿入] の順に選択します。
- 2 [コマンド変換 - 部署別のコンテナの作成 - パート 1] を選択し、[OK] をクリックします。>



- 3 ポリシービルダ内で右クリックして、[新規作成] > [事前定義されたルール] > [事前定義されたルールを前に挿入] または [事前定義されたルールを後に挿入] 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、[OK] をクリックします。 >



- 5 [ファイル]> [保存] の順にクリックして、ルールを保存します。

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。

8.1.3 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」イベントが想定されます。「追加」イベントが発生すると、2つのローカル変数が設定されます。最初のローカル変数は、`target-container` という名前になります。`target-container` の値が、ターゲット DN に設定されます。2つ目のローカル変数は、`does-target-exist` という名前になります。`does-target-exist` の値は、`objectclass` のターゲット属性値に設定されます。クラスは `OrganizationalUnit` に設定されます。`OrganizationalUnit` の DN は、ローカル変数 `target-container` に設定されます。



パート 2 では、ローカル変数 `does-target-exist` が使用可能かどうかを確認されます。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を確認されます。値が空白である場合、部門オブジェクトが作成されます。部門の DN は、ローカル変数 `target-container` の値に設定されます。また、OU 属性の値も追加されます。OU 属性の値

は、ローカル変数 `target-container` に設定されます。これは、ソース形式をターゲット DN として使用します。ターゲット形式はドット形式です。

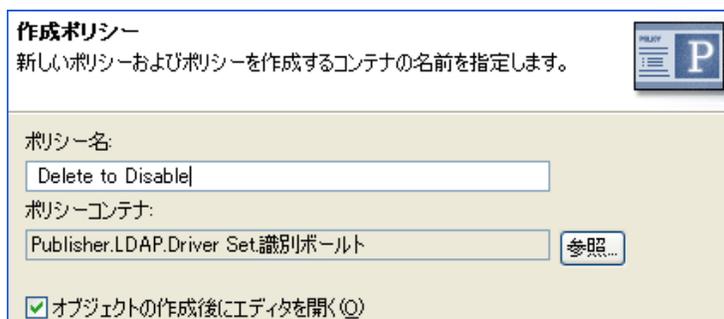
8.2 コマンド変換 - 無効にする発行者の削除

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

この事前定義されたルールを使用する手順には、コマンド変換ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの2つがあります。このルールに追加するコマンド変換ポリシーがすでにある場合は、[110 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。

8.2.1 ポリシーの作成

- 1 **[Outline (アウトライン)]** ビューまたは **[Policy Flow (ポリシーフロー)]** ビューから発行者チャンネルを選択します。
- 2 **[ポリシーセット]** ビューでコマンド変換ポリシーセットを選択し、**[ポリシーセットに新しいポリシーを作成または追加します]** アイコン  をクリックして、新しいポリシーを作成します。
- 3 **[新しいポリシーの作成]** をクリックし、**[次へ]** をクリックします。
- 4 ポリシーに名前を付けます。
- 5 デフォルトの場所を使用するか、または別の場所を参照して使用して、ドライバにポリシーを配置します。



- 6 **[Open Editor after creating policy (ポリシーの作成後にエディタを開く)]** を選択し、**[次へ]** をクリックします。
- 7 ポリシーのタイプとして、**[DirXML スクリプト]** を選択し、**[終了]** をクリックします。
- 8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか?)」というメッセージが表示されたら、**[はい]** をクリックします。ポリシービルダが起動され、新しいコマンド変換ポリシーが保存されます。
- 9 [110 ページのセクション 8.2.2 「事前定義されたルールのインポート」](#) に進みます。

8.2.2 事前定義されたルールのインポート

- 1 ポリシービルダ内で、右クリックして [New (新規作成)] > [Predefined Rule (事前定義されたルール)] > [Insert Predefined Rule Before (事前定義されたルールを前に挿入)] または [Insert Predefined Rule After (事前定義されたルールを後に挿入)] の順に選択します。>
- 2 [コマンド変換 - 無効にする発行者の削除] を選択し、[OK] をクリックします。>



- 3 [ファイル] > [保存] の順にクリックして、ルールを保存します。

このルールには、環境に応じて変更すべき情報はありません。

8.2.3 ルールの動作

このルールは、接続データストアで「削除」イベントが発生したときに使用されます。ユーザオブジェクトは識別ポルトで削除される代わりに、無効になります。ユーザオブジェクトに対して削除イベントが発生するときはいつでも、「ログインの無効化」のターゲット属性値が True に設定され、ユーザオブジェクトから関連付けが削除されます。ユーザオブジェクトは、Novell® eDirectory™ ツリーへはログインできなくなりますが、ユーザオブジェクトは削除されません。

8.3 作成 - 属性が必要

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

この事前定義されたルールを使用するには、作成ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの2つの手順を実行します。このルールに追加する作成ポリシーがすでにある場合は、[111 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。

8.3.1 ポリシーの作成

- 1 [アウトライン] ビューまたは [ポリシーフロー] ビューから発行者チャンネルまたは購読者チャンネルを選択します。
- 2 [ポリシーセット] ビューで作成ポリシーセットを選択し、[ポリシーセットに新しいポリシーを作成または追加します] アイコン  をクリックして、新しいポリシーを作成します。

- 3 [新しいポリシーの作成] をクリックし、[次へ] をクリックします。
- 4 ポリシーに名前を付けます。
- 5 デフォルトの場所を使用するか、または別の場所を参照して使用して、ドライバにポリシーを配置します。

- 6 [Open Editor after creating policy (ポリシーの作成後にエディタを開く)] を選択し、[次へ] をクリックします。
- 7 ポリシーのタイプとして、[DirXML スクリプト] を選択し、[終了] をクリックします。
- 8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか?)」というメッセージが表示されたら、[はい] をクリックします。ポリシービルダが起動され、新しい作成ポリシーが保存されます。
- 9 111 ページのセクション 8.3.2 「事前定義されたルールのインポート」に進みます。

8.3.2 事前定義されたルールのインポート

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

- 3 [アクション] タブをダブルクリックし、アクションを編集します。
- 4 [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 作成するユーザオブジェクトに必要な属性を参照して選択し、[OK] をクリックします。

- 6 [OK] をクリックします。
- 7 [ファイル]> [保存] の順に選択して、ルールを保存します。

8.3.3 ルールの動作

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

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

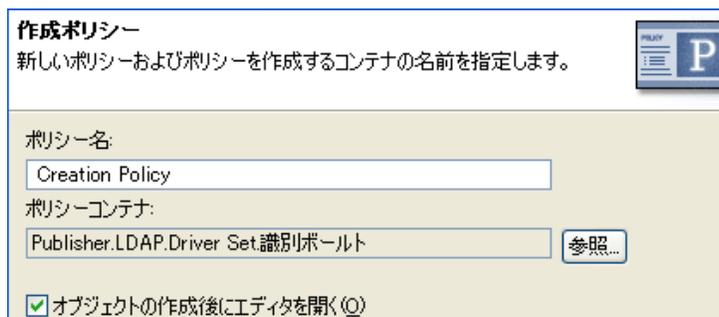
8.4 作成 - 発行者 - テンプレートの使用

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します。このルールは、ドライバ内の発行者作成ポリシーに実装します。このルールは、発行者チャ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、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、[113 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。

8.4.1 ポリシーの作成

- 1 [アウトライン] ビューまたは [ポリシーフロー] ビューから、発行者チャンネルを選択します。
- 2 [ポリシーセット] ビューで作成ポリシーセットを選択し、[ポリシーセットに新しいポリシーを作成または追加します] アイコン  をクリックして、新しいポリシーを作成します。
- 3 [新しいポリシーの作成] をクリックし、[次へ] をクリックします。
- 4 ポリシーに名前を付けます。
- 5 デフォルトの場所を使用するか、または別の場所を参照して使用して、ドライバにポリシーを配置します。



作成ポリシー
新しいポリシーおよびポリシーを作成するコンテナの名前を指定します。

ポリシー名:
Creation Policy

ポリシーコンテナ:
Publisher.LDAP.Driver Set 識別ポールド

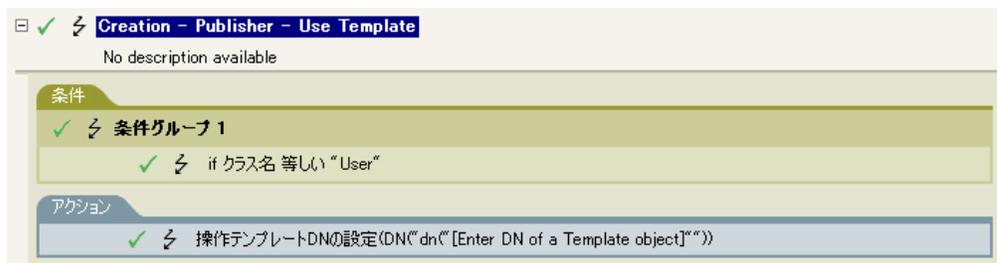
オブジェクトの作成後にエディタを開く(O)

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

- 7 ポリシーのタイプとして、[DirXML スクリプト] を選択し、[終了] をクリックします。
- 8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか?)」というメッセージが表示されたら、[はい] をクリックします。ポリシービルダが起動され、新しい作成ポリシーが保存されます。
- 9 113 ページのセクション 8.4.2 「事前定義されたルールのインポート」に進みます。

8.4.2 事前定義されたルールのインポート

- 1 ポリシービルダ内で、右クリックして [新規作成] > [事前定義されたルール] > [事前定義されたルールを前に挿入] または [事前定義されたルールを後に挿入] の順に選択します。
- 2 [作成 - 発行者 - テンプレートの使用] を選択し、[OK] をクリックします。>



- 3 [アクション] タブをダブルクリックし、アクションを編集します。
- 4 [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 [Noun (名詞)] リストの [テキスト] を選択します。
- 7 [テキスト] をダブルクリックして、引数に追加します。
- 8 エディタで、参照アイコンをクリックしてテンプレートオブジェクトを参照して選択し、[OK] をクリックします。
- 9 [OK] をクリックします。
- 10 [ファイル]> [保存] の順にクリックして、ルールを保存します。

8.4.3 ルールの動作

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

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

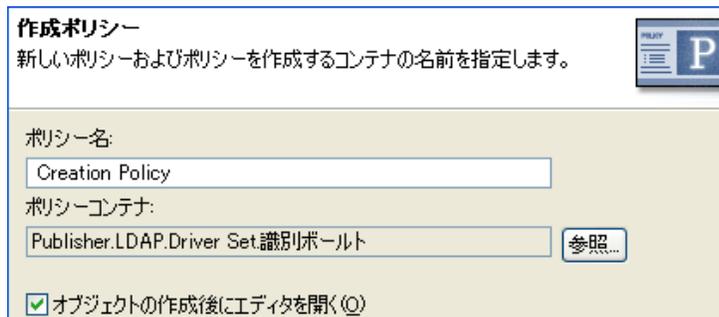
8.5 作成 - デフォルト属性値の設定

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、[115 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。

8.5.1 ポリシーの作成

- 1 [アウトライン] ビューまたは [ポリシーフロー] ビューから発行者チャンネルまたは購読者チャンネルを選択します。
- 2 [ポリシーセット] ビューで作成ポリシーセットを選択し、[ポリシーセットに新しいポリシーを作成または追加します] アイコン  をクリックして、新しいポリシーを作成します。
- 3 [新しいポリシーの作成] をクリックし、[次へ] をクリックします。
- 4 ポリシーに名前を付けます。
- 5 デフォルトの場所を使用するか、または別の場所を参照して使用して、ドライバにポリシーを配置します。



作成ポリシー
新しいポリシーおよびポリシーを作成するコンテナの名前を指定します。

ポリシー名:
Creation Policy

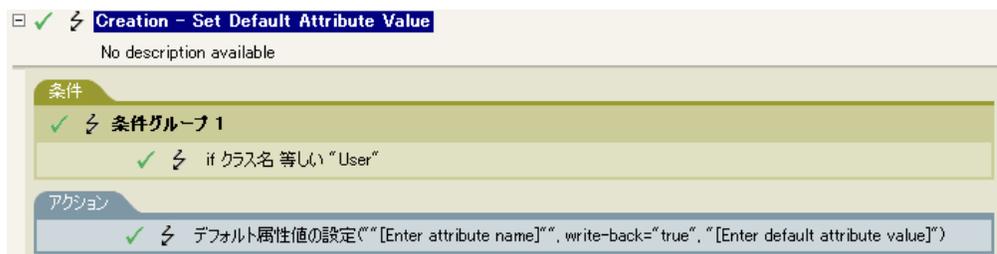
ポリシーコンテナ:
Publisher.LDAP.Driver Set.識別ポールド

オブジェクトの作成後にエディタを開く(O)

- 6 [Open Editor after creating policy (ポリシーの作成後にエディタを開く)] を選択し、[次へ] をクリックします。
- 7 ポリシーのタイプとして、[DirXML スクリプト] を選択し、[終了] をクリックします。
- 8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか?)」というメッセージが表示されたら、[はい] をクリックします。ポリシービルダが起動され、新しい作成ポリシーが保存されます。
- 9 [115 ページのセクション 8.5.2 「事前定義されたルールのインポート」](#)に進みます。

8.5.2 事前定義されたルールのインポート

- 1 ポリシービルダ内で、右クリックして [新規作成] > [事前定義されたルール] > [事前定義されたルールを前に挿入] または [事前定義されたルールを後に挿入] の順に選択します。
- 2 [作成 - デフォルト属性値の設定] を選択し、[OK] をクリックします。



- 3 [アクション] タブをダブルクリックし、アクションを編集します。
- 4 [属性名を入力してください] を [属性名の入力] フィールドから削除します。
- 5 参照アイコンをクリックして、作成する属性を参照して選択します。
- 6 [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、属性の値を作成し、[OK] をクリックします。
- 11 [OK] をクリックします。
- 12 [ファイル]> [保存] の順にクリックして、ルールを保存します。

8.5.3 ルールの動作

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

複数の属性値を定義する場合は、アクションを右クリックして [新規作成] > [アクションの追加] の順にクリックします。アクションを選択し、デフォルト属性値を設定し、[115 ページのステップ 1](#) から [115 ページのステップ 12](#) の手順を実行して属性に値を割り当てます。

8.6 作成 - デフォルトパスワードの設定

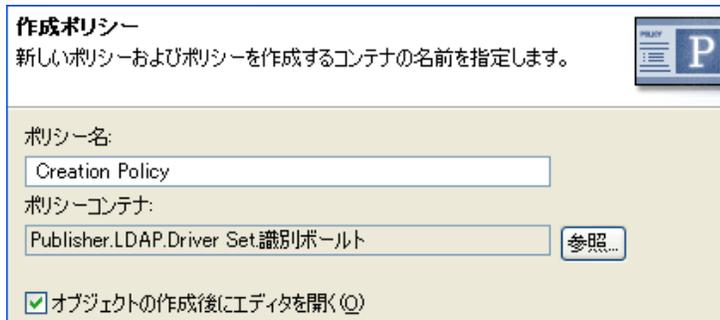
ユーザオブジェクトの作成中に、ルールによってユーザオブジェクトのデフォルトパスワードが設定されます。このルールは、ドライバ内の作成ポリシーに実装します。ルールは、発行者と購読者のどちらのチャンネルにも、また両方のチャンネルにも設定できます。

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

する作成ポリシーがすでにある場合は、116 ページの「事前定義されたルールインポート」に進みます。

8.6.1 ポリシーの作成

- 1 [アウトライン] ビューまたは [ポリシーフロー] ビューから発行者または購読者チャンネルを選択します。
- 2 [ポリシーセット] ビューで作成ポリシーセットを選択し、[ポリシーセットに新しいポリシーを作成または追加します] アイコン  をクリックして新しいポリシーを作成します。
- 3 [新しいポリシーの作成] をクリックし、[次へ] をクリックします。
- 4 ポリシーに名前を付けます。
- 5 デフォルトの場所を使用するか、または別の場所を参照して使用して、ドライバにポリシーを配置します。



作成ポリシー

新しいポリシーおよびポリシーを作成するコンテナの名前を指定します。

ポリシー名:
Creation Policy

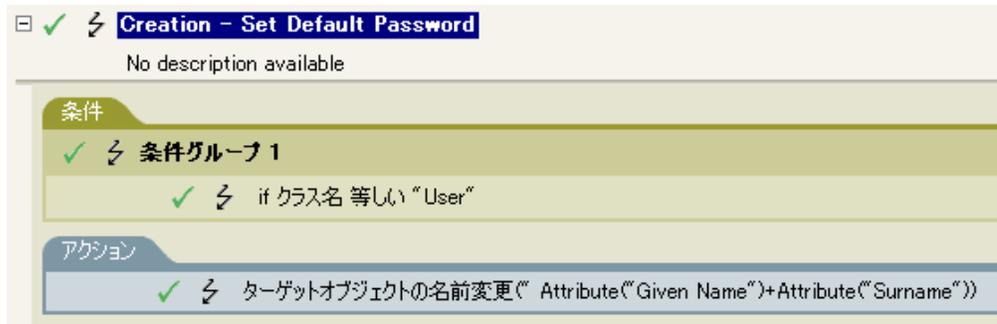
ポリシーコンテナ:
Publisher.LDAP.Driver Set 識別ポールド

オブジェクトの作成後にエディタを開く(O)

- 6 [Open Editor after creating policy (ポリシーの作成後にエディタを開く)] を選択し、[次へ] をクリックします。
- 7 ポリシーのタイプとして、[DirXML スクリプト] を選択し、[終了] をクリックします。
- 8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか?)」というメッセージが表示されたら、[はい] をクリックします。ポリシービルダが起動され、新しい作成ポリシーが保存されます。
- 9 116 ページのセクション 8.6.2 「事前定義されたルールインポート」に進みます。

8.6.2 事前定義されたルールインポート

- 1 ポリシービルダ内で、右クリックして [新規作成] > [事前定義されたルール] > [事前定義されたルールを前に挿入] または [事前定義されたルールを後に挿入] の順に選択します。
- 2 [作成 - デフォルトパスワードの設定] を選択し、[OK] をクリックします。



3 [ファイル]> [保存] の順にクリックして、ルールを保存します。

このルールには、環境に応じて変更すべき情報はありません。

8.6.3 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。

8.7 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

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

この事前定義されたルールを使用するには、イベント変換ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの2つの手順を実行します。このルールに追加するイベント変換ポリシーがすでにある場合は、[\(118 ページ\) 事前定義されたルールのインポート](#)へ進みます。

8.7.1 ポリシーの作成

- 1 [Outline (アウトライン)] ビューまたは [Policy Flow (ポリシーフロー)] ビューから発行者チャンネルまたは購読者チャンネルを選択します。
- 2 [ポリシーセット] ビューでイベント変換ポリシーセットを選択し、[ポリシーセットに新しいポリシーを作成または追加します] アイコン  をクリックして新しいポリシーを作成します。
- 3 [新しいポリシーの作成] をクリックし、[次へ] をクリックします。
- 4 ポリシーに名前を付けます。
- 5 デフォルトの場所を使用するか、または別の場所を参照して使用して、ドライバにポリシーを配置します。

作成ポリシー
新しいポリシーおよびポリシーを作成するコンテナの名前を指定します。

ポリシー名:
Event Transformation

ポリシーコンテナ:
Publisher.LDAP.Driver Set 識別ポールド

オブジェクトの作成後にエディタを開く(O)

- 6 [Open Editor after creating policy (ポリシーの作成後にエディタを開く)] を選択し、[次へ] をクリックします。
- 7 ポリシーのタイプとして、[DirXML スクリプト] を選択し、[終了] をクリックします。
- 8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか?)」というメッセージが表示されたら、[はい] をクリックします。ポリシービルダが起動され、新しいイベント変換ポリシーが保存されます。
- 9 118 ページのセクション 8.7.2 「事前定義されたルールのインポート」に進みます。

8.7.2 事前定義されたルールのインポート

- 1 ポリシービルダ内で、右クリックして [New (新規作成)] > [Predefined Rule (事前定義されたルール)] > [Insert Predefined Rule Before (事前定義されたルールを前に挿入)] または [Insert Predefined Rule After (事前定義されたルールを後に挿入)] の順にクリックします。>
- 2 [イベント変換 - スコープフィルタリング - サブツリーの組み込み] を選択し、[OK] をクリックします。

Event Transformation - Scope Filtering - Include subtree(s)
No description available

条件

条件グループ 1

if ソースDN サブツリー内になし "[Enter a subtree to include]"

アクション

拒否0

- 3 [条件] タブをダブルクリックし、条件を編集します。
- 4 [組み込むサブツリーを入力してください] を [値] 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識別ポールドを参照し、イベントを同期させるツリーの部分を選択し、[OK] をクリックします。
- 6 [OK] をクリックします。
- 7 [ファイル] > [保存] の順にクリックして、ルールを保存します。

8.7.3 ルールの動作

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

8.8 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、ドライブ内のイベント変換ポリシーに実装します。ルールは、発行者と購読者のどちらのチャンネルにも、また両方のチャンネルにも設定できます。

この事前定義されたルールを使用するには、イベント変換ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの2つの手順を実行します。このルールに追加するイベント変換ポリシーがすでにある場合は、[120 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。

8.8.1 ポリシーの作成

- 1 [アウトライン] ビューまたは [ポリシーフロー] ビューから発行者チャンネルまたは購読者チャンネルを選択します。
- 2 [ポリシーセット] ビューでイベント変換ポリシーセットを選択し、[ポリシーセットに新しいポリシーを作成または追加します] アイコン  をクリックして、新しいポリシーを作成します。
- 3 [新しいポリシーの作成] をクリックし、[次へ] をクリックします。
- 4 ポリシーに名前を付けます。
- 5 デフォルトの場所を使用するか、または別の場所を参照して使用して、ドライブにポリシーを配置します。

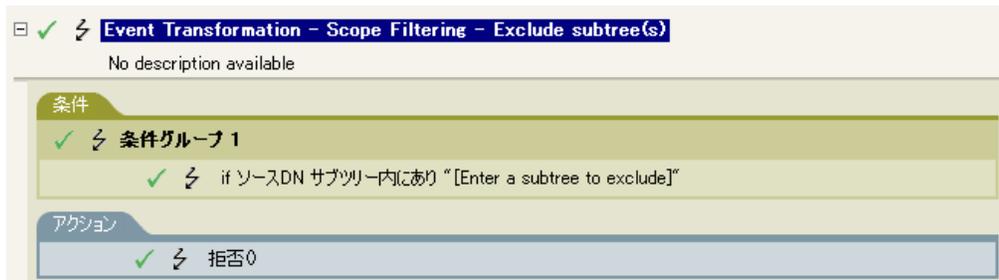
- 6 [Open Editor after creating policy (ポリシーの作成後にエディタを開く)] を選択し、[次へ] をクリックします。
- 7 ポリシーのタイプとして、[DirXML スクリプト] を選択し、[終了] をクリックします。
- 8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります)」

ます。エディタの変更内容を保存して続行しますか?)」というメッセージが表示されたら、[はい] をクリックします。ポリシービルダが起動され、新しいイベント変換ポリシーが保存されます。

9 120 ページのセクション 8.8.2 「事前定義されたルールのインポート」に進みます。

8.8.2 事前定義されたルールのインポート

- 1 ポリシービルダを右クリックし、[New (新規作成)] > [Predefined Rule (事前定義されたルール)] の順にクリックします。
- 2 [イベント変換 - スコープフィルタリング - サブツリーの除外] を選択し、[OK] をクリックします。



- 3 [条件] タブをダブルクリックし、条件を編集します。
- 4 [除外するサブツリーを入力してください] を [値] 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アイデンティティボールドを参照し、イベントを同期から除外するツリーの部分を選択し、[OK] をクリックします。
- 6 [OK] をクリックします。
- 7 [ファイル] > [保存] の順にクリックして、ルールを保存します。

8.8.3 ルールの動作

このルールは、識別ボールドの一部を同期から除外する場合に使用されます。これによって、フィルタを使用しなくても、同期するオブジェクトと同期しないオブジェクトを区別できます。識別ボ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す。

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

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

この事前定義されたルールを使用するには、入出力変換ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの 2 つの手順を実行します。このルールに追加する入出力変換ポリシーがすでにある場合は、121 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。

8.9.1 ポリシーの作成

- 1 [アウトライン] ビューまたは [ポリシーフロー] ビューから発行者チャンネルまたは購読者チャンネルを選択します。
- 2 [ポリシーセット] ビューで入出力変換ポリシーセットを選択し、[ポリシーセットに新しいポリシーを作成または追加します] アイコン  をクリックして、新しいポリシーを作成します。
- 3 [新しいポリシーの作成] をクリックし、[次へ] をクリックします。
- 4 ポリシーに名前を付けます。
- 5 デフォルトの場所を使用するか、または別の場所を参照して使用して、ドライバにポリシーを配置します。

作成ポリシー

新しいポリシーおよびポリシーを作成するコンテナの名前を指定します。



ポリシー名:

ポリシーコンテナ:

オブジェクトの作成後にエディタを開く (O)

- 6 [Open Editor after creating policy (ポリシーの作成後にエディタを開く)] を選択し、[次へ] をクリックします。
- 7 ポリシーのタイプとして、[DirXML スクリプト] を選択し、[終了] をクリックします。
- 8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか?)」というメッセージが表示されたら、[はい] をクリックします。ポリシービルダが起動され、新しい入出力変換ポリシーが保存されます。
- 9 [121 ページのセクション 8.9.2 「事前定義されたルールのインポート」](#) に進みます。

8.9.2 事前定義されたルールのインポート

- 1 ポリシービルダ内で、右クリックして [新規作成] > [事前定義されたルール] > [事前定義されたルールを前に挿入] または [事前定義されたルールを後に挿入] の順に選択します。
- 2 [入出力変換 - 電話番号の形式を (nnn) nnn-nnnn から nnn-xxx-nnnn に変更] を選択し、[OK] をクリックします。



- 3 [条件] タブをダブルクリックし、条件を編集します。
- 4 電話番号の形式変更が実行されるときに条件を定義します。
- 5 [OK] をクリックします。
- 6 [ファイル]> [保存] の順にクリックして、ルールを保存します。

8.9.3 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。電話番号の形式変更が実行されるときに指定する条件を定義します。

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

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

この事前定義されたルールを使用するには、入出力変換ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの2つの手順を実行します。このルールに追加する入出力変換ポリシーがすでにある場合は、[123 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。

8.10.1 ポリシーの作成

- 1 [アウトライン] ビューまたは [ポリシーフロー] ビューから発行者チャンネルまたは購読者チャンネルを選択します。
- 2 [ポリシーセット] ビューで入力出力変換ポリシーセットを選択し、[ポリシーセットに新しいポリシーを作成または追加します] アイコン  をクリックして、新しいポリシーを作成します。
- 3 [新しいポリシーの作成] をクリックし、[次へ] をクリックします。
- 4 ポリシーに名前を付けます。
- 5 デフォルトの場所を使用するか、または別の場所を参照して使用して、ドライブにポリシーを配置します。

作成ポリシー

新しいポリシーおよびポリシーを作成するコンテナの名前を指定します。



ポリシー名:

ポリシーコンテナ:

オブジェクトの作成後にエディタを開く(O)

- 6 [Open Editor after creating policy (ポリシーの作成後にエディタを開く)] を選択し、[次へ] をクリックします。
- 7 ポリシーのタイプとして、[DirXML スクリプト] を選択し、[終了] をクリックします。
- 8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか?)」というメッセージが表示されたら、[はい] をクリックします。ポリシービルダが起動され、新しい入出力変換ポリシーが保存されます。
- 9 123 ページのセクション 8.10.2 「事前定義されたルールのインポート」に進みます。

8.10.2 事前定義されたルールのインポート

- 1 ポリシービルダ内で、右クリックして [New (新規作成)] > [Predefined Rule (事前定義されたルール)] > [Insert Predefined Rule Before (事前定義されたルールを前に挿入)] または [Insert Predefined Rule After (事前定義されたルールを後に挿入)] の順に選択します。>
- 2 [入出力変換 - 電話番号の形式を nnn-xxx-xxxx から (xxx) nnn-xxxx に変更] をクリックし、[OK] をクリックします。

Input or Output Transformation - Reformat Telephone number from nnn-xxx-xxxx to (xxx) nnn-xxxx

No description available

条件

✓ ↻ 条件グループ 1

新しい条件をここで定義

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%d%d)-(%d%d%d%d)\$", "(%1)\$2-\$3", ローカル変数("current-value"))

- 3 [条件] タブをダブルクリックし、条件を編集します。
- 4 電話番号の形式変更が実行されるときに条件を定義します。
- 5 [OK] をクリックします。
- 6 [ファイル] > [保存] の順にクリックして、ルールを保存します。

8.10.3 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。電話番号の形式変更が実行されるときに指定する条件を定義します。

8.11 一致 - 発行者 (ミラーリング)

このルールは、指定したポイントからデータストア内のミラー化された構造を使用して、識別ボルト内のオブジェクトを照合します。このルールは、ドライバ内の一致ポリシーに実装します。このルールは、発行者チャ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、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、[\(125 ページ\) 事前定義されたルールのインポート](#)へ進みます。

8.11.1 ポリシーの作成

- 1 [アウトライン] ビューまたは [ポリシーフロー] ビューから、発行者チャンネルを選択します。
- 2 [ポリシーセット] ビューで一致ポリシーセットを選択し、[ポリシーセットに新しいポリシーを作成または追加します] アイコン  をクリックして、新しいポリシーを作成します。
- 3 [新しいポリシーの作成] をクリックし、[次へ] をクリックします。
- 4 ポリシーに名前を付けます。
- 5 デフォルトの場所を使用するか、または別の場所を参照して使用して、ドライバにポリシーを配置します。

作成ポリシー

新しいポリシーおよびポリシーを作成するコンテナの名前を指定します。



ポリシー名:

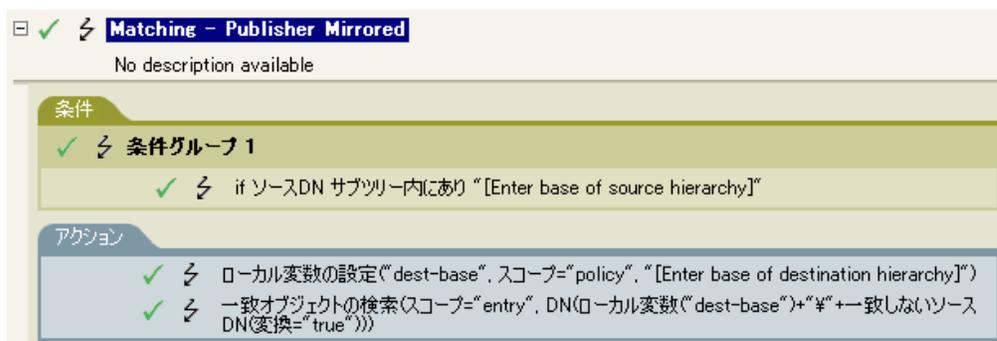
ポリシーコンテナ:

オブジェクトの作成後にエディタを開く (O)

- 6 [Open Editor after creating policy (ポリシーの作成後にエディタを開く)] を選択し、[次へ] をクリックします。
- 7 ポリシーのタイプとして、[Dir.XML スクリプト] を選択し、[終了] をクリックします。
- 8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか?)」というメッセージが表示されたら、[はい] をクリックします。ポリシービルダが起動され、新しい一致ポリシーが保存されます。
- 9 [125 ページのセクション 8.11.2 「事前定義されたルールのインポート」](#)に進みます。

8.11.2 事前定義されたルールのインポート

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



- 3 [条件] タブをダブルクリックし、条件を編集します。
- 4 [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 照合作業を開始するソース階層内のコンテナを参照して選択し、[OK] をクリックします。
- 6 [OK] をクリックします。
- 7 [アクション] タブをダブルクリックし、アクションを編集します。
- 8 [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。
- 9 [引数の編集] アイコンをクリックして、引数ビルダを起動します。
- 10 [Noun (名詞)] リストで [テキスト] を選択します。
- 11 [テキスト] をダブルクリックして、引数に追加します。
- 12 エディタで、参照ボタンをクリックして、ソース構造を一致させる宛先階層内のコンテナを参照し、[OK] をクリックします。
- 13 [OK] をクリックします。
- 14 [ファイル] > [保存] の順にクリックして、ルールを保存します。

8.11.3 ルールの動作

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

8.12 一致 - 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アイデンティティボールド内のミラー化された構造を使用して、データストア内のオブジェクトを照合します。このルールは、ドライバ内の一致ポリシーに実装します。このルールは、購読者チャ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、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、[127 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。

8.12.1 ポリシーの作成

- 1 [アウトライン] ビューまたは [ポリシーフロー] ビューから購読者チャンネルを選択します。
- 2 [ポリシーセット] ビューで一致ポリシーセットを選択し、[ポリシーセットに新しいポリシーを作成または追加します] アイコン  をクリックして、新しいポリシーを作成します。
- 3 [新しいポリシーの作成] をクリックし、[次へ] をクリックします。
- 4 ポリシーに名前を付けます。
- 5 デフォルトの場所を使用するか、または別の場所を参照して使用して、ドライバにポリシーを配置します。

作成ポリシー

新しいポリシーおよびポリシーを作成するコンテナの名前を指定します。



ポリシー名:

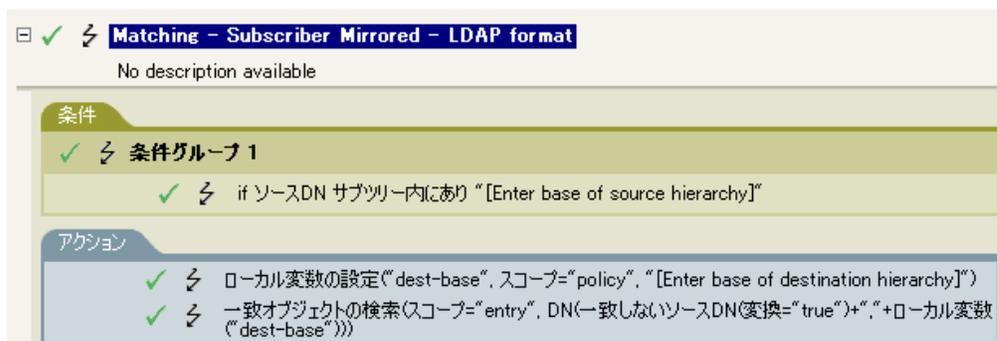
ポリシーコンテナ:

オブジェクトの作成後にエディタを開く(O)

- 6 [Open Editor after creating policy (ポリシーの作成後にエディタを開く)] を選択し、[次へ] をクリックします。
- 7 ポリシーのタイプとして、[DirXML スクリプト] を選択し、[終了] をクリックします。
- 8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか?)」というメッセージが表示されたら、[はい] をクリックします。ポリシービルダが起動され、新しい一致ポリシーが保存されます。
- 9 [127 ページのセクション 8.12.2「事前定義されたルールのインポート」](#)に進みます。

8.12.2 事前定義されたルールのインポート

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



- 3 [条件] タブをダブルクリックし、条件を編集します。
- 4 [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 照合作業を開始するソース階層内のコンテナを参照して選択し、[OK] をクリックします。
- 6 [OK] をクリックします。
- 7 [アクション] タブをダブルクリックし、アクションを編集します。
- 8 [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。
- 9 [引数の編集] アイコンをクリックして、引数ビルダを起動します。
- 10 [Noun (名詞)] リストの [テキスト] を選択します。
- 11 [テキスト] をダブルクリックして、引数に追加します。
- 12 エディタで、参照アイコンをクリックして、ソース構造を照合する宛先階層内のコンテナを参照し、選択したら [OK] をクリックします。
- 13 [OK] をクリックします。
- 14 [ファイル] > [保存] の順にクリックして、ルールを保存します。

8.12.3 ルールの動作

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

8.13 一致 - 属性値別

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

この事前定義されたルールを使用するには、一致ポリシーセット内へのポリシーの作成、および事前定義されたルールのインポートの2つの手順を実行します。このルールに追加する一致ポリシーがすでにある場合は、[129 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。

8.13.1 ポリシーの作成

- 1 [Outline (アウトライン)] ビューまたは [Policy Flow (ポリシーフロー)] ビューから発行者チャンネルまたは購読者チャンネルを選択します。
- 2 [ポリシーセット] ビューで一致ポリシーセットを選択し、[ポリシーセットに新しいポリシーを作成または追加します] アイコン  をクリックして、新しいポリシーを作成します。
- 3 [新しいポリシーの作成] をクリックし、[次へ] をクリックします。
- 4 ポリシーに名前を付けます。
- 5 デフォルトの場所を使用するか、または別の場所を参照して使用して、ドライバにポリシーを配置します。

作成ポリシー

新しいポリシーおよびポリシーを作成するコンテナの名前を指定します。

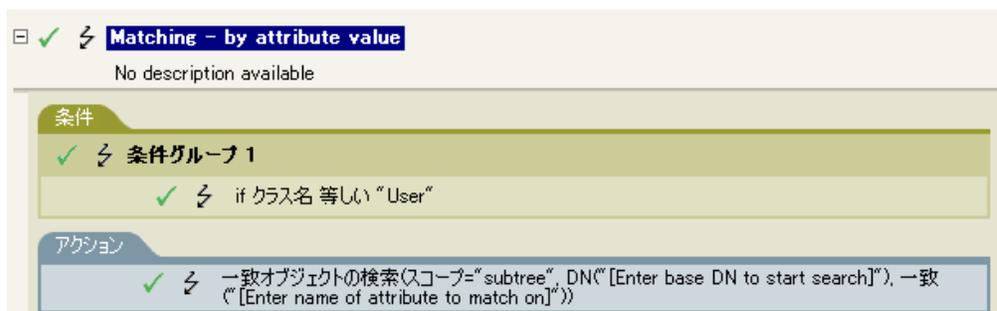


ポリシー名:	<input type="text" value="Matching"/>
ポリシーコンテナ:	<input type="text" value="Publisher.LDAP.Driver Set識別ポ-ルト"/> <input type="button" value="参照..."/>
<input checked="" type="checkbox"/> オブジェクトの作成後にエディタを開く(O)	

- 6 [Open Editor after creating policy (ポリシーの作成後にエディタを開く)] を選択し、[次へ] をクリックします。
- 7 ポリシーのタイプとして、[DirXML スクリプト] を選択し、[終了] をクリックします。
- 8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか?)」というメッセージが表示されたら、[はい] をクリックします。ポリシービルダが起動され、新しい一致ポリシーが保存されます。
- 9 [129 ページのセクション 8.13.2「事前定義されたルールのインポート」](#)に進みます。

8.13.2 事前定義されたルールのインポート

- 1 ポリシービルダ内で、右クリックして [新規作成] > [事前定義されたルール] > [事前定義されたルールを前に挿入] または [事前定義されたルールを後に挿入] の順に選択します。
- 2 [一致-属性値別] を選択し、[OK] をクリックします。



- 3 [アクション] タブをダブルクリックし、アクションを編集します。
- 4 [検索を開始するベース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] リストの [テキスト] を選択します。
- 7 [テキスト] をダブルクリックして、引数に追加します。
- 8 エディタで、参照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。
- 9 [一致させる属性名を入力] を [一致属性を入力] フィールドから削除します。
- 10 [引数の編集] アイコンをクリックして、一致属性ビルダを起動します。
- 11 参照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。
- 12 [OK] をクリックします。
- 13 [ファイル]> [保存] の順にクリックして、ルールを保存します。

8.13.3 ルールの動作

このルールは、属性によってユーザオブジェクトと一致します。ユーザオブジェクトを同期する場合、ドライバはルールを使用して指定した属性が存在するかどうかを確認します。属性が存在しない場合、新しいユーザオブジェクトが作成されます。

8.14 配置 - 発行者 (ミラーリング)

このルールは、指定したポイントからデータストア内のミラー化された構造を使用して、識別ポート内にオブジェクトを配置します。このルールは、ドライバ内の配置ポリシーに実装します。このルールは、発行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、130 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。

8.14.1 ポリシーの作成

- 1 [アウトライン] ビューまたは [ポリシーフロー] ビューから、発行者チャンネルを選択します。
- 2 ポリシーセット内で配置ポリシーセットを選択し、[ポリシーセットに新しいポリシーを作成または追加します] アイコン  をクリックして、新しいポリシーを作成します。
- 3 [新しいポリシーの作成] をクリックし、[次へ] をクリックします。
- 4 ポリシーに名前を付けます。
- 5 デフォルトの場所を使用するか、または別の場所を参照して使用して、ドライバにポリシーを配置します。

作成ポリシー

新しいポリシーおよびポリシーを作成するコンテナの名前を指定します。



ポリシー名:

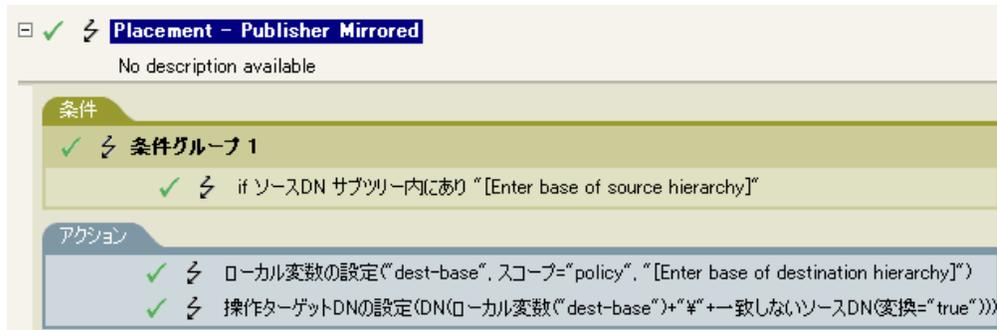
ポリシーコンテナ:

オブジェクトの作成後にエディタを開く(O)

- 6 [Open Editor after creating policy (ポリシーの作成後にエディタを開く)] を選択し、[次へ] をクリックします。
- 7 ポリシーのタイプとして、[DirXML スクリプト] を選択し、[終了] をクリックします。
- 8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか?)」というメッセージが表示されたら、[はい] をクリックします。ポリシービルダが起動され、新しい配置ポリシーが保存されます。
- 9 130 ページのセクション 8.14.2 「事前定義されたルールのインポート」に進みます。

8.14.2 事前定義されたルールのインポート

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



- 3 [条件] タブをダブルクリックし、条件を編集します。
- 4 [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 [アクション] タブをダブルクリックし、アクションを編集します。
- 7 [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] リストの [テキスト] を選択します。
- 10 [テキスト] をダブルクリックして、引数に追加します。
- 11 エディタで、参照ボタンをクリックして、オブジェクトを配置する宛先階層内のコンテナを参照し、選択したら [OK] をクリックします。
- 12 [OK] をクリックします。
- 13 [ファイル]> [保存] の順にクリックして、ルールを保存します。

8.14.3 ルールの動作

ユーザオブジェクトがソース階層内にある場合、オブジェクトは、データストアからミラー化された構造に配置されます。配置はローカル変数 `dest-base` が定義されているポイントから開始します。ユーザオブジェクトは「`dest-base\一致しないソース DN`」に配置されます。このルールではスラッシュ形式を使用します。

8.15 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、[132 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。

8.15.1 ポリシーの作成

- 1 [アウトライン] ビューまたは [ポリシーフロー] ビューから購読者チャンネルを選択します。
- 2 [ポリシーセット] ビューで配置ポリシーセットを選択し、[ポリシーセットに新しいポリシーを作成または追加します] アイコン  をクリックして、新しいポリシーを作成します。
- 3 [新しいポリシーの作成] をクリックし、[次へ] をクリックします。
- 4 ポリシーに名前を付けます。
- 5 デフォルトの場所を使用するか、または別の場所を参照して使用して、ドライバにポリシーを配置します。

作成ポリシー

新しいポリシーおよびポリシーを作成するコンテナの名前を指定します。



ポリシー名:

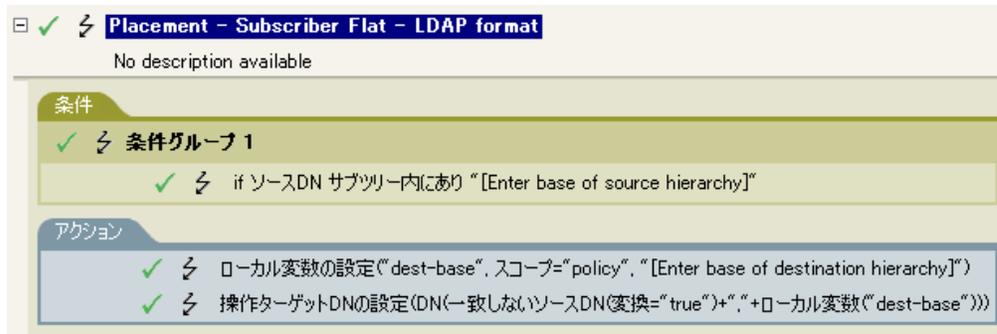
ポリシーコンテナ:

オブジェクトの作成後にエディタを開く(O)

- 6 [Open Editor after creating policy (ポリシーの作成後にエディタを開く)] を選択し、[次へ] をクリックします。
- 7 ポリシーのタイプとして、[DirXML スクリプト] を選択し、[終了] をクリックします。
- 8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか?)」というメッセージが表示されたら、[はい] をクリックします。ポリシービルダが起動され、新しい配置ポリシーが保存されます。
- 9 [132 ページのセクション 8.15.2 「事前定義されたルールのインポート」](#) に進みます。

8.15.2 事前定義されたルールのインポート

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



- 3 [条件] タブをダブルクリックし、条件を編集します。
- 4 [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 [アクション] タブをダブルクリックし、アクションを編集します。
- 7 [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] リストの [テキスト] を選択します。
- 10 [テキスト] をダブルクリックして、引数に追加します。
- 11 エディタで、参照ボタンをクリックして、オブジェクトを配置する宛先階層内のコンテナを参照し、選択したら [OK] をクリックします。
- 12 [OK] をクリックします。
- 13 [ファイル]> [保存] の順にクリックして、ルールを保存します。

8.15.3 ルールの動作

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

8.16 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ポールドの 1 つのコンテナ内に配置します。このルールは、ドライバ内の配置ポリシーに実装します。このルールは、発行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、[134 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。

8.16.1 ポリシーの作成

- 1 [アウトライン] ビューまたは [ポリシーフロー] ビューから、発行者チャンネルを選択します。
- 2 [ポリシーセット] ビューで配置ポリシーセットを選択し、[ポリシーセットに新しいポリシーを作成または追加します] アイコン  をクリックして、新しいポリシーを作成します。
- 3 [新しいポリシーの作成] をクリックし、[次へ] をクリックします。
- 4 ポリシーに名前を付けます。
- 5 デフォルトの場所を使用するか、または別の場所を参照して使用して、ドライバにポリシーを配置します。

作成ポリシー

新しいポリシーおよびポリシーを作成するコンテナの名前を指定します。



ポリシー名:

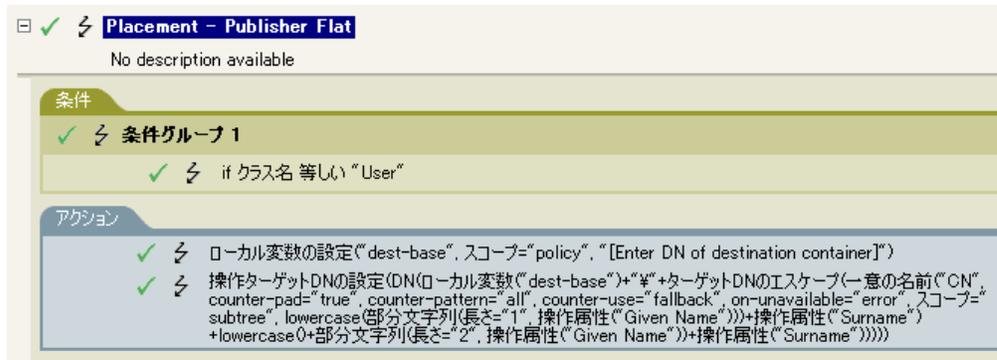
ポリシーコンテナ:

オブジェクトの作成後にエディタを開く(O)

- 6 [Open Editor after creating policy (ポリシーの作成後にエディタを開く)] を選択し、[次へ] をクリックします。
- 7 ポリシーのタイプとして、[DirXML スクリプト] を選択し、[終了] をクリックします。
- 8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか?)」というメッセージが表示されたら、[はい] をクリックします。ポリシービルダが起動され、新しい配置ポリシーが保存されます。
- 9 [134 ページのセクション 8.16.2 「事前定義されたルールのインポート」](#) に進みます。

8.16.2 事前定義されたルールのインポート

- 1 ポリシービルダ内で、右クリックして [新規作成] > [事前定義されたルール] > [事前定義されたルールを前に挿入] または [事前定義されたルールを後に挿入] の順に選択します。
- 2 [配置 - 発行者 (フラット)] を選択し、[OK] をクリックします。 >



- 3 [アクション] タブをダブルクリックし、アクションを編集します。
- 4 [宛先コンテナのDN 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。
- 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] リストの [テキスト] を選択します。
- 7 [テキスト] をダブルクリックして、引数に追加します。
- 8 エディタで、参照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。
- 9 [OK] をクリックします。
- 10 [ファイル]> [保存] の順にクリックして、ルールを保存します。

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

8.17 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、ドライブ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、[136 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。

8.17.1 ポリシーの作成

- 1 [アウトライン] ビューまたは [ポリシーフロー] ビューから、発行者チャンネルを選択します。
- 2 [ポリシーセット] ビューで配置ポリシーセットを選択し、[ポリシーセットに新しいポリシーを作成または追加します] アイコンをクリックして、新しいポリシーを作成します。

- 3 [新しいポリシーの作成] をクリックし、[次へ] をクリックします。
- 4 ポリシーに名前を付けます。
- 5 デフォルトの場所を使用するか、または別の場所を参照して使用して、ドライブにポリシーを配置します。

作成ポリシー

新しいポリシーおよびポリシーを作成するコンテナの名前を指定します。



ポリシー名:

ポリシーコンテナ:

オブジェクトの作成後にエディタを開く(O)

- 6 [Open Editor after creating policy (ポリシーの作成後にエディタを開く)] を選択し、[次へ] をクリックします。
- 7 ポリシーのタイプとして、[DirXML スクリプト] を選択し、[終了] をクリックします。
- 8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか?)」というメッセージが表示されたら、[はい] をクリックします。ポリシービルダが起動され、新しい配置ポリシーが保存されます。
- 9 136 ページのセクション 8.17.2 「事前定義されたルールのインポート」に進みます。

8.17.2 事前定義されたルールのインポート

- 1 ポリシービルダ内で、右クリックして [新規作成] > [事前定義されたルール] > [事前定義されたルールを前に挿入] または [事前定義されたルールを後に挿入] の順に選択します。
- 2 [配置 - 購読者 (フラット) - LDAP 形式] を選択し、[OK] をクリックします。

☐ ✓ ⚡ Placement - Subscriber Flat - LDAP format
 No description available

条件

✓ ⚡ 条件グループ 1

✓ ⚡ if クラス名 等しい "User"

アクション

✓ ⚡ ローカル変数の設定 ("dest-base", スコープ="policy", "[Enter DN of destination container]")

✓ ⚡ 操作ターゲットDNの設定(DN("uid="+ターゲットDNのエスケープ(一意の名前("uid", counter-pattern="last", counter-use="fallback", on-unavailable="error", lowercase(部分文字列(長さ="1", 操作属性("Given Name")+操作属性("Surname")+lowercase(部分文字列(長さ="2", 操作属性("Given Name")+操作属性("Surname")))+".", +ローカル変数("dest-base")))))

- 3 [アクション] タブをダブルクリックし、アクションを編集します。
- 4 [宛先コンテナのDN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。

- 5 [引数の編集] アイコンをクリックして、引数ビルダを起動します。
- 6 [Noun (名詞)] リストの [テキスト] を選択します。
- 7 [テキスト] をダブルクリックして、引数に追加します。
- 8 エディタで、すべてのユーザオブジェクトを配置するターゲットコンテナを追加します。コンテナが LDAP 形式で指定されていることを確認し、[OK] をクリックします。
- 9 [OK] をクリックします。
- 10 [ファイル]> [保存] の順にクリックして、ルールを保存します。

8.17.3 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。その後で、ターゲット DN を `uid=一意の名前、dest-base` に設定します。ユーザオブジェクトの `uid` 属性は、名前属性および名字属性の最初の 2 文字 (小文字) になります。このルールでは LDAP 形式を使用します。

8.18 配置 - 部署別発行者

このルールは、オブジェクトを、データストア内の 1 つのコンテナから識別ボールド内の複数のコンテナ内に配置します。このルールは、ドライバ内の配置ポリシーに実装します。このルールは、発行者チャ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、[138 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。

8.18.1 ポリシーの作成

- 1 [アウトライン] ビューまたは [ポリシーフロー] ビューから、発行者チャンネルを選択します。
- 2 [ポリシーセット] ビューで配置ポリシーセットを選択し、[ポリシーセットに新しいポリシーを作成または追加します] アイコンをクリックして、新しいポリシーを作成します。
- 3 [新しいポリシーの作成] をクリックし、[次へ] をクリックします。
- 4 ポリシーに名前を付けます。
- 5 デフォルトの場所を使用するか、または別の場所を参照して使用して、ドライバにポリシーを配置します。

作成ポリシー

新しいポリシーおよびポリシーを作成するコンテナの名前を指定します。



ポリシー名:

ポリシーコンテナ:

オブジェクトの作成後にエディタを開く(O)

- 6 [Open Editor after creating policy (ポリシーの作成後にエディタを開く)] を選択し、[次へ] をクリックします。
- 7 ポリシーのタイプとして、[DirXML スクリプト] を選択し、[終了] をクリックします。
- 8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか?)」というメッセージが表示されたら、[はい] をクリックします。ポリシービルダが起動され、新しい配置ポリシーが保存されます。
- 9 138 ページのセクション 8.18.2 「事前定義されたルールのインポート」に進みます。

8.18.2 事前定義されたルールのインポート

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

☐ ✓ ⚡ Placement - Publisher By Dept
No description available

条件

✓ ⚡ 条件グループ 1

- ✓ ⚡ if クラス名 等しい "User"
- And(N) ✓ ⚡ if 属性 'OU' 使用可能

アクション

- ✓ ⚡ ローカル変数の設定 ("dest-base", スコープ="policy", "[Enter DN of destination Organization]")
- ✓ ⚡ 操作ターゲットDNの設定(DN(ローカル変数("dest-base")+属性("OU")+ターゲットDNのエスケープ(一意の名前("CN", counter-pad="true", counter-pattern="all", counter-use="fallback", on-unavailable="error", スコープ="subtree", lowercase(部分文字列(長さ="1", 操作属性("Given Name"))+操作属性("Surname")+lowercase(部分文字列(長さ="2", 操作属性("Given Name"))+操作属性("Surname"))))))))

- 3 [アクション] タブをダブルクリックし、アクションを編集します。
- 4 [宛先組織のDN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。
- 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] リストの [テキスト] を選択します。
- 7 [テキスト] をダブルクリックして、引数に追加します。

- 8 エディタで、参照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。
- 9 **[OK]** をクリックします。
- 10 **[ファイル]> [保存]** の順にクリックして、ルールを保存します。

8.18.3 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を適切な部署に配置します。配置する必要があり、使用可能な OU 属性を持っているユーザオブジェクトの場合は、「`dest-base\OU 属性\CN 属性の値`」に配置されます。

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

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

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

8.19 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、オブジェクトを、アイデンティティボールド内の 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せん。

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、[140 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。

8.19.1 ポリシーの作成

- 1 **[アウトライン]** ビューまたは **[ポリシーフロー]** ビューから購読者チャンネルを選択します。
- 2 **[ポリシーセット]** ビューで配置ポリシーセットを選択し、**[ポリシーセットに新しいポリシーを作成または追加します]** アイコン  をクリックして、新しいポリシーを作成します。
- 3 **[新しいポリシーの作成]** をクリックし、**[次へ]** をクリックします。
- 4 ポリシーに名前を付けます。
- 5 デフォルトの場所を使用するか、または別の場所を参照して使用して、ドライバにポリシーを配置します。

作成ポリシー

新しいポリシーおよびポリシーを作成するコンテナの名前を指定します。



ポリシー名:

ポリシーコンテナ:

オブジェクトの作成後にエディタを開く(O)

- 6 [Open Editor after creating policy (ポリシーの作成後にエディタを開く)] を選択し、[次へ] をクリックします。
- 7 ポリシーのタイプとして、[DirXML スクリプト] を選択し、[終了] をクリックします。
- 8 ファイルの衝突ウィンドウに、「Before editing this item you need to save. Do you wish to save the editor's changes and continue? (この項目を編集する前に保存する必要があります。エディタの変更内容を保存して続行しますか?)」というメッセージが表示されたら、[はい] をクリックします。ポリシービルダが起動され、新しい配置ポリシーが保存されます。
- 9 140 ページのセクション 8.19.2 「事前定義されたルールのインポート」に進みます。

8.19.2 事前定義されたルールのインポート

- 1 ポリシービルダ内で、右クリックして [新規作成] > [事前定義されたルール] > [事前定義されたルールを前に挿入] または [事前定義されたルールを後に挿入] の順に選択します。
- 2 [配置 - 部署別購読者-LDAP 形式] を選択し、[OK] をクリックします。

Placement - Publisher Mirrored

No description available

条件

✓ 条件グループ 1

- ✓ if クラス名 等しい "User"
- And(N) ✓ if 属性 'OU' 使用可能

アクション

- ローカル変数の設定 ("dest-base", スコープ="policy", "[Enter DN of destination Organization]")
- 操作ターゲットDNの設定 (DN("uid="+ターゲットDNのエスケープ(一意の名前("uid", counter-pattern="last", counter-use="fallback", on-unavailable="error", lowercase(部分文字列(長さ="1", 操作属性("Given Name")+操作属性("Surname")+lowercase(部分文字列(長さ="2", 操作属性("Given Name")+操作属性("Surname"))))+属性("OU")+"+ローカル変数("dest-base"))))

- 3 [アクション] タブをダブルクリックし、アクションを編集します。
- 4 [宛先組織の DN 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。
- 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] リストの [テキスト] を選択します。
- 7 [テキスト] をダブルクリックして、引数に追加します。

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

8.19.3 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を適切な部署に配置します。配置する必要があり、使用可能な OU 属性を持っているユーザオブジェクトの場合は、「uid=一意の名前,ou=OU 属性の値,dest-base」に配置されます。

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

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

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

ポリシーシミュレータを使用したポリシーのテスト

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

ポリシーシミュレータで一般的なタスクについては、次の節を参照してください。

- ◆ 143 ページのセクション 9.1 「ポリシーシミュレータへのアクセス」
- ◆ 145 ページのセクション 9.2 「ポリシーシミュレータの使用」
- ◆ 149 ページのセクション 9.3 「Java 拡張を使用したポリシーのシミュレート」

ポリシーシミュレータは、XML を使用します。eDirectory™ ドキュメントタイプ定義ファイル (nds.dtd) では、メタディレクトリエンジンが処理できる XML ドキュメントのスキーマを定義します。このスキーマに準拠しない XML ドキュメントでは、エラーが発生します。ドキュメントが nds.dtd に準拠しているかどうかを確認し、エラーが発生している理由に関する情報を見つけるには、『*Identity Manager 3.5.1 DTD リファレンス*』の「**NDS DTD**」を参照してください。

ポリシーがマッピングテーブルオブジェクトまたは ECMAScript オブジェクトを使用している場合は、ポリシーシミュレータは、ポリシーのテスト時にこれらのオブジェクトをテストします。

ポリシーシミュレータでは、SOAP および区切りテキストなどのアプリケーションドライバの最初のポリシーセットのシミュレートはできません。これらのドライバは、カンマ区切りのファイルまたはテキストファイルを入力として使用し、XML または XDS をポリシーチェーン内のポリシーから派生します。現在、ポリシーシミュレータが入力として受け付けるのは、有効な XML または XDS だけです。将来のリリースで、機能が追加される可能性があります。

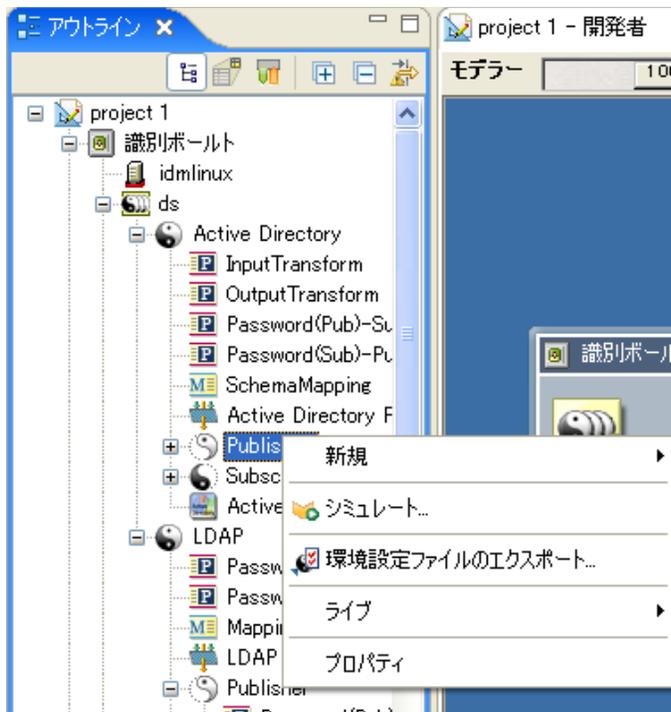
9.1 ポリシーシミュレータへのアクセス

ポリシーシミュレータには、次の 3 つの方法でアクセスできます。

- ◆ 143 ページの 「[Outline (アウトライン)] ビュー」
- ◆ 144 ページの 「[Policy Flow (ポリシーフロー)] ビュー」
- ◆ 144 ページの 「エディタ」

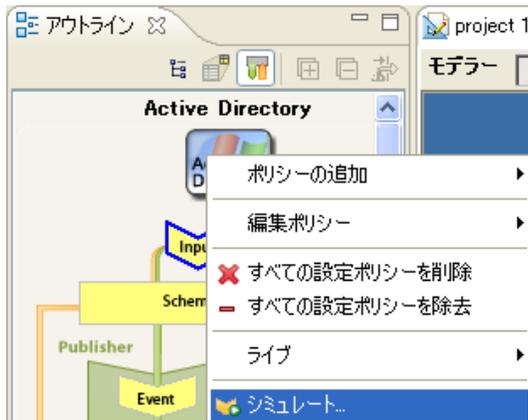
9.1.1 [Outline (アウトライン)] ビュー

- 1 [モデルアウトアランの表示] アイコン  をクリックします。
- 2 シミュレートするドライバ、発行者、購読者、マッピングルール、フィルタ、または任意のポリシーを右クリックし、[Simulate (シミュレート)] をクリックします。



9.1.2 [Policy Flow (ポリシーフロー)] ビュー

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



9.1.3 エディタ

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

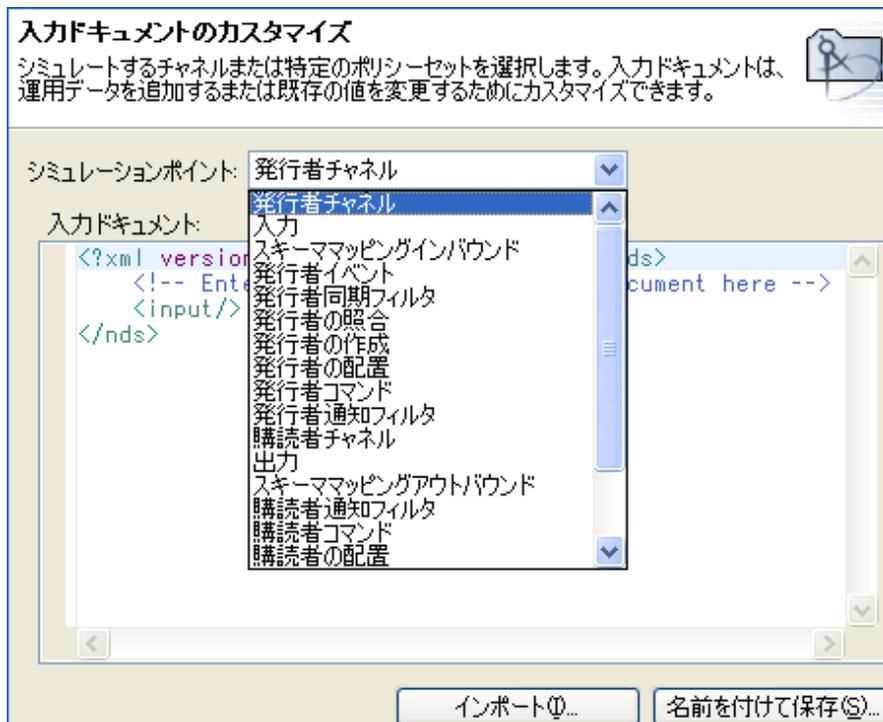
9.2 ポリシーシミュレータの使用

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

ポリシーシミュレータを使用するには

- 1 [Simulation Point (シミュレーションポイント)] ドロップダウンリストで、ポリシーをテストするドライバフロー内の場所を選択します。

[発行者チャンネル]、[購読者チャンネル]、[入力]、[スキーマのマッピング]、[イベント]、[Sync Filter (同期フィルタ)]、[一致]、[作成]、[配置]、[コマンド]、および [Notify Filter (通知フィルタ)] の任意の項目を選択できます。



テストする特定のポリシーまたはルールを選択する場合は、[シミュレーションポイント] オプションには、[識別ポールドへ] または [識別ポールドから] だけが表示されます。

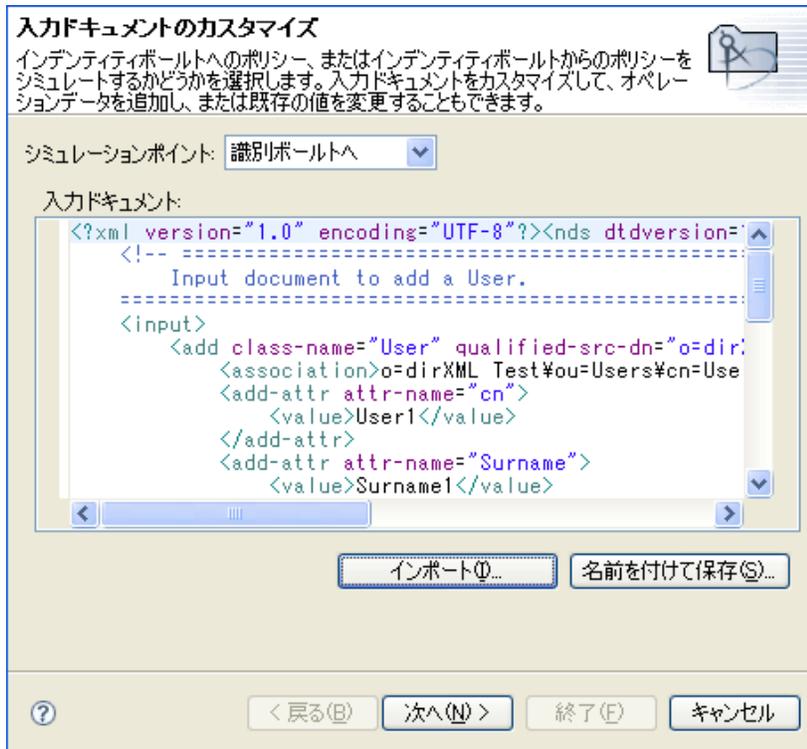
- 2 [Import (インポート)] を選択し、テストするファイルを参照して選択します。

Designer にはサンプルのイベントファイルが付属します。ファイルは、プラグイン `com.novell.designer.idm.policy.simulation` にあります。イベントは、追加、関連付け、削除、インスタンス、変更、移動、クエリ、名前変更、およびステータスです。

- 3 フォルダをダブルクリックして、使用可能なイベントを表示します。

選択できるファイルは、イベントごとに異なります。たとえば、[追加] を選択した場合は、[Organization.xml]、[OrganizationalUnit.xml]、および [User.xml] の3つのオプションがあります。ファイルはイベントを示します。User.xml を選択した場合、これはユーザオブジェクトの追加イベントになります。

- 4 ファイルを選択し、[開く] をクリックして、ウィンドウに入力ドキュメントを表示します。
- 5 [次へ] をクリックして、シミュレーションを開始します。



- 6 [トレース] タブを選択して、DSTRACE に表示されるとおりの追加イベントの結果を表意 j します。

変換結果の表示

シミュレーションの詳細を確認するには[トレース]を、変換されたドキュメントを確認するには[出力]を、入力ドキュメントと出力ドキュメントの差異を確認するには[比較]を選択します。

トレース 出力 比較

```
LDAP :Applying policy: %+C%14CCreation%-C.  
LDAP : Applying to add #1.  
LDAP :Policy returned:  
LDAP :  
<nds dtdversion="1.0" ndsversion="8.5" xml:space  
<!-- =====  
      Input document to add a User.  
      =====  
<input>  
  <add class-name="User" qualified-src-dn="o=d  
    <association>o=dirXML Test%ou=Users%cn=Use  
    <add-attr attr-name="cn">  
      <value>User1</value>  
    </add-attr>  
    <add-attr attr-name="Surname">
```

ログのクリア(C)

繰り返し

? <戻る(B) 次へ(N)> 終了(F) キャンセル

[ログのクリア] をクリックし、[繰り返し] をクリックして、新しいトレースログでシミュレーションを再び実行します。

- 7 [出力] タブを選択し、入力ドキュメントに対してポリシーが実行されたときに生成される出力ドキュメントを表示します。入力ドキュメントはユーザの「追加」イベントです。



入力ドキュメントおよび出力ドキュメントを編集できます。変更を保持する場合は、[名前を付けて保存] をクリックします。

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



- 9 [Repeat (繰り返し)] をクリックして、別の入力ドキュメントを選択し、イベントの結果を表示します。
- 10 テストが済んだら、[終了] をクリックして、ポリシーシミュレータを閉じます。

9.3 Java 拡張を使用したポリシーのシミュレート

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

拡張ディレクトリを決定または変更するには

- 1 ツールバーで [Windows (ウィンドウ)] > [初期設定] の順に選択します。
- 2 [Designer for IDM (IDM の Designer)] > [Simulation (シミュレーション)] の順に移動します。
- 3 Java クラスを含む jar ファイルを、指定したディレクトリにコピーし、ポリシーをシミュレートします。
-
- 4 [適用] をクリックして変更内容を保存するか、または [OK] をクリックして変更内容を保存してウィンドウを閉じます。

注: [サポートされない実験用のプレリリース機能を有効にします] オプションを使用すると、ポリシーシミュレータで、ライブの識別ボールドまたは接続システムに対するポリ

シーをテストできます。このオプションは、Designer 2.1 ではサポートされていないため、説明はありません。

Designer では、外部 Java クラスを含む複数のディレクトリを指定することができます。追加ディレクトリを指定するには、次の手順を実行します。

- 1 [追加] をクリックします。
- 2 目的のディレクトリを参照して選択してから、[OK] をクリックします。
- 3 ディレクトリを削除するには、[削除] をクリックします。

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、異なるタイプの利をオブジェクトを含みます。

- ◆ 151 ページのセクション 10.1 「汎用リソースオブジェクト」
- ◆ 153 ページのセクション 10.2 「マッピングテーブルオブジェクト」
- ◆ 157 ページのセクション 10.3 「ECMAScript オブジェクト」
- ◆ 158 ページのセクション 10.4 「アプリケーションオブジェクト」
- ◆ 158 ページのセクション 10.5 「リポジトリオブジェクト」
- ◆ 158 ページのセクション 10.6 「ライブラリオブジェクト」

10.1 汎用リソースオブジェクト

汎用リソースオブジェクトは、ポリシーが使用す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。

- ◆ 151 ページのセクション 10.1.1 「リソースオブジェクトの作成」
- ◆ 153 ページのセクション 10.1.2 「汎用リソースオブジェクトの使用」

10.1.1 リソースオブジェクトの作成

- 1 [アウトライン] ビューで、リソースオブジェクトを作成する場所を右クリックし、[新規]> [リソース] を選択します。
- 2 リソースオブジェクトの名前を指定します。
- 3 コンテンツタイプ [XML] または [テキスト] を選択します。
- 4 [オブジェクトの作成後にエディタを開く] のチェックボックスを選択して、[OK] をクリックします。

リソース名の設定
新しいリソースの名前を入力します。



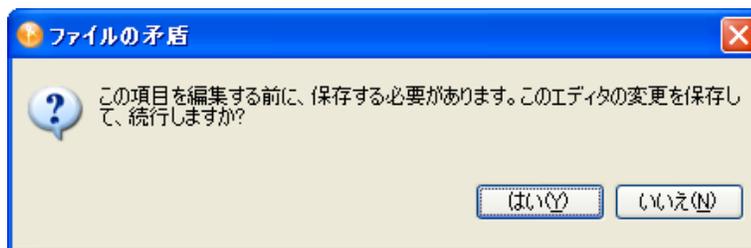
名前(N):

コンテンツタイプ:

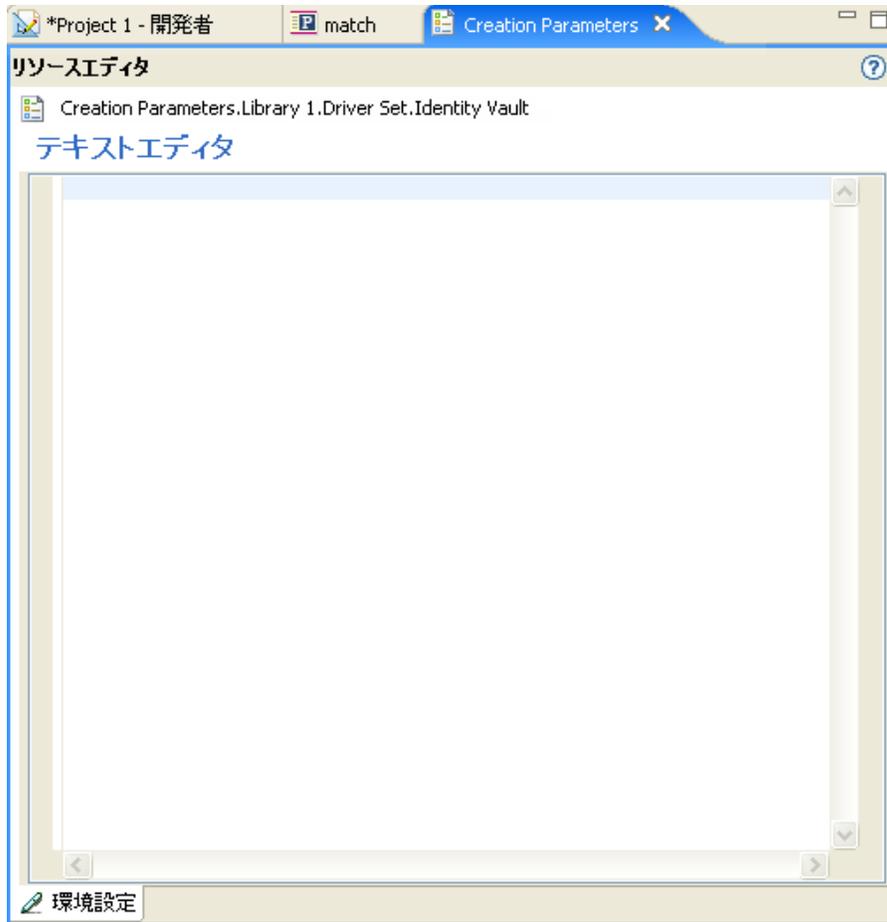
オブジェクトの作成後にエディタを開く(O)

OK キャンセル

5 ファイル衝突メッセージで [はい] をクリックします。



6 目的のテキストまたは XML を指定して、<Ctrl>+<S> キーを押してリソースオブジェクトを保存します。



10.1.2 汎用リソースオブジェクトの使用

リソースオブジェクトは、情報を保存するボックスです。これは eDirectory オブジェクトで、オブジェクト内の情報を使用するには、他の eDirectory オブジェクトと同様に処理します。属性 DirXML-Data は、リソースオブジェクトに情報を保存し、属性 DirXML-Content タイプは、情報のラベルを保存します。

リソースオブジェクト内に保存されている情報を読むには、(325 ページ) ソース属性または (303 ページ) ターゲット属性トークンを使用します。オブジェクトに情報を書き込むには、以下のアクションを使用します。

- (235 ページ) ターゲット属性値のクリア
- (237 ページ) ソース属性値のクリア
- (267 ページ) デフォルト属性値の設定
- (280 ページ) ソース属性値の設定

10.2 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、(346 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、以下の手順を完了してください。

1. [154 ページのセクション 10.2.1 「マッピングテーブルオブジェクトの作成」](#)
2. [155 ページのセクション 10.2.2 「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」](#)

マッピングテーブルを編集するには、[156 ページのセクション 10.2.3 「マッピングテーブルオブジェクトの編集」](#)を参照してください。

10.2.1 マッピングテーブルオブジェクトの作成

マッピングテーブルオブジェクトは、ライブラリ、ドライバオブジェクト、発行者チャンネル、または購読者チャンネルで作成できます。

- 1 [アウトライン] ビューで、マッピングテーブルを作成する場所を右クリックして、[新規]> [マッピングテーブル] を選択します。
- 2 マッピングテーブルオブジェクトの名前を指定します。
- 3 [オブジェクトの作成後にエディタを開く] のチェックボックスを選択して、[OK] をクリックします。

マッピングテーブル名を選択する
マッピングテーブルリソースを作成する

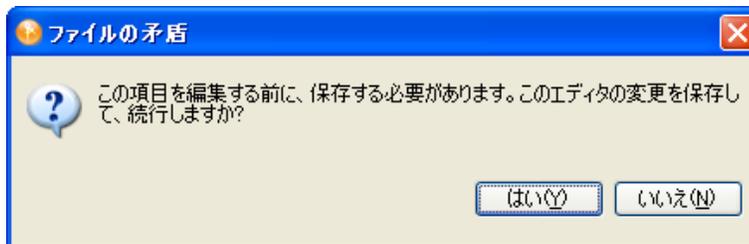


名前(N): 新規マッピングテーブル

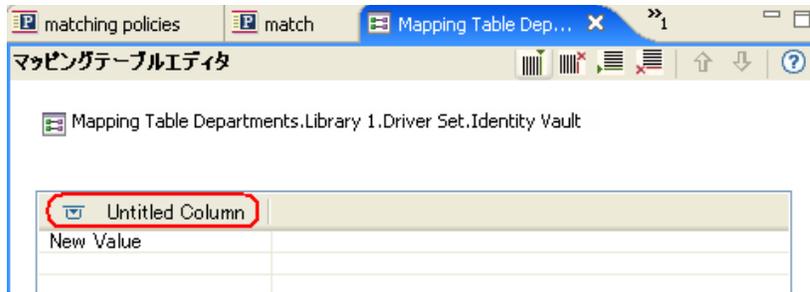
オブジェクトの作成後にエディタを開く(O)

OK キャンセル

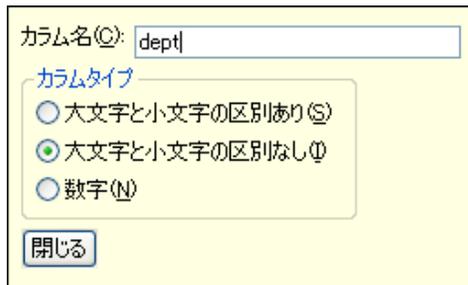
- 4 ファイル衝突メッセージ内で [はい] をクリックしてマッピングテーブルを保存します。



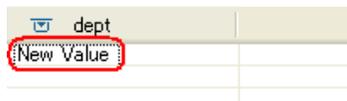
- 5 タイトルのないカラムをクリックします。



- 6 カラムの名前を指定して、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] かどうかを選択します。



- 7 [閉じる] をクリックします。
- 8 [新しい値] をクリックして、行の値を指定します。



- 9 (オプション) 追加カラムを追加するには、マッピングテーブルエディタ内で右クリックして [カラムの追加.] を選択します。
または
[カラムの追加] アイコンをクリックして、**ステップ 5** から **155 ページのステップ 7** を繰り返します。
- 10 (オプション) 追加行を追加するには、マッピングテーブルエディタ内で右クリックして [行の追加] を選択します。
または
[行の追加] アイコンをクリックして、**ステップ 8** を繰り返します。
- 11 <Ctrl>+<S> キーを押してマッピングテーブルオブジェクトを保存します。
- 12 **155 ページのセクション 10.2.2 「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」** に進みます。

10.2.2 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、または既存のポリシーを編集用に選択します。
- 2 引数ビルダをポリシービルダ内で起動する

- 3 動詞のリストから [マップ] をダブルクリックして、式パネルに追加します。
- 4 [True] または [False] を選択して、このマッピングテーブルをトレースするかどうかを示します。
- 5 [エディタ] フィールドで、154 ページのセクション 10.2.1 「マッピングテーブルオブジェクトの作成」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。
- 6 ソースカラム名を指定します。
- 7 ターゲットカラム名を指定します。

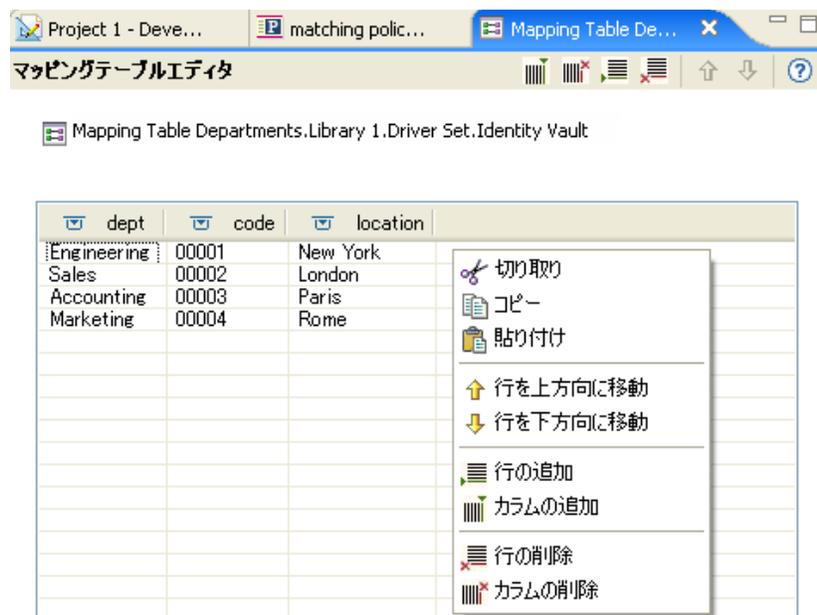
マッピングテーブルは、この時点では任意の方法で使用できます。この例では、OU 属性にマッピングテーブルから派生した値が指定されます。

`マップ(dest="code", src="dept", table=".%Library 1%Mapping Table Departments")`
 操作属性("OU")

10.2.3 マッピングテーブルオブジェクトの編集

Designer は、マッピングテーブルを編集するために以下のオプションを提供しています。

図 10-1 マッピングテーブルの編集



- ◆ [切り取り] : 選択した行を切り取ります。
- ◆ コピー : 選択した行をコピーします。
- ◆ 貼り付け : 選択した行を貼り付けます。
- ◆ 行を上方向に移動 : 選択した行を 1 行上に移動します。
- ◆ 行を下方向に移動 : 選択した行を 1 行下に移動します。
- ◆ 行の追加 : マッピングテーブルに行を追加します。
- ◆ カラムの追加 : マッピングテーブルにカラムを追加します。
- ◆ 行の削除 : マッピングテーブルから行を削除します。
- ◆ カラムの削除 : マッピングテーブルからカラムを削除します。

10.2.4 マッピングテーブルオブジェクトのテスト

マッピングテーブルの機能をポリシーシミュレータを使用してテストできます。ポリシーシミュレータは、マッピングテーブルを使用するポリシーをテストすることによってマッピングテーブルをテストします。詳細については、[143 ページの第 9 章「ポリシーシミュレータを使用したポリシーのテスト」](#)を参照してください。

10.3 ECMAScript オブジェクト

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、[161 ページの第 11 章「ポリシーでの ECMAScript の使用」](#)を参照してください。

10.4 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用の認証パラメータ値を保存します。Novell SecureLogin および Novell SecretStore® 用のアプリケーションオブジェクトがあります。SecureLogin 用のアプリケーションオブジェクトを作成する方法については、『*Identity Manager 3.5.1 用 Novell 資格情報プロビジョニングポリシー*』の「“Novell SecureLogin 用アプリケーションオブジェクトの作成”」を参照してください。SecretStore 用のアプリケーションオブジェクトを作成する方法については、『*Identity Manager 3.5.1 用 Novell 資格情報プロビジョニングポリシー*』の「“Novell SecretStore 用アプリケーションオブジェクトの作成”」を参照してください。

10.5 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静的設定情報を保存します。Novell SecureLogin および Novell SecretStore 用のリポジトリオブジェクトがあります。SecureLogin 用のリポジトリオブジェクトの作成方法については、『*Identity Manager 3.5.1 用 Novell 資格情報プロビジョニングポリシー*』の「“Novell SecureLogin 用リポジトリオブジェクトの作成”」を参照してください。SecretStore 用のリポジトリオブジェクトを作成する方法については、『*Identity Manager 3.5.1 用 Novell 資格情報プロビジョニングポリシー*』の「“Novell SecretStore 用リポジトリオブジェクトの作成”」を参照してください。

10.6 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ブ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、ドライブ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。ドライブは、ドライブを実行しているサーバがライブラリオブジェクトの読み取り/書き込みまたはマスタレプリカを保持している場合は、ツリー内の任意のライブラリを参照できます。

スタイルシート、ポリシー、ルール、およびその他のリソースオブジェクトは、ライブラリに保存でき、1つまたは複数のドライブによって参照できます。

- ◆ 158 ページのセクション 10.6.1 「ライブラリオブジェクトの作成」
- ◆ 159 ページのセクション 10.6.2 「ライブラリオブジェクトへのポリシーの追加」
- ◆ 160 ページのセクション 10.6.3 「ライブラリオブジェクト内でのポリシーでの使用」

10.6.1 ライブラリオブジェクトの作成

- 1 [アウトライン] ビュー内でドライブセットまたは識別ポールドオブジェクトをクリックします。次に、[新規] > [ライブラリ] をクリックします。
- 2 ライブラリオブジェクトの名前を指定し、[OK] をクリックします。

ライブラリ名の設定

新しいライブラリの名前を入力します。



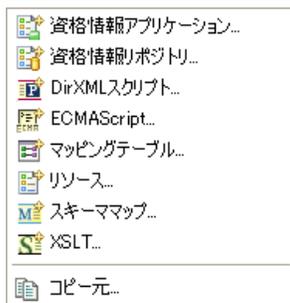
名前(N):

OK キャンセル

10.6.2 ライブラリオブジェクトへのポリシーの追加

ライブラリは、任意のポリシー、XSLT スタイルシート、または任意のタイプのリソースオブジェクトを保持することができます。

- 1 ライブラリオブジェクトを右クリックして、[新規] を選択し、ライブラリに保存したいオブジェクトのタイプを選択します。オプションは次のとおりです。



- ◆ **資格情報アプリケーション** : Novell 資格情報プロビジョニングポリシー用のアプリケーション認証パラメータ値を保存します。詳細については、『[Identity Manager 3.5.1 用 Novell 資格情報プロビジョニングポリシー](#)』を参照してください。
- ◆ **資格情報リポジトリ** : Novell 資格情報プロビジョニングポリシーの静的設定情報を保存します。詳細については、『[Identity Manager 3.5.1 用 Novell 資格情報プロビジョニングポリシー](#)』を参照してください。
- ◆ **DirXML スクリプト** : ポリシーセットを作成します。詳細については、[30 ページのセクション 3.3 「ポリシーの作成」](#)を参照してください。
- ◆ **ECMAScript** : ECMAScript オブジェクトを作成します。詳細については、[161 ページのセクション 11.1 「ECMAScript オブジェクトの作成」](#)を参照してください。
- ◆ **マッピングテーブル** : マッピングテーブルオブジェクトを作成します。詳細については、[153 ページのセクション 10.2 「マッピングテーブルオブジェクト」](#)を参照してください。
- ◆ **リソース** : 汎用リソースオブジェクトを作成します。詳細については、[151 ページのセクション 10.1 「汎用リソースオブジェクト」](#)を参照してください。
- ◆ **スキーママップ** : スキーママップオブジェクトを作成します。詳細については、[75 ページの第 6 章 「スキーママッピングポリシーの定義」](#)を参照してください。

- ◆ **XSLT:** ライブラリ内で XSLT スタイルシートを作成します。詳細については、『*Identity Manager 3.5.1 用ポリシーの理解*』の「**XSLT スタイルシートを使用したポリシーの定義**」を参照してください。
- ◆ **コピー元:** 既存オブジェクトのコピーを作成します。

10.6.3 ライブラリオブジェクト内でのポリシーでの使用

ライブラリを作成した後に、ライブラリ内に保存されている任意のリソースを任意のポリシーで使用できます。

- 1 [アウトライン] ビュー内で目的のポリシーをダブルクリックします。
- 2 ポリシービルダ内を右クリックし、[新規]> [インクルード]> [インクルードを前に挿入] または [インクルードを後に挿入] を選択します。
- 3 ライブラリオブジェクト内に保存されている目的のリソースを参照して選択し、[OK] を 2 回クリックします。



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。これはしばしば JavaScript[®] または JScript^{*} と呼ばれますが、これらは ECMAScript の実装です。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[®] スクリプトまたは XSLT スタイルシートが提供できない高度な機能を提供します。

Identity Manager は ECMAScript オブジェクトを、プロビジョニング要求定義エディタでカスタムフォームを作成する、または ECMAScript 機能をポリシー内で呼び出す、という 2 つの異なる方法で使用します。カスタムフォームの詳細については、「[カスタムフォームの作成 \(http://www.novell.com/documentation/idm35/dgpro/data/prdefcreateformschapter.html\)](http://www.novell.com/documentation/idm35/dgpro/data/prdefcreateformschapter.html)」を参照してください。

この節では、ECMAScript エディタの使用方法、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。

- 161 ページのセクション 11.1 「ECMAScript オブジェクトの作成」
- 162 ページのセクション 11.2 「ECMAScript エディタの使用」
- 170 ページのセクション 11.3 「ポリシーを備えた ECMAScripts の例」
- 172 ページのセクション 11.4 「JavaScript ファイル初期設定の変更」

11.1 ECMAScript オブジェクトの作成

ECMAScript オブジェクトは、ライブラリ、ドライバオブジェクト、発行者チャンネル、または購読者チャンネルで作成できます。

- 1 [アウトライン] ビューで、ECMAScript オブジェクトを作成する場所を右クリックして、[新規]> [ECMAScript] を選択します。
- 2 ECMAScript オブジェクトの名前を指定します。
- 3 [オブジェクトの作成後にエディタを開く] のチェックボックスを選択して、[OK] をクリックします。

新規ECMAScript
新規ECMAScriptの作成



名前(N):

オブジェクトの作成後にエディタを開く(O)

- 4 ファイル衝突メッセージ内で [はい] をクリックして ECMAScript オブジェクトを保存します。



- 5 ECMAScript を入力するか、または ECMAScript を既存のファイルからエディタにコピーします。
- 6 ECMAScript を保存するには、ECMAScript が完了したら <Ctrl>+<S> キーを押します。

ECMAScript エディタの使用方法については、[162 ページのセクション 11.2 「ECMAScript エディタの使用」](#) を参照してください。

11.2 ECMAScript エディタの使用

ECMAScript オブジェクトは、Identity Manager 3.5.1 バージョンを持つサーバでのみサポートされます。選択したドライバセット内のサーバが Identity Manager 3.5.1 以前のバージョンの場合、エラーメッセージが表示され、Designer はオブジェクトの作成を許可しません。サーバのプロパティでサーバのバージョンを Identity Manager 3.5.1 に変更すると、ECMAScript オブジェクトを作成できます。

Designer は、ECMA 式ビルダを含む ECMAScript エディタを提供します。両方を使用して ECMAScript を作成します。

ECMAScript エディタにアクセスする

- 1 [アウトライン] ビュー内で ECMAScript オブジェクトを右クリックし、[編集] を選択します。

または

ECMAScript オブジェクトを作成するときに、[オブジェクトの作成後にエディタを開く] チェックボックスを選択します。

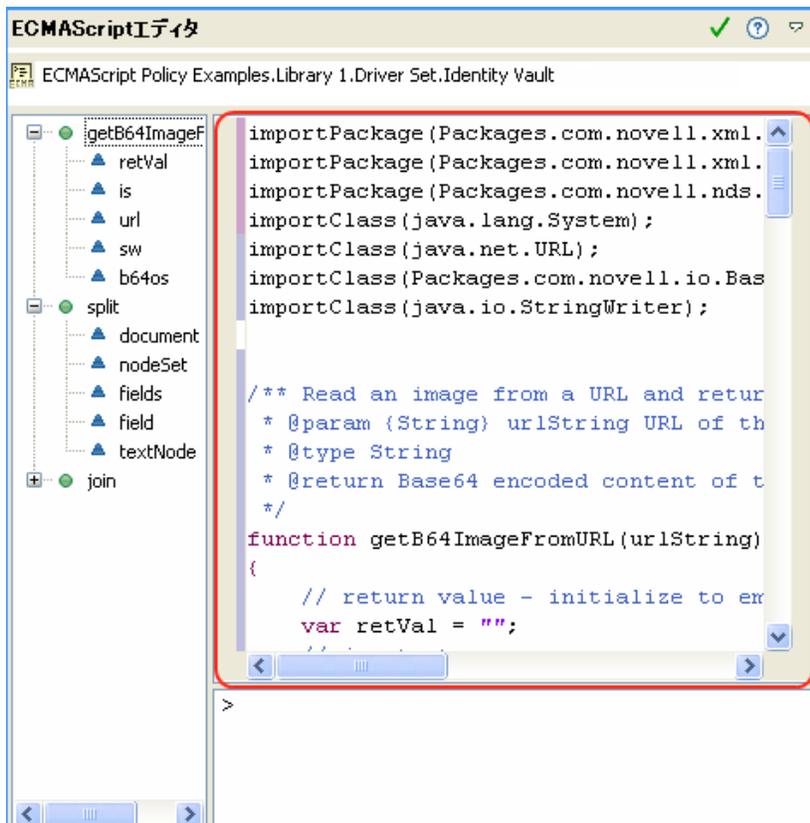
ECMAScript エディタは、使用しているセクションによって異なるタイプの機能を提供します。

- ◆ [163 ページのセクション 11.2.1 「メインスクリプト記述エリア」](#)
- ◆ [165 ページのセクション 11.2.2 「式ビルダ」](#)
- ◆ [167 ページのセクション 11.2.3 「関数と変数」](#)
- ◆ [168 ページのセクション 11.2.4 「エラー表示」](#)
- ◆ [169 ページのセクション 11.2.5 「シェルエリア」](#)

11.2.1 メインスクリプト記述エリア

ECMAScript エディタは、ECMAScript が作成されるメインスクリプト記述エリアを提供します。新しいスクリプトを入力するか、または既存のスクリプトをコピーできます。

図 11-1 メインスクリプト記述エリア



- ◆ 163 ページの「既存の ECMAScript の使用」
- ◆ 163 ページの「ECMAScript の編集」
- ◆ 164 ページの「ECMAScript のコーディングヘルプ」

既存の ECMAScript の使用

- 1 ECMAScript をテキストエディタで開いて、スクリプトをコピーします。
- 2 ECMAScript を ECMAScript エディタに貼り付けます。
- 3 <Ctrl>+<S> キーを押して ECMAScript を保存します。

ECMAScript の編集

- 1 メインスクリプト記述エリア内で右クリックして、目的のオプションを選択します。

 元に戻す(U) 入力	Ctrl+Z
 やり直し(R)	Ctrl+Y
 切り取り(T)	Ctrl+X
 コピー(C)	Ctrl+C
 貼り付け(P)	Ctrl+V
 削除(D)	Delete
<hr/>	
すべて選択(A)	Ctrl+A
検索/置換(F)...	Ctrl+F
<hr/>	
 Expression Builderを表示	

- ◆ **入力を元に戻す** : 行われた入力を元に戻します。
- ◆ **やり直し** : 最後のアクションをやり直します。
- ◆ **[切り取り]** : 選択したエリアを切り取ります。
- ◆ **コピー** : 選択したエリアをコピーします。
- ◆ **貼り付け** : クリップボード内の情報をメインスクリプト記述エリアに貼り付けます。
- ◆ **削除** : 選択した情報をメインスクリプト記述エリアから削除します。
- ◆ **すべて選択** : メインスクリプト記述エリア内の情報すべてを選択します。
- ◆ **検索 / 置換** : 指定した情報を検索および置換します。
- ◆ **Expression Builder を表示** : 式ビルダを起動します。詳細については、[165 ページのセクション 11.2.2 「式ビルダ」](#) を参照してください。

ECMAScript のコーディングヘルプ

- 1 メインスクリプト記述エリアの左余白を右クリックして、目的のオプションを選択します。

ブレイクポイントの切り替え(B)	
ブレイクポイントを使用可能にする(E)	
ブレイクポイントのプロパティ(P)...	
<hr/>	
ブックマークの追加(Q)...	
タスクの追加(T)...	
ブロックを元に戻す(U)	
追加した行を削除する(L)	
✓ クイック Diff の表示(Q)	Ctrl+Shift+Q
行番号の表示(N)	
<hr/>	
設定(F)...	

- ◆ **ブレイクポイントの切り替え** : 実装予定です。
- ◆ **ブレイクポイントの有効化** : ECMAScript にブレイクポイントを設定します。
- ◆ **ブレイクポイントのプロパティ** : ブレイクポイントのプロパティを表示します。
- ◆ **ブックマークの追加** : ECMAScript エディタ内の行にブックマークアイコンを配置します。
- ◆ **タスクの追加** : 完了する必要がある追加作業のリマインダとして行内にタスクアイコンを配置します。ツールバーから [タスク] ビューを開く場合、[ウィンドウ] > [ビューの表示] > [タスク] を選択するとタスクが表示されます。
- ◆ **Quick Diff の表示** : 実装予定です。

- ◆ **行番号を表示** : メインスクリプト記述領域に行番号を表示します。
- ◆ **初期設定** : 行区切りを設定して、ECMAScript エディタ内で作成したファイルのサフィックスを設定します。デフォルトでは、行区切りには変換はありません。サフィックスは js になります。

11.2.2 式ビルダ

式ビルダは、ECMAScript 式の作成の支援となります。式ビルダは、ECMAScript エディタを介して 2 つの方法でアクセスできます。ポリシービルダおよび引数ビルダからもアクセスできます。

ECMAScript エディタ内で式ビルダにアクセスする

- 1 ECMAScript エディタのメインスクリプト記述エリアで右クリックします。

または

ECMAScript エディタのシェルエリアで右クリックします。

ポリシービルダを介して式ビルダにアクセスする

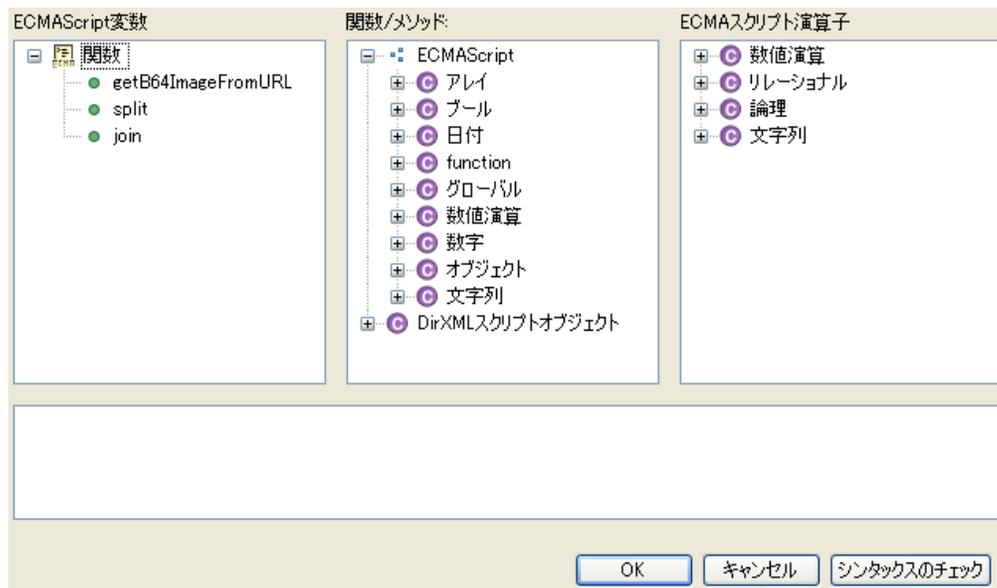
- 1 次のアクションまたは条件の隣にある [*ECMA Expression Builder* の起動] アイコンをクリックします。
 - ◆ XPath 式
 - ◆ XML 要素の追加
 - ◆ XML テキストの追加
 - ◆ XPath 式によるクローン
 - ◆ XML 属性の設定
 - ◆ ストリップ XPath 式

引数ビルダを介して式ビルダにアクセスする

- 1 XPath 名詞トークンをダブルクリックします。
- 2 引数ビルダ内の [*ECMA Expression Builder* の起動] アイコンをクリックします。

式ビルダには、[ECMAScript/ 変数]、[関数/ メソッド]、および [ECMAScript 演算子] の 3 つのペインがあります。

図 11-2 式ビルダ



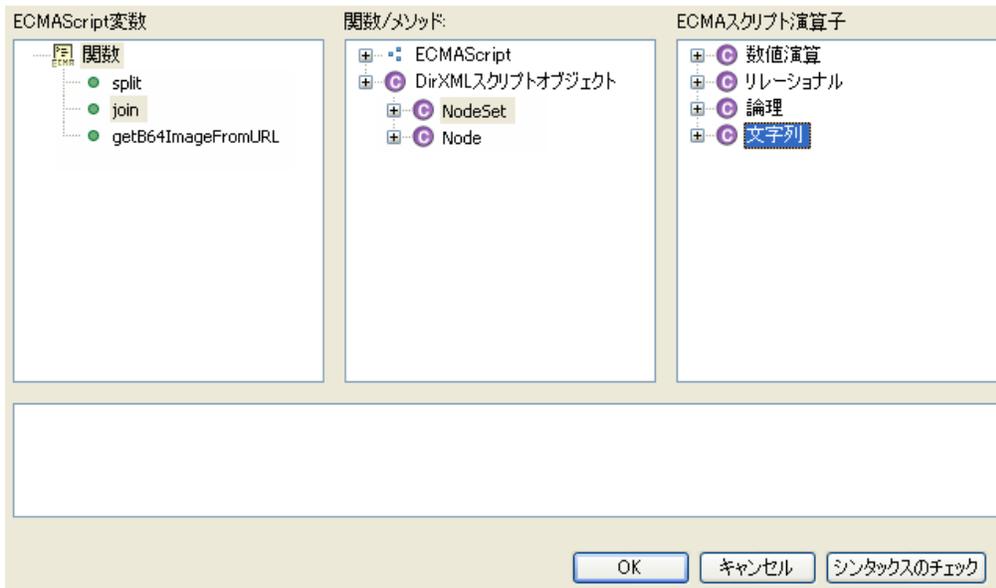
[*ECMAScript/ 変数*] は、ECMAScript 内の現在の定義済関数すべてを一覧表示します。
[*関数/ メソッド*] は、標準 ECMAScript 関数と DirXML スクリプト関数を含んでいます。
[*ECMAScript 演算子*] は、標準 ECMAScript 演算子を表示します。

式ビルダを使用する

- 1 (オプション) 目的の [*ECMAScript/ 変数*] をクリックします。
- 2 (オプション) 目的の [*関数/ メソッド*] をクリックします。
- 3 (オプション) 目的の [*ECMAScript 演算子*] をクリックします。
- 4 [*シンタックスのチェック*] をクリックして、式を評価します。
- 5 [*OK*] をクリックして式ビルダを閉じます。

次の例では、結合 ECMAScript 変数が NodeSet 関数またはメソッドと一緒に使用されていますが、ECMAScript 演算子は選択されていません。

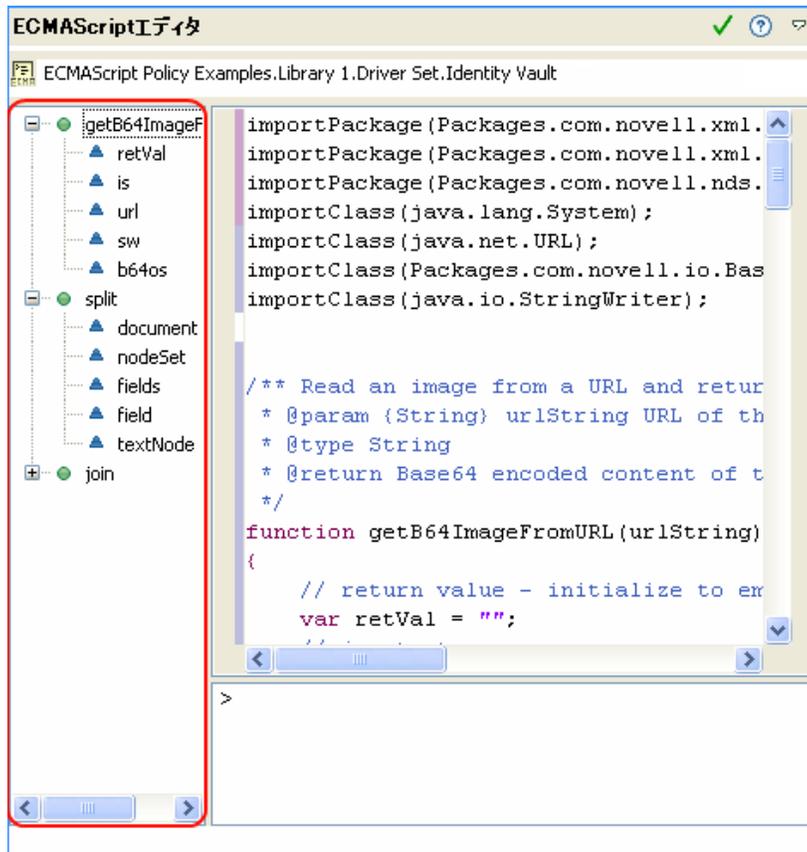
図 11-3 式ビルダの例



11.2.3 関数と変数

関数および変数が ECMAScript に定義されているため、ECMAScript エディタの左側に表示されます。

図 11-4 関数と変数

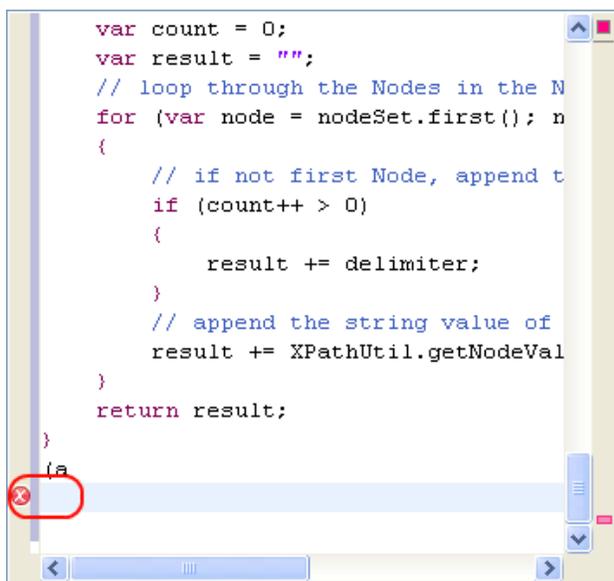


関数に格納されているすべての変数はグループ化されています。+アイコン (Linux の場合は矢印アイコン) をクリックすることによって、関数を展開してすべての変数を表示することができます。-アイコン (Linux の場合は矢印アイコン) をクリックすることによって、変数を持たない関数を表示することができます。

11.2.4 エラー表示

ECMAScript が作成されると、メインスクリプト記述エリアおよび [問題] ビューにエラーが表示されます。メインスクリプト記述エリアは、エラーが発生した行に赤い X でエラーを表示します。

図 11-5 メインスクリプト記述エリアのエラー



[問題] ビューは、ECMAScript が入力されるときにエラーを累積し、エラーの原因を表示します。

[問題] ビュー内のエラーをダブルクリックします。メインスクリプト記述エリア内の問題行にカーソルが移動します。

[問題] ビューにアクセスする

- 1 ツールバーで、[ウィンドウ] > [ビューの表示] > [その他] > [一般的な問題] を選択します。

[問題] ビューは ECMAScript エディタの下に表示されます。



11.2.5 シェルエリア

ECMAScript エリアのシェルエリアでは、ECMAScript を実行できます。ECMAScript が作成された後に、スクリプトの機能をテストすることができます。

図 11-6 シェルエリア

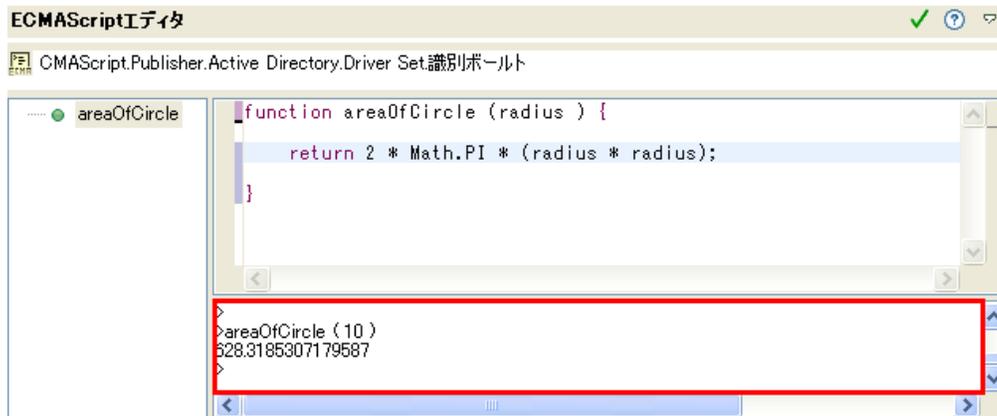


図 11-6 は、サークルエリアを決定する関数の例を含みます。関数は、`areaOfCircle(10)` 値を指定することによってテストされます。シェルは、値 `628.3185307179587` を表示します。

式を実行するには、`<Enter>` キーを押します。コンソールに複数のコード行を入力する場合は、数値キーパッドの `<Enter>` を押します。

11.3 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。

- ◆ 170 ページのセクション 11.3.1 「ECMAScript 関数を呼び出す DirXML スクリプトポリシー」
- ◆ 171 ページのセクション 11.3.2 「ECMAScript 関数をドライバレベルで呼び出す XSLT ポリシー」
- ◆ 172 ページのセクション 11.3.3 「ECMAScript 関数をスタイルシート内で呼び出す XSLT ポリシー」

11.3.1 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()` を呼び出すことによって、写真への URL リファレンスである属性を Base64 エンコード化写真データへ変換します。このポリシーは、入力変換または出力変換ポリシーとして使用できます。

関数は、URL からイメージを読み込み、コンテンツを Base64 エンコード化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 スクリプト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 11-7 写真の再フォーマットの例



ECMAScript は、getB64ImageFromURL 関数を呼び出し、この関数は現在の値を文字列として返します。

11.3.2 ECMAScript 関数をドライバレベルで呼び出す XSLT ポリシー

XSLT ポリシーは、単一カンマ区切り値を複数値に分割するか、または複数の値を単一カンマ区切り値に結合します。XSLT ポリシーは、ドライバレベルで定義され、入力変換または出力変換ポリシーとして使用できます。

注 : DirXML スクリプトは、分割および結合機能を備えていますが、XSLT にはありません。このタイプの関数は、XSLT が分割および結合機能を持つことを許可します。

次の 2 つの関数があります。

- ◆ [171 ページの「参加」](#)
- ◆ [172 ページの「分割」](#)

参加

結合関数は、NodeSet 内のノードのテキスト値を単一文字列に結合します。

```
<!-- 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 (入力 NodeSet) および区切り記号 ((optional: default = none) で分割する区切り記号)

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.xsl \(./samples/SplitJoin.xsl\)](#) は、XSLT スタイルシート内で結合または分割関数を呼び出します。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。

11.3.3 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、ECMAScript 関数定義の XSLT スタイルシートへの埋め込みをデモンストレーションします。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.xsl \(./samples/uppercase.xsl\)](#) 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。

11.4 JavaScript ファイル初期設定の変更

Designer は、JavaScript ファイルの表示方法および使用方法を変更できます。ECMAScript は、JavaScript であり、これらの初期設定は ECMAScript エディタに影響します。初期設定を変更する

- 1 Designer ツールバーで、[ウィンドウ]> [初期設定]> [Web およびXML]> [JavaScript ファイル] を選択します。
- 2 目的の設定を変更して、[OK] をクリックします。

すべての初期設定リストは、[173 ページのセクション 11.4.1 「JavaScript ファイルの初期設定」](#)を参照してください。

11.4.1 JavaScript ファイルの初期設定

Designer による JavaScript ファイルの処理方法を変更します。変更するための複数のオプションがあります。

- ◆ 173 ページの「JavaScript ファイル」
- ◆ 173 ページの「[JavaScript ファイル] > [JavaScript ソース]」
- ◆ 174 ページの「[JavaScript ファイル] > [JavaScript スタイル]」
- ◆ 176 ページの「JavaScript 検証」

JavaScript ファイル

JavaScript ファイルの作成方法を変更します。

図 11-8 JavaScript ファイル



表 11-1 JavaScript ファイル

設定	説明
ファイルの作成または保存: 線区切り記号	どのタイプの線区切り記号をファイルに適用するかを設定します。オプションは次のとおりです。 <ul style="list-style-type: none">◆ 変換なし◆ UNIX◆ Mac◆ Windows
ファイルの作成: このサフィックスを追加 (指定されていない場合)	ファイルにサフィックスを設定します。デフォルト値は <code>js</code> です。任意の値に設定できます。

[JavaScript ファイル] > [JavaScript ソース]

JavaScript ファイルのフォーマットを変更します。

図 11-9 [JavaScript ファイル] > [JavaScript ソース]

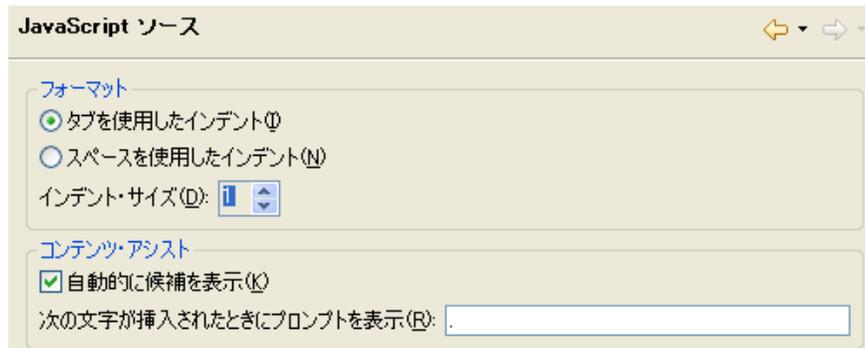


表 11-2 [JavaScript ファイル] > [JavaScript ソース]

設定	説明
フォーマット	<p>エディタのフォーマットを設定します。オプションは次のとおりです。</p> <ul style="list-style-type: none"> ◆ タブを使用したインデント ◆ スペースを使用したインデント ◆ インデントサイズ: 数値を設定することによってインデントサイズを変更します。
コンテンツ支援	<p>ファイルを作成するときのプロンプトをヘルプします。</p> <p><i>自動的に提案:</i> チェックボックスが選択されているかで有効または無効にすることができます。</p> <p><i>これらの文字が挿入されたときにプロンプト表示する:</i> 指定した文字が入力されたときにプロンプトを表示できるようにします。</p>

[JavaScript ファイル] > [JavaScript スタイル]

JavaScript ファイルのコンテンツをカスタマイズします。

図 11-10 [JavaScript ファイル] > [JavaScript スタイル]



表 11-3 [JavaScript ファイル] > [JavaScript スタイル]

設定	説明
コンテンツタイプ	<p>カスタマイズするコンテンツを選択します。それぞれの要素を個別に変更できます。変更可能なセクションは次のとおりです。</p> <ul style="list-style-type: none"> ◆ コメント ◆ デフォルトコード ◆ Keywords - ◆ リテラル文字列 ◆ 未完了の文字列およびコメント <p>サンプルテキストフィールド内の要素を選択した場合、コンテンツタイプは選択したタイプに変更されます。</p>
フォアグラウンド	<p>フォアグラウンドに設定された色を表示します。色フィールドをダブルクリックして別の色を選択します。</p>
背景	<p>バックグラウンドに設定された色を表示します。色フィールドをダブルクリックして別の色を表示します。</p>
太字	<p>要素を太字にすることができます。要素を選択して、[太字] チェックボックスを選択します。</p>

設定	説明
サンプルテキスト	変更を確認するためのサンプルファイルを表示します。

JavaScript 検証

エディタが JavaScript が入力されたときに検証できるようにします。[スクリプトを自動的に検証する] を選択して、スクリプトを自動的に検証します。選択されていない場合は、JavaScript は検証されません。

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。

- ◆ 178 ページの「If 関連付け」
- ◆ 180 ページの「If 属性」
- ◆ 183 ページの「If クラス名」
- ◆ 186 ページの「If ターゲット属性」
- ◆ 189 ページの「If ターゲット DN」
- ◆ 191 ページの「If エンタイトルメント」
- ◆ 194 ページの「If グローバル構成値」
- ◆ 196 ページの「If ローカル変数」
- ◆ 199 ページの「If 名前付きパスワード」
- ◆ 200 ページの「If 操作属性」
- ◆ 204 ページの「If 操作プロパティ」
- ◆ 206 ページの「If 操作」
- ◆ 209 ページの「If パスワード」
- ◆ 212 ページの「If ソース属性」
- ◆ 214 ページの「If ソース DN」
- ◆ 216 ページの「If XML 属性」
- ◆ 218 ページの「If XPath 式」
- ◆ 220 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

選択演算子に定義されている値を含みます。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、(220 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

条件は、比較が実行される方法を示す比較モードパラメータを持っています。

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。

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、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。

条件 **関連付け** ▼ ?

オペレータ* **使用可能** ▼

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

選択演算子に定義されている値を含みます。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、[\(220 ページ\) 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

条件は、比較が実行される方法を示す比較モードパラメータを持っています。

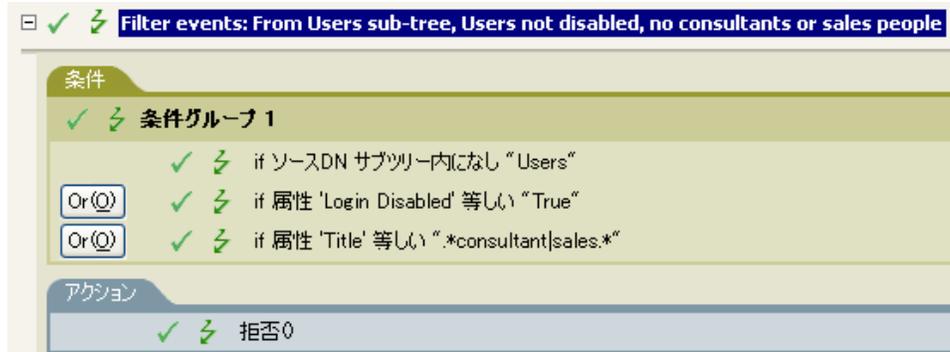
モード	説明
大文字と小文字の 区別あり	1文字ずつ比較する (大文字と小文字の区別あり)。
大文字と小文字の 区別なし	1文字ずつ比較する (大文字と小文字の区別なし)。
正規表現	正規表現が文字列全体に一致します。デフォルトで「大文字と小文字の区別あり」と設定されていますが、式の中でエスケープすることで変更できます。 Sun の Web サイト (http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html) を参照してください。 [パターン] のオプションには CASE_INSENSITIVE 、 DOTALL 、お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

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、[Identity Manager ポリシーのダウンロード \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html) 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 \(../samples/001-Event-FilterByContainerDisabledOrTitle.xml\)](http://www.novell.com/documentation/idm35/samples/001-Event-FilterByContainerDisabledOrTitle.xml) を参照してください。



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

条件

名前 *

オペレータ *

モード

値

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

選択演算子に定義されている値を含みます。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、[\(220 ページ\) 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

条件は、比較が実行される方法を示す比較モードパラメータを持っています。

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」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、Identity Manager ポリシーのダウンロード (<http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html>) を参照してください。ポリシーを XML で表示するには、004-Command-GroupChangeOnTitleChange.xml (../samples/004-Command-GroupChangeOnTitleChange.xml) を参照してください。

現在のオブジェクトのクラス名が「User」であるかどうかを確認します。

条件

オペレータ*

モード

値

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

選択演算子に定義されている値を含みます。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、[\(220 ページ\) 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

条件は、比較が実行される方法を示す比較モードパラメータを持っています。

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

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」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、[Identity Manager ポリシーのダウンロード \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html) 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 \(../samples/004-Command-GroupChangeOnTitleChange.xml\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html) を参照してください。

☐ ✓ ⚡ **User changing from Manager to Employee**

条件

✓ ⚡ 条件グループ 1

And(N) ✓ ⚡ if クラス名 等しい "User"

And(N) ✓ ⚡ if ターゲット属性 'Title' 等しい "*manager*"

And(N) ✓ ⚡ 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")

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。

条件 ?

名前 * 🔍

オペレータ * ▼

モード ▼

値 🔍

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

選択演算子に定義されている値を含みます。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、[\(220 ページ\) 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

条件 

オペレータ * 

値 

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

選択演算子に定義されている値を含みます。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、(220 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

条件は、比較が実行される方法を示す比較モードパラメータを持っています。

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

条件

名前 *

オペレータ *

モード

値

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 \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policycomponents.html\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policycomponents.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

選択演算子に定義されている値を含みます。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、[\(220 ページ\) 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

条件は、比較が実行される方法を示す比較モードパラメータを持っています。

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 <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
- ◆ 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

条件 

名前* 

オペレータ* 

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。

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 \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policycomponents.html\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policycomponents.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

選択演算子に定義されている値を含みます。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、[\(220 ページ\) 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

条件は、比較が実行される方法を示す比較モードパラメータを持っています。

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 <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
- ◆ 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」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、「Identity Manager ポリシーのダウンロード (<http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html>)」を参照してください。ポリシーを XML で表示するには、`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

条件グループ 1

- if ローカル変数 'manager-group-info' 使用可能
- 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", 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

条件グループ 1

- if クラス名 等しい "User"

And(N)

条件グループ 2

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

Or(O)

アクション

- ローカル変数の設定(" manager-group-dn", "Users#ManagerGroup")
- ローカル変数の設定(" manager-group-info", ソース属性("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", ソース属性("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。

条件

名前 *

オペレータ *

モード

値

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

条件

名前 *

オペレータ *

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。

演算子は次の場合に **True** を返します ...

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、内容は一組の <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

選択演算子に定義されている値を含みます。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、(220 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

条件は、比較が実行される方法を示す比較モードパラメータを持っています。

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

- ◆ 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

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、[Identity Manager ポリシーのダウンロード \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html) を参照してください。ポリシーを XML で表示するには、[003-Command-Add-CreateGroups.xml \(../samples/003-Command-AddCreateGroups.xml\)](#) を参照してください。

The screenshot shows a policy configuration window with the following sections:

- Set local variables to test existence of groups and for placement** (checked)
- Create ManagersGroup, if needed** (checked)
- Create EmployeesGroup, if needed** (checked)
- If Title indicates Manager, add to ManagerGroup and set rights** (unchecked)

The **条件** (Conditions) section is expanded to show:

- 条件グループ 1** (checked)
 - if クラス名 等しい "User" (checked)
 - And(N) if 操作属性 'Title' 等しい ".*manager.*" (checked)

The **アクション** (Actions) section is expanded to show:

- ターゲット属性値の設定 ("Group Membership", ローカル変数 ("manager-group-dn")) (checked)
- 操作属性のクローン ("Group Membership", "Security Equals") (checked)

At the bottom, another condition is visible: **If Title does not indicate Manager, add to EmployeeGroup and set rights** (unchecked).

条件

名前 *

オペレータ *

モード

値

条件は、役職の属性が正規表現である `*manager*` と等しいかどうかを確認します。つまり、`manager` の前に 0 個以上の文字を持ち、`manager` の後に 1 文字を持つ役職を検索して

います。ユーザオブジェクトの役職が `sales managers` であった場合、一致として検出されます。

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

選択演算子に定義されている値を含みます。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、[\(220 ページ\) 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

条件は、比較が実行される方法を示す比較モードパラメータを持っています。

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 <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
- ◆ 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

条件

名前 *

オペレータ *

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

選択演算子に定義されている値を含みます。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、[\(220 ページ\) 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
- ◆ チェックパスワード
- ◆ 削除
- ◆ 名前付きパスワードの取得
- ◆ パラメータの開始
- ◆ インスタンス
- ◆ 変更

- ◆ 関連付けの変更
- ◆ パスワード変更
- ◆ 移動
- ◆ 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

条件は、比較が実行される方法を示す比較モードパラメータを持っています。

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
- ◆ 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」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、[Identity Manager ポリシーのダウンロード \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html) 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。

The screenshot shows a policy configuration window titled "Set local variables to test existence of groups and for placement". It is divided into three sections: "条件" (Conditions), "アクション" (Actions), and a configuration area below.

- 条件 (Conditions):**
 - 条件グループ 1 (Condition Group 1):**
 - if クラス名 等しい "User"
 - 条件グループ 2 (Condition Group 2):**
 - if 操作 等しい "add"
 - if 操作 等しい "modify"
- アクション (Actions):**
 - ローカル変数の設定 ("manager-group-dn", "Users#ManagerGroup")
 - ローカル変数の設定 ("manager-group-info", ソース属性 ("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", ソース属性 ("Object Class", DN(ローカル変数("employee-group-dn"))))

条件

オペレータ*

モード

値

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

選択演算子に定義されている値を含みます。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、[\(220 ページ\) 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

条件は、比較が実行される方法を示す比較モードパラメータを持っています。

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

例

Novell 資格情報プロビジョニングポリシー Identity Manager 3.5.1 用 Novell 資格情報プロビジョニングポリシー (http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_credprov/data/bookinfo.html) を実行している場合、パスワード条件をし世するサンプルの購読者コマンド変換ポリシーがあります。サンプルファイルは SampleSubCommandTransform.xml と呼ばれます。このファイルは Identity Manager メディアの DirXML[®] Utilities フォルダにあります。詳細については、[サンプル資格情報プロビジョニングポリシー \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_credprov/data/credprovnlexample.html\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_credprov/data/credprovnlexample.html) を参照してください。ポリシーを XML で表示するには、[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 operations (if needed)
- ☑ ✓ ⚡ Add payload data to modify-password subscribe operations
- ☑ ✓ ⚡ Add payload data to add subscribe operations

条件

✓ ⚡ 条件グループ 1

- ✓ ⚡ if 操作 等しい "add"
- And(N) ✓ ⚡ if パスワード 使用可能

アクション

- ✓ ⚡ XMLエレメントの追加("sso-sync-data", "operation-data")
- ✓ ⚡ XMLエレメントの追加("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", ソース属性("DirXML-ADContext"))
- ✓ ⚡ XMLエレメントの追加("sso-app-username", "operation-data/sso-sync-data")
- ✓ ⚡ XMLテキストの追加("operation-data/sso-sync-data/sso-app-username", ソース属性("CN"))
- ✓ ⚡ XMLエレメントの追加("password", "operation-data/sso-sync-data")
- ✓ ⚡ XMLテキストの追加("operation-data/sso-sync-data/password", パスワード)
- ✓ ⚡ XMLエレメントの追加("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", ソース属性("workforceID"))

条件 ?

オペレータ*

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

選択演算子に定義されている値を含みます。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、[\(220 ページ\) 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

条件は、比較が実行される方法を示す比較モードパラメータを持っています。

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。
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

条件

名前 *

オペレータ *

モード

値

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

選択演算子に定義されている値を含みます。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、[\(220 ページ\) 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、[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 \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_designer/data/prfilterexcludesubtree.html#prfilterexcludesubtree\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_designer/data/prfilterexcludesubtree.html#prfilterexcludesubtree) を参照してくだ

さい。ポリシーを XML で表示するには、[predef_transformation_exclude_subtree.xml](#) ([../samples/predef_transformation_exclude_subtrees.xml](#)) を参照してください。

The screenshot shows a configuration window titled "Event Transformation - Scope Filtering - Exclude subtree(s)". Below the title, it says "No description available". There are two main sections: "条件" (Condition) and "アクション" (Action). Under "条件", there is a sub-section "条件グループ 1" (Condition Group 1) containing a single condition: "if ソースDN サブツリー内にあり "[Enter a subtree to exclude]"". Under "アクション", there is a single action: "拒否" (Deny).

条件

オペレータ*

値

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る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

選択演算子に定義されている値を含みます。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、[\(220 ページ\) 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

条件は、比較が実行される方法を示す比較モードパラメータを持っています。

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。

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

条件

名前 *

オペレータ *

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

ポリシーで XPath 式を使用する詳細については、[XPath 1.0 式 \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policyxpathexpression.html#policyxpathexpression\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policyxpathexpression.html#policyxpathexpression) を参照してください。

例

Novell 資格情報プロビジョニングポリシーを実装している場合、XPath 式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あります。サンプルファイルは SampleSubCommandTransform.xml と呼ばれます。このファイルは Identity Manager メディアの DirXML Utilities フォルダにあります。詳細については、[サンプル資格情報プロビジョニングポリシー \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_credprov/data/credprovnlexample.html\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_credprov/data/credprovnlexample.html) を参照してください。ポリシーを XML で表示するには、[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 operations (if needed)

条件

✓ ↺ 条件グループ 1

- ✓ ↺ if 操作 等しい "add"
- And(N) ✓ ↺ if パスワード 使用可能
- And(N) ✓ ↺ if XPath式 Trueではない "operation-data"

Or(O)

✓ ↺ 条件グループ 2

- ✓ ↺ if 操作 等しい "modify-password"
- And(N) ✓ ↺ 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

条件 XPath式

オペレータ * Trueではない

値

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、形式が `$<variable-name>$` の埋め込みリファレンスが指定した名前を持つローカルまたはグローバル変数の値と置き換えられます。`$<variable-name>$` は、正しい変数名でなければなりません。正しい XML 名の詳細については、[W3C Extensible Markup Language \(XML\) \(http://www.w3.org/TR/2006/REC-xml11-20060816/#sec-suggested-names\)](http://www.w3.org/TR/2006/REC-xml11-20060816/#sec-suggested-names) を参照してください。

指定した変数が存在しない場合は、リファレンスは空の文字列と置き換えられます。単一の `$` を使用して、それを変数リファレンスとして解釈しない場合は、`$` を追加してエスケープする必要があります (例 : You owe me \$\$100.00)。

アクションは、包含するルールを条件を満たした場合に実行されます。いくつかのアクションには [モード] フィールドがあります。モードは、ポリシーを実行しているコンテキストと選択したモードに互換性がない場合には、実行時に無視されます。

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

- ◆ 223 ページの「関連付けの追加」
- ◆ 224 ページの「ターゲット属性値の追加」
- ◆ 226 ページの「ターゲットオブジェクトの追加」
- ◆ 228 ページの「ソース属性値の追加」
- ◆ 229 ページの「ソースオブジェクトの追加」
- ◆ 230 ページの「XML 要素の追加」
- ◆ 232 ページの「XML テキストの追加」
- ◆ 234 ページの「中断」
- ◆ 235 ページの「ターゲット属性値のクリア」
- ◆ 236 ページの「操作プロパティのクリア」
- ◆ 237 ページの「ソース属性値のクリア」
- ◆ 238 ページの「SSO 資格情報のクリア」
- ◆ 239 ページの「XPath 式によるクローン」
- ◆ 240 ページの「操作属性のクローン」
- ◆ 241 ページの「ターゲットオブジェクトの削除」
- ◆ 242 ページの「ソースオブジェクトの削除」
- ◆ 243 ページの「一致オブジェクトの検索」
- ◆ 246 ページの「毎」
- ◆ 247 ページの「イベントの生成」
- ◆ 250 ページの「If」
- ◆ 252 ページの「エンタイトルメントの実装」
- ◆ 253 ページの「ターゲットオブジェクトの移動」
- ◆ 255 ページの「ソースオブジェクトの移動」
- ◆ 256 ページの「操作属性値の再フォーマット」
- ◆ 257 ページの「関連付けの削除」
- ◆ 258 ページの「ターゲット属性値の削除」
- ◆ 259 ページの「ソース属性値の削除」
- ◆ 260 ページの「ターゲットオブジェクトの名前変更」
- ◆ 261 ページの「操作属性の名前変更」
- ◆ 262 ページの「ソースオブジェクトの名前変更」

- ◆ 263 ページの「電子メールの送信」
- ◆ 265 ページの「テンプレートから電子メールを送信」
- ◆ 267 ページの「デフォルト属性値の設定」
- ◆ 269 ページの「ターゲット属性値の設定」
- ◆ 271 ページの「ターゲットパスワードの設定」
- ◆ 272 ページの「ローカル変数の設定」
- ◆ 274 ページの「操作関連付けの設定」
- ◆ 275 ページの「操作クラス名の設定」
- ◆ 276 ページの「操作ターゲット DN」
- ◆ 277 ページの「操作プロパティの設定」
- ◆ 278 ページの「操作ソース DN の設定」
- ◆ 279 ページの「操作テンプレート DN の設定」
- ◆ 280 ページの「ソース属性値の設定」
- ◆ 282 ページの「ソースパスワードの設定」
- ◆ 283 ページの「SSO 資格情報の設定」
- ◆ 284 ページの「SSO パスフレーズの設定」
- ◆ 285 ページの「XML 属性の設定」
- ◆ 286 ページの「ステータス」
- ◆ 287 ページの「ワークフローの開始」
- ◆ 289 ページの「操作属性のストリップ」
- ◆ 290 ページの「XPath のストリップ」
- ◆ 291 ページの「メッセージのトレース」
- ◆ 293 ページの「拒否」
- ◆ 294 ページの「操作属性値がない場合は拒否」
- ◆ 295 ページの「While」
- ◆ 296 ページの「変数の拡張」

関連付けの追加

関連付けの追加コマンドを指定した関連付けとともに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。

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 ▼

? 現在のオブジェクトを使用するには、以下のDNフィールドの空白をそのままにします。

DNの指定: ソースDN0 📄

関連付けを指定してください: * 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、[\(296 ページ\) 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、[\(296 ページ\) 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の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

DN、関連付け、または現在のオブジェクトをターゲットオブジェクトとして指定します。

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、「[コマンド変換 - 部署別コンテナの作成 - パート 1 およびパート 2 \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_designer/data/prdeptcontainer.html#prdeptcontainer\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_designer/data/prdeptcontainer.html#prdeptcontainer)」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 \(../samples/predef_command_create_dept_container1.xml\)](#) 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。

Command Transformation - Create Department Container - Part 2

条件

- 条件グループ 1
 - if 操作 等しい ""add""

アクション

- ローカル変数の設定("target-container", スコープ="policy", ターゲットDN(長さ="-2")+ソースDN)
- ローカル変数の設定("does-target-exist", スコープ="policy", ターゲット属性("objectclass", クラス名="Organizational Unit", DN(ローカル変数("target-container"))))

Command Transformation - Create Departmental Container - Part 2

No description available

条件

- 条件グループ 1
 - if ローカル変数 'does-target-exist' 使用可能
 - And(N) 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="Organizational Unit", 直接="true", DN(ローカル変数("target-container")))
- ターゲット属性値の設定("ou", クラス名="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 ?

属性名を指定してください: * 🔍

クラス名を指定してください: 🔍

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の指定: * Local Variable("target-container") 📄

値のタイプを指定してください: string ▼

文字列を入力: * Parse DN("dest-dn", "dot", length="1", start="-1", Local \ 📄

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ターゲットデータストアに作成します。名前と場所には、[DN を入力] フィールドで指定した名前と場所を使用します。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、[\(296 ページ\) 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、同じ DN を使用して以降の [ターゲット属性値の追加](#)アクションで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。詳細については、事前定義されたルールで「[コマンド変換 - 部署別コンテナの作成 - パート 1 およびパート 2 \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_designer/data/prdeptcontainer.html#prdeptcontainer\)](#)」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 \(./samples/predef_command_create_dept_container1.xml\)](#) 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。



Command Transformation - Create Departmental Container - Part 2

No description available

条件

条件グループ 1

- if ローカル変数 'does-target-exist' 使用可能
- And(N) 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="Organizational Unit", 直接="true", DN(ローカル変数("target-container")))
- ターゲット属性値の設定("ou", クラス名="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

クラス名を指定してください: * 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の指定: * Local Variable("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、[\(296 ページ\) 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、[\(296 ページ\) 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の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

DN、関連付け、または現在のオブジェクトをターゲットオブジェクトとして指定します。

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

実行内容 | ソース属性値の追加

属性名を指定してください: *	<input type="text" value="Title"/>	<input type="button" value="🔍"/>
クラス名を指定してください:	<input type="text" value="User"/>	<input type="button" value="🔍"/>
オブジェクトを選択:	<input type="button" value="関連付け"/>	<input type="button" value="▼"/>
関連付けを指定してください: *	<input type="text" value="Source Name0"/>	<input type="button" value="📄"/>
値のタイプを指定してください:	<input type="button" value="string"/>	<input type="button" value="▼"/>
文字列を入力: *	<input type="text" value="Manager"/>	<input type="button" value="📄"/>

ソースオブジェクトの追加

[DN] フィールドで指定した名前と場所を使用して、指定したタイプのオブジェクトを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、同じ DN を使用して以降の **ソース属性値の追加** アクションで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、[\(296 ページ\) 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**  

クラス名を指定してください: * 

DNの指定: * 

XML 要素の追加

XPath 式で選択された要素のセットに、[名前] フィールドで指定した名前を使用してカスタム要素を追加します。[XPath 式の前] を指定していない場合、新しい要素は選択した要素の既存の子の後に追加されます。[XPath 式の前] を指定した場合、式によって選択された各要素と比較して評価され、どの子の前に挿入するか決定されます。[XPath 式の前] が空のノードセット、または選択された要素の子を含まないノードセットに評価された場合、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、新しい要素は、すでに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、ネームスペースプレフィックスを含めることができます (プレフィックスがすでにこのポリシーで定義されている場合)。変数の拡張がサポートされます。詳細については、(296 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

新しい要素が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に対して相対的に評価される XPath 1.0 式を指定します。

備考

ポリシーで XPath 式を使用する場合の詳細については、「[XPath 1.0 式 \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policyxpathexpression.html#policyxpathexpression\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policyxpathexpression.html#policyxpathexpression)」を参照してください。

例

Novell 資格情報プロビジョニングポリシーを実装している場合、XPath 式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あります。サンプルファイルは SampleSubCommandTransform.xml という名前です。このファイルは Identity Manager のメディアの DirXML[®] Utilities フォルダにあります。詳細については、「[資格情報プロビジョニングポリシーの例 \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_credprov/data/credprovnlsexample.html\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_credprov/data/credprovnlsexample.html)」を参照してください。このポリシーを XML で表示するには、[SampleSubCommandTransform.xml \(./samples/SampleSubCommandTransform.xml\)](#) を参照してください。

サンプルファイルは、XML 要素の追加アクションを使用して、プロビジョニング時に Novell[®] SecureLogin または Novell SecretStore[®] の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 operations (if needed)
- ☑️ ⚡️ Add payload data to modify-password subscribe operations
- ☑️ ⚡️ Add payload data to add subscribe operations

条件

- ☑️ ⚡️ 条件グループ 1
 - ☑️ ⚡️ if 操作 等しい "add"
 - ☑️ ⚡️ if パスワード 使用可能

アクション

- ☑️ ⚡️ XMLエレメントの追加("sso-sync-data", "operation-data")
- ☑️ ⚡️ XMLエレメントの追加("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", ソース属性("DirXML-ADContext"))
- ☑️ ⚡️ XMLエレメントの追加("sso-app-username", "operation-data/sso-sync-data")
- ☑️ ⚡️ XMLテキストの追加("operation-data/sso-sync-data/sso-app-username", ソース属性("CN"))
- ☑️ ⚡️ XMLエレメントの追加("password", "operation-data/sso-sync-data")
- ☑️ ⚡️ XMLテキストの追加("operation-data/sso-sync-data/password", パスワード)
- ☑️ ⚡️ XMLエレメントの追加("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", ソース属性("workforceID"))

実行内容 XMLエレメントの追加 ▼ ?

要素名の入力: * 🔍

XPath式を指定してください: * 🏠 📄 🗑️

挿入: ▼

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] を指定していない場合、テキストは選択した要素の既存の子の後に追加されます。[XPath 式の前] を指定した場合、式によって選択された各要素と比較して評価され、どの子の前に挿入するか決定されます。[XPath 式の前] が空のノードセット、または選択された要素の子を含まないノードセットに評価された場合、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、テキストは、選択されたノードの子である「before」で選択されたノードセット内の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

テキストが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に対して相対的に評価される XPath 1.0 式を指定します。

文字列

追加するテキストを指定します。

備考

ポリシーで XPath 式を使用する場合の詳細については、「[XPath 1.0 式 \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policyxpathexpression.html#policyxpathexpression\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policyxpathexpression.html#policyxpathexpression)」を参照してください。

例

Novell 資格情報プロビジョニングポリシーを実装している場合、XPath 式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あります。サンプルファイルは SampleSubCommandTransform.xml という名前です。このファイルは Identity Manager のメディアの DirXML Utilities フォルダにあります。詳細については、「[資格情報プロビジョニングポリシーの例 \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_credprov/data/credprovnlsexample.html\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_credprov/data/credprovnlsexample.html)」を参照してください。このポリシーを XML で表示するには、[SampleSubCommandTransform.xml \(../samples/SampleSubCommandTransform.xml\)](#) を参照してください。

この例では、XML テキストの追加アクションを使用して、Novell SecureLogin または Novell SecretStore のアプリケーションユーザ名を検索します。アプリケーション名を取得することによって、プロビジョニング時にユーザオブジェクトに対して資格情報を設定できます。

- ☑ ✓ ⚡ Add operation-data element to password subscribe operations (if needed)
- ☑ ✓ ⚡ Add payload data to modify-password subscribe operations
- ☑ ✓ ⚡ Add payload data to add subscribe operations

条件

- ✓ ⚡ 条件グループ 1
 - ✓ ⚡ if 操作 等しい "add"
 - And(N) ✓ ⚡ if パスワード 使用可能

アクション

- ✓ ⚡ XMLエレメントの追加("sso-sync-data", "operation-data")
- ✓ ⚡ XMLエレメントの追加("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", ソース属性("DirXML-ADContext"))
- ✓ ⚡ XMLエレメントの追加("sso-app-username", "operation-data/sso-sync-data")
- ✓ ⚡ XMLテキストの追加("operation-data/sso-sync-data/sso-app-username", ソース属性("CN"))
- ✓ ⚡ XMLエレメントの追加("password", "operation-data/sso-sync-data")
- ✓ ⚡ XMLテキストの追加("operation-data/sso-sync-data/password", パスワード)
- ✓ ⚡ XMLエレメントの追加("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", ソース属性("workforceID"))

実行内容 XMLテキストの追加 ▼ ?

XPath式を指定してください: * 🔍 📄 🗑️

文字列を指定してください: * 📄

挿入: ▼

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

実行内容 中断  

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、[\(296 ページ\) 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、[\(296 ページ\) 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の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

DN、関連付け、または現在のオブジェクトをターゲットオブジェクトとして選択します。

例

実行内容 | ターゲット属性値のクリア  

属性名を指定してください: * 

クラス名を指定してください: 

モードを選択: 

オブジェクトを選択: 

DNの指定: * 

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 / 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、[\(296 ページ\) 変数の拡張](#)を参照してください。

例

実行内容 操作プロパティのクリア ?

プロパティ名を指定してください: *

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。詳細については、[\(296 ページ\) 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、[\(296 ページ\) 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の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

DN、関連付け、または現在のオブジェクトをターゲットオブジェクトとして選択します。

例

実行内容 ソース属性値のクリア ▼ ?

属性名を指定してください: * 🔍

クラス名を指定してください: 🔍

オブジェクトを選択: ▼

DNの指定: * 📄

SSO 資格情報のクリア

Single Sign On 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。使用される文字列の数と名前は、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 資格情報プロビジョニングポリシー (http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_credprov/data/bookinfo.html)」を参照してください。

フィールド

資格情報リポジトリオブジェクトの 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、(296 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、(296 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各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を指定してください: * 

DN相対ポリシーを設定する

ターゲットユーザDNを指定してください: * 

[アプリケーションオブジェクトから次を入力](#)

アプリケーション資格情報IDを指定してください: *

ログインパラメータ文字列を指定してください: 

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] を指定していない場合、非属性のクローンノードは選択した要素の既存の子の後に追加されます。[XPath 式の前] を指定した場合、式によって選択された各要素と比較して評価され、どの子の前に挿入するか決定されます。[XPath 式の前] が空のノードセット、または選択された要素の子を含まないノードセットに評価された場合、非属性のクローンノードは既存の子の後に追加されます。それ以外の場合、非属性のクローンノードは、すでに選択されているノードセット内にあり、選択し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。

挿入

XPath 式を、ソース XPath 式の前に挿入するか、または XPath 式をターゲット XPath 式内の現在のノードの後に追加するかを選択します。

備考

ポリシーで XPath 式を使用する場合の詳細については、「XPath 1.0 式 (<http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policyxpathexpression.html#policyxpathexpression>)」を参照してください。

例

実行内容 XPath式によるクローン  

ソースXPath式を指定してください: *   

宛先XPath式を指定してください: *   

挿入: 

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、[\(296 ページ\) 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、[\(296 ページ\) 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、「[Identity Manager ポリシーのダウンロード \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html)」を参照してください。このポリシーを XML で表示するには、[003-AddCreateGroups.xml \(../samples/003-AddCreateGroups.xml\)](#) を参照してください。

The screenshot shows a policy configuration window with the following sections:

- Conditions:**
 - Condition Group 1
 - if クラス名 等しい "User"
 - And(N)
 - if 操作属性 'Title' 等しい ".*manager.*"
- Actions:**
 - ターゲット属性値の設定 ("Group Membership", ローカル変数 ("manager-group-dn"))
 - 操作属性のクローン ("Group Membership", "Security Equals")

Below the configuration, there are two input fields:

- ソース名を指定してください: * Group Membership
- 宛先名を指定してください: Security Equals

実行内容 操作属性のクローン

ソース名を指定してください: * Group Membership

宛先名を指定してください: 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にある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

DN、関連付け、または現在のオブジェクトをターゲットオブジェクトとして選択します。

例

実行内容 ターゲットオブジェクトの削除 ?

クラス名を指定してください: 

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の指定: * 

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にある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

DN、関連付け、または現在のオブジェクトをターゲットオブジェクトとして選択します。

例

実行内容 

クラス名を指定してください: 

オブジェクトを選択: 

DNの指定: * 

一致オブジェクトの検索

ターゲットデータストア内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] です。

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] で、一致属性が指定されている場合には、結果は定義されません。ターゲットデータストアが接続されたアプリケーションの場合は、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ルトの場合は、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 は「￼」という 1 文字に設定されます。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 は「󿿽」という 1 文字に設定されます。

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、「一致 - 属性値別 (http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_designer/data/prmatchattrvalue.html#prmatchattrvalue)」を参照してください。このポリシーを XML で参照するには、[predef_match_by_attribute.xml \(../samples/predef_match_by_attribute.xml\)](#) を参照してください。

Matching - by attribute value
 No description available

条件
 条件グループ 1
 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(スコープ="subtree", DN("Enter base DN to start search"), 一致 ("Enter name of attribute to match on"))

実行内容

スコープを選択:

DNの指定:

一致属性を指定してください:

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

一致属性

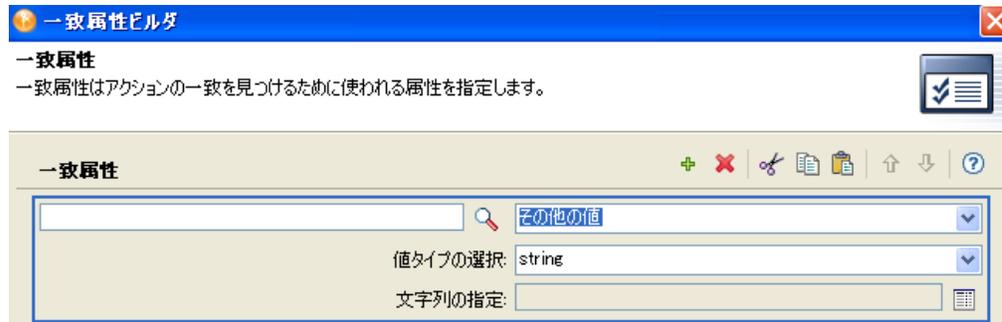
<input type="text" value="CN"/>	<input type="button" value="現在のオブジェクトの値を使用する"/>
<input type="text" value="L"/>	<input type="button" value="現在のオブジェクトの値を使用する"/>

左のフィールドには一致する属性が保存されます。右のフィールドでは、一致する現在のオブジェクトの値または別の値を使用するよう指定できます。[その他の値] を選択した場合は、複数の値タイプから指定します。

- ◆ 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

[その他の値] を使用する

- 1 一致属性ビルダを起動して、[その他の値] を選択します。



- 2 目的の値タイプを選択します。
- 3 値を指定して [OK] をクリックします。

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。

ノードセット内の現在のノードがエンタイトルメント要素の場合、アクションは同時に**エンタイトルメントの実装**アクションでも囲まれているときと同様にマークされます。現在のノードがクエリによって返されたクエリ要素の場合、そのトークンを使用して、自動的に次のクエリ結果のバッチを取得して処理します。

例

実行内容 | 毎 

ノードセットを指定してください: * 

アクションを指定してください: * 

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 

属性名を指定してください: * 

クラス名を指定してください: 

モードを選択: 

オブジェクトを選択: 

DNの指定: * 

値のタイプを指定してください: 

文字列を入力: * 

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せん。詳細については、[\(296 ページ\) 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。
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 す。
value3	ここに入力される任意の数字は、 value3 イベントフィールドに格納され ます。
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**) で構成されます。テキストフィールドは 256 バイトに制限されています。データフィールドには 3KB までの情報を含めることができます。ただし、環境によってはこれより大きいデータフィールドを使用できる場合もあります。

例

この例には 4 つのルールがあり、名字属性の最初の文字に基づいてユーザオブジェクト用の配置ポリシーを実装しています。トレースメッセージおよび Novell Audit または Sentinel カスタム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」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、「Identity Manager ポリシーのダウンロード (<http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html>)」を参照してください。このポリシーを XML で表示するには、001-Placement-BySurname.xml (../samples/001-Placement-BySurname.xml) を参照してください。

The screenshot shows a configuration window for a policy. At the top, there are two expandable sections: 'Setup Local Variables' and 'Surname A-I: place in Users1'. The 'Surname A-I: place in Users1' section is expanded to show its configuration. It has a 'Conditions' (条件) section with a 'Condition Group 1' (条件グループ 1) containing two conditions: 'if class name equals "User"' and 'if operation attribute "Surname" match "[a-i]*"'. Below the conditions is an 'Actions' (アクション) section with three actions: 'Set operation target DN' (操作ターゲットDNの設定), 'Trace message' (メッセージのトレース), and 'Generate event' (イベントの生成). At the bottom, there are three more expandable sections: 'Surname J-R: place in Users2', 'Surname S-Z: place in Users3', and 'Surname S-Z: place in Users3'.

実行内容 イベントの生成 ?

IDを指定してください: * 1000

レベルを選択: 情報

文字列を指定してください: 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。

名前	文字列の値:
text1	ローカル変数("LVUsers1")

イベントの生成は、ID 1000 を持つイベントを作成し、ローカル変数 LVUser1 によって生成されたテキストを表示します。ローカル変数 LVUser1 は、ユーザ: 操作属性 “cn” + “+” Training\Users\Active\Users1” +” container” に追加したという文字列です。このイベントは、ユーザ :jsmith を Training\Users\Active\Users1 コンテナに追加したことを示します。

If

条件付きでアクションのセットを実行します。

フィールド

条件

目的の条件を指定します。

アクションを実行する

条件が True の場合に、目的のアクションを指定します。

他のアクションを実行する

(オプション)条件が False の場合に、目的のアクションを指定します。

例

追加または変更操作中に、役職の属性がマネージャと等しい場合は、ユーザオブジェクトが **ManagerGroup** グループに追加されます。役職がマネージャでない場合は、ユーザオブジェクトは **UsersGroup** グループに追加されます。このポリシーを XML で表示するには、[if.xml \(../samples/if.xml\)](#) を参照してください。

実行内容	if
条件:	and(if 操作属性 'Title' 等しい "manager")
アクションを実行する:	do-set-dest-attr-value
他のアクションを実行する:	do-set-default-attr-value

ifアクションを作成するときには、条件と1つのアクションを追加する必要があります。この例では、2つの別個のアクションがあります。この条件は、ユーザオブジェクトがマネージャの役職を持っているかどうかです。

条件リスト

✓ ↺ if 操作属性 'Title' 等しい "manager"

このアクションは、ユーザオブジェクトを ManagerGroup グループに追加するためのものです。

アクションリスト

✓ ↺ ターゲット属性値の設定 ("Group Membership", クラス名="User", "Novell#Users#ManagerGroup")

役職がマネージャでない場合は、ユーザオブジェクトは UsersGroup グループ内に配置されます。

アクションリスト

✓ ↺ デフォルト属性値の設定 ("Group Membership", write-back="false", "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を、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

実行内容 

ノードセットを指定してください: * 

アクションを指定してください: * 

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 

属性名を指定してください: * 

クラス名を指定してください: 

モードを選択: 

オブジェクトを選択: 

DNの指定: * 

値のタイプを指定してください: 

文字列を入力: * 

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。

移動先のコンテナ

オブジェクトを受け取るコンテナを選択します。このコンテナは、DN または関連付けによって指定します。

DN または関連付け

コンテナの DN または関連付けを使用するかどうかを指定します。

例

この例は1つのルールで構成されています。このルールは、説明属性がユーザアカウント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」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、「Identity Manager ポリシーのダウンロード (<http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html>)」を参照してください。このポリシーを XML で表示するには、005-Command-DisableMoveOnTermination (../samples/005-Command-DisableMoveOnTermination.xml) を参照してください。

The screenshot shows a configuration window for a rule titled "On Termination, disable user and move to Disabled container". The interface is divided into two main sections: "条件" (Conditions) and "アクション" (Actions).

条件 (Conditions):

- 条件グループ 1 (Condition Group 1) is expanded, showing three conditions:
 - if 操作 等しい "modify" (if operation is "modify")
 - if クラス名 等しい "User" (if class name is "User")
 - if 操作属性 'Description' 等しい "~terminated.*" (if operation attribute 'Description' is "~terminated.*")

アクション (Actions):

- ターゲット属性値の設定 ("Login Disabled", クラス名="true", "True") (Set target attribute value ("Login Disabled", class name="true", "True"))
- ターゲットオブジェクトの移動 (when="after", DN("Users#Disabled")) (Move target object (when="after", DN("Users#Disabled")))

実行内容 ターゲットオブジェクトの移動 ?

クラス名を指定してください:

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の指定: * Users\Disabled

このポリシーは、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

クラス名を指定してください:

移動するオブジェクトを選択:

DNの指定: *

移動するコンテナを選択:

DNの指定: *

操作属性値の再フォーマット

パターンを使用して、現在の操作内にある属性の値すべて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、[\(296 ページ\) 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、(nnn)-nnn-nnnn から nnn-nnn-nnnn 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、「[入出力変換 - 電話番号の形式を \(nnn\) nnn-nnnn から nnn-nnn-nnnn に変更 \(http://www.novell.com/documentation/idm35/index.html?page=documentation/idm35/policy_designer/data/prreformattele1.html#prreformattele1\)](http://www.novell.com/documentation/idm35/index.html?page=documentation/idm35/policy_designer/data/prreformattele1.html#prreformattele1)」を参照してください。このポリシーを XML で表示するには、`predef_transformation_reformat_telephone1.xml` (`../samples/predef_transformation_reformat_telephone1.xml`) を参照してください。

The screenshot shows the configuration for a transformation rule titled "Input or Output Transformation - Reformat Telephone number from (nnn) nnn-nnnn to nnn-nnn-nnnn". The rule has no description available. It contains one condition group named "条件グループ 1" and one action named "操作属性の再フォーマット". The action configuration is as follows:

- 名前を指定: * phone
- 値のタイプを指定してください: string
- 文字列を入力: * 最初を置換("^(%d%d%d)-(d%d%d)-(d%d%d)d\$", "(%1) \$2-\$3", ローカル変数("current-value"))

操作属性の再フォーマットアクションにより、電話番号のフォーマットが変更されます。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%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、削除操作を使用して代わりにユーザオブジェクトを無効にし、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、「[コマンド変換 - 無効にする発行者の削除 \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_designer/data/prdeletetodisable.html#prdeletetodisable\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_designer/data/prdeletetodisable.html#prdeletetodisable)」を参照してください。このポリシーを XML で表示するには、`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 titled "Command Transformation - Publisher Delete to Disable". It contains a "Condition" section with two rules: "if 操作 等しい 'delete'" and "if クラス名 等しい 'User'", connected by an "Or()" operator. The "Action" section contains two actions: "ターゲット属性値の追加 ('Login Disabled', 'true')" and "関連付けを削除 (関連付け (関連付け))". Below the configuration, the "実行内容" (Execution Content) is set to "関連付けを削除". The "モードを選択" (Select Mode) dropdown is set to "現在の操作に追加する" (Add to current operation). The "関連付けを指定してください" (Specify Association) field contains "Association0".

ユーザオブジェクトに対して削除操作が実行されると、「ログインの無効化」属性の値が True に設定され、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、[\(296 ページ\) 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、[\(296 ページ\) 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 ▼ ?

属性名を指定してください: *	<input type="text" value="Member"/>	
クラス名を指定してください:	<input type="text"/>	
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の指定: *	<input type="text" value="Novell#Users#ManagerGroup"/>	
値のタイプを指定してください:	string	▼
文字列を入力: *	<input type="text" value="Destination DN0"/>	

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、[\(296 ページ\) 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、[\(296 ページ\) 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指定します。

文字列

削除する属性値を指定します。

例

実行内容 **ソース属性値の削除**  

属性名を指定してください: * 

クラス名を指定してください: 

オブジェクトを選択: 

DNの指定: * 

値のタイプを指定してください: 

文字列を入力: * 

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

実行内容 ターゲットオブジェクトの名前変更 ?

クラス名を指定してください: 🔍

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の指定: * 📄

文字列を指定してください: * 📄

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、[\(296 ページ\) 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、[\(296 ページ\) 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**  

ソース名を指定してください: * 

宛先名を指定してください: 

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

実行内容 | ソースオブジェクトの名前変更  

クラス名を指定してください: 

オブジェクトを選択: 

DNの指定: * 

文字列を指定してください: * 

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、(296 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、(296 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 サーバアカウントのパスワードを名前付きパスワードとしてドライバオブジェクトに保存できます。これにより、パスワードが暗号化されます。それ以外の場合は、入力したパスワードはクリアテキストとして保存されます。名前付きパスワードの詳細については、『[Novell Identity Manager 管理ガイド \(http://www.novell.com/documentation/idm35/index.html\)](http://www.novell.com/documentation/idm35/index.html)』の「Using Named Password (名前付きパスワードの使用)」を参照してください。

文字列

複数の電子メールアドレス、件名、およびメッセージが含まれる値を指定します。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
>	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。受信者のカンマ区切り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。受信者のカンマ区切り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。受信者のカンマ区切りリストを含めることができます。
送信者	電子メールの発信アドレスとして使用されるアドレスを指定します。
reply-to	電子メールメッセージの返信アドレスとして使用されるアドレスを指定します。
件名	電子メールの件名を指定します。
メッセージ	電子メールメッセージの内容を指定します。

文字列名	説明
エンコード	電子メールメッセージで使用する文字エンコードを指定します。
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を指定してください:

サーバを指定してください: *

メッセージタイプを選択: テキスト

パスワードを指定してください:

文字列を指定してください:

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

名前	文字列の値:
<input type="text" value="to"/>	<input type="text" value="ManagerGroup@digitalairlines.com"/>
<input type="text" value="subject"/>	<input type="text" value="This is the e-mail subject"/>
<input type="text" value="message"/>	<input type="text" value="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、[\(296 ページ\) 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、[\(296 ページ\) 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 サーバアカウントのパスワードを名前付きパスワードとしてドライバオブジェクトに保存できます。これにより、パスワードが暗号化されます。それ以外の場合は、入力したパスワードはクリアテキストとして保存されます。名前付きパスワードの詳細については、『[Novell Identity Manager 管理ガイド \(http://www.novell.com/documentation/idm35/index.html\)](http://www.novell.com/documentation/idm35/index.html)』の「Using Named Password (名前付きパスワードの使用)」を参照してください。

文字列

電子メールメッセージの追加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
>	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。受信者のカンマ区切り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。受信者のカンマ区切り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。受信者のカンマ区切り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されるアドレスを指定します。
エンコード	電子メールメッセージで使用する文字エンコードを指定します。
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を指定してください: * 

テンプレートDNを指定してください: * 

パスワードを指定してください: 

文字列を指定してください: 

次に示すのは、名前付き文字列ビルダの例で、文字列の引数を指定する場合に使用されます。

名前	文字列の値:
<input type="text" value="to"/>	<input type="text" value="ManagerGroup@digitalairlines.com"/>
<input type="text" value="cc"/>	<input type="text" value="cc_SalesGroup@digitalairlines.com"/>

デフォルト属性値の設定

属性にまだ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 (オブションで、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、[\(296 ページ\) 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、「作成 - デフォルト属性値の設定 (http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_designer/data/prdefaultattr.html#prdefaultattr)」を参照してください。このポリシーを XML で表示するには、[predef_creation_set_default_attribute_value.xml \(../samples/predef_creation_set_default_attribute_value.xml\)](#) を参照してください。

Creation - Set Default Attribute Value
No description available

条件

- 条件グループ 1
 - if クラス名 等しい "User"

アクション

- デフォルト属性値の設定 (" [Enter attribute name]", write-back="true", " [Enter default attribute value]")

実行内容 デフォルト属性値の設定 ?

属性名を指定してください: * 🔍

ライトバック: ▼

引数値を指定してください: * 📄

タイプ	引数値
<input type="text" value="string"/> ▼	<input type="text" value="Digital Airlines"/>

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、「[引数値リストビルダ](http://www.novell.com/documentation/idm35/index.html?page=/) (<http://www.novell.com/documentation/idm35/index.html?page=/>)

documentation/idm35/policy_designer/data/pbvaluebuilder.html#pbvaluebuilder)」を参照してください。値は必要な内容に設定できます。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、[\(296 ページ\) 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、[\(296 ページ\) 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、「[コマンド変換 - 無効にする発行者の削除 \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_designer/data/prdeletetodisable.html#prdeletetodisable\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_designer/data/prdeletetodisable.html#prdeletetodisable)」を参照してください。このポリシーを XML で表示するには、[predef_command_delete_to_disable.xml \(./samples/predef_command_delete_to_disable.xml\)](#) を参照してください。

Command Transformation - Publisher Delete to Disable
No description available

条件

条件グループ 1

- if 操作 等しい "delete"
- Or(Q) if クラス名 等しい "User"

アクション

- ターゲット属性値の追加("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

属性名を指定してください: * Login Disabled

クラス名を指定してください:

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

値のタイプを指定してください: string

文字列を入力: * true

このルールは、「ログインの無効化」の属性値を True に設定し、引数ビルダを使用して、この属性の値として「True」というテキストを追加します。このビルダの詳細については、「引数ビルダ (http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_designer/data/pbargbuilder.html#pbargbuilder)」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内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) 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

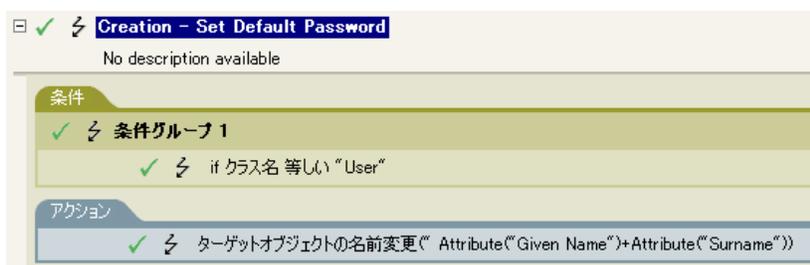
ターゲットオブジェクトを選択します。このオブジェクトは現在のオブジェクトを指定することも、DN または関連付けによって指定することもできます。

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、「作成 - デフォルトパスワードの設定 (http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_designer/data/prdefaultpassword.html#prdefaultpassword)」を参照してください。このポリシーを XML で表示するには、`prdef_creation_set_default_password.xml` (`../samples/prdef_creation_set_default_password.xml`) を参照してください。



実行内容 | ターゲットパスワードの設定

クラス名を指定してください:

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

文字列を指定してください: * Attribute("Given Name")+Attribute("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、[\(296 ページ\) 変数の拡張](#)を参照してください。

変数スコープ

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、[\(296 ページ\) 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セキュリティを設定します。ポリシーの名前は「Govern Groups for User Based on Title」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、「[Identity Manager ポリシーのダウンロード \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html\)](#)」を参照してください。このポリシーを XML で表示するには、[003-AddCreateGroups.xml \(../samples/003-Command-AddCreateGroups.xml\)](#) を参照してください。

The screenshot shows a policy configuration window titled "Set local variables to test existence of groups and for placement". It is divided into two main sections: "条件" (Conditions) and "アクション" (Actions).

条件 (Conditions):

- 条件グループ 1 (Condition Group 1):**
 - if クラス名 等しい "User"
- 条件グループ 2 (Condition Group 2):**
 - if 操作 等しい "add"
 - if 操作 等しい "modify"

The two condition groups are connected by an "And(N)" operator.

アクション (Actions):

- ローカル変数の設定 ("manager-group-dn", "Users#ManagerGroup")
- ローカル変数の設定 ("manager-group-info", ソース属性 ("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", ソース属性 ("Object Class", DN(ローカル変数("employee-group-dn"))))

実行内容 ローカル変数の設定 ▼ ?

変数名を入力してください: * 

スコープを選択: ▼

変数タイプの選択: ▼

文字列を指定してください: * 

ローカル変数は、ユーザオブジェクトのターゲット属性 (オブジェクトクラスとローカル変数 `manager-group-info`) の値に設定されます。ローカル変数を作成するには、引数ビルダを使用します。詳細については、「[引数ビルダ \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_designer/data/pbargbuilder.html#pbargbuilder\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_designer/data/pbargbuilder.html#pbargbuilder)」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 ▼ ?

関連付けを指定してください。* ☰

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

実行内容 

文字列を指定してください。* 

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、「[配置 - 発行者 \(ミラーリング\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_designer/data/prplacepubmirrored.html#prplacepubmirrored) (http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_designer/data/prplacepubmirrored.html#prplacepubmirrored)」を参照してください。このポリシーを XML で表示するには、[predef_place_pub_mirrored.xml](#) ([../samples/predef_place_pub_mirrored.xml](#)) を参照してください。

Placement - Publisher Mirrored
No description available

条件

✓ ↻ 条件グループ 1

✓ ↻ if ソースDN サブツリー内にあり "[Enter base of source hierarchy]"

アクション

✓ ↻ ローカル変数の設定 ("dest-base", スコープ="policy", "[Enter base of destination hierarchy]")

✓ ↻ 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の指定: * Local Variable ("dest-base")+ "*" +Unmatched Source DN

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、[\(296 ページ\) 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  

プロパティ名を指定してください: *

文字列を指定してください: * 

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

実行内容  

DNの指定: * 

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」という単語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」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、「Identity Manager ポリシーのダウンロード (<http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html>)」を参照してください。このポリシーを XML で表示するには、[003-Create_AssignTemplateByTitle.xml](#) ([../samples/003-Create_AssignTemplateByTitle.xml](#)) を参照してください。

The screenshot shows a policy configuration window titled "Assign Manager template if Title contains 'Manager'". It is divided into two sections: "条件" (Conditions) and "アクション" (Actions). Under "条件", there is a "条件グループ 1" (Condition Group 1) containing three conditions connected by "And(N)" operators: "if クラス名 等しい 'User'", "if 操作属性 'Title' 使用可能", and "if 操作属性 'Title' 等しい '*.manager.*'". Under "アクション", there is one action: "操作テンプレートDNの設定 (DN('Users\ManagerTemplate'))". Below this window, another policy configuration window titled "Assign Employee template if Title does not contain 'Manager'" is partially visible.

実行内容
DNの指定: *

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」という単語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、[\(296 ページ\) 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、[\(296 ページ\) 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の変更を検出し、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」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、「[Identity Manager ポリシーのダウンロード \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html\)](#)」を参照してください。このポリシーを XML で表示するには、[001-Input_PushBackOnEmail \(../samples/001-Input-PushBackOnEmail.xml\)](#) を参照してください。

The screenshot shows a configuration window for a policy named "Push back on email changing". The window has a title bar with a close button, a checkmark, and a refresh icon. Below the title bar, it says "No description available". The main content is divided into two sections: "条件" (Conditions) and "アクション" (Actions). Under "条件", there is a "条件グループ 1" (Condition Group 1) which contains two conditions: "if クラス名 等しい 'User'" and "if 操作属性 'Email' 変更あり". These two conditions are connected by an "And(N)" button. Under "アクション", there are two actions: "ソース属性値の追加('Email', ターゲット属性('Internet EMail Address'))" and "操作属性のストリップ('Email')". Each condition and action has a checkmark and a refresh icon next to it.

実行内容 **ソース属性値の設定**  

属性名を指定してください: * 

クラス名を指定してください: 

オブジェクトを選択: 

値のタイプを指定してください: 

文字列を入力: * 

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内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を指定することも、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  

クラス名を指定してください: 

オブジェクトを選択: 

文字列を指定してください: * 

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 資格情報プロビジョニングポリシー \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_credprov/data/bookinfo.html\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_credprov/data/bookinfo.html)」を参照してください。

フィールド

資格情報リポジトリオブジェクトの 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、[\(296 ページ\) 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、[\(296 ページ\) 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を指定してください: * 🔍

DN相対ポリシーを設定する

ターゲットユーザDNを指定してください: * 📄

[アプリケーションオブジェクトから次を入力](#)

アプリケーション資格情報IDを指定してください: *

ログインパラメータ文字列を指定してください: 📄

SSO パスフレーズの設定

ユーザオブジェクトをプロビジョニングするときの Novell SecureLogin のパスフレーズと回答を設定します。このアクションは、資格情報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 資格情報プロビジョニングポリシー (http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_credprov/data/bookinfo.html)」を参照してください。

フィールド

資格情報リポジトリオブジェクトの 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、[\(296 ページ\) 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。

質問文字列

SecureLogin パスフレーズの質問を指定します。

回答文字列

SecureLogin パスフレーズの回答を指定します。

例

実行内容 SSOパスフレーズの設定

資格情報リポジトリオブジェクトDNを指定してください: *

DN相対ポリシーを設定する

ターゲットユーザDNを指定してください: *

質問文字列: *

回答文字列: *

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。

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、ネームスペースプレフィックスを含めることができます (プレフィックスがすでにこのポリシーで定義されている場合)。変数の拡張がサポートされます。詳細については、[\(296 ページ\) 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

実行内容 XML属性の設定  

属性名の入力: * 

XPath式を指定してください: *   

文字列を指定してください: * 

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、[\(296 ページ\) 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」の場合、この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」の場合、この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、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上で、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、開始するワークフローによって異なります。

注

一部の名前は特別な意味を持ち、開始されているワークフローに関係なく利用できます。

- ◆ **:InitiatorOverrideDN:** 認証に使用されたユーザ以外の場合におけるワークフローのニシエータの LDAP フォーマット DN。
- ◆ **:CorrelationID:** 関連するワークフローを相互関連させる識別子。

ワークフローの開始中にいずれかのタイプのエラーが発生した場合、エラー文字列は `error.do-start-workflow` という名前のローカル変数内にある包含するポリシーで利用できます。それ以外の場合は、そのローカル変数は利用できません。

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 フォーマットで指定します。変数の拡張がサポートされます。詳細については、[\(296 ページ\) 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の 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、[\(296 ページ\) 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上でワークフローを開始する権限を持つユーザの DN を LDAP フォーマットで指定します。変数の拡張がサポートされます。詳細については、[\(296 ページ\) 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を開始する権限を持つユーザのパスワードを指定します。パスワードを名前付きパスワードとしてドライバオブジェクトに保存します。これにより、保存時にパスワードを暗号化できます。

受信者 DN

ワークフローの受信者の DN を LDAP フォーマットで指定します。

追加の引数

ワークフローの引数を指定します。引数はワークフロー上で定義します。一部の引数は、ワークフローを開始するために必須です。これはワークフローの定義内容によって変わります。

例

次の例では、入力および追加操作が行われるたびにワークフロープロセスが開始されます。このワークフローは携帯電話の要求です。このポリシーをXMLで表示するには、[start_workflow.xml](#) (./samples/start_workflow.xml) を参照してください。

Start Workflow

条件

- 条件グループ 1
 - if 操作 等しい "add"

アクション

- ワークフローの開始 (id="cn=WorkflowAdmin,o=People", url="http://localhost:8080/IDMProv", workflow-id="CN=ApproveCellPhone,CN=RequestDefs,CN=AppConfig,CN=UserApplication,CN=DriverSet,O=novell", 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** ?

プロビジョニング要求DNを指定してください: * 🔍

ユーザアプリケーションURLを指定してください: *

承認されたユーザDNを指定してください: * 🔍

承認されたユーザパスワードを指定してください: * 📄

受信者DNを指定してください: * 📄

追加の引数を指定してください: 📄

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、[\(296 ページ\) 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の変更を検出し、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」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、「[Identity Manager ポリシーのダウンロード \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html\)](#)」を参照してください。このポリシーを XML で表示するには、[001-Input-PushBackOnEmail.xml \(../samples/001-Input-PushBackOnEmail.xml\)](#) を参照してください。

☐ ✓ ↻ **Push back on email changing**
No description available

条件

✓ ↻ 条件グループ 1

- ✓ ↻ if クラス名 等しい "User"
- And(N) ✓ ↻ if 操作属性 "Email" 変更あり

アクション

- ✓ ↻ ソース属性値の追加 ("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 ?

名前を指定: * 🔍

このアクションは、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

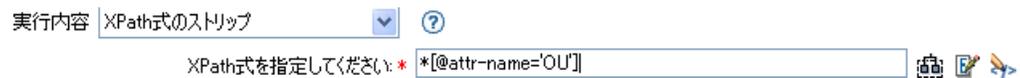
XPath 式

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

ポリシーで XPath 式を使用する場合の詳細については、「XPath 1.0 式 (<http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policyxpathexpression.html#policyxpathexpression>)」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。デフォルトのレベルは0です。メッセージは、指定したトレースレベルがドライバで設定されているトレースレベル以下である場合にのみ表示されます。

ドライバのトレースレベルの設定方法の詳細については、『[Novell Identity Manager 管理ガイド \(http://www.novell.com/documentation/idm35/index.html\)](http://www.novell.com/documentation/idm35/index.html)』の「バージョン情報」を参照してください。

色

トレースメッセージの色を選択します。

文字列

トレースメッセージの値を指定します。

例

この例には4つのルールがあり、名字属性の最初の文字に基づいてユーザオブジェクト用の配置ポリシーを実装しています。トレースメッセージおよび Novell Audit または Sentinel カスタム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を使用して、DSTRACE へのトレースメッセージを送信しています。これは、「Policy to Place by Surname」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、「[Identity Manager ポリシーのダウンロード \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html)」を参照してください。このポリシーを XML で表示するには、001-Placement-BySurname.xml (./samples/001-Placement-BySurname.xml) を参照してください。

The screenshot shows a configuration window for a policy named "Surname A-I: place in Users1". It is divided into two sections: "条件" (Conditions) and "アクション" (Actions).

- 条件 (Conditions):**
 - 条件グループ 1 (Condition Group 1)
 - if クラス名 等しい "User" (if class name is "User")
 - And(N) if 操作属性 'Surname' match "[a-]*" (if operation attribute 'Surname' matches "[a-]*")
- 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"))

Below the configuration window, there are three more policy entries listed:

- Surname J-R: place in Users2
- Surname S-Z: place in Users3

実行内容 

レベルを指定してください:

色を選択:

文字列を指定してください: * 

DSTRACE 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。詳細については、事前定義されたルールの「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 (http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_designer/data/prfilterexcludesubtree.html#prfilterexcludesubtree)」を参照してください。このポリシーを XML で表示するには、`predef_transformation_filter_exclude_subtree.xml` (`../samples/predef_transformation_filter_exclude_subtrees.xml`) を参照してください。

Event Transformation - Scope Filtering - Exclude subtree(s)
No description available

条件

✓ ↺ 条件グループ1

✓ ↺ if ソースDN サブツリー内にあり "[Enter a subtree to exclude]"

アクション

✓ ↺ 拒否0

実行内容

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、[\(296 ページ\) 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」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、「[Identity Manager ポリシーのダウンロード \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html)」を参照してください。このポリシーを XML で表示するには、001-Create-RequiredAttrs.xml (../samples/001-Create-RequiredAttrs.xml) を参照してください。



実行内容
名前を指定: *

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を繰り返します。

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 と評価される場合に繰り返すアクションを指定します。

例



The screenshot shows a configuration window for a 'While' loop. It is divided into two main sections: '条件' (Conditions) and 'アクション' (Actions).

条件 (Conditions):

- 条件グループ 1
 - if 操作 等しい "add"

アクション (Actions):

- ローカル変数の設定 ("counter", スコープ="policy", "1")
- 次の間
 - if ローカル変数 'counter' より大きくない "10"
 - 実行内容
 - 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", スコープ="policy", XPath ("counter + 1"))

Below the configuration window, there is a dropdown menu for '実行内容' (Execution Content) set to '次の間' (Between). Below this, there are two input fields for configuration:

- 条件: * and(if ローカル変数 'counter' より大きくない "")
- アクションを指定してください: * 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ます。

注

多くのアクションは、属性またはコンテンツ内での動的な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、`$<variable-name>$` という形式の埋め込み参照は、指定した名前を持つローカル変数またはグローバル変数の値に置き換えられます。`$<variable-name>$` は、有効な変数名でなければなりません。有効な XML 名の詳細については、[W3C Extensible Markup Language \(XML\) \(http://www.w3.org/TR/2006/REC-xml11-20060816/#sec-suggested-names\)](http://www.w3.org/TR/2006/REC-xml11-20060816/#sec-suggested-names) を参照してください。

指定した変数が存在しない場合、参照は空の文字列に置き換えられます。1つの\$を使用するときに、変数参照として解釈しない場合は、\$を追加してエスケープ処理する必要があります(例:「`$$100.00` 借りている」)。

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。

- ◆ 298 ページの「追加されたエンタイトルメント」
- ◆ 299 ページの「関連付け」
- ◆ 300 ページの「属性」
- ◆ 301 ページの「文字」
- ◆ 302 ページの「クラス名」
- ◆ 303 ページの「ターゲット属性」
- ◆ 305 ページの「ターゲット DN」
- ◆ 307 ページの「ターゲット名」
- ◆ 308 ページの「ドキュメント」
- ◆ 309 ページの「エンタイトルメント」
- ◆ 310 ページの「パスワードの生成」
- ◆ 311 ページの「グローバル構成値」
- ◆ 312 ページの「ローカル変数」
- ◆ 314 ページの「名前付きパスワード」
- ◆ 316 ページの「説明」
- ◆ 317 ページの「操作属性」
- ◆ 319 ページの「操作プロパティ」
- ◆ 320 ページの「[パスワード]」
- ◆ 321 ページの「クエリ」
- ◆ 322 ページの「削除された属性」
- ◆ 323 ページの「削除されたエンタイトルメント」
- ◆ 324 ページの「解決」
- ◆ 325 ページの「ソース属性」
- ◆ 326 ページの「ソース DN」
- ◆ 327 ページの「ソース名」
- ◆ 328 ページの「[時刻]」
- ◆ 329 ページの「テキスト」
- ◆ 331 ページの「一意の名前」
- ◆ 334 ページの「一致しないソース DN」
- ◆ 335 ページの「XPath」
- ◆ 336 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、[\(336 ページ\) 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。事前定義ルールの詳細については、[コマンド変換 - 無効にする発行者の削除 \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_designer/data/prdeletetodisable.html#prdeletetodisable\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_designer/data/prdeletetodisable.html#prdeletetodisable) 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。ポリシーをXML で表示するには、[predef_command_delete_to_disable.xml \(../samples/predef_command_delete_to_diabile.xml\)](#) を参照してください。

The screenshot displays the configuration for a rule named "Command Transformation - Publisher Delete to Disable". The rule has no description available. It is divided into two sections: "条件" (Conditions) and "アクション" (Actions).

条件 (Conditions):

- 条件グループ 1
 - if 操作 等しい "delete"
 - Or(Q) if クラス名 等しい "User"

アクション (Actions):

- ターゲット属性値の追加 ("Login Disabled", "true")
- 関連付けを削除 (関連付け (関連付け 0))

..... 関連付け 0

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、[\(336 ページ\) 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、[作成 - デフォルトパスワードの設定 \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_designer/data/prdefaultpassword.html#prdefaultpassword\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_designer/data/prdefaultpassword.html#prdefaultpassword) 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。

..... 属性("Given Name")

..... 属性("Surname")

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、[\(336 ページ\) 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**

↶ トレースしない: ▼

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

.....品 クラス名0

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、[\(336 ページ\) 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、[\(336 ページ\) 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、[Identity Manager ポリシーのダウンロード](#) (<http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html>) 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。

The screenshot shows a configuration window titled "Set local variables to test existence of groups and for placement". It is divided into two main sections: "条件" (Conditions) and "アクション" (Actions).

条件 (Conditions):

- 条件グループ 1 (Condition Group 1):** Contains the condition "if クラス名 等しい 'User'" (if class name is equal to 'User').
- 条件グループ 2 (Condition Group 2):** Contains two conditions: "if 操作 等しい 'add'" (if operation is equal to 'add') and "if 操作 等しい 'modify'" (if operation is equal to 'modify'). These two conditions are grouped together with an "Or" operator.

アクション (Actions):

- ローカル変数の設定 ("manager-group-dn", "Users%ManagerGroup")
- ローカル変数の設定 ("manager-group-info", ソース属性 ("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", ソース属性 ("Object Class", DN(ローカル変数 ("employee-group-dn"))))

.....  ターゲット属性("Object Class", DN)

 **エディタ**

トレースしない: false 

名前: * 

クラス名: 

オブジェクトを選択: 

DNの指定: * 

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、[コマンド変換 - 部署別コンテナの作成 - パート 1 およびパート 2 \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_designer/data/prdeptcontainer.html#prdeptcontainer\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_designer/data/prdeptcontainer.html#prdeptcontainer) 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。

The screenshot shows a configuration window titled "Command Transformation - Create Department Container - Part 2". It contains a "Condition" section with a rule: "if 操作 等しい 'add'". Below it is an "Action" section with two rules: "ローカル変数の設定 'target-container', スコープ='policy', ターゲットDN(長さ='-2')+ソースDN0" and "ローカル変数の設定 'does-target-exist', スコープ='policy', ターゲット属性('objectclass', クラス名='Organizational Unit', DN(ローカル変数('target-container'))))". At the bottom, there is a label "ターゲットDN(長さ='-2')".

 **エディタ**

↶ トレースしない:

開始:

長さ:

ソースDN形式に変換:

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

.....  ターゲット名0

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。

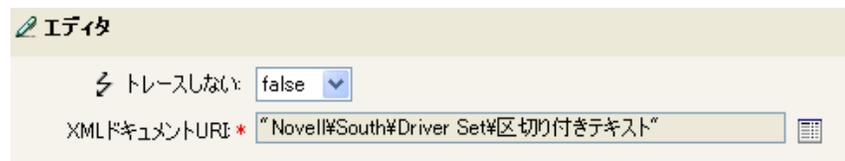
フィールド

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#区切り付きテキスト")



エディタ

↳ トレースしない: false

XMLドキュメントURI * "Novell#South#Driver Set#区切り付きテキスト"

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、[\(336 ページ\) 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

.....  エンタイトルメント("manager")



 **エディタ**

↳ トレースしない:

名前: * 

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、[\(336 ページ\) 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、[\(336 ページ\) 変数の拡張](#)を参照してください。

例

.....  グローバル構成値("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、[\(336 ページ\) 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、[Identity Manager ポリシーのダウンロード](#) (<http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html>) 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。

The screenshot displays a configuration window for a policy. At the top, there are two main sections: "Set local variables to test existence of groups and for placement" and "Create ManagersGroup, if needed". Below these, a "条件" (Conditions) section is expanded, showing "条件グループ 1" (Condition Group 1). This group contains two conditions: "if ローカル変数 'manager-group-info' 使用可能" (if local variable 'manager-group-info' is usable) and "if ローカル変数 'manager-group-info' 等しくない 'group'" (if local variable 'manager-group-info' is not equal to 'group'). An "And(N)" button is visible between the conditions. Below the conditions is an "アクション" (Actions) section with one action: "ターゲットオブジェクトの追加(クラス名='Group', DN(ローカル変数('manager-group-dn')))" (Add target object (class name='Group', DN(local variable('manager-group-dn')))).

..... ローカル変数("manager-group-dn")

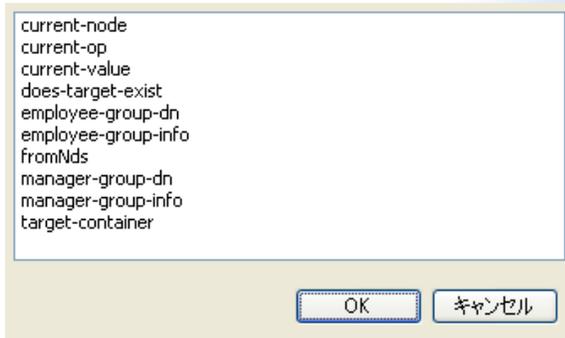
エディタ

トレースしない: false

変数名: *

ローカル変数セレクタ

リストからローカル変数を選択します。



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

ローカル変数の値は、`group-manager-dn` です。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

名前付きパスワードを指定します。変数の拡張がサポートされます。詳細については、[\(336 ページ\) 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すると、セキュリティリスクが発生します。名前付きパスワードの詳細については、『[Novell Identity Manager Administration Guide \(http://www.novell.com/documentation/idm35/index.html\)](http://www.novell.com/documentation/idm35/index.html)』を参照してください。

例では、[\(287 ページ\) 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。ポリシーを XML で表示するには、[start_workflow.xml \(./samples/start_workflow.xml\)](#) を参照してください。

Start Workflow

条件

- 条件グループ 1
 - if 操作 等しい "add"

アクション

- ワークフローの開始 (id="cn=WorkflowAdmin,o=People", url="http://localhost:8080/IDMProv", workflow-id="CN=ApproveCellPhone,CN=RequestDefs,CN=AppConfig,CN=UserApplication,CN=DriverSet,O=novell", 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** ?

プロビジョニング要求DNを指定してください: *

ユーザアプリケーションURLを指定してください: *

承認されたユーザDNを指定してください: *

承認されたユーザパスワードを指定してください: *

受信者DNを指定してください: *

追加の引数を指定してください:

名前付きパスワード(" workflow-admin")

エディタ

トレースしない:

パスワード名: *

名前付きパスワードブラウザの選択

選択した名前付きパスワードは式に渡されます。

サーバ	
server ▼	
名前	表示名
smtp-admin	smtp-admin
workflow-admin	workflow-admin

説明

現在の操作の名前に展開します。

例

.....  操作0

操作属性

現在の操作から属性の値に展開します。操作は <add-attr>、<add-value>、または <attr> になります。このトークンが、ノードセット結果が期待されているコンテキスト内で評価される場合、すべての変数値はノードセット内でノードして返されます。それ以外の場合は、最初の変数値が文字列として返されます。

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、[\(336 ページ\) 変数の拡張](#)を参照してください。

例

例には、名字属性の最初の文字に基づいたユーザオブジェクト用の配置ポリシーを実装する4つのルールがあります。トレースメッセージおよびカスタム 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、[Identity Manager ポリシーのダウンロード \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html) を参照してください。ポリシーを XML で表示するには、[001-Placement-BySurname.xml \(../samples/001-Placement-BySurname.xml\)](#) を参照してください。

⊕ ✓ ↻ Setup Local Variables

⊖ ✓ ↻ Surname A-I: place in Users1

条件

✓ ↻ 条件グループ 1

- ✓ ↻ if クラス名 等しい "User"
- And(N) ✓ ↻ if 操作属性 "Surname" match "[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

Training#Users#Active#Users1

#

操作属性("CN")

エディタ

トレースしない: false

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、[\(336 ページ\) 変数の拡張](#)を参照してください。

例

.....  操作プロパティ("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

.....  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、[\(336 ページ\) 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

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

The screenshot shows a query configuration interface with the following fields and values:

- データストア: 宛先
- スコープ: サブツリー
- 最大結果カウント: (empty)
- 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

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、[\(336 ページ\) 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、[\(336 ページ\) 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", DN0)

 **エディタ**

↺ トレースしない: false ▼

データストア: ソース ▼

解決タイプ: DNから関連付けへ ▼

DN: * "manager" 

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、[\(336 ページ\) 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、[\(336 ページ\) 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")

 **エディタ**

↳ トレースしない: false 

名前: * 

クラス名: 

オブジェクトを選択: 

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 の部分で使用されます。

例

.....  ソースDN(長さ="-2")

 **エディタ**

 トレースしない:	<input type="checkbox"/> false	
開始:	<input type="text" value="0"/>	
長さ:	<input type="text" value="-2"/>	
ターゲットDN形式に変換:	<input type="checkbox"/> false	

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

.....  ソース名0

[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、[\(336 ページ\) 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、[\(336 ページ\) 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、[\(336 ページ\) 変数の拡張](#)を参照してください。

例

.....  時間(format="!CTIME", lang="ja-JP", tz="JST")

 **エディタ**

↳ トレースしない:	<input type="checkbox"/> false
書式 *	1970年1月1日からの秒数[CTIME]   {x}
使用言語:	日本語 (日本)[ja-JP] 
タイムゾーン:	日本標準時[JST] 

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、[\(336 ページ\) 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、「[Identity Manager ポリシーのダウンロード \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html\)](#)」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。

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。

The screenshot shows a configuration window titled "Set local variables to test existence of groups and for placement". It is divided into two main sections: "条件" (Conditions) and "アクション" (Actions).

条件 (Conditions):

- 条件グループ 1 (Condition Group 1):** Contains one condition: "if クラス名 等しい 'User'" (if class name is equal to 'User').
- 条件グループ 2 (Condition Group 2):** Contains two conditions: "if 操作 等しい 'add'" (if operation is equal to 'add') and "if 操作 等しい 'modify'" (if operation is equal to 'modify').

アクション (Actions):

- ローカル変数の設定 ("manager-group-dn", "Users%ManagerGroup")
- ローカル変数の設定 ("manager-group-info", ソース属性 ("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", ソース属性 ("Object Class", DN(ローカル変数("employee-group-dn"))))

..... "Users%ManagersGroup"

The screenshot shows the "エディタ" (Editor) section. It includes a "トレースしない:" (Do not trace:) checkbox with a dropdown menu set to "false". Below it is a "テキスト:" (Text:) label followed by a text input field containing "Users%ManagersGroup".

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
- ◆ never
- ◆ fallback

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。

- ◆ first
- ◆ last
- ◆ all

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ルトであり、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、名前は必須です。

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、カウンタは使い果たされたと考えられます。

指定された名前が一意的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。

指定された名前の順番は次のようにテストされます。

- ◆ 各パターンは指定された順にテストされます。**counter-use="always"** で、パターンが **counter-pattern** で示されているパターンのいずれかを使用している場合、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。
- ◆ パターンをすべて使っても一意な名前が見つからず、**counter-use="fallback"** の場合は、**counter-pattern** で示されたパターンがカウンタを使用して再試行されます。

指定されたパターンとカウンタの組み合わせすべてが消費された場合は、**on-unavailable** によって指定されたアクションが実行されます。

例

```
一意の名前<"CN", counter-pattern="first", counter-use="fallback", on-unavailable="error", uppercase0+uppercase0+upperca
```

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。

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

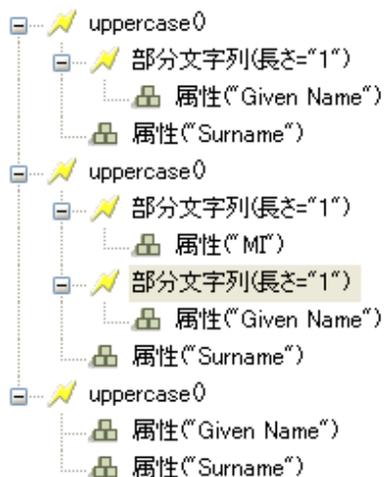
パターン: * uppercase(部分文字列(長さ="1", 属性("Given Name"))+属|

カウンタの使用タイミング: fallback

カウンタとともに使用するパターン: first

カウンタの開始: 1 桁: 1 カウンタに先行ゼロを埋め込む

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、一致 - 購読者 (ミラーリング)-LDAP 形式 (http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_designer/data/prmatchsubmirror.html#prmatchsubmirror) 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。

Matching - Subscriber Mirrored - LDAP format
No description available

条件

- 条件グループ 1
 - if ソースDN サブツリー内にあり "[Enter base of source hierarchy]"

アクション

- ローカル変数の設定 ("dest-base", スコープ="policy", "[Enter base of destination hierarchy])"
- 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

トレースしない: false

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

ポリシーで XPath 式を使用する詳細については、[XPath 1.0 式 \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policyxpathexpression.html#policyxpathexpression\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policyxpathexpression.html#policyxpathexpression) を参照してください。

例

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

 **エディタ**

トレースしない: 

XPath式: *   

変数の拡張

名詞トークンでの動的変数の使用を許可

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す。サポートされている場合は、形式が `<variable-name>$` の埋め込みリファレンスが指定した名前を持つローカルまたはグローバル変数の値と置き換えられます。`<variable-name>$` は、正しい変数名でなければなりません。正しい XML 名の詳細については、[W3C Extensible Markup Language \(XML\) \(http://www.w3.org/TR/2006/REC-xml11-20060816/#sec-suggested-names\)](http://www.w3.org/TR/2006/REC-xml11-20060816/#sec-suggested-names) を参照してください。

指定した変数が存在しない場合は、リファレンスは空の文字列と置き換えられます。単一の `$` を使用して、それを変数リファレンスとして解釈しない場合は、`$` を追加してエスケープする必要があります (例 : You owe me \$\$100.00)。

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。

- ◆ 338 ページの「Base64 デコード」
- ◆ 339 ページの「Base64 エンコード」
- ◆ 340 ページの「変換時間」
- ◆ 342 ページの「ターゲット DN のエスケープ」
- ◆ 343 ページの「ソース DN のエスケープ」
- ◆ 344 ページの「参加」
- ◆ 345 ページの「小文字」
- ◆ 346 ページの「マップ」
- ◆ 347 ページの「DN の解析」
- ◆ 350 ページの「すべて置換」
- ◆ 351 ページの「最初を置換」
- ◆ 353 ページの「分割」
- ◆ 354 ページの「部分文字列」
- ◆ 356 ページの「大文字」
- ◆ 357 ページの「XML 解析」
- ◆ 358 ページの「XML シリアライズ」
- ◆ 359 ページの「変数の拡張」

Base64 デコード

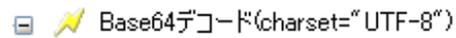
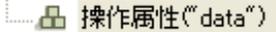
Base64 エンコードデータからの格納されたトークンの結果をバイト単位にデコードして、指定された文字セットを使用して、それらのバイトを文字列に変換します。

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。フィールドが空白のままの場合は、文字セットは `file.encoding` システムプロパティによって指定されたシステムエンコーディングにデフォルト設定されます。変数の拡張がサポートされます。詳細については、[\(359 ページ\) 変数の拡張](#)を参照してください。

例

 Base64デコード(charset=" UTF-8")
 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。フィールドが空白のままの場合は、文字セットは `file.encoding` システムプロパティによって指定されたシステムエンコーディングにデフォルト設定されます。変数の拡張がサポートされます。詳細については、[\(359 ページ\) 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、[\(359 ページ\) 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、[\(359 ページ\) 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、[\(359 ページ\) 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、[\(359 ページ\) 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、[\(359 ページ\) 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、[\(359 ページ\) 変数の拡張](#)を参照してください。

例

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

ソースの形式: *	MM/dd/yy	▼	🗄️ {x}
ソースの言語:	日本語 (日本)[ja-JP]	▼	
ソースのタイムゾーン:	日本標準時[Asia/Tokyo]	▼	
宛先の形式: *	dd/MM/yy	▼	🗄️ {x}
宛先の言語:	日本語 (日本)[ja-JP]	▼	
宛先のタイムゾーン:	日本標準時[Asia/Tokyo]	▼	

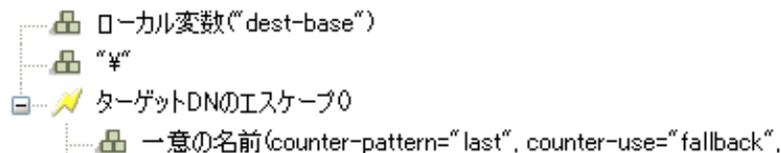
ターゲット DN のエスケープ

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

例は、識別マネージャ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、[配置 - 発行者 \(フラット\) \(http://www.novell.com/documentation/idm35/index.html?page=documentation/idm35/policy_designer/data/prplacepubflat.html#prplacepubflat\)](http://www.novell.com/documentation/idm35/index.html?page=documentation/idm35/policy_designer/data/prplacepubflat.html#prplacepubflat) 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 \(../sample/predef_place_pub_flat.xml\)](#) 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

☐ ⚡ ソースDNのエスケープ0
└── ☐ 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、[\(359 ページ\) 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 (csv="true", delimiter=",")
└─ 属性 ("Member")

エディタ

トレースしない: false

区切り記号:

CSV引用ルールを適用する

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、[Identity Manager ポリシーのダウンロード \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html) 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

- 条件グループ 1
 - if クラス名 等しい "User"
 - And(N) if 操作属性 'Given Name' 使用可能
 - And(N) 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(部分文字列(長さ="8", 部分文字列(長さ="1", 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、[\(359 ページ\) 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、[\(359 ページ\) 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、[\(359 ページ\) 変数の拡張](#)を参照してください。

例

マップ(dest="code", src="dept", table=".%Department Table")
操作属性("OU")

エディタ

トレースしない: false

マッピングテーブルDN: * .%Department Table

DN相対ポリシーを設定する

ソース列の名前: * dept

宛先の列の名前: * 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、0xffffe、0xffffd、および 0xffff は使用できません。

- ◆ 相対 RDN 区切り文字
- ◆ RDN 区切り文字
- ◆ 名前ディバイダ
- ◆ 名前の値の区切り文字
- ◆ 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、[コマンド変換 - 部署別コンテナの作成 - パート 1 およびパート 2 \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_designer/data/prdeptcontainer.html#prdeptcontainer\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_designer/data/prdeptcontainer.html#prdeptcontainer) 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` (`./sample/predef_command_create_dept_container2.xml`) を参照してください。

☐ ✓ ⚡ **Command Transformation - Create Departmental Container - Part 2**
No description available

条件

✓ ⚡ **条件グループ 1**

- ✓ ⚡ if ローカル変数 'does-target-exist' 使用可能
- And(N) ✓ ⚡ 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="Organizational Unit", 直接="true", DN(ローカル変数("target-container")))
- ✓ ⚡ ターゲット属性値の設定("ou", クラス名="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")))

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

エディタ

⚡ トレースしない: false ▼

開始: -1

長さ: 1

ソースDNのフォーマット: ターゲットDN ▼

宛先DNのフォーマット: 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。変数拡張をサポートします。詳細については、[\(359 ページ\) 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、[\(359 ページ\) 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

すべて置換("()", "\$1")
ターゲットDN0

✎ **エディタ**

🔄 トレースしない: false ▼

正規表現: *

置換文字列:

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、[\(359 ページ\) 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、[\(359 ページ\) 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\(java.lang.String\)](http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Matcher.html#replaceAll(java.lang.String)))

[パターン] のオプションには `CASE_INSENSITIVE`、`DOTALL`、および `UNICODE_CASE` が使用されますが、適切な埋め込みエスケープを使用して逆の意味を指定することができます。

例

この例では、電話番号 (nnn)-nnn-nnnn を nnn-nnn-nnnn に再フォーマットします。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、[入出力変換 - 電話番号の形式を \(nnn\) nnn-nnnn から nnn-nnn-nnnn に変更 \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_designer/data/prreformattel1.html#prreformattel1\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_designer/data/prreformattel1.html#prreformattel1) を参照してください。ポリシーを XML で表示するには、[predef_transformation_reformat_telephone1 \(../samples/predef_transformation_reformat_telephone1.xml\)](#) 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。

Input or Output Transformation - Reformat Telephone number from (nnn) nnn-nnnn to nnn-nnn-nnnn
No description available

条件

✓ ↻ 条件グループ 1
新しい条件をここで定義

アクション

✓ ↻ 操作属性の再フォーマット("phone", 最初を置換("^(?[0-9])%s*(?[0-9])-(?[0-9])\$", "\$1-\$2-\$3", ローカル変数("current-value")))

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")

エディタ

トレースしない: false

正規表現: *

置換文字列:

正規表現 `^(\\d\\d\\d)\\s*(\\d\\d\\d)-(\\d\\d\\d\\d)$` は、`(nnn) nnn-nnnn` を、正規表現 `$1-$2-$3` は `nnn` を示しています。このルールでは、電話番号の形式を `(nnn) nnn-nnnn` から `nnn-nnn-nnnn` に変更します。

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、[\(359 ページ\) 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(csv="true", delimiter=",")  
"Doe,John,Doe, John"
```

エディタ

トレースしない:

区切り記号: *
 CSV引用ルールを適用する

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、[Identity Manager ポリシーのダウンロード \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html) 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。

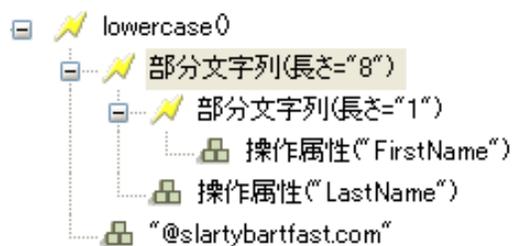
The screenshot shows a configuration window for a policy. The title bar reads: "Set email address: name@slartybartfast.com; name = (1 char of Given Name + Surname) <= 8 chars".

条件 (Conditions):

- 条件グループ 1
 - if クラス名 等しい "User"
 - And(N) if 操作属性 'Given Name' 使用可能
 - And(N) if 操作属性 'Surname' 使用可能

アクション (Actions):

- 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(部分文字列(長さ="8", 部分文字列(長さ="1", 操作属性("FirstName"))+操作属性("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、Identity Manager ポリシーのダウンロード (<http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policydownloadable.html>) 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。

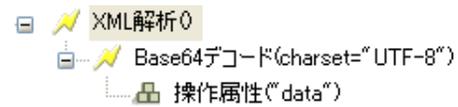
The screenshot shows the configuration for a policy named "Convert First/Last name to uppercase". It is divided into three sections: "条件" (Conditions), "アクション" (Actions), and "操作属性" (Operation Properties). The "条件" section contains two groups: "条件グループ 1" with the condition "if クラス名 等しい 'User'", and "条件グループ 2" with two conditions: "if 操作属性 'Given Name' 変更あり" and "if 操作属性 'Surname' 変更あり". The "アクション" section contains two actions: "操作属性の再フォーマット('Given Name', uppercase(操作属性('Given Name')))" and "操作属性の再フォーマット('Surname', uppercase(操作属性('Surname')))".

The screenshot shows the configuration for the operation property "uppercase0". It is a simple text field containing the value "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。格納されたトークンの結果が正しく形成された XML ではない、または別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、形式が `<variable-name>$` の埋め込みリファレンスが指定した名前を持つローカルまたはグローバル変数の値と置き換えられます。`<variable-name>$` は、正しい変数名でなければなりません。正しい XML 名の詳細については、[W3C Extensible Markup Language \(XML\) \(http://www.w3.org/TR/2006/REC-xml11-20060816/#sec-suggested-names\)](http://www.w3.org/TR/2006/REC-xml11-20060816/#sec-suggested-names) を参照してください。

指定した変数が存在しない場合は、リファレンスは空の文字列と置き換えられます。単一の `$` を使用して、それを変数リファレンスとして解釈しない場合は、`$` を追加してエスケープする必要があります (例: `You owe me $$100.00`)。

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。

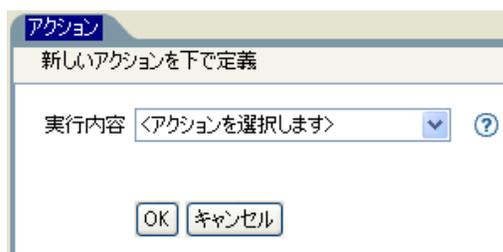
- ◆ 361 ページのセクション 16.1 「アクションビルダ」
- ◆ 362 ページのセクション 16.2 「アクションビルダ」
- ◆ 363 ページのセクション 16.3 「引数ビルダ」
- ◆ 367 ページのセクション 16.4 「アクションの引数コンポーネントビルダ」
- ◆ 367 ページのセクション 16.5 「条件ビルダ」
- ◆ 369 ページのセクション 16.6 「条件の引数コンポーネントビルダ」
- ◆ 369 ページのセクション 16.7 「一致属性ビルダ」
- ◆ 371 ページのセクション 16.8 「名前付き文字列ビルダ」
- ◆ 372 ページのセクション 16.9 「パターン文字列ビルダ」
- ◆ 373 ページのセクション 16.10 「引数値リストビルダ」
- ◆ 374 ページのセクション 16.11 「ネームスペースエディタ」

16.1 アクションビルダ

アクションビルダでは、ルールを構成するアクションを追加、表示、および削除できます。アクションは他のアクションを含むことができます。

16.1.1 アクションの作成

- 1 ポリシービルダでは、新しいルールを作成するかまたは既存のルールを編集します。
- 2 [アクション] タブをダブルクリックして、アクションビルダを起動します。



- 3 ドロップダウンリストから目的のアクションを選択し、[OK] をクリックします。

16.1.2 アクションビルダの追加オプション

- 1 アクションを右クリックして追加オプションを表示します。



- ◆ [新規] > [アクションを前に挿入] : 現在のアクションの前に新しいアクションを追加します。
- ◆ [新規] > [アクションを後に挿入] : 現在のアクションの後に新しいアクションを追加します。
- ◆ **編集** : アクションビルダを起動します。
- ◆ **選択した項目を上に移動** : 選択したアクションを実行順序内の上に移動します。
- ◆ **選択した項目を下に移動** : 選択したアクションを実行順序内の下に移動します。
- ◆ **アクションの切り取り、コピー、貼り付け、または削除** : アクションを切り取り、コピー、貼り付け、または削除します。
- ◆ **元に戻す、または、やり直し** : 最後のアクションを元に戻す、またはやり直します。
- ◆ **初期設定** : ポリシービルダ内でデフォルト機能を設定できます。
- ◆ **ヘルプ** : アクションを選択して、[ヘルプ] アイコンをクリックし、そのアクションに固有の情報を参照します。

16.2 アクションビルダ

アクションビルダでは、別のアクション内にアクションを作成できます。アクションビルダを起動するには、以下のいずれかのアクションを選択して、[引数を編集する] アイコンをクリックします。

- ◆ (424 ページ) 毎
- ◆ (427 ページ) エンタイトルメントの実装

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

図 16-1 For Each アクション

実行内容 毎 ?

ノードセットを指定してください: *

アクションを指定してください: *

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

図 16-2 アクションビルダ

実行内容 ターゲット属性値の追加 ?

属性名を指定してください: *

クラス名を指定してください:

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の指定: *

値のタイプを指定してください: string

文字列を入力: *

16.3 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。

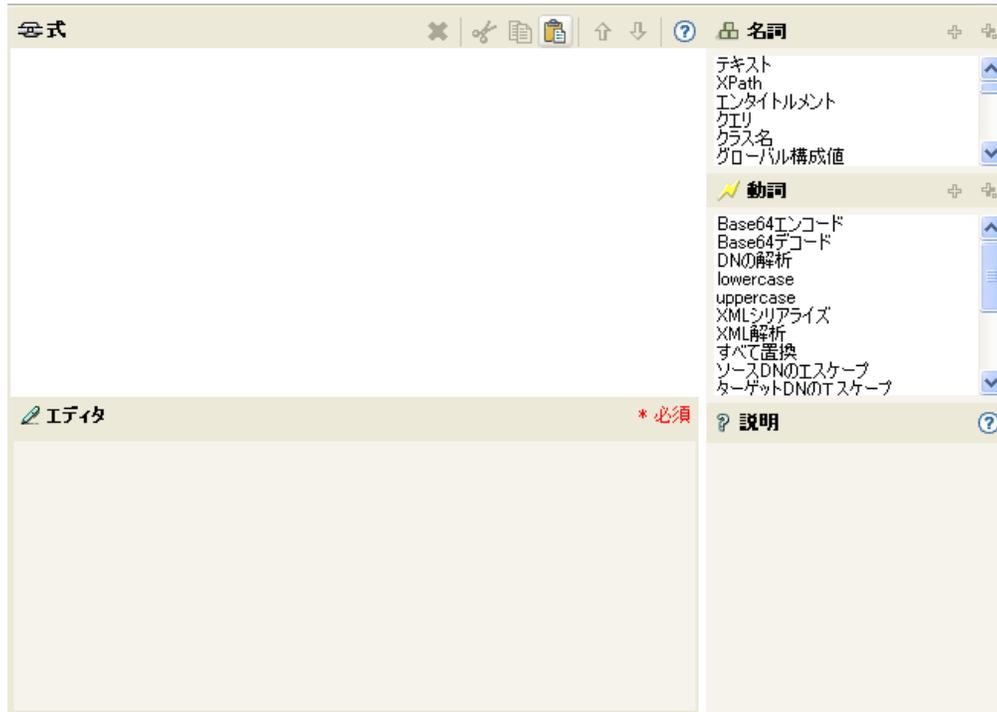
引数ビルダは 5 つの別々のセクションで構成されます。

- ◆ **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]** を選択して名詞トークンを **[式]** ペインに追加します。詳細については、[461 ページの「プレ Identity Manager 3.5 名詞トークン」](#) を参照してください。
- ◆ **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]** を選択して動詞トークンを **[式]** ペインに追加します。詳細については、[487 ページの「プレ Identity Manager 3.5 動詞トークン」](#) を参照してください。
- ◆ **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]** ペイン内で名詞および動詞に値を提供します。

図 16-3 プレ Identity Manager 3.5 引数ビルダ

引数の作成および編集

引数を構成するために式領域にコンポーネントを追加または式領域からコンポーネントを削除します。エディタでコンポーネント値を指定します。



- ◆ 364 ページのセクション 16.3.1 「引数ビルダの起動」
- ◆ 365 ページのセクション 16.3.2 「引数ビルダの例」

16.3.1 引数ビルダの起動

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

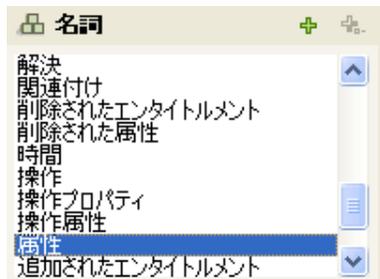
- ◆ (407 ページ) 関連付けの追加
- ◆ (408 ページ) ターゲット属性値の追加
- ◆ (409 ページ) ターゲットオブジェクトの追加
- ◆ (410 ページ) ソース属性値の追加
- ◆ (413 ページ) XML テキストの追加
- ◆ (415 ページ) ターゲット属性値のクリア (選択されたオブジェクトが [DN] または [関連付け] である場合)。
- ◆ (417 ページ) ソース属性値のクリア (選択されたオブジェクトが [DN] または [関連付け] である場合)。
- ◆ (421 ページ) ターゲットオブジェクトの削除 (選択されたオブジェクトが [DN] または [関連付け] である場合)。
- ◆ (422 ページ) ソースオブジェクトの削除 (選択されたオブジェクトが [DN] または [関連付け] である場合)。

- ◆ (423 ページ) 一致オブジェクトの検索
- ◆ (424 ページ) 毎
- ◆ (428 ページ) ターゲットオブジェクトの移動
- ◆ (429 ページ) ソースオブジェクトの移動
- ◆ (430 ページ) 操作属性の再フォーマット値
- ◆ (431 ページ) 関連付けの削除
- ◆ (432 ページ) ターゲット属性値の削除
- ◆ (433 ページ) ソース属性値の削除
- ◆ (434 ページ) ターゲットオブジェクトの名前変更 (選択されたオブジェクトが [DN] または [関連付け] および [文字列を入力] である場合)。
- ◆ (436 ページ) ソースオブジェクトの名前変更 (選択されたオブジェクトが [DN] または [関連付け] および [文字列を入力] である場合)。
- ◆ (440 ページ) ターゲット属性値の設定 (選択されたオブジェクトが [DN] または [関連付け] であり、[値タイプを入力] が指定されていない場合)。
- ◆ (441 ページ) ターゲットパスワードの設定
- ◆ (442 ページ) ローカル変数の設定
- ◆ (443 ページ) 操作関連付けの設定
- ◆ (444 ページ) 操作クラス名の設定
- ◆ (445 ページ) 操作ターゲット DN
- ◆ (446 ページ) 操作プロパティの設定
- ◆ (447 ページ) 操作ソース DN の設定
- ◆ (448 ページ) 操作テンプレート DN の設定
- ◆ (449 ページ) ソース属性値の設定
- ◆ (450 ページ) ソースパスワードの設定
- ◆ (453 ページ) XML 属性の設定
- ◆ (454 ページ) ステータス
- ◆ (457 ページ) メッセージのトレース

16.3.2 引数ビルダの例

次の例では、名と姓全体の最初の文字からユーザ名の引数を作成します。

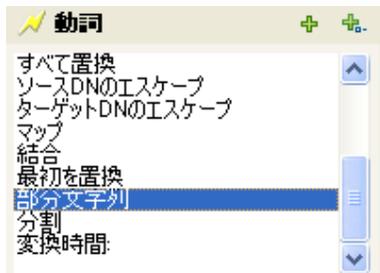
- 1 名詞のリストから [属性] をダブルクリックします。



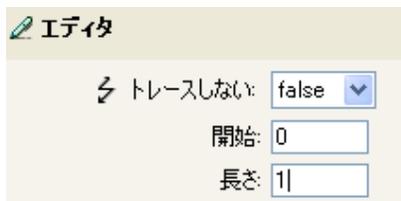
- 2 名の属性を指定または選択します。



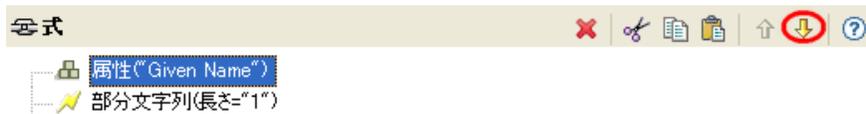
- 3 動詞のリストから [部分文字列] をダブルクリックします。



- 4 [長さ] フィールドに「1」を入力します。

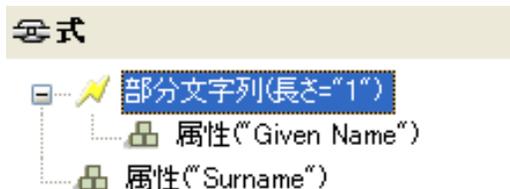


- 5 [名前] 属性を選択して、[下へ移動] アイコンをクリックします。



- 6 名詞リストから [属性] をダブルクリックします。

- 7 [名字] 属性を指定または参照します。



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。

- 8 [OK] をクリックして、引数を保存します。

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

- ◆ (408 ページ) ターゲット属性値の追加
- ◆ (410 ページ) ソース属性値の追加
- ◆ (430 ページ) 操作属性の再フォーマット値
- ◆ (432 ページ) ターゲット属性値の削除
- ◆ (433 ページ) ソース属性値の削除
- ◆ (440 ページ) ターゲット属性値の設定
- ◆ (449 ページ) ソース属性値の設定

図 16-4 ターゲット属性値の追加アクション

実行内容 ターゲット属性値の追加 

属性名を指定してください: * 

クラス名を指定してください: 

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: 現在のオブジェクト: 

値のタイプを指定してください: structured 

コンポーネントを入力: * 

- 1 値タイプが「構造」に設定されているときに [コンポーネントを編集する] アイコンをクリックします。
- 2 アクションコンポーネントの値を作成します。
値を入力するか、または [引数を編集する]  アイコンをクリックして、引数ビルダ内で値を作成します。

引数コンポーネント

引数コンポーネントは構造化された引数値です。



名前	値							
<input type="text" value="value"/>	<input type="text" value="user"/>							

- 3 [完了] をクリックします。

16.5 条件ビルダ

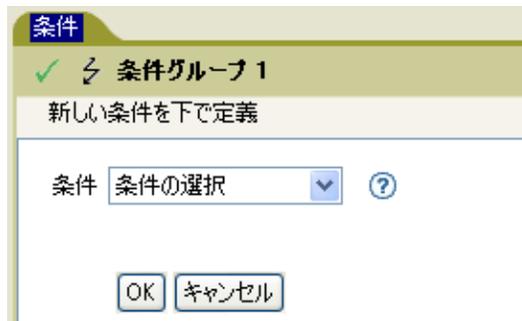
条件ビルダでは、ルールを構成する条件を追加、表示、および削除できます。条件には、1つ以上の条件と1つ以上の条件グループが含まれます。条件グループには2つの異なる条件構造が含まれます。条件構造は、条件グループの論理を定義します。2つの条件構造は次のとおりです。

- ◆ OR 条件、AND グループ

- ◆ AND 条件、OR グループ
- ◆ 368 ページのセクション 16.5.1 「条件の作成」
- ◆ 368 ページのセクション 16.5.2 「条件ビルダの追加オプション」

16.5.1 条件の作成

- 1 ポリシービルダでは、新しいルールを作成するかまたは既存のルールを編集します。
- 2 [条件] タブをダブルクリックして、条件ビルダを起動します。



- 3 ドロップダウンリストから目的の条件を選択し、[OK] をクリックします。

16.5.2 条件ビルダの追加オプション

- 1 条件を右クリックして追加オプションを表示します。



- ◆ [新規] > [条件を前に挿入] : 現在の条件の前に新しい条件を追加します。
- ◆ [新規] > [条件を後に挿入] : 現在の条件の後に新しい条件を追加します。
- ◆ 編集 : 条件ビルダを起動します。
- ◆ 選択した項目を上に移動 : 選択した条件を実行順序内の上に移動します。
- ◆ 選択した項目を下に移動 : 選択した条件を実行順序内の下に移動します。

- ◆ **切り取り、コピー、貼り付け、または削除**：条件を切り取り、コピー、貼り付け、または削除します。
- ◆ **元に戻す、または、やり直し**：最後のアクションを元に戻す、またはやり直します。
- ◆ **初期設定**：ポリシービルダ内でデフォルト機能を設定できます。
- ◆ **ヘルプ**：条件を選択して、[ヘルプ] アイコンをクリックし、その条件に固有の情報を参照します。

条件ビルダとルールの詳細については、[37 ページのセクション 3.4 「ルールの作成」](#) を参照してください。

16.6 条件の引数コンポーネントビルダ

条件の引数コンポーネントビルダを起動するには、次のいずれかの条件を選択し、[モード] で [構造] を選択して [ArgComponent ビルダの起動] アイコンを表示します。

- ◆ [\(379 ページ\)](#) If 属性
- ◆ [\(383 ページ\)](#) If ターゲット属性
- ◆ [\(378 ページ\)](#) If 関連付け

図 16-5 If 属性モード

条件 **属性** 

名前 * 

オペレータ * **等しい** 

モード **構造** 

値 

- 1 条件コンポーネントの名前および値を指定します。

引数コンポーネント

条件引数コンポーネントは名前/値のペアです。



名前	値							
<input type="text"/>	<input type="text"/>							

- 2 [完了] をクリックします。

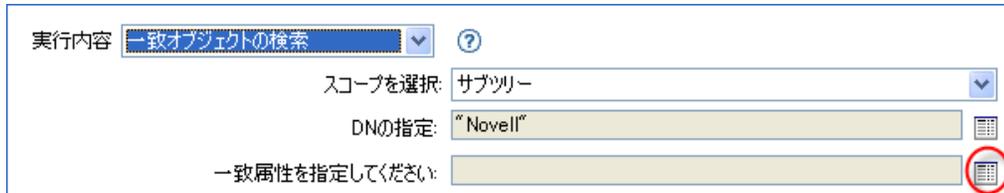
16.7 一致属性ビルダ

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、[\(423 ページ\)](#) **一致オブジェクトの検索**によって使用される属性および値を選択できます。

たとえば、共通名と場所に基づいてユーザを一致させるには

- 1 [一致オブジェクトの検索] のアクションを選択します。

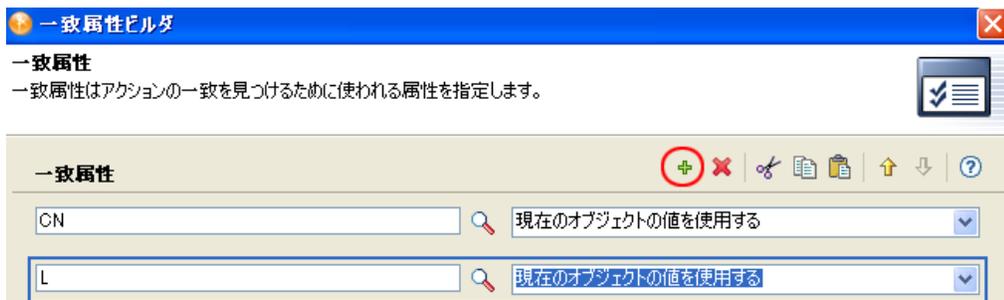
- 一致オブジェクトの検索の範囲を選択します。[エントリ]、[サブオーディネート]、または [サブツリー] から選択します。
- 検索の開始点となる DN を指定します。
- [一致属性の編集] アイコン  をクリックして、一致属性ビルダを起動します。



- [属性の参照]  アイコンをクリックして、スキーマブラウザを起動します。
- [属性] タブをクリックして、属性を参照して選択します。



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



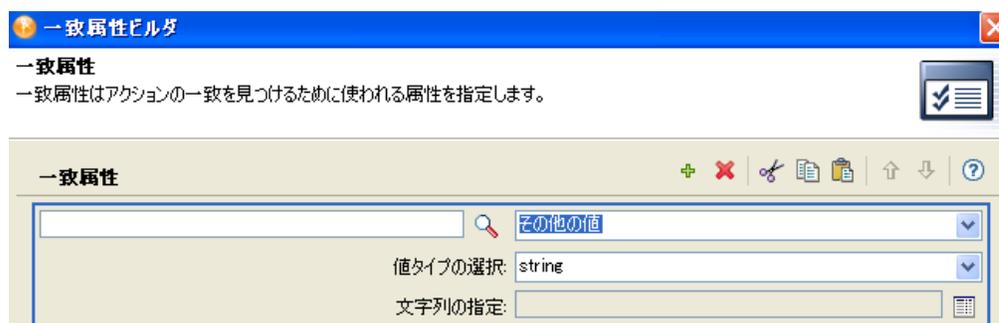
- [完了] をクリックします。

一致属性ビルダでは、現在のオブジェクトからの値を使用する代わりに別の値を指定することができます。別の値を使用するには、[現在のオブジェクトの値を使用する] の代わりに [その他の値] を選択します。指定できる値タイプは複数あります。

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

その他の値を使用する

- 1 一致属性ビルダを起動して、[その他の値] を選択します。



- 2 目的の値タイプを選択します。
- 3 値を指定して [OK] をクリックします。

16.8 名前付き文字列ビルダ

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

- ◆ (425 ページ) イベントの生成
- ◆ (437 ページ) 電子メールの送信
- ◆ (438 ページ) テンプレートから電子メールを送信

- 1 ドロップダウンリストから、文字列の名前を選択します。
- 2 [引数を編集する] アイコン  をクリックして引数ビルダを起動し、文字列の値を作成します。

名前付き文字列ビルダ
文字列要素は引数の値を指定します。



名前	文字列の値:	+	×	✂	📄	📂	↑	↓	?
<input type="text"/>	<input type="text"/>								

3 [完了] をクリックします。

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

名前付き文字列ビルダ
文字列要素は引数の値を指定します。



名前	文字列の値:	+	×	✂	📄	📂	↑	↓	?
to	<input type="text"/>								
subject	<input type="text"/>								
message	<input type="text"/>								

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

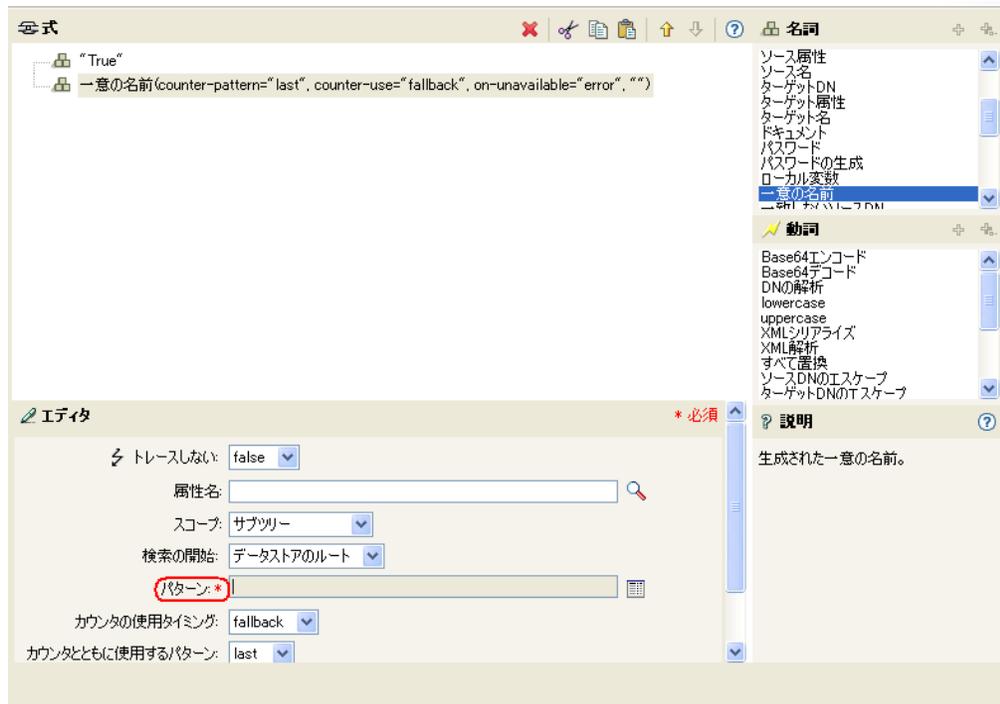
16.9 パターン文字列ビルダ

パターン文字列ビルダは、(483 ページ) 一意の名前トークンが選択されているときに引数ビルダのエディタから起動できます。引数ビルダのエディタペインに [パターン] フィールドが表示されるので、このフィールドをクリックして、パターン文字列ビルダを起動します。

図 16-6 引数ビルダ内の一意的名前トークン

引数の作成および編集

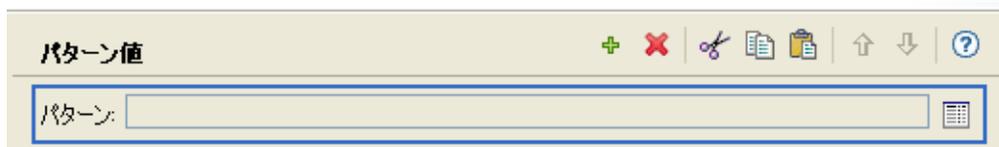
引数を構成するために式領域にコンポーネントを追加または式領域からコンポーネントを削除します。エディタでコンポーネント値を指定します。



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

パターンビルダ

パターンリストの定義



- 3 [完了] をクリックします。

16.10 引数値リストビルダ

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

- ◆ (439 ページ) デフォルト属性値の設定

図 16-7 デフォルト属性値の設定

実行内容 デフォルト属性値の設定 ?

属性名を指定してください: * 🔍

ライトバック: false ▼

引数値を指定してください: * 📄

- 1 値のタイプ ([counter (カウンタ)], [DN], [int (整数)], [間隔], [octet (オクテット)], [state (状態)], [文字列], [構造], [teleNumber (電話番号)], [時間]) を選択します。> > >
- 2 [値リストを編集する] アイコン  をクリックします。

引数値

引数値は属性に使用される値を指定します。



タイプ 引数値 + × ✂ 📄 📄 ↑ ↓ ?

structured ▼ 📄

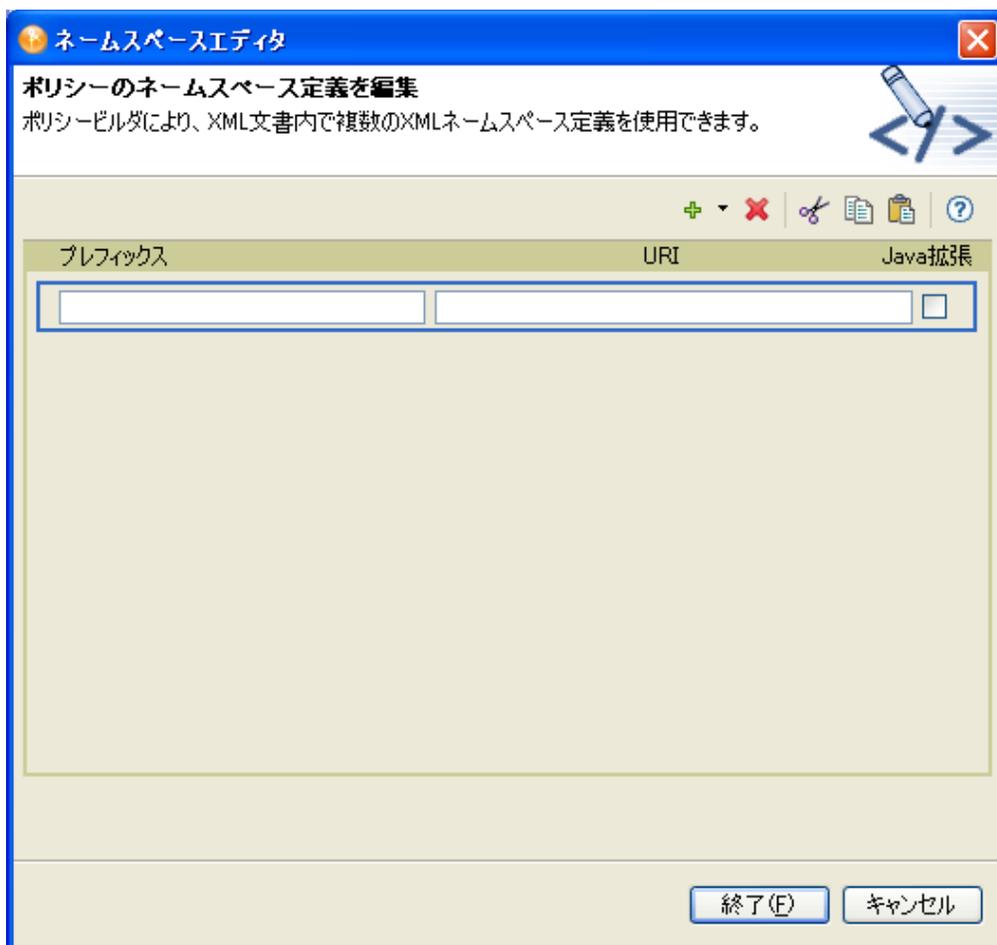
- 3 [引数を編集する] アイコン  をクリックします。
- 4 アクションコンポーネントの値を作成します。
値を入力するか、または [引数を編集する]  アイコンをクリックして引数ビルダ内で値を作成します。
- 5 [完了] をクリックします。

16.11 ネームスペースエディタ

ポリシービルダでは、XML ドキュメント内で、複数の XML ネームスペースを使用できます。ネームスペースを定義するには、[名前] フィールドでネームスペースのプレフィックスを指定して、[URI] フィールドで URI を指定します。[Java 拡張] チェックボックスはオフのままにします。

XML ネームスペースを使用し、Xpath を介して Java* クラスにアクセスすることもできます。Java クラスのネームスペースを作成するには、ネームスペースプレフィックスを [名前] フィールドに指定し、クラス名を [URI] フィールドに指定して、[Java 拡張] チェックボックスを選択します。

図 16-8 ネームスペースエディタ



16.11.1 ネームスペースを使用した Java クラスへのアクセス

Novell では、ポリシービルダから XPath 式を使用して呼び出すことができる Identity Manager Java* クラスが複数用意されています。次のリンクから、これらの Java クラスの Javadoc リファレンスを開くことができます。

- ◆ [com.novell.nds.dirxml.driver.XdsQueryProcessor](http://developer.novell.com/documentation/dirxml/dirxmlbk/api/com/novell/nds/dirxml/driver/XdsQueryProcessor.html) (<http://developer.novell.com/documentation/dirxml/dirxmlbk/api/com/novell/nds/dirxml/driver/XdsQueryProcessor.html>)
- ◆ [com.novell.nds.dirxml.driver.XdsCommandProcessor](http://developer.novell.com/documentation/dirxml/dirxmlbk/api/com/novell/nds/dirxml/driver/XdsCommandProcessor.html) (<http://developer.novell.com/documentation/dirxml/dirxmlbk/api/com/novell/nds/dirxml/driver/XdsCommandProcessor.html>)
- ◆ [com.novell.nds.dirxml.driver.DNConverter](http://developer.novell.com/documentation/dirxml/dirxmlbk/api/com/novell/nds/dirxml/driver/DNConverter.html) (<http://developer.novell.com/documentation/dirxml/dirxmlbk/api/com/novell/nds/dirxml/driver/DNConverter.html>)

Java Developer Kit (JDK*) にも、`java.lang.String` や `java.lang.System` など、役に立つクラスがいくつかあります。これらのクラスのリファレンスは JDK で使用できます。

XPath の使用およびリストされている Novell Java クラスの詳細については、[DirXML Driver Developer Kit](http://developer.novell.com/documentation/dirxml/dirxmlbk/ref/dirxmlfaq.html) (<http://developer.novell.com/documentation/dirxml/dirxmlbk/ref/dirxmlfaq.html>) を参照してください。

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、プレ Identity Manager 3.5 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。

- ◆ 378 ページの「If 関連付け」
- ◆ 379 ページの「If 属性」
- ◆ 381 ページの「If クラス名」
- ◆ 383 ページの「If ターゲット属性」
- ◆ 385 ページの「If ターゲット DN」
- ◆ 386 ページの「If エンタイトルメント」
- ◆ 388 ページの「If グローバル構成値」
- ◆ 390 ページの「If ローカル変数」
- ◆ 392 ページの「If 名前付きパスワード」
- ◆ 393 ページの「If 操作属性」
- ◆ 395 ページの「If 操作プロパティ」
- ◆ 397 ページの「If 操作」
- ◆ 399 ページの「If パスワード」
- ◆ 400 ページの「If ソース属性」
- ◆ 402 ページの「If ソース DN」
- ◆ 403 ページの「If XPath 式」

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。

値

選択演算子に定義されている値を含みます。この値は条件によって使用されます。

- ◆ 等しい
- ◆ 等しくない

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。

値

選択演算子に定義されている値を含みます。この値は条件によって使用されます。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

比較モード

条件は、比較が実行される方法を示す比較モードパラメータを持っています。

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

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。

値

選択演算子に定義されている値を含みます。この値は条件によって使用されます。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

比較モード

条件は、比較が実行される方法を示す比較モードパラメータを持っています。

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。

モード	説明
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

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。

値

選択演算子に定義されている値を含みます。この値は条件によって使用されます。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

比較モード

条件は、比較が実行される方法を示す比較モードパラメータを持っています。

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。

モード	説明
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 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

選択演算子に定義されている値を含みます。この値は条件によって使用されます。

- ◆ 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

If エンタイトルメント

現在の操作または識別ポータル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タルで、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。
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

選択演算子に定義されている値を含みます。この値は条件によって使用されます。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

比較モード

条件は、比較が実行される方法を示す比較モードパラメータを持っています。

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

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 \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policycomponents.html\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policycomponents.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。

値

選択演算子に定義されている値を含みます。この値は条件によって使用されます。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

比較モード

条件は、比較が実行される方法を示す比較モードパラメータを持っています。

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

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。

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 \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policycomponents.html\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policycomponents.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。

値

選択演算子に定義されている値を含みます。この値は条件によって使用されます。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

比較モード

条件は、比較が実行される方法を示す比較モードパラメータを持っています。

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

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。

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。

演算子は次の場合に **True** を返します ...

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。
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

選択演算子に定義されている値を含みます。この値は条件によって使用されます。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

比較モード

条件は、比較が実行される方法を示す比較モードパラメータを持っています。

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 <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
- ◆ 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

f 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。

値

選択演算子に定義されている値を含みます。この値は条件によって使用されます。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

比較モード

条件は、比較が実行される方法を示す比較モードパラメータを持っています。

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。

モード	説明
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

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。

値

選択演算子に定義されている値を含みます。この値は条件によって使用されます。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

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
- ◆ チェックパスワード
- ◆ 削除
- ◆ 名前付きパスワードの取得
- ◆ パラメータの開始
- ◆ インスタンス
- ◆ 変更
- ◆ 関連付けの変更
- ◆ パスワード変更
- ◆ 移動
- ◆ 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

条件は、比較が実行される方法を示す比較モードパラメータを持っています。

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

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。

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。

値

選択演算子に定義されている値を含みます。この値は条件によって使用されます。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

比較モード

条件は、比較が実行される方法を示す比較モードパラメータを持っています。

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。

モード	説明
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 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

選択演算子に定義されている値を含みます。この値は条件によって使用されます。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

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

ポリシーで XPath 式を使用する詳細については、[XPath 1.0 式 \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policyxpathexpression.html#policyxpathexpression\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policyxpathexpression.html#policyxpathexpression) を参照してください。

プレ Identity Manager 3.5 のアクション

18

アクションは、格納されたルールを満たした場合に実行されます。いくつかのアクションには [Mode] フィールドがあります。モードは、ポリシーを実行しているコンテキストが選択したモードと互換性がある場合には実行時に無視されます。

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

- ◆ 407 ページの「関連付けの追加」
- ◆ 408 ページの「ターゲット属性値の追加」
- ◆ 409 ページの「ターゲットオブジェクトの追加」
- ◆ 410 ページの「ソース属性値の追加」
- ◆ 411 ページの「ソースオブジェクトの追加」
- ◆ 412 ページの「XML 要素の追加」
- ◆ 413 ページの「XML テキストの追加」
- ◆ 414 ページの「中断」
- ◆ 415 ページの「ターゲット属性値のクリア」
- ◆ 416 ページの「操作プロパティのクリア」
- ◆ 417 ページの「ソース属性値のクリア」
- ◆ 418 ページの「SSO 資格情報のクリア」
- ◆ 419 ページの「XPath 式によるクローン」
- ◆ 420 ページの「操作属性のクローン」
- ◆ 421 ページの「ターゲットオブジェクトの削除」
- ◆ 422 ページの「ソースオブジェクトの削除」
- ◆ 423 ページの「一致オブジェクトの検索」
- ◆ 424 ページの「毎」
- ◆ 425 ページの「イベントの生成」
- ◆ 427 ページの「エンタイトルメントの実装」
- ◆ 428 ページの「ターゲットオブジェクトの移動」
- ◆ 429 ページの「ソースオブジェクトの移動」
- ◆ 430 ページの「操作属性の再フォーマット値」
- ◆ 431 ページの「関連付けの削除」
- ◆ 432 ページの「ターゲット属性値の削除」
- ◆ 433 ページの「ソース属性値の削除」
- ◆ 434 ページの「ターゲットオブジェクトの名前変更」
- ◆ 435 ページの「操作属性の名前変更」

- ◆ 436 ページの「ソースオブジェクトの名前変更」
- ◆ 437 ページの「電子メールの送信」
- ◆ 438 ページの「テンプレートから電子メールを送信」
- ◆ 439 ページの「デフォルト属性値の設定」
- ◆ 440 ページの「ターゲット属性値の設定」
- ◆ 441 ページの「ターゲットパスワードの設定」
- ◆ 442 ページの「ローカル変数の設定」
- ◆ 443 ページの「操作関連付けの設定」
- ◆ 444 ページの「操作クラス名の設定」
- ◆ 445 ページの「操作ターゲット DN」
- ◆ 446 ページの「操作プロパティの設定」
- ◆ 447 ページの「操作ソース DN の設定」
- ◆ 448 ページの「操作テンプレート DN の設定」
- ◆ 449 ページの「ソース属性値の設定」
- ◆ 450 ページの「ソースパスワードの設定」
- ◆ 451 ページの「SSO 資格情報の設定」
- ◆ 452 ページの「SSO パスフレーズの設定」
- ◆ 453 ページの「XML 属性の設定」
- ◆ 454 ページの「ステータス」
- ◆ 455 ページの「操作属性のストリップ」
- ◆ 456 ページの「XPath のストリップ」
- ◆ 457 ページの「メッセージのトレース」
- ◆ 458 ページの「拒否」
- ◆ 459 ページの「操作属性値がない場合は拒否」

関連付けの追加

関連付けの追加コマンドを指定した関連付けとともに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。

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。

ターゲット属性値の追加

ターゲットデータストア内のオブジェクトの属性に値を追加します。

フィールド

属性名

属性の名前を指定します。

クラス名

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

モード

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

オブジェクト

ターゲットオブジェクトの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

DN、関連付け、または現在のオブジェクトをターゲットオブジェクトとして指定します。

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。

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。

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。

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。

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。

オブジェクト

ターゲットオブジェクトの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

DN、関連付け、または現在のオブジェクトをターゲットオブジェクトとして指定します。

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。

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 **ソース属性値の追加** アクションで同じ DN を使って追加する必要があります。

フィールド

クラス名

追加するオブジェクトのクラス名を指定します。

DN

追加するオブジェクトの DN を指定します。

XML 要素の追加

XPath 式で選択されたエレメントのセットに、[名前] フィールドで指定された名前を使用してカスタムエレメントを追加します。

フィールド

名前

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

XPath 式

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

備考

ポリシーで XPath 式を使用する詳細については、[XPath 1.0 式 \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policyxpathexpression.html#policyxpathexpression\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policyxpathexpression.html#policyxpathexpression) を参照してください。

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。

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。

文字列

追加するテキストを指定します。

備考

ポリシーで XPath 式を使用する詳細については、[XPath 1.0 式 \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policyxpathexpression.html#policyxpathexpression\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policyxpathexpression.html#policyxpathexpression) を参照してください。

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。

フィールド

中断アクションにはフィールドがありません。

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。

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。

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の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

DN、関連付け、または現在のオブジェクトをターゲットオブジェクトとして選択します。

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 / 値ペアです。

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。

ソース属性値のクリア

ソースデータストア内の1つのオブジェクトから、すべての属性値を削除します。

フィールド

属性名

属性の名前を指定します。

クラス名

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

オブジェクト

ターゲットオブジェクトの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

DN、関連付け、または現在のオブジェクトをターゲットオブジェクトとして選択します。

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 資格情報プロビジョニングポリシー \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_credprov/data/bookinfo.html\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_credprov/data/bookinfo.html) を参照してください。

フィールド

資格情報リポジトリオブジェクトの DN

リポジトリオブジェクトの DN を指定します。

ターゲットユーザの DN

ターゲットユーザの DN を指定します。

アプリケーションのアクティベーションキー ID

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

ログインパラメータの文字列

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

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。

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。

備考

ポリシーで XPath 式を使用する詳細については、[XPath 1.0 式 \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policyxpathexpression.html#policyxpathexpression\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policyxpathexpression.html#policyxpathexpression) を参照してください。

操作属性のクローン

現在の操作で属性に行った内容を、現在の操作内の別の属性にコピーします。

フィールド

ソース名

コピー元の属性の名前を指定します。

ターゲット名

コピー先の属性の名前を指定します。

ターゲットオブジェクトの削除

ターゲットデータストア内のオブジェクトを削除します。

フィールド

モード

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

オブジェクト

ターゲットデータストア内の削除するターゲットオブジェクト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

DN、関連付け、または現在のオブジェクトをターゲットオブジェクトとして選択します。

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

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

DN、関連付け、または現在のオブジェクトをターゲットオブジェクトとして選択します。

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。

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

ローカル変数を使用される場合、アクションを繰り返すと、現在のノードは異なる値になります。

ノードセット内の現在のノードがエンタイトルメント要素の場合、アクションは**エンタイトルメントの実装**アクションに格納されているかのようにマークされます。現在のノードがクエリによって返されたクエリ要素の場合、トークンは自動的に次のクエリ結果のバッチを取得して処理するよう使用されます。

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。

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 '\8b\7d 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 '\95\5c 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。
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す。
value3	ここに入力される任意の数字は、 value3 イベントフィールドに格納されます。
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 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。

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。

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

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。

コンテナ移動する

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るかどうかを指定します。

ソースオブジェクトの移動

ソースデータストア内のオブジェクトを移動します。

フィールド

移動するオブジェクト

移動するオブジェクトを選択します。このオブジェクト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。

操作属性の再フォーマット値

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

フィールド

名前

属性の名前を指定します。

値タイプ

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。

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。

ターゲット属性値の削除

ターゲットデータストア内のオブジェクトから、属性値を1つ削除します。

フィールド

属性名

属性の名前を指定します。

クラス名

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

モード

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

オブジェクトの選択

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

値タイプ

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。

ソース属性値の削除

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

フィールド

属性名

属性の名前を指定します。

クラス名

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

オブジェクト

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

値タイプ

削除する属性値の構文を指定します。

文字列

削除する属性値を指定します。

ターゲットオブジェクトの名前変更

ターゲットデータストア内のオブジェクトの名前を変更します。

フィールド

モード

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

オブジェクト

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

文字列

オブジェクトの新しい名前を指定します。

操作属性の名前変更

現在の操作内で出現したすべての属性の名前を変更します。

フィールド

ソース名

変更前の属性名を指定します。

ターゲット名

新しい属性名を指定します。

ソースオブジェクトの名前変更

ソースデータストア内のオブジェクトの名前を変更します。

フィールド

オブジェクトの選択

ターゲットオブジェクトを選択します。このオブジェクトは現在のオブジェクトにすることも、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。

電子メールの送信

電子メール通知を送信します。

フィールド

ID

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

サーバ

SMTP サーバ名を指定します。

[パスワード]

(オプション)SMTP サーバのアカウントのパスワードを指定します。

重要: SMTP サーバアカウントパスワードを名前付きパスワードとしてドライバオブジェクトに保存できます。これにより、パスワードが暗号化されます。それ以外の場合は、パスワードを入力するとクリアテキストとして保存されます。名前付きパスワードの詳細については、『[Novell Identity Manager Administration Guide \(http://www.novell.com/documentation/idm35/index.html\)](http://www.novell.com/documentation/idm35/index.html)』の「Using Named Password (名前付きパスワードの使用)」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。

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
>	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。受信者のカンマ区切りリストを含むことができます。
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。受信者のカンマ区切りリストを含むことができます。
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。受信者のカンマ区切りリストを含むことができます。
送信者	電子メールの発信アドレスとして使用されるアドレスを指定します。
reply-to	電子メールメッセージの返信アドレスとして使用されるアドレスを指定します。
件名	電子メールの件名を指定します。
メッセージ	電子メールメッセージの内容を指定します。
エンコード	電子メールメッセージで使用する文字エンコードを指定します。

テンプレートから電子メールを送信

テンプレートを使用して、電子メール通知を生成します。

フィールド

通知 DN

SMTP 通知設定オブジェクトのスラッシュ形式の DN を指定します。

テンプレート DN

電子メールテンプレートオブジェクトのスラッシュ形式の DN を指定します。

[パスワード]

(オプション)SMTP サーバのアカウントのパスワードを指定します。

重要 : SMTP サーバアカウントパスワードを名前付きパスワードとしてドライバオブジェクトに保存できます。これにより、パスワードが暗号化されます。それ以外の場合は、パスワードを入力するとクリアテキストとして保存されます。名前付きパスワードの詳細については、『[Novell Identity Manager Administration Guide \(http://www.novell.com/documentation/idm35/index.html\)](http://www.novell.com/documentation/idm35/index.html)』の「Using Named Password (名前付きパスワードの使用)」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
>	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。受信者のカンマ区切りリストを含むことができます。
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。受信者のカンマ区切りリストを含むことができます。
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。受信者のカンマ区切りリストを含むことができます。
reply-to	電子メールメッセージの返信アドレスとして使用されるアドレスを指定します。
エンコード	電子メールメッセージで使用する文字エンコードを指定します。

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

デフォルト属性値の設定

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

フィールド

属性名

デフォルト属性の名前を指定します。

ライトバック

デフォルト値をソースデータストアにもライトバックするかどうかを選択します。

値

属性のデフォルト値を指定します。

ターゲット属性値の設定

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

フィールド

属性名

属性の名前を指定します。

クラス名

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

モード

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

オブジェクト

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

値タイプ

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。

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オブジェクトのパスワードを設定します。

フィールド

モード

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

文字列

設定するパスワードを指定します。

ローカル変数の設定

指定した名前で、指定した文字列値、XPath 1.0 ノードセット、または Java* オブジェクトのローカル変数を設定します。

フィールド

変数名

新しいローカル変数の名前を指定します。

変数タイプ

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

文字列

[*変数タイプ*] フィールドで要求される形式で、変数の値を指定します。

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。

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。

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。

操作プロパティの設定

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

フィールド

プロパティ名

操作プロパティの名前を指定します。

文字列

文字列の名前を指定します。

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。

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。

ソース属性値の設定

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

フィールド

属性名

属性の名前を指定します。

クラス名

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

オブジェクト

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

値タイプ

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。

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

文字列

設定するパスワードを指定します。

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 資格情報プロビジョニングポリシー \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_credprov/data/bookinfo.html\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_credprov/data/bookinfo.html) を参照してください。

フィールド

資格情報リポジトリオブジェクトの DN

リポジトリオブジェクトの DN を指定します。

ターゲットユーザの DN

ターゲットユーザの DN を指定します。

アプリケーションのアクティベーションキー ID

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

ログインパラメータの文字列

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

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 のパスワード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 資格情報プロビジョニングポリシー \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_credprov/data/bookinfo.html\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_credprov/data/bookinfo.html) を参照してください。

フィールド

資格情報リポジトリオブジェクトの DN

リポジトリオブジェクトの DN を指定します。

ターゲットユーザの DN

ターゲットユーザの DN を指定します。

質問文字列

SecureLogin パスフレーズの質問を指定します。

回答文字列

SecureLogin パスフレーズの回答を指定します。

XML 属性の設定

XPath 式で選択された要素のセットに XML 属性を設定します。

フィールド

名前

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

XPath 式

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

文字列

XML 属性の値を指定します。

備考

ポリシーで XPath 式を使用する詳細については、[XPath 1.0 式 \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policyxpathexpression.html#policyxpathexpression\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policyxpathexpression.html#policyxpathexpression) を参照してください。

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。

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。

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。

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

ポリシーで XPath 式を使用する詳細については、[XPath 1.0 式 \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policyxpathexpression.html#policyxpathexpression\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policyxpathexpression.html#policyxpathexpression) を参照してください。

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 管理ガイド \(http://www.novell.com/documentation/idm35/index.html\)](http://www.novell.com/documentation/idm35/index.html)』の「バージョン情報」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。

拒否

現在の操作を拒否します。

フィールド

フィールドはありません。

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。

プレ Identity Manager 3.5 名詞トークン

19

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、プレ Identity Manager 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。

- ◆ 462 ページの「追加されたエンタイトルメント」
- ◆ 463 ページの「関連付け」
- ◆ 464 ページの「属性」
- ◆ 465 ページの「クラス名」
- ◆ 466 ページの「ターゲット属性」
- ◆ 467 ページの「ターゲット DN」
- ◆ 468 ページの「ターゲット名」
- ◆ 469 ページの「エンタイトルメント」
- ◆ 470 ページの「グローバル構成値」
- ◆ 471 ページの「ローカル変数」
- ◆ 472 ページの「名前付きパスワード」
- ◆ 473 ページの「説明」
- ◆ 474 ページの「操作属性」
- ◆ 475 ページの「操作プロパティ」
- ◆ 476 ページの「[パスワード]」
- ◆ 477 ページの「削除された属性」
- ◆ 478 ページの「削除されたエンタイトルメント」
- ◆ 479 ページの「ソース属性」
- ◆ 480 ページの「ソース DN」
- ◆ 481 ページの「ソース名」
- ◆ 482 ページの「テキスト」
- ◆ 483 ページの「一意の名前」
- ◆ 485 ページの「一致しないソース DN」
- ◆ 486 ページの「XPath」

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。

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。

関連付け

現在の操作から関連付けの値に展開します。

フィールド

フィールドはありません。

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。

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。

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。

フィールド

フィールドはありません。

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。

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。

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。

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。

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。

フィールド

フィールドはありません。

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。

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。

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。

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。

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

名前付きパスワードを指定します。

説明

現在の操作の名前に展開します。

フィールド

フィールドはありません。

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。

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。

[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。

フィールド

フィールドはありません。

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

削除する属性の名前を指定します。

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。

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。

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。

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。

名前

属性の名前。

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。

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 の部分で使用されます。

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。

フィールド

フィールドはありません。

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。

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。

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ボールドであり、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、名前は必須です。

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、カウンタは使い果たされたと考えられます。

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。

指定された名前の順番は次のようにテストされます。

- ◆ 各パターンは指定された順にテストされます。 `counter-use="always"` で、パターンが `counter-pattern` で示されているパターンのいずれかを使用している場合、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。
- ◆ パターンをすべて使っても一意な名前が見つからず、`counter-use="fallback"` の場合は、`counter-pattern` で示されたパターンがカウンタを使用して再試行されます。

指定されたパターンとカウンタの組み合わせすべてが消費された場合は、`on-unavailable` によって指定されたアクションが実行されます。

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。

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

ポリシーで XPath 式を使用する詳細については、[XPath 1.0 式 \(http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policyxpathexpression.html#policyxpathexpression\)](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy/data/policyxpathexpression.html#policyxpathexpression) を参照してください。

プレ Identity Manager 3.5 動詞トークン

20

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、プレ Identity Manager 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。

- ◆ [488 ページの「ターゲット DN のエスケープ」](#)
- ◆ [489 ページの「ソース DN のエスケープ」](#)
- ◆ [490 ページの「小文字」](#)
- ◆ [491 ページの「DN の解析」](#)
- ◆ [493 ページの「すべて置換」](#)
- ◆ [494 ページの「最初を置換」](#)
- ◆ [495 ページの「部分文字列」](#)
- ◆ [496 ページの「大文字」](#)

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。

フィールド

フィールドはありません。

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。

フィールド

フィールドはありません。

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。

フィールド

フィールドはありません。

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 区切り文字
- ◆ RDN 区切り文字
- ◆ 名前ディバイダ
- ◆ 名前の値の区切り文字
- ◆ 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。

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。

置換文字列

置換する文字列を指定します。

備考

正規表現の作成についての詳細は、次を参照してください。

- ◆ Sun の Java Web サイト (<http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html>)
- ◆ Sun の Java Web サイト (<http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Matcher.html#replaceAll> (java.lang.String))

[パターン] のオプションには `CASE_INSENSITIVE`、`DOTALL`、および `UNICODE_CASE` が使用されますが、適切な埋め込みエスケープを使用して逆の意味を指定することができます。

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。

置換文字列

置換する文字列を指定します。

備考

一致したインスタンスは、`[置換文字列]` フィールドで指定された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\(java.lang.String\)](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。

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。負の数値は (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。

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。

フィールド

フィールドはありません。