

Novell Identity Manager

3.5.1

September 28, 2007

DTD リファレンス

www.novell.com



Novell®

保証と著作権

米国 Novell, Inc. およびノベル株式会社は、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	37
1 Identity Manager DTD リファレンスの概要	39
2 フィルタ DTD	41
2.1 フィルタ DTD の要素	41
フィルタ	42
例 42	
許可されるコンテンツ	42
属性	42
内容の規則	42
親要素	42
filter-attr	43
備考	43
例 44	
内容に使用できる要素	44
属性	45
内容の宣言	45
親要素	45
filter-class	46
備考	46
例 46	
内容に使用できる要素	46
属性	46
内容の規則	47
親要素	47
3 NDS DTD	49
3.1 NDS DTD の要素	49
追加	53
備考	53
例 53	
内容に使用できる要素	53
属性	54
内容の規則	55
親要素	55
関連付けの追加	56
例 56	
内容に使用できる要素	56
属性	56
内容の規則	56
親要素	57

add-attr	58
例 58	
内容に使用できる要素 58	
属性 58	
内容の規則 58	
親要素 59	
add-value	60
例 60	
内容に使用できる要素 60	
属性 60	
親要素 60	
allow-attr	61
例 61	
内容に使用できる要素 61	
属性 61	
内容の宣言 61	
親要素 61	
allow-class	62
例 62	
内容に使用できる要素 62	
属性 62	
内容の規則 62	
親要素 62	
app-name	63
例 63	
内容に使用できる要素 63	
属性 63	
内容の規則 63	
親要素 63	
関連付け	64
例 64	
内容に使用できる要素 64	
属性 64	
内容の規則 64	
親要素 64	
attr	66
例 66	
内容に使用できる要素 66	
属性 66	
内容の規則 66	
親要素 66	
attr-def	67
例 67	
内容に使用できる要素 67	
属性 67	
内容の宣言 68	
親要素 68	
attr-name	69

例 69	
内容に使用できる要素 69	
属性 69	
内容の規則 69	
親要素 69	
attr-name-map	70
備考 70	
例 70	
内容に使用できる要素 70	
属性 70	
内容の規則 71	
親要素 71	
authentication-info	72
例 72	
内容に使用できる要素 72	
属性 72	
内容の規則 72	
親要素 72	
オブジェクトパスワードの確認	73
備考 73	
例 73	
内容に使用できる要素 73	
属性 73	
内容の規則 73	
親要素 74	
チェックパスワード	75
例 75	
内容に使用できる要素 75	
属性 75	
内容の規則 75	
親要素 75	
class-def	76
例 76	
内容に使用できる要素 76	
属性 76	
内容の規則 76	
親要素 76	
class-name	77
例 77	
内容に使用できる要素 77	
属性 77	
内容の規則 77	
親要素 77	
component	78
例 78	
内容に使用できる要素 78	
属性 78	

内容の規則 78	
親要素 79	
config-object	80
備考 80	
例 80	
内容に使用できる要素 80	
属性 80	
内容の規則 80	
親要素 80	
contact	81
例 81	
内容に使用できる要素 81	
属性 81	
内容の規則 81	
親要素 81	
copy-attr	82
備考 82	
例 82	
内容に使用できる要素 82	
属性 82	
内容の宣言 82	
親要素 82	
copy-name	83
備考 83	
例 83	
内容に使用できる要素 83	
属性 83	
内容の宣言 83	
親要素 83	
copy-path	84
備考 84	
例 84	
内容に使用できる要素 84	
属性 84	
内容の宣言 84	
親要素 84	
copy-path-suffix	85
備考 85	
例 85	
内容に使用できる要素 85	
属性 85	
内容の宣言 85	
親要素 85	
create-rule	86
備考 86	
例 86	
内容に使用できる要素 86	

属性 87	
内容の規則 87	
親要素 87	
create-rules	88
備考 88	
例 88	
内容に使用できる要素 88	
属性 89	
内容の規則 89	
親要素 89	
削除	90
備考 90	
例 90	
内容に使用できる要素 90	
属性 90	
内容の規則 91	
親要素 91	
driver-config	92
備考 92	
例 92	
内容に使用できる要素 92	
属性 93	
内容の規則 93	
親要素 93	
driver-filter	94
例 94	
内容に使用できる要素 94	
属性 94	
内容の規則 94	
親要素 94	
driver-options	95
例 95	
内容に使用できる要素 95	
属性 95	
内容の規則 95	
親要素 95	
driver-state	96
例 96	
内容に使用できる要素 96	
属性 96	
内容の規則 96	
親要素 96	
名前付きパスワードの取得	97
例 97	
内容に使用できる要素 97	
属性 97	
内容の規則 97	

親要素 97	
パラメータの開始	98
備考 98	
例 98	
内容に使用できる要素 100	
属性 100	
内容の規則 100	
親要素 100	
input	101
例 101	
内容に使用できる要素 101	
属性 101	
内容の規則 102	
親要素 102	
インスタンス	103
例 103	
内容に使用できる要素 103	
属性 103	
内容の規則 104	
親要素 104	
match-attr	105
備考 105	
例 105	
内容に使用できる要素 105	
属性 105	
内容の規則 105	
親要素 105	
match-class	106
備考 106	
例 106	
内容に使用できる要素 106	
属性 106	
内容の宣言 106	
親要素 106	
match-path	107
備考 107	
例 107	
内容に使用できる要素 107	
属性 107	
内容の宣言 107	
親要素 108	
matching-rule	109
備考 109	
例 109	
内容に使用できる要素 109	
属性 109	
内容の規則 109	

親要素 110	
matching-rules.....	111
備考 111	
例 111	
内容に使用できる要素 112	
属性 112	
内容の規則 112	
親要素 112	
変更.....	113
備考 113	
例 113	
内容に使用できる要素 113	
属性 113	
内容の規則 114	
親要素 114	
関連付けの変更.....	115
例 115	
内容に使用できる要素 115	
属性 115	
内容の規則 115	
親要素 115	
modify-attr.....	116
備考 116	
例 116	
内容に使用できる要素 116	
属性 116	
内容の規則 116	
親要素 117	
パスワード変更.....	118
備考 118	
例 118	
内容に使用できる要素 118	
属性 118	
内容の規則 119	
親要素 119	
移動.....	120
備考 120	
例 120	
内容に使用できる要素 120	
属性 120	
内容の規則 121	
親要素 121	
nds.....	122
備考 122	
例 122	
内容に使用できる要素 122	
属性 123	

内容の規則	123
親要素	123
nds-name	124
例	124
内容に使用できる要素	124
属性	124
内容の規則	124
親要素	124
new-name	125
例	125
内容に使用できる要素	125
属性	125
内容の規則	125
親要素	125
old-password	126
例	126
内容に使用できる要素	126
属性	126
内容の規則	126
親要素	126
operation-data	127
備考	127
例	127
内容に使用できる要素	127
属性	127
内容の規則	127
親要素	127
output	129
例	129
内容に使用できる要素	129
属性	129
内容の規則	129
親要素	129
parent	130
備考	130
例	130
内容に使用できる要素	130
属性	130
内容の規則	131
親要素	131
パスワード	132
例	132
内容に使用できる要素	132
属性	132
内容の規則	132
親要素	132
placement	133
備考	133

例 133	
内容に使用できる要素 133	
属性 133	
内容の規則 133	
親要素 133	
placement-rule.....	134
備考 134	
例 134	
内容に使用できる要素 134	
属性 134	
内容の規則 134	
親要素 135	
placement-rules.....	136
備考 136	
例 137	
内容に使用できる要素 137	
属性 137	
内容の規則 137	
親要素 137	
publisher-options.....	138
例 138	
内容に使用できる要素 138	
属性 138	
内容の規則 138	
親要素 138	
publisher-state.....	139
備考 139	
例 139	
内容に使用できる要素 139	
属性 139	
内容の規則 139	
親要素 139	
クエリー.....	140
備考 140	
例 141	
内容に使用できる要素 141	
属性 141	
内容の規則 142	
親要素 142	
query-ex.....	143
備考 143	
例 143	
内容に使用できる要素 144	
属性 144	
内容の規則 145	
親要素 145	
クエリスキーマ.....	146

備考 146	
例 146	
内容に使用できる要素 146	
属性 146	
内容の規則 146	
親要素 146	
query-token.	147
備考 147	
内容に使用できる要素 147	
属性 147	
内容の規則 147	
親要素 147	
read-attr	148
例 148	
内容に使用できる要素 148	
属性 148	
内容の宣言 148	
親要素 148	
read-parent.	149
例 149	
内容に使用できる要素 149	
属性 149	
内容の宣言 149	
親要素 149	
remove-all-values.	150
例 150	
内容に使用できる要素 150	
属性 150	
内容の宣言 150	
親要素 150	
関連付けの削除	151
例 151	
内容に使用できる要素 151	
属性 151	
内容の規則 151	
親要素 151	
remove-value	152
例 152	
内容に使用できる要素 152	
属性 152	
内容の規則 152	
親要素 152	
名前を変更する	153
備考 153	
例 153	
内容に使用できる要素 153	
属性 153	

内容の規則	154
親要素	154
required-attr	155
備考	155
例	155
内容に使用できる要素	155
属性	155
内容の規則	155
親要素	156
スキーマ定義	157
備考	157
例	157
内容に使用できる要素	158
属性	158
内容の規則	159
親要素	159
search-attr	160
例	160
内容に使用できる要素	160
属性	160
内容の規則	160
親要素	160
search-class	161
例	161
内容に使用できる要素	161
属性	161
内容の宣言	161
親要素	161
server	162
例	162
内容に使用できる要素	162
属性	162
内容の規則	162
親要素	162
source	163
例	163
内容に使用できる要素	163
属性	163
内容の規則	163
親要素	163
status	164
備考	164
例	164
内容に使用できる要素	165
属性	165
内容の規則	165
親要素	165
subscriber-options	166

例 166	
内容に使用できる要素 166	
属性 166	
内容の規則 166	
親要素 166	
subscriber-state	167
備考 167	
例 167	
内容に使用できる要素 167	
属性 167	
内容の規則 167	
親要素 167	
同期	168
備考 168	
例 168	
内容に使用できる要素 168	
属性 168	
内容の規則 169	
親要素 169	
template	170
例 170	
内容に使用できる要素 170	
属性 170	
内容の規則 170	
親要素 170	
user.....	171
例 171	
内容に使用できる要素 171	
属性 171	
内容の規則 171	
親要素 171	
値	172
備考 172	
例 174	
内容に使用できる要素 174	
属性 174	
内容の規則 174	
親要素 175	
4 マップ DTD	177
4.1 マップ DTD の要素	177
col.....	178
例 178	
内容に使用できる要素 178	
属性 178	
内容の規則 178	
親要素 178	

col-def	179
備考 179	
例 179	
内容に使用できる要素 179	
属性 179	
内容の宣言 179	
親要素 180	
mapping-table	181
例 181	
内容に使用できる要素 181	
属性 181	
内容の規則 181	
親要素 181	
row	182
例 182	
内容に使用できる要素 182	
属性 182	
内容の規則 182	
親要素 182	
5 DirXML スクリプト DTD	183
5.1 DirXML スクリプト DTD の要素	183
actions	189
備考 189	
例 189	
内容に使用できる要素 189	
属性 191	
内容の規則 191	
親要素 191	
and	192
例 192	
内容に使用できる要素 192	
属性 192	
内容の規則 193	
親要素 193	
arg-actions	194
例 194	
内容に使用できる要素 194	
属性 196	
内容の規則 196	
親要素 196	
arg-association	197
例 197	
内容に使用できる要素 197	
属性 198	
内容の規則 198	
親要素 199	

arg-component	200
例 200	
内容に使用できる要素 200	
属性 201	
内容の規則 201	
親要素 202	
arg-conditions	203
例 203	
内容に使用できる要素 203	
属性 203	
内容の規則 203	
親要素 203	
arg-dn	204
例 204	
内容に使用できる要素 204	
属性 205	
内容の規則 205	
親要素 206	
arg-match-attr	208
例 208	
内容に使用できる要素 208	
属性 208	
内容の規則 208	
親要素 208	
arg-node-set	209
例 209	
内容に使用できる要素 209	
属性 210	
内容の規則 210	
親要素 211	
arg-object	212
例 212	
内容に使用できる要素 212	
属性 212	
内容の規則 212	
親要素 212	
arg-password	213
例 213	
内容に使用できる要素 213	
属性 216	
内容の規則 216	
親要素 216	
arg-string	217
例 217	
内容に使用できる要素 217	
属性 218	
内容の規則 218	
親要素 219	

arg-value	220
例 220	
内容に使用できる要素 220	
属性 222	
内容の規則 222	
親要素 222	
comment	223
備考 223	
例 223	
内容に使用できる要素 223	
属性 223	
内容の規則 223	
親要素 223	
.....	224
例 224	
内容に使用できる要素 224	
属性 224	
内容の規則 224	
親要素 224	
conditions	225
備考 225	
例 226	
内容に使用できる要素 226	
属性 226	
内容の規則 226	
親要素 226	
description	227
例 227	
内容に使用できる要素 227	
属性 227	
内容の規則 227	
親要素 227	
do-add-association	228
例 228	
内容に使用できる要素 228	
属性 228	
内容の規則 229	
親要素 229	
do-add-dest-attr-value	230
例 230	
内容に使用できる要素 230	
属性 230	
内容の規則 231	
親要素 231	
do-add-dest-object	232
例 232	
内容に使用できる要素 232	
属性 232	

内容の規則	233
親要素	233
do-add-src-attr-value	234
例	234
内容に使用できる要素	234
属性	234
内容の規則	234
親要素	235
do-add-src-object	236
例	236
内容に使用できる要素	236
属性	236
内容の規則	236
親要素	236
do-append-xml-element	237
例	237
内容に使用できる要素	237
属性	237
内容の宣言	238
親要素	238
do-append-xml-text	239
例	239
内容に使用できる要素	239
属性	239
内容の規則	239
親要素	239
do-break	240
例	240
内容に使用できる要素	240
属性	240
内容の宣言	240
親要素	240
do-clear-dest-attr-value	241
例	241
内容に使用できる要素	241
属性	241
内容の規則	242
親要素	242
do-clear-op-property	243
例	243
内容に使用できる要素	243
属性	243
内容の宣言	243
親要素	243
do-clear-src-attr-value	244
例	244
内容に使用できる要素	244
属性	244

内容の規則 244	
親要素 244	
do-clear-sso-credential	245
例 245	
内容に使用できる要素 245	
属性 245	
内容の規則 246	
親要素 246	
do-clone-op-attr	247
例 247	
内容に使用できる要素 247	
属性 247	
内容の規則 247	
親要素 247	
do-clone-xpath	248
例 248	
内容に使用できる要素 248	
属性 248	
内容の宣言 248	
親要素 249	
do-delete-dest-object	250
例 250	
内容に使用できる要素 250	
属性 250	
内容の規則 251	
親要素 251	
do-delete-src-object	252
例 252	
内容に使用できる要素 252	
属性 252	
内容の規則 252	
親要素 252	
do-find-matching-object	253
備考 253	
例 253	
内容に使用できる要素 253	
属性 254	
内容の規則 254	
親要素 254	
do-for-each	255
例 255	
内容に使用できる要素 255	
属性 255	
内容の規則 255	
親要素 255	
do-generate-event	256
備考 256	
例 257	

内容に使用できる要素 257	
属性 257	
内容の規則 257	
親要素 257	
do-if.....	258
例 258	
内容に使用できる要素 258	
属性 258	
内容の規則 258	
親要素 258	
do-implement-entitlement.....	259
例 259	
内容に使用できる要素 259	
属性 259	
内容の規則 259	
親要素 259	
do-move-dest-object.....	260
例 260	
内容に使用できる要素 260	
属性 260	
内容の規則 261	
親要素 261	
do-move-src-object.....	262
例 262	
内容に使用できる要素 262	
属性 262	
内容の規則 262	
親要素 262	
do-reformat-op-attr.....	263
例 263	
内容に使用できる要素 263	
属性 263	
内容の規則 263	
親要素 263	
do-remove-association.....	264
例 264	
内容に使用できる要素 264	
属性 264	
内容の規則 264	
親要素 265	
do-remove-dest-attr-value.....	266
例 266	
内容に使用できる要素 266	
属性 266	
内容の規則 267	
親要素 267	
do-remove-src-attr-value.....	268
例 268	

内容に使用できる要素 268	
属性 268	
内容の規則 268	
親要素 269	
do-rename-dest-object.	270
例 270	
内容に使用できる要素 270	
属性 270	
内容の規則 271	
親要素 271	
do-rename-op-attr.	272
例 272	
内容に使用できる要素 272	
属性 272	
内容の規則 272	
親要素 272	
do-rename-src-object.	273
例 273	
内容に使用できる要素 273	
属性 273	
内容の規則 273	
親要素 273	
do-send-email.	274
備考 274	
例 274	
内容に使用できる要素 275	
属性 275	
内容の規則 275	
親要素 275	
do-send-email-from-template.	276
備考 276	
例 276	
内容に使用できる要素 276	
属性 277	
内容の規則 277	
親要素 277	
do-set-default-attr-value.	278
例 278	
内容に使用できる要素 278	
属性 278	
内容の規則 278	
親要素 278	
do-set-dest-attr-value.	280
例 280	
内容に使用できる要素 280	
属性 280	
内容の規則 281	

親要素 281	
do-set-dest-password	282
例 282	
内容に使用できる要素 282	
属性 282	
内容の規則 283	
親要素 283	
do-set-local-variable	284
例 284	
内容に使用できる要素 284	
属性 284	
内容の規則 285	
親要素 285	
do-set-op-association	286
例 286	
内容に使用できる要素 286	
属性 286	
内容の規則 286	
親要素 286	
do-set-op-class-name	287
例 287	
内容に使用できる要素 287	
属性 287	
内容の規則 287	
親要素 287	
do-set-op-dest-dn	288
例 288	
内容に使用できる要素 288	
属性 288	
内容の規則 288	
親要素 288	
do-set-op-property	289
例 289	
内容に使用できる要素 289	
属性 289	
内容の規則 289	
親要素 289	
do-set-op-src-dn	290
例 290	
内容に使用できる要素 290	
属性 290	
内容の規則 290	
親要素 290	
do-set-op-template-dn	291
例 291	
内容に使用できる要素 291	
属性 291	
内容の規則 291	

親要素 291	
do-set-src-attr-value	292
例 292	
内容に使用できる要素 292	
属性 292	
内容の規則 292	
親要素 293	
do-set-src-password	294
例 294	
内容に使用できる要素 294	
属性 294	
内容の規則 294	
親要素 294	
do-set-ssso-credential	295
例 295	
内容に使用できる要素 295	
属性 295	
内容の規則 296	
親要素 296	
do-set-ssso-passphrase	297
例 297	
内容に使用できる要素 297	
属性 297	
内容の規則 297	
親要素 297	
do-set-xml-attr	299
例 299	
内容に使用できる要素 299	
属性 299	
内容の規則 299	
親要素 300	
do-start-workflow	301
301	
例 301	
内容に使用できる要素 301	
属性 302	
内容の規則 302	
親要素 302	
do-status	303
備考 303	
例 303	
内容に使用できる要素 303	
属性 303	
内容の規則 303	
親要素 304	
do-strip-op-attr	305
例 305	
内容に使用できる要素 305	

属性 305	
内容の規則 305	
親要素 305	
do-strip-xpath	306
例 306	
内容に使用できる要素 306	
属性 306	
内容の宣言 306	
親要素 306	
do-trace-message	307
例 307	
内容に使用できる要素 307	
属性 307	
内容の規則 307	
親要素 307	
do-veto	309
例 309	
内容に使用できる要素 309	
属性 309	
内容の規則 309	
親要素 309	
do-veto-if-op-attr-not-available	310
例 310	
内容に使用できる要素 310	
属性 310	
内容の規則 310	
親要素 310	
do-while	311
例 311	
内容に使用できる要素 311	
属性 311	
内容の規則 311	
親要素 311	
if-association	312
備考 312	
例 312	
内容に使用できる要素 312	
属性 313	
内容の規則 313	
親要素 313	
if-attr	314
備考 314	
例 314	
内容に使用できる要素 314	
属性 315	
内容の規則 315	
親要素 315	
if-class-name	316

備考 316	
例 316	
内容に使用できる要素 316	
属性 316	
内容の規則 317	
親要素 317	
if-dest-attr	318
備考 318	
例 318	
内容に使用できる要素 318	
属性 319	
内容の規則 319	
親要素 319	
if-dest-dn	320
備考 320	
例 320	
内容に使用できる要素 320	
属性 320	
内容の規則 321	
親要素 321	
if-entitlement	322
備考 322	
例 323	
内容に使用できる要素 323	
属性 323	
内容の規則 323	
親要素 324	
if-global-variable	325
備考 325	
例 325	
内容に使用できる要素 325	
属性 325	
内容の規則 326	
親要素 326	
if-local-variable	327
備考 327	
例 327	
内容に使用できる要素 327	
属性 328	
内容の規則 328	
親要素 328	
if-named-password	329
備考 329	
例 329	
内容に使用できる要素 329	
属性 329	
内容の宣言 329	

親要素 330	
if-op-attr	331
備考 331	
例 332	
内容に使用できる要素 332	
属性 332	
内容の規則 332	
親要素 333	
if-op-property	334
備考 334	
例 334	
内容に使用できる要素 334	
属性 334	
内容の規則 335	
親要素 335	
if-operation	336
備考 336	
例 336	
内容に使用できる要素 336	
属性 336	
内容の規則 337	
親要素 337	
if-password	338
備考 338	
例 338	
内容に使用できる要素 338	
属性 338	
内容の規則 339	
親要素 339	
if-src-attr	340
備考 340	
例 340	
内容に使用できる要素 340	
属性 341	
内容の規則 341	
親要素 341	
if-src-dn	342
備考 342	
例 342	
内容に使用できる要素 342	
属性 342	
内容の規則 343	
親要素 343	
if-xml-attr	344
備考 344	
例 344	
内容に使用できる要素 344	

属性	344
内容の規則	345
親要素	345
if-xpath	346
備考	346
例	346
内容に使用できる要素	346
属性	346
内容の規則	346
親要素	346
include	347
備考	347
例	347
内容に使用できる要素	347
属性	347
内容の宣言	347
親要素	347
または	348
例	348
内容に使用できる要素	348
属性	348
内容の規則	349
親要素	349
policy	350
備考	350
変数	350
例	353
内容に使用できる要素	354
属性	354
内容の規則	354
親要素	354
rule	355
例	355
内容に使用できる要素	355
属性	355
内容の規則	355
親要素	355
token-added-entitlement	356
例	356
内容に使用できる要素	356
属性	356
内容の宣言	356
親要素	356
token-association	358
例	358
内容に使用できる要素	358
属性	358

内容の宣言 358	
親要素 358	
token-attr	360
例 360	
内容に使用できる要素 360	
属性 360	
内容の宣言 360	
親要素 360	
token-base64-decode	362
例 362	
内容に使用できる要素 362	
属性 363	
内容の規則 363	
親要素 363	
token-base64-encode	365
例 365	
内容に使用できる要素 365	
属性 366	
内容の規則 366	
親要素 367	
token-char	368
例 368	
内容に使用できる要素 368	
属性 368	
内容の宣言 368	
親要素 368	
token-class-name	370
例 370	
内容に使用できる要素 370	
属性 370	
内容の宣言 370	
親要素 370	
token-convert-time	372
例 372	
内容に使用できる要素 372	
属性 373	
内容の規則 374	
親要素 374	
token-dest-attr	376
例 376	
内容に使用できる要素 376	
属性 376	
内容の規則 376	
親要素 376	
token-dest-dn	378
例 378	
内容に使用できる要素 378	
属性 378	

内容の宣言	378
親要素	379
token-dest-name	380
例	380
内容に使用できる要素	380
属性	380
内容の規則	380
親要素	380
token-document	382
例	382
内容に使用できる要素	382
属性	382
内容の規則	382
親要素	382
token-entitlement	384
例	384
内容に使用できる要素	384
属性	384
内容の宣言	384
親要素	384
token-escape-for-dest-dn	386
例	386
内容に使用できる要素	386
属性	387
内容の規則	387
親要素	388
token-escape-for-src-dn	389
例	389
内容に使用できる要素	389
属性	390
内容の規則	390
親要素	391
token-generate-password	392
例	392
内容に使用できる要素	392
属性	392
内容の宣言	392
親要素	392
token-global-variable	394
例	394
内容に使用できる要素	394
属性	394
内容の宣言	394
親要素	394
token-join	396
例	396
内容に使用できる要素	396
属性	397

内容の規則 398	
親要素 398	
token-local-variable	400
例 400	
内容に使用できる要素 400	
属性 400	
内容の宣言 400	
親要素 400	
token-lower-case	402
例 402	
内容に使用できる要素 402	
属性 403	
内容の規則 403	
親要素 404	
token-map.....	405
備考 405	
例 405	
内容に使用できる要素 405	
属性 407	
内容の規則 407	
親要素 407	
token-named-password	409
例 409	
内容に使用できる要素 409	
属性 409	
内容の宣言 409	
親要素 409	
token-op-attr	411
例 411	
内容に使用できる要素 411	
属性 411	
内容の宣言 411	
親要素 411	
token-op-property	413
例 413	
内容に使用できる要素 413	
属性 413	
内容の宣言 413	
親要素 413	
token-operation.....	415
例 415	
内容に使用できる要素 415	
属性 415	
内容の宣言 415	
親要素 415	
token-parse-dn	417
備考 417	
例 417	

内容に使用できる要素 417	
属性 419	
内容の規則 420	
親要素 420	
token-password	422
例 422	
内容に使用できる要素 422	
属性 422	
内容の宣言 422	
親要素 422	
token-query	424
備考 424	
例 424	
内容に使用できる要素 424	
属性 425	
内容の規則 425	
親要素 425	
token-removed-attr	427
例 427	
内容に使用できる要素 427	
属性 427	
内容の宣言 427	
親要素 427	
token-removed-entitlement	429
例 429	
内容に使用できる要素 429	
属性 429	
内容の宣言 429	
親要素 429	
token-replace-all	431
備考 431	
例 431	
内容に使用できる要素 431	
属性 433	
内容の規則 433	
親要素 433	
token-replace-first	435
備考 435	
例 435	
内容に使用できる要素 435	
属性 437	
内容の規則 437	
親要素 437	
token-resolve	439
例 439	
内容に使用できる要素 439	
属性 439	

内容の規則 439	
親要素 439	
token-split	441
例 441	
内容に使用できる要素 441	
属性 442	
内容の規則 443	
親要素 443	
token-src-attr	444
例 444	
内容に使用できる要素 444	
属性 444	
内容の規則 444	
親要素 444	
token-src-dn	446
例 446	
内容に使用できる要素 446	
属性 446	
内容の宣言 446	
親要素 447	
token-src-name	448
例 448	
内容に使用できる要素 448	
属性 448	
内容の宣言 448	
親要素 448	
token-substring	450
例 450	
内容に使用できる要素 450	
属性 451	
内容の規則 452	
親要素 452	
token-text	454
例 454	
内容に使用できる要素 454	
属性 454	
内容の規則 454	
親要素 454	
token-time	456
例 456	
内容に使用できる要素 456	
属性 456	
内容の宣言 456	
親要素 456	
token-unique-name	458
備考 458	
例 458	
内容に使用できる要素 459	

属性	459
内容の規則	460
親要素	460
token-unmatched-src-dn	462
例	462
内容に使用できる要素	462
属性	462
内容の宣言	462
親要素	462
token-upper-case	464
例	464
内容に使用できる要素	464
属性	465
内容の規則	465
親要素	466
token-xml-parse	467
例	467
内容に使用できる要素	467
属性	468
内容の規則	468
親要素	469
token-xml-serialize	470
例	470
内容に使用できる要素	470
属性	471
内容の規則	471
親要素	472
token-xpath	473
例	473
内容に使用できる要素	473
属性	473
内容の宣言	473
親要素	473
6 DirXML エンタイトルメント DTD	475
6.1 DirXML エンタイトルメント DTD の要素	476
description	478
例	478
内容に使用できる要素	478
内容の規則	478
親要素	478
display-name	479
例	479
内容に使用できる要素	479
属性	479
内容の規則	479
親要素	479

dn	480
例 480	
内容に使用できる要素 480	
属性 480	
内容の規則 480	
親要素 480	
ent-value	481
例 481	
内容に使用できる要素 481	
属性 481	
内容の規則 481	
親要素 481	
entitlement	482
例 482	
内容に使用できる要素 482	
属性 483	
内容の規則 483	
親要素 483	
entitlement-impl	484
例 484	
内容に使用できる要素 484	
属性 484	
内容の規則 484	
親要素 485	
id	486
例 486	
内容に使用できる要素 486	
属性 486	
内容の規則 486	
親要素 486	
item	487
例 487	
内容に使用できる要素 487	
属性 487	
内容の規則 487	
親要素 487	
item-description	488
例 488	
内容に使用できる要素 488	
属性 488	
内容の規則 488	
親要素 488	
item-display-name	489
例 489	
内容に使用できる要素 489	
属性 489	
内容の規則 489	
親要素 489	

item-value	490
例 490	
内容に使用できる要素 490	
属性 490	
内容の規則 490	
親要素 490	
項目	491
例 491	
内容に使用できる要素 491	
属性 491	
内容の規則 491	
親要素 491	
msg	492
例 492	
内容に使用できる要素 492	
属性 492	
内容の規則 492	
親要素 492	
param.....	493
例 493	
内容に使用できる要素 493	
属性 493	
内容の規則 493	
親要素 493	
query-app.....	494
例 494	
内容に使用できる要素 494	
属性 494	
内容の規則 494	
親要素 494	
query-xml.....	495
例 495	
内容に使用できる要素 495	
属性 495	
内容の規則 495	
親要素 495	
ref.....	496
例 496	
内容に使用できる要素 496	
属性 496	
内容の規則 496	
親要素 496	
result	497
例 497	
内容に使用できる要素 497	
属性 497	
内容の規則 497	
親要素 497	

result-set.....	498
例 498	
内容に使用できる要素 498	
属性 498	
内容の規則 498	
親要素 498	
src.....	499
例 499	
内容に使用できる要素 499	
属性 499	
内容の規則 499	
親要素 499	
state.....	500
例 500	
内容に使用できる要素 500	
属性 500	
内容の規則 500	
親要素 500	
status.....	501
例 501	
内容に使用できる要素 501	
属性 501	
内容の規則 501	
親要素 501	
timestamp.....	502
例 502	
内容に使用できる要素 502	
属性 502	
内容の規則 502	
親要素 502	
token-association.....	503
例 503	
内容に使用できる要素 503	
属性 503	
内容の規則 503	
親要素 503	
token-attr.....	504
例 504	
内容に使用できる要素 504	
属性 504	
内容の規則 504	
親要素 504	
token-src-dn.....	505
例 505	
内容に使用できる要素 505	
属性 505	
内容の宣言 505	
親要素 505	

value	506
例 506	
内容に使用できる要素 506	
属性 506	
内容の規則 506	
親要素 506	
values	507
例 507	
内容に使用できる要素 507	
属性 507	
内容の規則 507	
親要素 507	

このガイドについて

このガイドは、Identity Manager で使用するドキュメントタイプ定義 (DTD) のリファレンスです。このガイドには、Identity Manager で使用する各要素の定義が含まれます。Identity Manager のコンポーネントごとに個別の DTD があります。

- ◆ 41 ページの「フィルタ DTD」
- ◆ 49 ページの「NDS DTD」
- ◆ 177 ページの「マップ DTD」
- ◆ 183 ページの「DirXML スクリプト DTD」
- ◆ 475 ページの「DirXML エンタイトルメント DTD」

対象読者

このガイドは、Identity Manager のコンサルタントがリファレンスとして使用することを目的としています。

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

『Identity Manager DTD リファレンス』の最新バージョンは、Identity Manager の Web サイト (<http://www.novell.com/documentation/idm35/>) で確認できます。

追加のマニュアル

Identity Manager のマニュアルについては、Identity Manager のマニュアルの Web サイト (<http://www.novell.com/documentation/idm35/>) 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。

パス名の表記に円記号 (l) を使用するプラットフォームとスラッシュ (/) を使用するプラットフォームがありますが、このマニュアルでは円記号を使用します。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。

Identity Manager DTD リファレンスの概要

1

これは、Identity Manager で使用するドキュメントタイプ定義 (DTD) のリファレンスセクションです。

- ◆ 41 ページの第 2 章「フィルタ DTD」
- ◆ 49 ページの第 3 章「NDS DTD」
- ◆ 177 ページの第 4 章「マップ DTD」
- ◆ 183 ページの第 5 章「DirXML スクリプト DTD」
- ◆ 475 ページの第 6 章「DirXML エンタイトルメント DTD」

次に、Identity Manager 3.5.1 の DTD に新たに加えられた項目のリストを示します。

- ◆ `:InitiatorOverrideDN` および `:CorrelationID` が指定されたパラメータの `do-start-workflow` への受け渡しのサポート。
- ◆ `do-generate-event` の `id` 属性の変数拡張のサポート。
- ◆ ノードセットタイプの変数の `if-local-variable` の動作を、複数値の他の条件の動作との整合性を向上するように変更。

フィルタ DTD

Identity Manager フィルタは、同期するオブジェクトクラスの制御と、そのオブジェクトクラスと同期する属性の制御を行うためのものです。Identity Manager 内のオブジェクトクラスおよび属性のその他の動作もフィルタによって制御します。

Identity Manager フィルタは、トップレベルの <filter> 要素と一連の <filter-class> 要素で構成されており、<filter-class> 要素のそれぞれには一連の <filter-attr> 要素が含まれています。特定のドライバのフィルタは、DirXML-Driver オブジェクトの DirXML-DriverFilter 属性に格納されます。

フィルタ DTD のすべての要素のリストについては、「[41 ページの「フィルタ DTD の要素」](#)」を参照してください。

2.1 フィルタ DTD の要素

要素	説明
フィルタ	Identity Manager ドライバのフィルタ
filter-attr	特定のオブジェクトクラスの属性の動作
filter-class	オブジェクトクラスの動作

フィルタ

Identity Manager ドライバの特定のインスタンスで使用されるオブジェクトクラスを記述する `<filter-class>` 要素のセットで構成されます。

例

```
<filter>          <filter-class class-name="User"
subscriber="sync"          publisher="sync">
<filter-attr attr-name="CN"          subscriber="sync"
publisher="ignore"          merge-authority="none"/>
<filter-attr attr-name="Surname"
subscriber="sync"/>          <filter-attr attr-name="Given name"
subscriber="sync"/>          <filter-attr attr-name="Internet EMail
Address"          publisher="sync"
publisher-optimize-modify="false"/>          <filter-attr attr-
name="Login Disabled"          subscriber="notify"/>
</filter-class>          <filter-class class-name="Group"
subscriber="sync"          publisher="sync">
publisher-create-homedir="false">          <filter-attr attr-
name="CN"          subscriber="sync"
merge-authority="none"/>          <filter-attr attr-name="Member"
subscriber="sync"          publisher="sync"
merge-authority="publisher"          publisher-optimize-
modify="false"/>          </filter-class>          </filter>
```

許可されるコンテンツ

要素	説明
----	----

`filter-class`

属性

なし

内容の規則

(`filter-class*`)

親要素

なし

filter-attr

Identity Manager ドライバの特定のインスタンスで使用される、格納している `<filter-class>` の属性を記述します。attr-name 属性によって、eDirectory™ の属性の名前が指定されます。次のテーブルに従い、publisher 属性と subscriber 属性によって、それぞれのチャンネルで対象の属性が同期されるかどうかを制御されます。オブジェクトクラスのチャンネル設定が全体として無視される場合は、個々の属性の設定が無視されます。

備考

次のテーブルに従い、publisher 属性と subscriber 属性によって、それぞれのチャンネルで対象の属性が同期されるかどうかを制御されます。

値	説明
ignore	対象の属性に対する変更は、レポートも自動的な同期化もされません。
通知	対象の属性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。
同期	対象の属性に対する変更は、レポートされ、自動的に同期化されます。
リセット	対象の属性に対する変更はレポートされ、この変更によって、属性が他方のチャンネルの値に自動的にリセットされます。発行者と購読者の両方を同一の属性にリセットすることは不正です。

次のテーブルに従い、merge-authority 属性によって、マージ操作の際の属性の動作が制御されます。

値	動作	有効
デフォルト	<p>1. 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。</p> <p>2. 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。ソースに値が複数ある場合にターゲットでは1つの値しか受け取ることができない場合、複数の値のうちの1つがターゲット側で使用されます。このとき、どの値が使用されるかは定義されていません。</p> <p>3. 属性が両方のチャンネルで同期されていて、両方の側で複数の値を受け取ることができる場合は、最終的に両方の側で、いずれかの側の値に結合します。</p> <p>4. 属性が両方のチャンネルで同期されていて、両方の側で1つの値しか受け取ることができない場合は、eDirectory に値があるかぎり、アプリケーションは最終的に eDirectory の値になります。この場合、eDirectory は最終的にアプリケーションの値になります (値がある場合)。</p> <p>5. 属性が両方のチャンネルで同期され、一方の側だけが複数の値を受け取ることができる場合は、単一値の側の値が複数値の側に存在していれば追加されます。★「if it is already there」は、「if it is not already there」の間違いではないかと思われ ★単一値の側に値がない場合は、複数の値のうちのいずれか (未定義) が単一値の側に追加されます。</p>	常時
edir	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です。	購読者チャンネルで同期または通知されている場合。
app	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です。	発行者チャンネルで同期または通知されている場合。
none (なし)	同期に関係なく、マージされません。	常時

publisher-optimize-modify 属性によって、識別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に対する変更を調べるかどうかを制御されます。

例

<フィルタ>を参照してください。

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
attr-name	CDATA 属性の名前です。	#REQUIRED
merge-authority	default edir app none 対象の属性がマージされる方法を制御するフラグ。	デフォルト
publisher	ignore notify sync reset 発行者チャンネルの同期化を制御するフラグ。	ignore
publisher-optimize-modify	true false 購読者チャンネルでの変更の最適化を制御するフラグ。	true
subscriber	 購読者チャンネルの同期化を制御するフラグ。	ignore

内容の宣言

なし

親要素

要素	説明
filter-class	

filter-class

Identity Manager ドライバの特定のインスタンスで使用されるオブジェクトクラスを記述します。class-name 属性によって、eDirectory で有効な (つまり、構造またはベース) クラスの名前が指定されます。この属性は、その特定のベースクラスのあるオブジェクトだけに適用されます。

備考

発行者および購読者の属性によって、対象のクラスがそれぞれのチャンネルで同期されるかどうかは制御されます。

値	説明
ignore	対象のクラス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。
同期	対象のクラス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。

publisher-track-template-member 属性によって、発行者チャンネルでテンプレートからオブジェクトが作成されるときに、テンプレートのメンバーの属性が保持されるかどうかは制御されます。publisher-create-homedir 属性によって、ホームディレクトリの属性を使用してユーザが作成されるときに、NetWare[®] ホームディレクトリが自動的に作成されるかどうかは制御されます。

例

<フィルタ>を参照してください。

内容に使用できる要素

要素	説明
<code>filter-attr</code>	

属性

属性	有効値	デフォルト値
class-name	CDATA オブジェクトクラスの名前。	#REQUIRED
	ignore sync	ignore

属性	有効値	デフォルト値
publisher-create-homedir	ホームディレクトリの自動作成を制御するフラグ。	true
publisher-track-template-member	オブジェクトの追跡を制御するフラグ。	false
		ignore

内容の規則

filter-attr

親要素

要素	説明
フィルタ	

NDS™ ドキュメントタイプ定義ファイル (nds.dtd) では、Identity Manager エンジンで処理できる 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。

nds.dtd ファイルでは次の項目を定義します。

- ◆ それぞれとともに含める必要があるデータおよびエントリに対して実行できる入力と出力のコマンドおよびイベント (追加、削除、変更、名前変更など)。
- ◆ ドライバシム、発行者シム、および購読者シムに対するドライバ初期化操作 (認証情報、ドライバフィルタ、環境設定オプション、状態など)、およびこれらの操作に必要なデータ。
- ◆ クラスと属性の定義を規定するためのスキーマ操作。
- ◆ スキーマのマッピング、一致、作成、および配置に関するルール。

DTD ファイルを読むときには次のことに留意してください。

マーカー	意味
?	これらのいずれも含めないか、1 つのみを含めることができます。
+	これらのうち 1 つ以上を含める必要があります。
*	これらのいずれも含めないか、1 つ以上を含めることができます。
CDATA	文字データ。
解析済み文字データ	解析済み文字データ。
<!	要素、エンティティ、または属性の定義の始まり。
>	要素、エンティティ、または属性の定義の終わり。

NDS DTD 内のすべての要素のリストについては、[49 ページの「NDS DTD の要素」](#)を参照してください。

3.1 NDS DTD の要素

要素	説明
追加	追加イベントが発生したときにオブジェクトを追加します。
関連付けの追加	関連付けを追加します。
add-attr	属性を追加します。
add-value	値を追加します。
allow-attr	フィルタで属性を許可します。

要素	説明
allow-class	フィルタでクラスを許可します。
app-name	アプリケーションネームスペース内の名前。
関連付け	アプリケーションオブジェクトの固有キー。
attr	属性の現在の状態。
attr-def	スキーマ属性の定義。
attr-name	属性名をマップします。
attr-name-map	スキーママッピングポリシーのトップレベル要素。
authentication-info	アプリケーションに対する接続と認証に関する情報。
オブジェクトパスワードの確認	パスワードを eDirectory オブジェクトと照らし合わせてチェックします。
チェックパスワード	パスワードを eDirectory ドライバオブジェクトと照らし合わせてチェックします。
class-def	スキーマクラスの定義。
class-name	クラス名をマップします。
component	構造化属性のコンポーネント。
config-object	追加の環境設定データ用に使用する eDirectory オブジェクト。
contact	元の製品の連絡窓口。
copy-attr	属性トークンをコピーします。 Identity Manager 2.0 から廃止されました。
copy-name	名前トークンをコピーします。
copy-path	パストークンをコピーします。 Identity Manager 2.0 から廃止されました。
copy-path-suffix	パストークンをコピーします。
create-rule	オブジェクト作成ルール。
create-rules	オブジェクト作成ルールのトップレベル要素。
削除	削除イベントが発生したときにオブジェクトを削除します。
driver-config	ドライバ固有の Driver Shim 環境設定オプション。
driver-filter	発行と購読のクラスおよび属性のイベントフィルタ。
driver-options	
driver-state	ドライバ固有の状態情報。
名前付きパスワードの取得	ドライバの名前付きパスワードを取得します。

要素	説明
パラメータの開始	DriverShim、SubscriptionShim、またはPublicationShimの初期化パラメータ。
tinput	イベントまたはコマンドを入力します。
インスタンス	オブジェクトのインスタンスの現在の状態。
match-attr	属性と一致します。
match-class	クラス名と一致します。
match-path	パスと一致します。
matching-rule	オブジェクト一致ルール。
matching-rules	オブジェクト一致ルールのトップレベル要素。
変更	変更イベントが発生したときにオブジェクトを変更します。
関連付けの変更	関連付けコマンドを変更します。
modify-attr	属性を変更します。
パスワード変更	パスワードの変更イベントが発生したときにオブジェクトパスワードを変更します。
移動	移動イベントが発生したときにオブジェクトを移動します。
nds	すべての Identity Manager 通信およびドライバ通信のトップレベル要素。
nds-name	eDirectory ネームスペース内の名前。
new-name	名前が変更されたオブジェクトの新しい名前。
old-password	古い認証パスワード。
operation-data	この操作によってさらなるカスタムデータが追加されます。
output	イベントまたはコマンドの結果。
parent	オブジェクトの親コンテナ。
パスワード	認証パスワード。
placement	オブジェクト配置指定子。
placement-rule	オブジェクト配置ルール。
placement-rules	オブジェクト配置ルールのトップレベル要素。
publisher-options	ドライバ固有の PublicationShim 環境設定オプション。
publisher-state	ドライバの PublicationShim 状態情報。
クエリー	query コマンド。
query-ex	結果カウント制限付きの query コマンド。

要素	説明
クエリスキーマ	query schema コマンド。
query-token	query-ex コマンドの不透明ハンドル。
rread-attr	指定されたオブジェクト属性値を返します。
read-parent	オブジェクトの親コンテナを返します。
remove-all-values	すべての属性値を削除します。
関連付けの削除	関連付けを削除します。
remove-value	指定された属性値を削除します。
名前を変更する	名前変更イベントが発生したときにオブジェクトの名前を変更します。
required-attr	必須の属性。
スキーマ定義	スキーマの定義。
search-attr	クエリ検索属性値フィルタ。
search-class	クエリ検索クラスフィルタ。
server	認証サーバ。
source	ドキュメントのソースまたは作成者。
status	コマンドまたはイベントの処理のステータス。
subscriber-options	ドライバ固有の SubscriptionShim 環境設定オプション。
subscriber-state	ドライバの SubscriptionShim 状態情報。
同期	再同期イベントまたは移行イベント。
template	テンプレートを指定します。
user	認証ユーザ名
値	属性値。

追加

アプリケーションでオブジェクトが追加されたときに、PublicationShim から Identity Manager へのイベント通知として使用されます。これが通知として使用されるときは、<関連付け>が必要です。これは、アプリケーションでオブジェクトを追加するための、Identity Manager から SubscriptionShim へのコマンドとしても使用されます。

備考

<add> には、追加されたオブジェクトの属性ごとに 1 つの <add-attr> が含まれています。

<add> には、追加されたオブジェクトの <パスワード> が含まれている場合があります。

<add> への応答は、<追加> が正常に処理されたかどうかを示す <status> である必要があります。コマンドとして使用された場合、<add> は、新たに追加されたオブジェクトの固有キーが含まれた <関連付けの追加> も返す必要があります。<関連付けの追加> の dest-dn 属性と dest-entry-id 属性は、<add> の src-dn と src-entry-id に設定する必要があります。

例

```
<add class-name="User" src-dn="\Sam"          <association>1012</association>
  <add-attr attr-name="cn">
<value>Sam</value>          </add-attr>          <add-attr attr-
name="Surname">          <value>Jones</value>          </add-attr>
<add-attr attr-name="Given Name">          <value>Sam</value>
</add-attr>          <add-attr attr-name="Telephone Number">
<value>555-1212</value>          </add-attr></add>
```

内容に使用できる要素

要素	説明
関連付け	
add-attr	属性を追加します。
パスワード	
operation-data	

属性

属性	有効値	デフォルト値
class-name	CDATA オブジェクトのベースクラスの名前。このクラス名は、スキーママッピングポリシーによってアプリケーションネームスペースと eDirectory ネームスペースの間でマップされているため、Identity Manager では eDirectory™ ネームスペース内の名前が認識され、ドライバではアプリケーションネームスペース内の名前が認識されます。	#REQUIRED
dest-dn	CDATA 受信者のネームスペース内のターゲットオブジェクトの識別名。 イベント通知については空のままにしておく必要があります。コマンドの配置ポリシーによって入力されます。	#IMPLIED
dest-entry-id	CDATA 受信者のネームスペース内のターゲットオブジェクトのエントリ ID。 予約済み。ドライバでは無視される必要があります。	#IMPLIED
event-id	CDATA イベントまたはコマンドの結果にタグ付けするための識別子。 結果の <status> 要素および <関連付けの追加> 要素の event-id 属性にコピーされる必要があります。	#IMPLIED
qualified-src-dn	CDATA src-dn の完全修飾バージョン。eDirectory 内のオブジェクトを説明するためにのみ使用されます。	#IMPLIED

属性	有効値	デフォルト値
src-dn	CDATA 送信者のネームスペース内でイベントを生成したソースオブジェクトの識別名。 コマンドの結果である < 関連付けの追加 > の dest-dn 属性にコピーされる必要があります。	#IMPLIED
src-entry-id	CDATA 送信者のネームスペース内でイベントを生成したソースオブジェクトのエントリ ID。 コマンドの結果である < 関連付けの追加 > の dest-entry-id 属性にコピーされる必要があります。	#IMPLIED
template-dn	CDATA オブジェクトを作成するためのベースとして使用される、受信者のネームスペース内のテンプレートの識別名。 コマンドの作成ポリシーによって入力されます。アプリケーションにとって適切である場合にのみ、ドライバはこれを実装する必要があります。	#IMPLIED
タイムスタンプ	CDATA 予約済み。	#IMPLIED

内容の規則

(association ? , add-attr * , password ? , operation-data ?)

親要素

要素	説明
input	

関連付けの追加

<追加> コマンドの結果として追加されたオブジェクトの固有キーを返すために使用されます。

例

```
<add-association dest-dn="\Users\Samuel"dest-entry-id="33974">{BC3E7155-CDF9-d311-9846-0008C76B16C2}</add-association>
```

内容に使用できる要素

#PCDATA

要素	説明
operation-data	操作によってさらなるカスタムデータが追加されます。

属性

属性	有効値	デフォルト値
dest-dn	CDATA	#REQUIRED
	<追加>のsrc-dnに設定される必要があります。	
	CDATA	#IMPLIED
	受信者のネームスペース内のターゲットオブジェクトのエントリ ID。	
	<追加>のsrc-entry-idに設定される必要があります。	
	CDATA	#IMPLIED
	<追加>のevent-idに設定される必要があります。	

内容の規則

(#PCDATA | operation-data)*

親要素

要素	説明
input	
output	

add-attr

<追加> 操作またはイベントの属性値を指定するために使用されます。各 <add-attr> には 1 つ以上の <値> が含まれている必要があります。

例

<追加> を参照してください。

内容に使用できる要素

要素	説明
値	

属性

属性	有効値	デフォルト値
attr-name	CDATA 属性の名前。 この名前は、スキーママッピングポリシーによってアプリケーション名前スペースと eDirectory 名前スペースの間でマップされているため、Identity Manager では eDirectory 名前スペース内の名前が認識され、ドライバではアプリケーション名前スペース内の名前が認識されます。マッピングルールでは、格納元のコマンドまたはイベントのクラス名属性を使用して、属性名をマップするために使用するクラスを決定します。	#REQUIRED
enforce-password-policy		#IMPLIED 予約済み。

内容の規則

(値+)

親要素

要素	説明
追加	

add-value

格納元の `<modify-attr>` で指定された属性に追加される値を指定するために使用されます。ドライバは、既存の値が指定された `<add-value>` を無視して、格納元である `<変更>` の残り部分の処理を続行する必要があります。

例

`<変更>` を参照してください。

内容に使用できる要素

要素	説明
<code>値</code>	

属性

なし

親要素

要素	説明
<code>modify-attr</code>	

allow-attr

格納元の `<allow-class>` で指定されたクラスのイベントフィルタで許可される属性を指定するために使用されます。

例

`<パラメータの開始>` を参照してください。

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
attr-name	CDATA	#REQUIRED
	この名前は、スキーママッピングポリシーによってアプリケーション名前スペースと eDirectory 名前スペースの間でマップされているため、Identity Manager では eDirectory 名前スペース内の名前が認識され、ドライバではアプリケーション名前スペース内の名前が認識されます。	
is-sensitive		false
	true の場合は、 <code><allow-attr></code> 要素が参照する属性値に、トレース情報で非表示にする必要のある機密データが含まれていることを指定します。	

内容の宣言

なし

親要素

要素	説明
<code>allow-class</code>	フィルタでクラスを許可します。

allow-class

格納元の <driver-filter> で指定されたイベントフィルタで許可されるクラスを指定するために使用されます。

例

<パラメータの開始> を参照してください。

内容に使用できる要素

要素	説明
<code>allow-attr</code>	フィルタで属性を許可します。

属性

属性	有効値	デフォルト値
<code>class-name</code>	CDATA オブジェクトのベースクラスの名前。 このクラス名は、スキーママッピングポリシーによってアプリケーション名前空間と eDirectory 名前空間の間でマップされているため、Identity Manager では eDirectory 名前空間内の名前が認識され、ドライバではアプリケーション名前空間内の名前が認識されます。	#REQUIRED

内容の規則

(allow-attr) *

親要素

要素	説明
<code>driver-filter</code>	

app-name

アプリケーションネームスペース内のクラス名または属性名を指定するために使用されます。

例

<attr-name-map> を参照してください。

内容に使用できる要素

#PCDATA

属性

なし

内容の規則

(#PCDATA)

親要素

要素	説明
----	----

attr-name	
-----------	--

class-name	
------------	--

関連付け

PublicationShim から Identity Manager へのイベント通知のソースであるアプリケーションオブジェクトの固有キー、Identity Manager から SubscriptionShim に送信されるコマンドのターゲット、または SubscriptionShim に送信される <クエリー> のベースオブジェクトを指定するために使用されます。

例

```
<associationstate="associated">{B43E7155-CDF9-d311-9846-0008C76B16C2}</association>
```

内容に使用できる要素

#PCDATA

属性

属性	有効値	デフォルト値
state	not-associated associated disabled migrate pending manual	#IMPLIED
	予約済み：ドライバでは無視される必要があります。	

内容の規則

#PCDATA

親要素

要素	説明
追加	
オブジェクトパスワードの確認	
削除	
インスタンス	
変更	
関連付けの変更	
パスワード変更	
移動	
parent	

要素	説明
クエリー	
query-ex	結果カウント制限付きの query コマンド。
名前を変更する	
同期	

attr

格納元の <インスタンス> で指定されたオブジェクトの属性値を指定するために使用されます。各 <attr> には 1 つ以上の <値> が含まれている必要があります。

例

<インスタンス> を参照してください。

内容に使用できる要素

要素	説明
値	

属性

属性	有効値	デフォルト値
attr-name	CDATA	#REQUIRED

内容の規則

(値*)

親要素

要素	説明
インスタンス	

attr-def

格納元の `<class-def>` で指定されたクラスのスキーマ属性を指定するために使用されます。

例

`<スキーマ定義>` を参照してください。

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
asn1id	CDATA 属性の ASN.1 オブジェクト ID。	#IMPLIED
attr-name	CDATA	#REQUIRED
case-sensitive	属性で大文字と小文字が区別されるかどうかを指定します。	false
multi-valued	属性が複数の値を持つことができるかどうかを指定します。	true
naming	属性を格納元クラスのオブジェクトの RDN の一部として使用できるかどうかを指定します。	false
読み込み専用	属性が読み取り専用かどうかを指定します。	false
required	属性が格納元クラスのオブジェクトによって要求されているかどうかを指定します。	false
type	string teleNumber int state counter dn interval octet time structured 属性のデータタイプ。	文字列

内容の宣言

なし

親要素

要素	説明
<code>class-def</code>	

attr-name

eDirectory ネームスペース内の属性名とアプリケーションネームスペース内の属性名の間でマッピングを指定するために使用されます。

例

<attr-name-map> を参照してください。

内容に使用できる要素

要素	説明
nds-name	eDirectory ネームスペース内の名前。指定された名前は、指定されたクラスで一意である必要があります。
app-name	アプリケーションネームスペース内の名前。

属性

属性	有効値	デフォルト値
class-name	CDATA	#IMPLIED

eDirectory ネームスペース内でこの属性名マッピングの対象となるクラスの名前。これが存在しないか空白の場合、マッピングは汎用と見なされ、クラス固有のマッピングがないすべてのクラスに適用されます。

内容の規則

(nds-name , app-name)

親要素

要素	説明
attr-name-map	

attr-name-map

スキーママッピングポリシーのトップレベル(ドキュメント)要素。スキーママッピングポリシーは、DirXML-Driver オブジェクトの DirXML-MappingRule 属性が指している DirXML-Rule オブジェクトの DirXML-XmlData 属性に保存されています。

備考

<attr-name-map> には <attr-name> 要素と <class-name> 要素が含まれており、これらの要素によって、eDirectory ネームスペース内のクラス名および属性名と、アプリケーション ネームスペース内のクラス名および属性名との 1 対 1 マッピングが指定されます。出力変換の適用前に、Identity Manager からドライバに XML が送信されるか返されるたびに、eDirectory ネームスペースからアプリケーションネームスペースへのマッピングのためにスキーママッピングポリシーが適用されます。

入力変換ポリシーの適用後に、ドライバから Identity Manager に XML が送信されるか返されるたびに、アプリケーションネームスペースから eDirectory ネームスペースへのマッピングのためにスキーママッピングポリシーが適用されます。

スキーママッピングポリシーは、ドキュメント内のすべての要素の <class-name> 属性と <attr-name> 属性をマップしようとします。属性名をマップするための <class-name> を見つけるには、<class-name> 属性を持つ最も近い先祖要素を参照します。

例

```
<attr-name-map>          <!-- map eDirectory class User application class
inetOrgPerson-->        <class-name>                <nds-name>User</nds-
name>                    <app-name>inetOrgPerson</app-name>          </class-
name>                    <!-- map NDS attribute Given Name to application attribute
givenNamefor class User -->    <attr-name class-name="User">
<nds-name>Given Name</nds-name>                <app-name>givenName</
app-name>                </attr-name>                <!-- map NDS attribute Surname to
application attribute sn for allclasses -->        <!-- that don't have
a class-specific mapping -->        <attr-name>                <nds-
name>Surname</nds-name>                <app-name>sn</app-name>        </
attr-name></attr-name-map>
```

内容に使用できる要素

要素	説明
attr-name	
class-name	

属性

なし

内容の規則

(attr-name | class-name) *

親要素

なし

authentication-info

ドライバがアプリケーションサーバに対して接続および認証するために必要なパラメータを指定するために使用されます。

例

< [パラメータの開始](#) > を参照してください。

内容に使用できる要素

要素	説明
server	
user	認証ユーザ名。
パスワード	

属性

なし

内容の規則

(server ? , user ? , password ?)

親要素

要素	説明
パラメータの開始	

オブジェクトパスワードの確認

パスワードを eDirectory オブジェクトと照らし合わせて検証するために使用されます。
<status> の結果が返され、成功または失敗が示されます。

備考

<関連付け> 要素、dest-dn 属性、または dest-entry-id 属性を使用して、パスワードを
チェックする際に照らし合わせる eDirectory オブジェクトを指定します。

eDirectory オブジェクトの Login Disabled 属性が true に設定されている場合は、パスワード
が正しい場合でも <status> はエラーを示します。

例

```
<check-object-passworddest-dn="container\object">abdc1234</check-  
object-password>
```

内容に使用できる要素

要素	説明
関連付け	
パスワード	
operation-data	操作の追加カスタムデータ。

属性

属性	有効値	デフォルト値
dest-dn	CDATA	#IMPLIED
	CDATA	#IMPLIED
event-id CDATA。イベントまたはコマンドの結果にタグ付けするための識別子。	CDATA	#IMPLIED

内容の規則

(association ? , password , operation-data ?)

親要素

要素	説明
input	

チェックパスワード

パスワードを eDirectory ドライバオブジェクトと照らし合わせて検証するために使用されます。**status**

例

```
<check-password>abdc1234</check-password>
```

内容に使用できる要素

#PCDATA

要素	説明
operation-data	

属性

なし

内容の規則

親要素

要素	説明
input	

class-def

格納元の <スキーマ定義> のスキーマクラスを指定するために使用されます。

例

<スキーマ定義> を参照してください。

内容に使用できる要素

要素	説明
attr-def	

属性

属性	有効値	デフォルト値
asn1id	CDATA クラスの ASN.1 オブジェクト ID。	#IMPLIED
class-name	CDATA スキーマクラスの名前。	#REQUIRED
コンテナ	このクラスのオブジェクトが他のオブジェクトのコンテナになれるかどうかを指定します。	false

内容の規則

(attr-def) *

親要素

要素	説明
スキーマ定義	

class-name

eDirectory ネームスペース内のクラス名とアプリケーションネームスペース内のクラス名の間でマッピングを指定するために使用されます。

例

<attr-name-map> を参照してください。

内容に使用できる要素

要素	説明
nds-name	指定された名前は、この <class-name> に対して一意である必要があります。
app-name	

属性

なし

内容の規則

親要素

要素	説明
attr-name-map	

component

値のデータタイプが structured の場合は、格納元の < 値 > の個別フィールドを指定するために使用されます。

例

< 値 > を参照してください。

内容に使用できる要素

#PCDATA

属性

属性	有効値	デフォルト値
association-ref	CDATA このコンポーネントが参照しているオブジェクトの関連付け値 (アプリケーションオブジェクトの固有キー)。コンポーネントがドライバからの通知イベントの一部である場合は、これは他のオブジェクトを参照するすべてのコンポーネントが必要です。コンポーネントが Identity Manager からのコマンドの一部である場合は、これは他のオブジェクトを参照するすべてのコンポーネントで存在しています。ただし、参照先のオブジェクトは eDirectory 内で確立された関連付けを保持していることが条件です。	#IMPLIED
名前	CDATA コンポーネントの名前。これは個別の属性構文に固有です。 < 値 > を参照してください。	#REQUIRED

内容の規則

(#PCDATA)

親要素

要素	説明
値	

config-object

追加の環境設定情報が取得されるオブジェクトと属性を指定するために使用されます。

備考

ドライバの起動時に、含まれている<クエリー>が処理されて、結果の<インスタンス>要素によって、DriverShim.init()、SubscriptionShim.init()、および PublicationShim.init() メソッドに渡された<パラメータの開始>内の<config-object>が置換されます。

例

<パラメータの開始>を参照してください。

内容に使用できる要素

要素	説明
クエリー	

属性

属性	有効値	デフォルト値
display-name	CDATA ConsoleOne®によって生成される インタフェースで表示する名前。	#IMPLIED

内容の規則

(クエリー)

親要素

なし

contact

格納元ドキュメントの作成者の連絡窓口を指定するために使用されます。

例

<n^{ds}> を参照してください。

内容に使用できる要素

#PCDATA

属性

なし

内容の規則

#PCDATA

親要素

要素	説明
----	----

source	
--------	--

copy-attr

Identity Manager 2.0 から廃止されました。代わりに DirXMLScript `<token-op-attr>` を使用してください。

備考

`<copy-attr>` は、格納元の `<placement>` によって生成される識別名内の文字列置換を指定するためのトークンとして使用されます。

置換文字列は、処理対象の `<追加>` イベントから、`attr-name` で指定された属性の最初の値をコピーすることによって生成されます。この属性が存在しない場合は、格納元の `<placement-rule>` はスキップされます。構造化された属性タイプはサポートされていません。

例

`<placement-rules>` を参照してください。

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
<code>attr-name</code>	CDATA	#REQUIRED

内容の宣言

なし

親要素

要素	説明
<code>placement</code>	

copy-name

token-src-dn

備考

<copy-name> は、格納元の <placement> によって生成される識別名内の文字列置換を指定するためのトークンとして使用されます。

置換文字列は、処理対象の <追加> イベントから、src-dn 属性の最末端コンポーネントの非修飾部分をコピーすることによって生成されます。src-dn が存在しない場合は、格納元の <placement-rule> はスキップされます。

例

<placement-rules> を参照してください。

内容に使用できる要素

なし

属性

なし

内容の宣言

なし

親要素

要素	説明
----	----

placement	
-----------	--

copy-path

token-src-dn

備考

<copy-path> は、格納元の <placement> によって生成される識別名内の文字列置換を指定するためのトークンとして使用されます。

置換文字列は、処理対象の <追加> イベントから src-dn 属性をコピーすることによって生成されます。格納元である <placement-rules> の src-dn-format と dest-dn-format が異なる場合は、前者から後者への変換が実行されます。ソースが eDirectory である場合を除いて、タイプなし (非修飾) フォーマットからタイプ付き (完全修飾) フォーマットへの変換はサポートされません。

例

<placement-rules> を参照してください。

内容に使用できる要素

なし

属性

なし

内容の宣言

なし

親要素

要素	説明
placement	

copy-path-suffix

token-unmatched-src-dn

備考

<copy-path-suffix> は、格納元の <placement> によって生成される識別名内の文字列置換を指定するためのトークンとして使用されます。

置換文字列は、処理対象の <追加> イベントから src-dn 属性をコピーして、格納元である <placement-rule> 内の <match-path> に一致する src-dn の部分を削除することによって生成されます。<match-path> が指定されていない場合は、src-dn 全体がコピーされます。

placement-rules

例

<placement-rules> を参照してください。

内容に使用できる要素

なし

属性

なし

内容の宣言

なし

親要素

要素	説明
----	----

placement	
-----------	--

create-rule

rule

備考

<create-rule> を使用して、<追加> イベントの結果として新しいオブジェクトを作成するための条件を指定します。

<create-rule> が評価される時は、これが問題の<追加> イベントに対して適切なルールであるかどうか最初にチェックされます。このために、このルールでクラス名が指定されているかどうかチェックされます。指定されている場合は、そのクラス名がイベント上のクラス名と一致する場合にのみ、このルールは適切となります。次に、<match-attr> 条件がこのルールで指定されているかどうかチェックされます。指定されている場合は、<match-attr> で要求されるすべての属性値が<追加> に含まれている場合にのみ、このルールは適切となります。適切でないと判定された場合、そのルールはスキップされます。

適切なルールが見つかり、<追加> が評価されて、デフォルト値を含んでいないすべての<required-attr> の値が保持されているかどうか確認されます。保持されていない場合は<追加> は拒否され、保持されている場合は許可されます。次に、<追加> に含まれていなかったデフォルト値を持つすべての必須属性が入力されます。<required-attr> 要素の write-back 属性が設定されている場合は、含まれていない値もソースオブジェクトに書き戻されます。<template> が指定されている場合は、template-dn 属性が入力されます。

例

<create-rules> を参照してください。

内容に使用できる要素

要素	説明
match-attr	
read-attr	
template	

属性

属性	有効値	デフォルト値
class-name	CDATA eDirectory ネームスペース内でこのルールが適用されるオブジェクトのベースクラスの名前。空であるか存在しない場合は、このルールはすべてのベースクラスに適用されます。	#IMPLIED
description	CDATA 主に ConoleOne で使用される、このルールの説明。	#IMPLIED

内容の規則

(match-attr * , required-attr * , template ?)

親要素

要素	説明
create-rules	

create-rules

policy

備考

<create-rules> は、オブジェクト作成ルールのトップレベル (ドキュメント) 要素です。オブジェクト作成ルールは、DirXML-Subscriber または DirXML-Publisher オブジェクトの DirXML-CreateRule 属性が指している DirXML-Rule オブジェクトの DirXML-XmlData 属性に保存されています。

購読者チャンネルでは、ソースは eDirectory であり、ターゲットはアプリケーションです。発行者チャンネルでは、ソースはアプリケーションであり、ターゲットは eDirectory です。

オブジェクト作成ルールを使用して、ソース内の <追加> イベントの結果としてターゲット内で新しいオブジェクトを作成するかどうかを決定します。(関連付けられていないオブジェクトからのイベントについては、Identity Manager は自動的に <変更> を <追加> に変換します)。既存の一致ルールが適用されたが、ターゲット内で一致オブジェクトが見つからなかった場合にのみ、その後でオブジェクト作成ルールが適用されます。

<create-rules> には、<create-rule> 要素が含まれていないか 1 つ以上含まれています。作成ルールプロセッサは、適切なルールが見つかるまで各 <create-rule> を順番に評価します。適切なルールが見つかったら、そのルールによってオブジェクト作成が拒否または許可されて、指定されたデフォルトの属性とテンプレートが入力されます。適切な <create-rule> が見つからなかった場合は、オブジェクト作成が許可されます。

例

```
<create-rules>          <!-- For all Users in the Defense organization
require Given Name-->    <!-- Surname, and Security Clearance.
Create using the -->    <!-- templates\Secure User template -->
<create-rule class-name="User">      <match-attr attr-
name="OU">                <value>Defense</value>          </
match-attr>              <required-attr attr-name="Given Name"/>
<required-attr attr-name="Surname"/>      <required-attr
attr-name="Security Clearance"/>          <template template-
dn="templates\Secure User"/>            </create-rule>      <!-- For all
other Users require Given Name and Surname. -->      <!-- Default the
value of Security Clearance to None -->      <!-- Don't use a
template for creation -->      <create-rule class-name="User">
<required-attr attr-name="Given Name"/>      <required-attr
attr-name="Surname"/>          <required-attr attr-name="Security
Clearance">                <value>None</value>          </
required-attr>          </create-rule></create-rules>
```

内容に使用できる要素

要素	説明
----	----

create-rule

属性

なし

内容の規則

(create-rule) *

親要素

なし

削除

アプリケーションでオブジェクトが削除されたときに、PublicationShim から Identity Manager へのイベント通知として使用されます。通知として使用される場合は、< 関連付け > が必要です。アプリケーションでオブジェクトを削除するための、Identity Manager から SubscriptionShim へのコマンドとしても使用されます。コマンドとして使用される場合は、削除するオブジェクトの固有キーである < 関連付け > が必要です。

備考

<delete> への応答は、<delete> が正常に処理されたかどうかを示す <status> である必要があります。

例

```
<delete class-name="User" src-dn="\Sam" >                <association>1012</  
association></delete>
```

内容に使用できる要素

要素	説明
関連付け	
operation-data	

属性

属性	有効値	デフォルト値
class-name	CDATA	#IMPLIED
dest-dn	CDATA	#IMPLIED
	イベント通知については空のままにしておく必要があります。	
	CDATA	#IMPLIED
	CDATA	#IMPLIED

属性	有効値	デフォルト値
	CDATA	#IMPLIED
src-dn	CDATA	#IMPLIED
src-entry-id	CDATA 送信者のネームスペース内でイベントを生成したソースオブジェクトのエントリ ID。 予約済み。	#IMPLIED
タイムスタンプ	CDATA 予約済み。	#IMPLIED

内容の規則

(association ? , operation-data ?)

親要素

要素	説明
input	

driver-config

ドライバ固有の環境設定オプションを指定するために使用されます。これは、eDirectory 内の DirXML-Driver オブジェクトの DirXML-ShimConfigInfo 属性に保存されている XML 内のトップレベル要素です。

備考

含まれている `<driver-options>`、`<subscriber-options>`、および `<publisher-options>` には、それぞれ任意数の `<config-object>` およびドライバ定義要素を格納できます。ドライバ定義要素には、それぞれテキストデータが格納されている場合があります。

各ドライバ定義要素は 1 つの `type` 属性を持つことができます。 `type` 属性に `password-ref` という値を割り当てることにより、その要素が名前付きパスワードを参照することを指定できます。名前付きパスワードの参照は、実行時に、Identity Manager の管理機能を使用して設定された名前付きパスワードの実際の値に置換されます。

Identity Manager の管理インタフェースでは、各ドライバ定義要素は、要素のコンテンツを編集できる編集コントロールとして表示されます。各 `<config-object>` は単一値の `dn` コントロールとして表示され、このコントロールを使用して、格納されている `<クエリー>` の `dest-dn` を入力するための `dn` を選択できます。各コントロールのラベルには、`display-name` 属性が存在する場合はこの属性の値が表示され、この属性が存在しない場合はタグ名が表示されます。

例

```
<driver-config name="Netscape DirXML Driver" > <driver-options>
<display-method display-name="Debug Output (0-none,1-Window, 2-
DSTrace)">1</display-method> </driver-options>
<subscriber-options> <config-object display-name="Super
driver configurationdata"> <query dest-
dn="novell/Driver Set/SuperDriver/Config Object" scope="entry" event-
id="config1"> <read-attr attr-name="Some
Attribute"/> <read-attr attr-
name="XmlData" type="xml"/> </query>
</config-object> </subscriber-options> <publisher-
options> <pollRate display-name="Poll rate
inseconds">5</pollRate> <changeLogSuffix display-
name="Netscape changelogsuffix">cn=changelog</changeLogSuffix>
<changeLogBegin display-name="Starting changelog (1-First,2-New, 3-
Continue)">2</changeLogBegin> </publisher-options></driver-
config>
```

内容に使用できる要素

要素	説明
<code>driver-options</code>	ドライバ固有の DriverShim 環境設定オプション。
<code>subscriber-options</code>	

要素	説明
----	----

publisher-options

属性

属性	有効値	デフォルト値
名前	CDATA 人間が読めるドライバシム名。	#IMPLIED

内容の規則

(driver-options ? , subscriber-options ? , publisher-options ?)

親要素

なし

driver-filter

特定のチャンネルで使用されているイベントフィルタを指定するために使用されます。これは、DirXML-Subscriber または DirXML-Publisher オブジェクトの DirXML-DriverFilter 属性から生成されます。

例

< [パラメータの開始](#) > を参照してください。

内容に使用できる要素

要素	説明
allow-class	

属性

属性	有効値	デフォルト値
type	publisher subscriber	#IMPLIED
	フィルタの対象となるチャンネルを指定します。	

内容の規則

([allow-class](#)) *

親要素

要素	説明
パラメータの開始	DriverShim、SubscriptionShim、または PublicationShim の初期化パラメータ。

driver-options

これは、eDirectory 内の DirXML-Driver オブジェクトの DirXML-ShimConfigInfo 属性から取得されます。

例

<driver-config> を参照してください。

内容に使用できる要素

ANY

属性

なし

内容の規則

ANY

親要素

要素	説明
driver-config	ドライバ固有の DriverShim 環境設定オプション。
パラメータの開始	

driver-state

ドライバ固有の状態情報を指定するために使用されます。

例

<パラメータの開始>を参照してください。

内容に使用できる要素

ANY

属性

なし

内容の規則

ANY

親要素

要素	説明
パラメータの開始	

名前付きパスワードの取得

ドライバの名前付きパスワードを取得するために使用されます。<status>の結果が返され、成功または失敗が示されます。ステータスが成功の場合は、パスワード値が含まれた<パスワード>要素も返されます。<get-namedpassword>のコンテンツは、取得されるパスワードの名前またはキーです。

例

```
<get-named-passwordevent-id="gnp37">web-password</get-named-password>
```

内容に使用できる要素

#PCDATA

要素	説明
operation-data	

属性

属性	有効値	デフォルト値
	CDATA	#IMPLIED

内容の規則

親要素

要素	説明
input	

パラメータの開始

DriverShim、SubscriptionShim、または PublicationShim の初期化パラメータを指定するために使用されます。

備考

<init-params> は、ドライバから Identity Manager へのすべての <output> または <input> にも含まれており、これにより、格納されている <driver-state>、<subscriber-state>、および <publisher-state> のコンテンツを eDirectory 内の DirXML-Driver オブジェクトの DirXML-DriverStorage 属性に保存するように Identity Manager に指示されます。状態は、ドライバ、購読者、または発行者の起動時に対応する init () 関数に送信された <init-params> に含まれています。

例

```
<!-- for DriverShim.init() --><init-params src-
dn="\MY_TREE\MyOrg\MyDriverSet\MyDriver"> <authentication-info>
<server>localhost</server> <user>Fred</user>
<password>foobar</password> </authentication-info>
<driver-options> <!-- some driver defined driver options
--> </driver-options> <driver-state> <!--
some driver defined driver state --> </driver-state></init-
params><!-- for SubscriptionShim.init() --><init-params src-
dn="\MY_TREE\MyOrg\MyDriverSet\MyDriver\Subscriber">
<authentication-info> <server>localhost</server>
<user>Fred</user> <password>foobar</password> </
authentication-info> <driver-filter type="subscriber">
<allow-class class-name="User"> <allow-attr attr-
name="Telephone Number"/> <allow-attr attr-
name="CN"/> <allow-attr attr-name="Surname"/>
<allow-attr attr-name="Given Name"/> <allow-attr
attr-name="Description"/> <allow-attr attr-
name="Title"/> <allow-attr attr-name="Postal
Address"/> <allow-attr attr-name="GUID"/>
<allow-attr attr-name="Full Name"/> </allow-class>
<allow-class class-name="Organizational Unit">
<allow-attr attr-name="OU"/> </allow-class>
<allow-class class-name="Organizational"> <allow-
attr attr-name="O"/> </allow-class> </driver-
filter> <subscriber-options> <!-- some driver
defined subscriber options --> </subscriber-options>
<subscriber-state> <!-- some driver defined subscriber
state --> </subscriber-state></init-params><!-- for
PublicationShim.init() --><init-params src-
dn="\MY_TREE\MyOrg\MyDriverSet\MyDriver\Publisher">
<authentication-info> <server>localhost</server>
<user>Fred</user> <password>foobar</password> </
authentication-info> <driver-filter type="publisher">
<allow-class class-name="User"> <allow-attr attr-
name="Telephone Number"/> <allow-attr attr-
name="CN"/> <allow-attr attr-name="Surname"/>
```

```

<allow-attr attr-name="Given Name"/>                                <allow-attr
attr-name="Description"/>                                        <allow-attr attr-
name="Title"/>                                                <allow-attr attr-name="Postal
Address"/>                                                    <allow-attr attr-name="GUID"/>
<allow-attr attr-name="Full Name"/>                                </allow-class>
<allow-class class-name="Organizational Unit">
<allow-attr attr-name="OU"/>                                    </allow-class>
<allow-class class-name="Organizational">                        <allow-
attr attr-name="O"/>                                          </allow-class>      </driver-
filter>      <publisher-options>                                <!-- some driver
defined publisher options -->      </publisher-options>
<publisher-state>      <!-- some driver defined publisher
state -->      </publisher-state></init-params><!-- for
DriverShim.getSchema() --><init-params>  <authentication-info>
<server>localhost</server>      <user>Fred</user>
<password>foobar</password>      </authentication-info>
<driver-filter type="subscriber">      <allow-class class-
name="User">      <allow-attr attr-name="Telephone
Number"/>      <allow-attr attr-name="CN"/>
<allow-attr attr-name="Surname"/>      <allow-attr
attr-name="Given Name"/>      <allow-attr attr-
name="Description"/>      <allow-attr attr-
name="Title"/>      <allow-attr attr-name="Postal
Address"/>      <allow-attr attr-name="GUID"/>
<allow-attr attr-name="Full Name"/>      </allow-class>
<allow-class class-name="Organizational Unit">
<allow-attr attr-name="OU"/>                                    </allow-class>
<allow-class class-name="Organizational">                        <allow-
attr attr-name="O"/>                                          </allow-class>      </driver-
filter>      <driver-filter type="publisher">      <allow-
class class-name="User">      <allow-attr attr-
name="Telephone Number"/>      <allow-attr attr-
name="CN"/>      <allow-attr attr-name="Surname"/>
<allow-attr attr-name="Given Name"/>      <allow-attr
attr-name="Description"/>      <allow-attr attr-
name="Title"/>      <allow-attr attr-name="Postal
Address"/>      <allow-attr attr-name="GUID"/>
<allow-attr attr-name="Full Name"/>      </allow-class>
<allow-class class-name="Organizational Unit">
<allow-attr attr-name="OU"/>                                    </allow-class>
<allow-class class-name="Organizational">                        <allow-
attr attr-name="O"/>                                          </allow-class>      </driver-
filter>      <driver-options>      <!-- some driver defined
driver options -->      </driver-options>      <subscriber-options>
<!-- some driver defined subscriber options -->      </subscriber-
options>      <publisher-options>      <!-- some driver
defined publisher options -->      </publisher-options>
<driver-state>      <!-- some driver defined driver state --
>      </driver-state>      <subscriber-state>      <!--
some driver defined subscriber state -->      </subscriber-state>
<publisher-state>      <!-- some driver defined publisher
state -->      </publisher-state></init-params>

```

内容に使用できる要素

要素	説明
authentication-info	
driver-filter	
driver-options	
subscriber-options	
publisher-options	
driver-state	
subscriber-state	
publisher-state	
operation-data	

属性

属性	有効値	デフォルト値
src-dn	CDATA DirXML-Driver、DirXML-Publisher、または DirXML-Subscriber の識別名。	#IMPLIED

内容の規則

([authentication-info](#) ? , [driver-filter](#) ? , [driver-options](#) ? , [subscriber-options](#) ? , [publisher-options](#) ? , [driver-state](#) ? , [subscriber-state](#) ? , [publisher-state](#) ? , [operation-data](#) ?)

親要素

要素	説明
input	
output	

input

ドライバまたは Identity Manager へ入力として送信されたイベントまたはコマンドをカプセル化するために使用されます。Identity Manager またはドライバのインタフェースメソッドへパラメータとして送信されたすべての <nds> ドキュメントには、1 つのみの <input> が含まれている必要があります。

例

<nds> を参照してください。

内容に使用できる要素

要素	説明
追加	
変更	
削除	
名前を変更する	
移動	移動イベントが発生したときにオブジェクトを移動します。
クエリー	
query-ex	
クエリスキーマ	
関連付けの追加	関連付けコマンドを追加します。
関連付けの変更	
関連付けの削除	関連付けコマンドを削除します。
パラメータの開始	
status	
チェックパスワード	パスワードを eDirectory ドライバオブジェクトと照らし合わせてチェックします。
パスワード変更	
オブジェクトパスワードの確認	パスワードを eDirectory オブジェクトと照らし合わせてチェックします。
同期	
名前付きパスワードの取得	

属性

なし

内容の規則

(add | modify | delete | rename | move | query | query-ex | query-schema | add-association | modify-association | remove-association | init-params | status | check-password | modify-password | check-object-password | sync | get-named-password) *

親要素

要素	説明
nds	

インスタンス

<クエリー> コマンドまたは <query-ex> コマンドへの応答の一部として、eDirectory 内またはアプリケーション内のオブジェクトを表すために使用されます。<instance> は、必ずしもオブジェクトの完全な状態を表すのではなく、<クエリー> または <query-ex> に よって要求された情報のみを表します。ドライバから返されたときは、<関連付け> が 必要です。

例

```
<instance class-name="User" src-dn="\Users\Samuel">
<association>1012</association>      <attr attr-name="Surname">
<value>Jones</value>                  </attr>      <attr attr-name="cn">
<value>Samuel</value>                 </attr>      <attr attr-name="Given
Name">
      <value>Samuel</value>           </attr>      <attr
attr-name="Telephone Number">
      <value>555-1212</value>
<value>555-1764</value>               </attr></instance>
```

内容に使用できる要素

要素	説明
関連付け	
parent	オブジェクトの親またはコンテナ。
attr	
operation-data	

属性

属性	有効値	デフォルト値
class-name	CDATA	#REQUIRED
	CDATA	#IMPLIED
	CDATA	#IMPLIED
src-dn	CDATA	#IMPLIED
	送信者のネームスペース内でイ ベントを生成したソースオブ ジェクトの識別名。	

属性	有効値	デフォルト値
src-entry-id	CDATA	#IMPLIED
<p>予約済み：ドライバでは無視される必要があります。</p>		

内容の規則

(association ? , parent ? , attr * , operation-data ?)

親要素

要素	説明
output	

match-attr

Identity Manager 2.0 から廃止されました。オブジェクト作成ポリシーおよびオブジェクト配置ポリシーについては DirXMLScript `<if-op-attr>` を使用し、オブジェクト一致ポリシーについては `<do-find-matching-object>` および `<arg-match-attr>` を使用してください。

備考

`<match-attr>` では以下を指定します。

- ◆ 格納元の `<create-rule>` または `<placement-rule>` のルール選択条件。これを指定する場合は、1 つ以上の `<値>` が含まれている必要があります。
- ◆ 格納元の `<matching-rule>` のオブジェクト選択条件。これを指定する場合は、`<値>` が含まれてはいけません。

例

`<create-rules>` を参照してください。

内容に使用できる要素

要素	説明
<code>値</code>	

属性

属性	有効値	デフォルト値
<code>attr-name</code>	CDATA eDirectory ネームスペース内の必須属性の名前。	#REQUIRED

内容の規則

(値)*

親要素

要素	説明
<code>create-rule</code>	
<code>matching-rule</code>	
<code>placement-rule</code>	

match-class

Identity Manager 2.0 から廃止されました。代わりに DirXMLScript <if-object-class> を使用してください。

備考

<match-class> を使用して、格納元の <matching-rule> または <placement-rule> のルール選択条件を指定します。

例

<matching-rules> および <placement-rules> を参照してください。

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
class-name	CDATA eDirectory ネームスペース内の ベースクラスの名前。	#REQUIRED

内容の宣言

なし

親要素

要素	説明
matching-rule	
placement-rule	

match-path

Identity Manager 2.0 から廃止されました。オブジェクト配置ポリシーについては DirXMLScript `<if-src-dn>` を使用し、オブジェクト一致ポリシーについては `<do-find-matching-object>` および `<arg-dn>` を使用してください。

備考

`<match-path>` では以下を指定します。

- ◆ 格納元の `<placement-rule>` のルール選択条件。これを指定する場合は、ソースの `<追加>` イベントの `src-dn` 属性はプレフィックスと比較され、ルートがプレフィックスであるサブツリー内に `src-dn` がある場合は、この属性は一致対象と見なされます。パスのネームスペースはイベントソースと同じです。
- ◆ `matching-rule` これを指定する場合は、プレフィックスは、格納元ルールによって生成される `<クエリー>` の `dest-dn` として使用されます。パスのネームスペースはイベントターゲットと同じです。

パスのネームスペースが eDirectory の場合は、フォーマットは円記号フォーマットです (例: `\treename\container\...\leaf`)。先頭の `\` を省略した場合は、パスはツリールートに対する相対パスと見なされます。

パスのネームスペースがアプリケーションネームスペースの場合は、パスのフォーマットはアプリケーションに依存し、ドライバのライターによってドキュメント化される必要があります。

例

`matching-rulesplacement-rules`

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
prefix	CDATA 一致するパスまたは識別名の最もルートに近い部分。	#REQUIRED

内容の宣言

なし

親要素

要素	説明
matching-rule	
placement-rule	

matching-rule

Identity Manager 2.0 から廃止されました。代わりに DirXMLScript `<rule>` および `<do-find-matching-object>` を使用してください。

備考

`<matching-rule>` を使用して、`<追加>` イベントの結果として新しいオブジェクトを自動的に関連付けるための一致オブジェクトを検索するための条件を指定します。

`<matching-rule>` が評価される時は、これが問題の `<追加>` イベントに対して適切なルールであるかどうか最初にチェックされます。このために、このルールで `<match-class>` 要素が指定されているかどうかチェックされます。指定されている場合は、このイベント上のクラス名がいずれかの `<match-class>` 要素上のクラス名と一致する場合にのみ、このルールは適切となります。 `modify-attr` 指定されている場合は、`<追加>` に `<match-attr>` で指定された各属性の属性値が含まれている場合にのみ、このルールは適切となります。

適切なルールが見つかり、ルール (`<match-attr>` および `<match-path>`) で指定された条件と、`<追加>` で指定されたクラス名および属性値に基づいて `<クエリー>` が生成されます。このクエリはターゲット (eDirectory またはドライバ) に送信されます。返されるすべての `<インスタンス>` 要素は一致対象と見なされます。

例

`<matching-rules>` を参照してください。

内容に使用できる要素

要素	説明
<code>match-class</code>	
<code>match-path</code>	
<code>match-attr</code>	

属性

属性	有効値	デフォルト値
description	CDATA	#IMPLIED

内容の規則

(`match-class *` , `match-path ?` , `match-attr *`)

親要素

要素	説明
matching-rules	

matching-rules

Identity Manager 2.0 から廃止されました。DirXMLScript `<policy>` を使用してください。

備考

`<matching-rules>` は、オブジェクト一致ルールの上レベル (ドキュメント) 要素です。オブジェクト一致ルールは、DirXML-Subscriber または DirXML-Publisher オブジェクトの DirXML-MatchingRule 属性が指している DirXML-Rule オブジェクトの DirXML-XmlData 属性に保存されています。

発行者チャンネルでは、ソースはアプリケーションであり、ターゲットは eDirectory です。

オブジェクト一致ルールを使用して、ソース内の `<追加>` イベントの結果として、ソース内の関連付けられていないオブジェクトのターゲット内で一致オブジェクトが検索されます。(関連付けられていないオブジェクトからのイベントについては、DirXML は `<変更>` を `<追加>` に自動的に変換します)。オブジェクト一致ルールは、ターゲット内で新しいオブジェクトを作成するかどうかを決定する前に適用されます。

`<matching-rules>` には、`<matching-rule>` 要素が含まれていないか 1 つ以上含まれています。一致ルールプロセッサは、ターゲット内の 1 つ以上の一致オブジェクトが見つかるまで各 `<matching-rule>` を順番に評価します。

一致オブジェクトが 1 つだけ見つかった場合は、そのオブジェクトはソースオブジェクトと自動的に関連付けられ、Identity Manager は、発行者と購読者のフィルタによって許可されたとおりに 2 つのオブジェクトの属性値の違いを調整しようとします。

複数の一致オブジェクトが見つかった場合は、エラーが通知され、オブジェクトを手動で関連付けるか、オブジェクト一致ルールをより具体的な内容に変更する必要があります。

一致オブジェクトが見つからなかった場合は、Identity Manager はイベントの処理を続行します。

例

```
<matching-rules>          <!-- for Users, first try to match on Surname,
Given Name andLocation -->          <matching-rule>          <match-
class class-name="User"/>          <match-attr attr-
name="Surname"/>          <match-attr attr-name="Given Name"/>
<match-attr attr-name="Location"/>          </matching-rule>          <!--
for Users, then try to match on Surname only in -->          <!-- the
o=novell subtree -->          <matching-rule>          <match-class
class-name="User"/>          <match-path prefix="o=novell"/>
<match-attr attr-name="Surname"/>          </matching-rule>          <!--
for all classes try to match on CN only -->          <matching-rule>
<match-attr attr-name="CN"/>          </matching-rule></matching-rules>
```

内容に使用できる要素

要素	説明
<code>matching-rule</code>	

属性

なし

内容の規則

(`matching-rule` *)

親要素

なし

変更

アプリケーションでオブジェクトが変更されたことを知らせる、PublicationShim から Identity Manager へのイベント通知として使用されます。**関連付け**これは、アプリケーションでオブジェクトを変更するための、Identity Manager から SubscriptionShim へのコマンドとしても使用されます。コマンドとして使用される時は、変更するオブジェクトの固有キーである **<関連付け>**が必要です。

備考

<追加>には、変更された属性ごとに1つの **<modify-attr>**が含まれています。

<modify>への応答は、**<modify>**が正常に処理されたかどうかを示す **<status>** である必要があります。

例

```
<modify class-name="User" src-dn="\Sam" >          <association>1012</association>
<modify-attr attr-name="Given Name">
<remove-all-values/>          <add-value>
<value>Samuel</value>          </add-value>          </modify-attr>
<modify-attr attr-name="Telephone Number">          <remove-value>
<value>555-1212</value>          </remove-value>          </modify-attr>
<modify-attr attr-name="Telephone Number">          <add-value>
<value>555-1764</value>          <value>555-1765</value>          </add-value>
</modify-attr></modify>
```

内容に使用できる要素

要素	説明
関連付け	
modify-attr	属性を変更します。
operation-data	

属性

属性	有効値	デフォルト値
class-name	CDATA	#IMPLIED

通知として使用される時は必要です。

属性	有効値	デフォルト値
dest-dn	CDATA	#IMPLIED
	予約済み。	
	CDATA	#IMPLIED
	予約済み。	
	CDATA	#IMPLIED
from-merge		false
	コマンドがマージの結果である場合は true です。	
	CDATA	#IMPLIED
src-dn	CDATA	#IMPLIED
src-entry-id	CDATA	#IMPLIED
	送信者のネームスペース内でイベントを生成したソースオブジェクトのエントリ ID。	
	予約済み。	
タイムスタンプ	CDATA	#IMPLIED
	予約済み。	

内容の規則

(association ? , modify-attr + , operation-data ?)

親要素

要素	説明
input	

関連付けの変更

アプリケーションオブジェクトの固有キーが変更されたことを Identity Manager に通知するために使用されます。SubscriptionShim または PublicationShim のイベントフィルタを通過したオブジェクトの固有キーが変更されたときは、<modify-association> が送信される必要があります。<modify-association> は、ドライバから Identity Manager への任意の <output> または <input> に含めることができます。

例

```
<modify-association>      <association>{BC3E7155-CDF9-d311-9846-
0008C76B16C2}</association>      <association>{CD3F7155-DE09-e311-
9846-0008D76C16D2}</association></modify-association>
```

内容に使用できる要素

要素	説明
関連付け operation-data	

属性

属性	有効値	デフォルト値
	CDATA	#IMPLIED

内容の規則

(association , association , operation-data ?)

親要素

要素	説明
input	
output	

modify-attr

<変更> 操作またはイベントの変更された属性値を指定するために使用されます。

備考

各 <modify-attr> には、1 つ以上の <add-value>、<remove-value>、または <remove-all-values> が含まれている必要があります。

これらの要素の順序は重要です。

例

<変更> を参照してください。

内容に使用できる要素

要素	説明
remove-value	指定された属性値を削除します。
remove-all-values	
add-value	

属性

属性	有効値	デフォルト値
attr-name	CDATA	#REQUIRED
enforce-password-policy	マッピングポリシーでは、格納元のコマンドまたはイベントのクラス名属性を使用して、属性名をマップするために使用するクラスを決定します。 予約済み。	#IMPLIED

内容の規則

(remove-value | remove-all-values | add-value) +

親要素

要素	説明
変更	

パスワード変更

アプリケーションでオブジェクトパスワードが変更されたことを知らせる、PublicationShim から Identity Manager へのイベント通知として使用されます。**関連付け**アプリケーションでオブジェクトパスワードを変更するための、Identity Manager から SubscriptionShim へのコマンドとしても使用されます。コマンドとして使用される時は、変更するオブジェクトの固有キーである **<関連付け>** が必要です。

備考

ターゲットが eDirectory であり、**<old-password>** が指定されている場合は、modifyPassword API を使用してパスワードが変更されます。指定されていない場合は、GenerateKeyPair API が使用されます。GenerateKeyPair を使用すると、ターゲットオブジェクトとして認証された既存セッションの認証資格情報が無効になる可能性があります。

ターゲットがアプリケーションの場合は、この機能をアプリケーションに適用可能かどうかに応じて、ドライバがこの機能を実装する場合としない場合があります。

<modify-password> への応答は、**<modify-password>** が正常に処理されたかどうかを示す **<status>** である必要があります。

例

```
<modify-password class-name="User" src-dn="\Sam">  
<association>1012</association>  
<password>mypassword<password></modify-password>
```

内容に使用できる要素

要素	説明
関連付け	
old-password	
パスワード	
operation-data	

属性

属性	有効値	デフォルト値
class-name	CDATA	#IMPLIED

属性	有効値	デフォルト値
dest-dn	CDATA	#IMPLIED
	CDATA	#IMPLIED
	予約済み。	
	CDATA	#IMPLIED
	CDATA	#IMPLIED
	src-dn の完全修飾バージョン。	
src-dn	CDATA	#IMPLIED
src-entry-id	CDATA	#IMPLIED
	予約済み。	
タイムスタンプ	CDATA	#IMPLIED
	予約済み。	

内容の規則

(association ? , old-password ? , password , operation-data ?)

親要素

要素	説明
input	

移動

アプリケーションでオブジェクトが異なるコンテナに移動されたときに、PublicationShim から Identity Manager へのイベント通知として使用されます。**関連付け**アプリケーションでオブジェクトを異なるコンテナに移動するための、Identity Manager から SubscriptionShim へのコマンドとしても使用されます。コマンドとして使用される場合は、移動するオブジェクトの固有キーである **<関連付け>** が必要です。

備考

<move> には、移動先のコンテナを指定する **<parent>** が含まれています。コマンドとして使用される場合は、**<parent>** には **<関連付け>** を含めることができます。これに関連付けが含まれていない場合は、ドライバはオブジェクトを移動しようとはならず、**<status> level="warning"** を返す必要があります。

<move> への応答は、**<move>** が正常に処理されたかどうかを示す **<status>** である必要があります。

例

```
<move class-name="User" src-dn="\Users\Samuel" old-src-dn="\Samuel">
<association>1012</association>      <parent src-dn="\Users\">
<association>1013</association>      </parent></move>
```

内容に使用できる要素

要素	説明
関連付け	
parent	
operation-data	この操作によってさらなるカスタムデータが追加されます。

属性

属性	有効値	デフォルト値
class-name	CDATA	#IMPLIED
dest-dn	CDATA	#IMPLIED
	予約済み。	

属性	有効値	デフォルト値
	CDATA	#IMPLIED
	予約済み。	
	CDATA	#IMPLIED
old-src-dn	CDATA	#IMPLIED
	送信者のネームスペース内でイベントを生成したソースオブジェクトの元の識別名。	
qualified-old-src-dn	CDATA	#IMPLIED
	old-src-dn の完全修飾バージョン。	
	CDATA	#IMPLIED
src-dn	CDATA	#IMPLIED
	送信者のネームスペース内でイベントを生成したソースオブジェクトの新しい識別名。	
src-entry-id	CDATA	#IMPLIED
	予約済み。	
タイムスタンプ	CDATA	#IMPLIED
	予約済み。	

内容の規則

(association ? , parent , operation-data ?)

親要素

要素	説明
input	

nds

Identity Manager またはドライバのインタフェースメソッドにパラメータとして送信されたか、またはこれらから返されたすべてのドキュメントのトップレベル(ドキュメント)要素。

備考

Identity Manager またはドライバのインタフェースメソッドにパラメータとして送信されたすべての `<nds>` ドキュメントには、`<input>` が 1 つだけ含まれている必要があります。

Identity Manager またはドライバのインタフェースメソッドから返されたすべての `<nds>` ドキュメントには、`<output>` が 1 つだけ含まれている必要があります。

例

```
DirXML sends:  <nds dtdversion="2.0">                <source>
<product version="2.0.0.0">DirXML</product>
<contact>Novell, Inc.</contact>                </source>                <input>
<modify class-name="User" event-id="0"    qualified-src-
dn="\T=ATREE\O=Users\CN=Julia" src-dn="\ATREE\Users\Julia"    src-
entry-id="33967">                                <association
state="associated">{B43E7155-CDF9-d311-9846-0008C76B16C2}</
association>                                <modify-attr attr-
name="Surname">                                <add-value>
<value    type="string">Gulia</value>
</add-value>                                </modify-attr>
</modify>                                </input>                </nds>
Shim returns:  <nds
dtdversion="2.0">                <source>                <product
version="2.0.0.0">Some Application    Driver</product>
<contact>Nobody in particular</contact>                </source>
<output>                <status event-id="0" level="success"/>
</output>                </nds>
```

内容に使用できる要素

要素	説明
<code>source</code>	
<code>input</code>	
<code>output</code>	

属性

属性	有効値	デフォルト値
dtdversion	CDATA ドライバの設計対象である Identity Manager の <メジャー>.<マイナー>バージョンに設定 される必要があります。	#REQUIRED
ndsversion	CDATA	#IMPLIED

内容の規則

(source ? , (input | output))

親要素

なし

nds-name

eDirectory ネームスペース内のクラス名または属性名を指定するために使用されます。

例

<attr-name-map> を参照してください。

内容に使用できる要素

#PCDATA

属性

なし

内容の規則

(#PCDATA)

親要素

要素	説明
----	----

attr-name	
-----------	--

class-name	
------------	--

new-name

格納元の<名前を変更する>イベントまたはコマンドで指定されたオブジェクトの新しい名前を指定するために使用されます。

例

<名前を変更する>を参照してください。

内容に使用できる要素

#PCDATA

属性

なし

内容の規則

(#PCDATA)

親要素

要素	説明
----	----

名前を変更する	
---------	--

old-password

格納元の <パスワード変更> の古いパスワードを指定するために使用されます。

例

<パスワード変更> を参照してください。

内容に使用できる要素

#PCDATA

属性

なし

内容の規則

(#PCDATA)

親要素

要素	説明
----	----

パスワード変更	
---------	--

operation-data

任意のイベントまたはコマンドとともに伝送される追加のカスタムデータペイロードを投入することをポリシーに許可するために使用されます。これは、アプリケーションシムに送信される前にイベントまたはコマンドから削除され、Identity Manager に返された後に対応する応答要素 (一致する event-id によって特定される) と再び関連付けられます。

備考

<operation-data> のコンテンツは任意の整形 XML にできますが、すべての要素と属性をカスタムネームスペースに配置して、これらが標準の Identity Manager 操作と混同されるのを防止することをお勧めします。

<operation-data> の一般的な用途は、操作の結果を処理するポリシーで必要となる可能性のあるその操作の追加コンテキストを提供するポリシーを作成することです。

コンテンツが通常は PCDATA である操作については、任意の文字データの後に 1 つだけの <operation-data> を配置する必要があります。これは、DTD で指定されるコンテンツルールとは対照的です。混合コンテンツ用の DTD ではより細かい指定ができないためです。

例

```
<operation-data xmlns:mystuff="http://mystuff.operation.data">
<mystuff:notify>admin@fred.com</mystuff:notify></operation-data>
```

内容に使用できる要素

ANY

属性

なし

内容の規則

ANY

親要素

要素	説明
----	----

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

要素	説明
名前付きパスワードの取得	
パラメータの開始	DriverShim、SubscriptionShim、またはPublicationShimの初期化パラメータ。
インスタンス	
変更	
関連付けの変更	
パスワード変更	
移動	
クエリー	
query-ex	
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	
同期	

output

ドライバまたは Identity Manager に返されたイベントまたはコマンドをカプセル化するために使用されます。Identity Manager またはドライバのインタフェースメソッドから返されたすべての <nds> ドキュメントには、<output> が 1 つだけ含まれている必要があります。

例

<nds> を参照してください。

内容に使用できる要素

要素	説明
status	
関連付けの追加	関連付けコマンドを追加します。
関連付けの変更	
関連付けの削除	
インスタンス	
スキーマ定義	
パラメータの開始	
パスワード	
query-token	query-ex コマンドの不透明ハンドル。

属性

なし

内容の規則

(status | add-association | modify-association | remove-association | instance | schema-def | init-params | password | query-token) *

親要素

要素	説明
nds	

parent

<移動> イベントまたはコマンドのターゲットコンテナ、または<インスタンス> で表されるオブジェクトの現在のコンテナを指定するために使用されます。

備考

ドライバから生成するときは、<parent> には、コンテナオブジェクトの固有キーが含まれた<関連付け>が含まれている必要があります。Identity Manager から生成するときは、コンテナオブジェクトが eDirectory 内の確立された関連付けを保持している場合にのみ、<parent> に<関連付け>が含まれます。

例

<移動> を参照してください。

内容に使用できる要素

要素	説明
関連付け	

属性

属性	有効値	デフォルト値
dest-dn	CDATA	#IMPLIED
	予約済み。 CDATA	#IMPLIED
	予約済み。 CDATA	#IMPLIED
src-dn	CDATA	#IMPLIED
src-entry-id	CDATA	#IMPLIED
	予約済み。	

内容の規則

(association ?)

親要素

要素	説明
----	----

インスタンス

移動

パスワード

<追加>内のオブジェクトの初期パスワードを指定するため、または<authentication-info>要素内のドライバの認証パスワードを指定するため (DirXML-Driver オブジェクトの DirXML-ShimAuthPassword 属性から取得される)、または<名前付きパスワードの取得>コマンドの処理結果として名前付きパスワードの値を返すために使用されます。

例

<パラメータの開始>を参照してください。

内容に使用できる要素

#PCDATA

属性

なし

内容の規則

(#PCDATA)

親要素

要素	説明
追加	
authentication-info	
オブジェクトパスワードの確認	
パスワード変更	
output	

placement

do-set-op-dest-dn

備考

<placement> を使用して、これから作成されるオブジェクトの識別名を指定して、格納元である <placement-rule> の選択条件を対応付けます。

DN は、含まれているトークン要素によって生成されたテキスト置換およびテキストを順番に連結することで生成されます。先頭および末尾のホワイトスペースはすべて、CDATA セクションに含まれていた場合を除いて削除されます。

例

<placement-rules> を参照してください。

内容に使用できる要素

#PCDATA

要素	説明
----	----

属性

なし

内容の規則

(#PCDATA | copy-name | copy-attr | copy-path | copy-path-suffix) *

親要素

要素	説明
placement-rule	

placement-rule

Identity Manager 2.0 から廃止されました。DirXMLScript `<rule>` を使用してください。

備考

`<placement-rule>` を使用して、これから作成されるオブジェクトの識別名を生成するための条件を指定します。

`<placement-rule>` が評価されるときは、これが問題の `<追加>` イベントに対して適切なルールであるかどうか最初にチェックされます。 `match-class` 指定されている場合は、このイベント上のクラス名がいずれかの `<match-class>` 要素上のクラス名と一致する場合にのみ、このルールは適切となります。 `match-attr` `追加` `match-attr``match-path` 指定されている場合は、`<追加>` の `src-dn` は、`<match-path>` で指定されたサブツリーの 1 つ以上に含まれている必要があります。

適切なルールが見つかり、`<placement>` が評価されて、`<追加>` の `dest-dn` 属性の値が生成されます。

適切なルールが見つからなかった場合は、`dest-dn` は空のままになり、`<追加>` の処理は続行されます。

例

`<placement-rules>` を参照してください。

内容に使用できる要素

要素	説明
<code>match-class</code>	
<code>match-path</code>	
<code>match-attr</code>	
<code>placement</code>	

属性

属性	有効値	デフォルト値
<code>description</code>	CDATA	#IMPLIED

内容の規則

(`match-class *` , `match-path *` , `match-attr *` , `placement`)

親要素

要素	説明
placement-rules	

placement-rules

policy

備考

<placement-rules> は、オブジェクト配置ルールの上レベル (ドキュメント) 要素です。オブジェクト一致ルールは、DirXML-Subscriber または DirXML-Publisher オブジェクトの DirXML-PlacementRule 属性が指している DirXML-Rule オブジェクトの DirXML-XmlData 属性に保存されています。

オブジェクト配置ルールを使用して、ソース内の <追加> イベントの結果としてこれから作成されるオブジェクトの識別名を生成します。変更追加既存の作成ルールが適用されて、<追加> が拒否されなかった場合にのみ、その後でオブジェクト配置ルールが適用されます。

<placement-rules> には、<placement-rule> 要素が含まれていないか 1 つ以上含まれています。配置ルールプロセッサは、適切なルールが見つかるまで各 <placement-rule> を順番に評価します。適切なルールが見つかったら、そのルールによって <追加> の dest-dn 属性が入力されます。

ソースイベントの識別名フォーマットは、src-dn-format または src-dn-delims のどちらかによって指定されており、ソースアプリケーションで実際に使用されているフォーマットと一致する必要があります。ターゲットイベントの識別名フォーマットは、dest-dn-format または dest-dn-delims のどちらかによって指定されており、ターゲットアプリケーションで実際に使用されているフォーマットと一致する必要があります。Identity Manager を通じて eDirectory で使用されるフォーマットは円記号です。

ほとんどのアプリケーションではいずれかの標準 DN フォーマットで十分ですが、場合によってはカスタム区切り記号セットを指定する必要があります。区切り記号セットを構成する 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 文字をエスケープ文字付きの 16 進数文字列 (\FEFF など) として出力も解釈もしないことを意味します。Unicode 文字の 0xfeff、0xffff、0xfffd、および 0xffff は、eDirectory では使用できません。
- ◆ 相対 RDN 区切り記号
- ◆ RDN 区切り記号
- ◆ 名前ディバイダ
- ◆ 名前の値の区切り記号
- ◆ ワイルドカード文字
- ◆ エスケープ文字

RDN 区切り記号と相対 RDN 区切り記号が同じ文字の場合は、名前の向きは右から左になり、異なる文字の場合は左から右になります。

例

```
<placement-rules src-dn-format="slash" dest-dn-format="ldap">
<!-- for Users coming from the subtree \Tree\novell in eDirectory-->
<!-- place them in the same relative hierarchy under o=novell -->
<placement-rule>          <match-class class-name="User"/>
<match-path prefix="\TREE\novell"/>          <placement><copy-
path-suffix/>,o=novell</placement>          </placement-rule>      <!--
- for all other users and groups -->          <!-- place them in the
department container under novell -->        <placement-rule>
<match-class class-name="User"/>          <match-class class-
name="Group"/>          <placement>cn=<copy-name/>,ou=<copy-
attrname="OU"/>,o=novell</placement>      </placement-rule>
<!-- for everything else, try to mirror the hierarchy -->
<placement-rule>          <placement><copy-path/></placement>
</placement-rule></placement-rules>
```

内容に使用できる要素

要素	説明
placement-rule	

属性

属性	有効値	デフォルト値
dest-dn-delims	CDATA dest-dn のカスタム区切り記号。	#IMPLIED
dest-dn-format	dot qualified-dot slash qualified-slash ldap src-dn のフォーマット。	slash
src-dn-delims	CDATA src-dn のカスタム区切り記号。	#IMPLIED
src-dn-format		slash

内容の規則

(placement-rule *)

親要素

なし

publisher-options

これは、eDirectory 内の DirXML-Driver オブジェクトの DirXML-ShimConfigInfo 属性から取得されます。コンテンツの詳細については、<driver-config> を参照してください。

例

<driver-config> を参照してください。

内容に使用できる要素

ANY

属性

なし

内容の規則

ANY

親要素

要素	説明
driver-config	
パラメータの開始	

publisher-state

PublicationShim のドライバ固有の状態情報を指定するために使用されます。

備考

<publisher-state> は、ドライバから Identity Manager への任意の <output> または <input> 内の <パラメータの開始> に含めることができ、これにより、eDirectory 内の DirXML-Driver オブジェクトの DirXML-DriverStorage 属性にコンテンツを保存するように Identity Manager に指示されます。

DirXML-Driver オブジェクトに保存されている <publisher-state> は、ドライバの起動時に PublicationShim.init() に送信された <パラメータの開始> に含まれています。

例

<パラメータの開始> を参照してください。

内容に使用できる要素

ANY

属性

なし

内容の規則

ANY

親要素

要素	説明
パラメータの開始	

クエリー

eDirectory またはアプリケーションからオブジェクトを検索または読み取るためのコマンド。Identity Manager のルールやオブジェクトマイグレーションなどの完全な機能を使用できるかどうかは、ドライバで `query` コマンドが完全に実装されているかどうかに依存します。

備考

`<query>` のベースオブジェクトは、`dest-dn` 属性または `<関連付け>` によって指定されます。両方とも存在する場合は、eDirectory をクエリする際は `dest-dn` 属性が優先され、接続されているアプリケーション (eDirectory 以外) をクエリする際は `<関連付け>` が優先されます。どちらも存在しない場合は、デフォルトのベースオブジェクトは、階層ネームスペースではルートオブジェクトになり、非階層ネームスペースではすべてのオブジェクトになります。

考えられるクエリのスコープは次のとおりです。

要素	説明
エントリ	ベースオブジェクトのみ。
サブオーディネート	ベースオブジェクトに含まれているオブジェクト。
サブツリー	ルートがベースオブジェクトであるサブツリー内のオブジェクト (ベースオブジェクトを含む)。

デフォルトでは、指定されたスコープ内のすべてのオブジェクトが選択されます。"entry" 以外のスコープについては、選択されるオブジェクトを `<search-class>` および `<search-attr>` によってさらに限定できます。"entry" スコープについては、`<search-attr>` および `<search-class>` の効果は定義されていません。

複数の `<search-class>` 要素が存在する場合は、いずれかの `<search-class>` 要素に一致するベースクラスを持つオブジェクトのみが選択されます。

複数の `<search-attr>` 要素が存在する場合は、すべての `<search-attr>` 要素で指定されたすべての値に一致する属性を持つオブジェクトのみが選択されます。

デフォルトでは、選択されたオブジェクトのすべてのオブジェクト属性が読み取られます。読み取られる属性は `<read-attr>` によって限定されます。どのオブジェクト属性も読み取らないようにするには、単一の名前なしの `<read-attr>` を指定してください。

`<read-parent>` が指定されている場合は、選択されたオブジェクトの `<parent>` も読み取られます。

parentread-parent

`<query>` への応答には、選択されたオブジェクトごとに1つの `<インスタンス>` が含まれている必要があります。

`<query>` への応答には、`<query>` が正常に処理されたかどうかを示す `<status>` も含まれている必要があります。検索条件に一致するオブジェクトがない場合は、これをエラーと見なしてはいけません。

例

```
<!-- search the whole application for a User object with the Surname
ofJones --><!-- don't read any attributes but read the parent --><query
class-name="User" event-id="0" scope="subtree">      <search-class
class-name="User"/>      <search-attr attr-name="Surname">
<value type="string">Jones</value>      </search-attr>      <read-
attr/>      <read-parent/></query><!-- read the User object whose
foreign key is 1011 --><!-- read Surname,cn,Given Name and Telephone
Number attributes --><query class-name="User" event-id="1"
scope="entry">      <association>1011</association>      <read-attr
attr-name="Surname"/>      <read-attr attr-name="cn"/>      <read-
attr attr-name="Given Name"/>      <read-attr attr-name="Telephone
Number"/></query>
```

内容に使用できる要素

要素	説明
関連付け	
search-class	
search-attr	
read-attr	指定されたオブジェクト属性値を返します。
read-parent	オブジェクトの親を返します。
operation-data	

属性

属性	有効値	デフォルト値
class-name	CDATA	#IMPLIED

これは、検索で指定された属性名の適切なスキーママッピングのために必要です。これを使用して検索を制限してはいけません。

dest-dn	CDATA	#IMPLIED
---------	-------	----------

	CDATA	#IMPLIED
--	-------	----------

予約済み。

属性	有効値	デフォルト値
	CDATA	#IMPLIED
scope	entry subordinates subtree クエリのスコープ。	

内容の規則

(association ? , (search-class | search-attr | read-attr | read-parent) * , operation-data ?)

親要素

要素	説明
config-object	
input	

query-ex

<クエリー>の派生コマンドであり、同時に返される検索結果の数を制限するために使用されます。検索とクエリの一般情報については、<クエリー>を参照してください。

備考

query-ex コマンドの結果には、<query-token>要素が含まれている場合があります。後続の query-ex コマンドでは query-token 要素が使用されて、追加の結果が取得されます。query-ex の結果に query-token 要素が含まれていない場合は、その query-ex コマンドの取得可能な結果がすべて返されます。結果セットごとに異なる query-token が返される場合があります。最初の結果セットで返された query-token のみを使用することは十分ではありません。

query-ex コマンドには、query コマンドにはない次の追加属性があります。

- ◆ max-result-count 属性は、単一の query-ex コマンドの結果として返す<インスタンス>要素の最大数を指定します。
- ◆ cancel 属性は、query-ex シーケンスを取り消すために使用され、その結果、検索に関連付けられたすべてのリソースが解放されます。

query-ex コマンドシーケンスは、連続して発行された複数の query-ex コマンドで構成される場合があります。最初の query-ex コマンドは、検索パラメータを設定して、初期結果セットを返しますが、初期結果セットで返すことができない追加の結果がある場合は、後続の query-ex コマンドで使用される も返します。後続の query-ex コマンドは query-token 要素を含んでおり、初期検索から追加の結果セットを取得するために繰り返し発行されます。このプロセスは、結果セット内に query-token 要素がなくなるまで継続されます。

前の query-ex から返されたトークンを使用している後続の query-ex コマンドは、属性や子要素にかかわらず検索パラメータを変更しません。

query-ex コマンドで cancel 属性を true に設定することで、すべての結果が返される前に query-ex シーケンスを中止できます。

すべてのアプリケーションシムが query-ex をサポートしているわけではありません。query-ex をサポートしているアプリケーションシムは、シムの起動時にメタディレクトリエンジンにそのことを報告します。この報告のために、ドライバ識別クエリへの応答として返された<インスタンス>要素の子として以下を返します。

```
<attr attr-name="query-ex-supported">          <value
type="state">true</value>    </attr>
```

例

```
<!-- search the whole application for all User objects --><!-- don't
read any attributes --><query-ex class-name="User" event-id="0"
scope="subtree" max-result-count="50">          <search-class class-
name="User"/>          <read-attr/></query-ex><!-- read additional
results based on the above query, assuming thatthe results of the above
included the query-token illustrated --><query-ex>          <query-
token>r00ABXNyACxjb20ubm92ZWxsLm5kcy5kaXJ4bWwuZW5naW5lLk5EU1JlYWwlcjRl
b2t1buWeJE0ga5xBAGCSgAFc3RhbXBjAA1zdGF0ZUhhc2h4cAAAAQQLokQbAbjW9w==</
query-token></query-ex><!-- cancel a query-ex before having read all
```

```
results --><query-ex cancel="true"> <query-
token>r00ABXNyACxjb20ubm92ZWxsLm5kcy5kaXJ4bWwuZW5naW5lLk5EU1JlYWRLciRU
b2t1buWeJE0ga5xBAGACsgAFc3RhbXBjAA1zdGF0ZUhhc2h4cAAAAQQLokQbAbjW9w==</
query-token></query-ex>
```

内容に使用できる要素

要素	説明
関連付け	
query-token	
search-class	
search-attr	クエリ検索属性値フィルタ
read-attr	
read-parent	
operation-data	

属性

属性	有効値	デフォルト値
キャンセル		#IMPLIED
	<query-token> 要素が含まれた query-ex コマンドで true に設定した場合は、検索は中止されて、すべての関連付けられたリソースは検索ターゲットによって解放されます。	
class-name	CDATA オブジェクトのベースクラスの名前。	#IMPLIED
dest-dn	CDATA	#IMPLIED
	CDATA	#IMPLIED
	予約済み。	
	CDATA	#IMPLIED

属性	有効値	デフォルト値
max-result-count	CDATA query-ex コマンドの結果として返す < インスタンス > 要素の最大数を指定します。有効な値は正の 10 進整数です。	#IMPLIED
scope	クエリのスコープ。entry スコープは、query-ex ではほとんど意味がありませんがサポートされています。	

内容の規則

(association ? , query-token ? , (search-class | search-attr | read-attr | read-parent) * , operation-data ?)

親要素

要素	説明
input	

クエリスキーマ

eDirectory またはアプリケーションからスキーマ定義を読み取るためのコマンド。現在はドライバはこのコマンドを実装する必要はありませんが、将来的には必要になる可能性があります。ドライバは、まったく同じ処理を実行する `DriverShim.getSchema()` メソッドを実装する必要があります。

備考

<query-schema> への応答は <スキーマ定義> である必要があります。

<クエリー> への応答には、<クエリー> が正常に処理されたかどうかを示す <status> も含まれている必要があります。

例

```
<query-schema/>
```

内容に使用できる要素

要素	説明
<code>operation-data</code>	

属性

属性	有効値	デフォルト値
	CDATA	#IMPLIED

内容の規則

(operation-data ?)

親要素

要素	説明
<code>input</code>	

query-token

query-ex コマンドと組み合わせて使用されます。**query-token** 要素のコンテンツは、**query-ex** コマンドの結果として返されていない検索結果を参照するための不透明トークンまたは不透明ハンドルです。

備考

<query-token> 要素は、**query-ex** コマンドの結果セットに含まれて返される場合があります、初期 **query-ex** コマンドによって選択された追加の結果を取得するために、<**query-ex**> 要素の子として使用されます。

内容に使用できる要素

ANY

属性

なし

内容の規則

ANY

親要素

要素	説明
output	
query-ex	

read-attr

格納元の<クエリー>コマンドによって読み取られるオブジェクト属性を指定するために使用されます。

例

<クエリー>を参照してください。

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
attr-name	CDATA	#IMPLIED
type	default xml type=xml の場合は、属性値は XML として解析されて XML として返されます。	デフォルト

内容の宣言

なし

親要素

要素	説明
クエリー	
query-ex	

read-parent

格納元の<クエリー>コマンドによって返されるすべてのインスタンスに<parent>が含まれる必要があることを指定するために使用されます。

例

<クエリー>を参照してください。

内容に使用できる要素

なし

属性

なし

内容の宣言

なし

親要素

要素	説明
クエリー	
query-ex	

remove-all-values

格納元の `<modify-attr>` で指定された属性から削除されるすべての値を指定するために使用されます。

例

`<変更>` を参照してください。

内容に使用できる要素

なし

属性

なし

内容の宣言

なし

親要素

要素	説明
<code>modify-attr</code>	属性を変更します。

関連付けの削除

特定の固有キーが無効であることを Identity Manager に通知するために使用されます。
<remove-association> は一般に、無効な関連付け値が Identity Manager からドライバに送信されたときに発生する必要があります。これはイベント変換ポリシーの一部としても便利です。これはドライバから Identity Manager への任意の <output> または <input> に含めることができます。

例

```
<remove-association>{BC3E7155-CDF9-d311-9846-0008C76B16C2}</remove-association>
```

内容に使用できる要素

#PCDATA

要素	説明
<code>operation-data</code>	

属性

属性	有効値	デフォルト値
	CDATA	#IMPLIED

内容の規則

親要素

要素	説明
<code>input</code>	
<code>output</code>	

remove-value

格納元の <modify-attr> で指定された属性から削除される値を指定するために使用されます。各 <add-attr> には 1 つ以上の <値> が含まれている必要があります。ドライバは、存在しない値が指定された <remove-value> を無視して、格納元である <変更> の残り部分の処理を続行する必要があります。

例

<変更> を参照してください。

内容に使用できる要素

要素	説明
値	

属性

なし

内容の規則

(値+)

親要素

要素	説明
modify-attr	

名前を変更する

アプリケーションでオブジェクトの名前が変更されたことを知らせる、PublicationShim から Identity Manager へのイベント通知として使用されます。**関連付け**アプリケーションでオブジェクトの名前を変更するための、Identity Manager から SubscriptionShim へのコマンドとしても使用されます。コマンドとして使用される場合は、名前を変更するオブジェクトの固有キーである < **関連付け** > が必要です。

備考

<rename> には、新しい名前を指定する <new-name> が含まれています。

<rename> への応答は、< **名前を変更する** > が正常に処理されたかどうかを示す <status> である必要があります。

例

```
<rename class-name="User" src-dn="\Samuel" old-src-dn="\Sam">  
<association>1012</association>          <new-name>Samuel</new-name></  
rename>
```

内容に使用できる要素

要素	説明
関連付け	
new-name	
operation-data	

属性

属性	有効値	デフォルト値
	CDATA	#IMPLIED
	予約済み。	
	CDATA	#IMPLIED
	CDATA	#IMPLIED
	CDATA	#IMPLIED

属性	有効値	デフォルト値
	CDATA	#IMPLIED
remove-old-name	古い名前を削除する場合は true 、 削除しない場合は false です。 通常は、オブジェクトの名前が そのオブジェクトの独立して存 在可能な属性でもある、 X.500 タ イプアプリケーションでのみ使 用されます。	true
src-dn	CDATA	#IMPLIED
src-entry-id	CDATA	#IMPLIED
タイムスタンプ	予約済み。 CDATA 予約済み。	#IMPLIED

内容の規則

(association ? , new-name , operation-data ?)

親要素

要素	説明
input	

required-attr

Identity manager 2.0 から廃止されました。代わりに DirXMLScript `<do-veto-if-op-attr-not-available>` または `<do-set-default-attr-value>` を使用してください。

備考

`<required-attr>` を使用して、格納元である `<create-rule>` の条件の一部としてオブジェクトを作成するために必要な属性を指定します。

`<required-attr>` には、1 つ以上の `<値>` 要素を含めることができます。これに値要素が含まれている場合に、`<追加>` イベントでその属性が指定されなかった場合は、これらの値要素はデフォルト値として使用されます。デフォルト値が指定されていない場合は、`<追加>` イベントは、`<required-attr>` の `attr-name` に対応する `<add-attr>` を含んでいない場合は拒否されます。

例

`<create-rules>` を参照してください。

内容に使用できる要素

要素	説明
<code>値</code>	

属性

属性	有効値	デフォルト値
<code>attr-name</code>	CDATA	#REQUIRED
<code>write-back</code>		false
	デフォルト値もソースオブジェクトに書き戻す必要がある場合は <code>true</code> に設定します。	

内容の規則

(値)*

親要素

要素	説明
create-rule	

スキーマ定義

query-schema コマンドおよび DriverShim.getSchema() メソッドへの応答の一部として使用されます。これにより、アプリケーションまたは eDirectory のスキーマを指定します。これは、DirXML-Driver オブジェクトの DirXML-ApplicationSchema 属性に保存されている XML のトップレベル要素でもあります。

備考

アプリケーションの識別名フォーマットは、dn-format または dn-delims のどちらかによって指定されており、ソースアプリケーションで実際に使用されているフォーマットに一致する必要があります。

ほとんどのアプリケーションではいずれかの標準 DN フォーマットで十分ですが、場合によってはカスタム区切り記号セットを指定する必要があります。区切り記号セットを構成する 8 文字は、次のように定義されます。

- ◆
- ◆ Unicode マップなし文字ブールフラグ:0 は、マップできない Unicode 文字をエスケープ文字付きの 16 進数文字列 (FEFF など) として出力も解釈もしないことを意味します。0xfeff、0xfffe、0xffffd、および 0xffff の各 Unicode 文字は、eDirectory では使用できません。
- ◆ 相対 RDN 区切り記号
- ◆ RDN 区切り記号
- ◆ 名前ディバイダ
- ◆ 名前の値の区切り記号
- ◆ ワイルドカード文字
- ◆ エスケープ文字

extra RDN 区切り記号と相対 RDN 区切り記号が同じ文字の場合は、名前の向きは右から左になり、異なる文字の場合は左から右になります。

区切り記号セットが 8 文字を超える場合、超過した文字はすべてエスケープ処理が必要な文字と見なされますが、Identity Manager 内ではそれ以外の特別な意味はありません。

例

```
<schema-def hierarchical="true">  <class-def class-  
name="Organization" container="true">    <attr-def attr-name="Name"  
case-sensitive="false" multi-valued="false"naming="true" read-  
only="false" required="false" type="string"/>    <attr-def attr-  
name="Object Path" case-sensitive="false"multi-valued="false"  
naming="false" read-only="false" required="true" type="string"/>  
<attr-def attr-name="Unique Id" case-sensitive="false"multi-  
valued="false" naming="false" read-only="false"  
required="true" type="string"/>  </class-def>  <class-def class-  
name="Organizational Unit" container="true">    <attr-def attr-  
name="Name" case-sensitive="false" multi-valued="false"naming="true"  
read-only="false" required="false" type="string"/>    <attr-def  
attr-name="Object Path" case-sensitive="false"multi-valued="false"
```

```

naming="false" read-only="false" required="true" type="string"/>
<attr-def attr-name="Unique Id" case-sensitive="false" multi-
valued="false" naming="false" read-only="false"
required="true" type="string"/> </class-def> <class-def class-
name="User" container="false"> <attr-def attr-name="cn" case-
sensitive="false" multi-valued="false" naming="true" read-only="false"
required="true" type="string"/> <attr-def attr-name="Surname"
case-sensitive="false" multi-valued="false" naming="false" read-
only="false" required="false" type="string"/> <attr-def attr-
name="Given Name" case-sensitive="false" multi-valued="false"
naming="false" read-only="false" required="false" type="string"/>
<attr-def attr-name="Telephone Number" case-sensitive="false" multi-
valued="true" naming="false" read-only="false"
required="false" type="string"/> <attr-def attr-name="Object Path"
case-sensitive="false" multi-valued="false" naming="false" read-
only="false" required="true" type="string"/> <attr-def attr-
name="Unique Id" case-sensitive="false" multi-valued="false"
naming="false" read-only="false" required="true" type="string"/> </
class-def> <class-def class-name="Bogus" container="false">
<attr-def attr-name="Whatever" case-sensitive="false" multi-
valued="true" naming="true" read-only="false"
required="false" type="string"/> <attr-def attr-name="Object Path"
case-sensitive="false" multi-valued="false" naming="false" read-
only="false" required="true" type="string"/> <attr-def attr-
name="Unique Id" case-sensitive="false" multi-valued="false"
naming="false" read-only="false" required="true" type="string"/> </
class-def></schema-def>

```

内容に使用できる要素

要素	説明
<code>class-def</code>	
<code>operation-data</code>	

属性

属性	有効値	デフォルト値
<code>application-name</code>	CDATA スキーマを使用するアプリケーションの名前を指定します。	#IMPLIED
<code>dn-delims</code>	CDATA アプリケーションで使用されるカスタム DN フォーマット。	#IMPLIED

属性	有効値	デフォルト値
dn-format	アプリケーションで使用される DN フォーマット。	slash
hierarchical	アプリケーションがデータを階 層構造で保存するかどうかを指 定します。	true

内容の規則

((class-def) * , operation-data ?)

親要素

要素	説明
output	

search-attr

格納元の<クエリー>コマンドによって検索されるオブジェクト属性を指定するために使用されます。各<search-attr>には1つ以上の<値>が含まれている必要があります。

例

<クエリー>を参照してください。

内容に使用できる要素

要素	説明
値	

属性

属性	有効値	デフォルト値
attr-name	CDATA	#REQUIRED

マッピングポリシーでは、格納元のコマンドまたはイベントのクラス名属性を使用して、属性名をマップするために使用するクラスを決定します。

内容の規則

(値)+

親要素

要素	説明
クエリー	
query-ex	

search-class

格納元の<クエリー>コマンドによって検索されるオブジェクトベースクラスを指定するために使用されます。

例

<クエリー>を参照してください。

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
class-name	CDATA	#REQUIRED

内容の宣言

なし

親要素

要素	説明
クエリー	
query-ex	

server

<authentication-info> 内のドライバ用のアプリケーションサーバを指定するために使用されます。これは、DirXML-Driver オブジェクトの DirXML-ShimAuthServer 属性から取得されます。

例

<パラメータの開始> を参照してください。

内容に使用できる要素

#PCDATA

属性

なし

内容の規則

(#PCDATA)

親要素

要素	説明
authentication-info	

source

格納元ドキュメントのソースを指定するために使用されます。

例

<nds> を参照してください。

内容に使用できる要素

要素	説明
product	このドキュメントの元となった製品。
contact	

属性

なし

内容の規則

(product ? , contact ?)

親要素

要素	説明
nds	すべての Identity Manager 通信およびドライバ通信のトップレベル要素。

status

操作またはイベントのステータスを返すために使用されます。操作またはイベントごとに、複数の <status> を返すことができます。ドライバのステータスを Identity Manager のログに記録する必要がある場合は、ドライバから Identity Manager への入力に <status> を含めることもできます。

備考

level 属性の可能な値は次のとおりです。

値	説明
成功	操作またはイベントは正常に実行されました。
警告	操作またはイベントは部分的に正常に実行されました。
エラー	操作またはイベントを実行できませんでした。
致命的	致命的なエラーが発生しました。ドライバをシャットダウンする必要があります。
再試行	アプリケーションサーバを使用できませんでした。このイベントまたは操作を後で送信してください。

type 属性の標準の値は次のとおりです。

値	説明
app-general	アプリケーション API からの一般的な応答。
app-authentication	アプリケーション認証要求に関連します。
app-connection	アプリケーションの可用性の変化を識別します。
driver-general	ドライバによって生成された応答を識別します。
driver-status	ドライバ状態の変化が発生しました。
password-set-operation	アプリケーションパスワードの設定イベントまたは変更イベントへの応答を識別します。
remoteloader	Identity Manager リモートローダによって生成されたメッセージ。

特定のイベントまたはコマンドについて <status> が返されない場合は、そのイベントまたはコマンドは正常に実行されたと見なされます。

例

```
<status event-id="0" level="success"/><status event-id="0" level="warning">Objects in the rear view mirror may appear closer than they are!</status><status event-id="0" level="warning" type="driver-status">Driver state changed to Stopped.</status>
```


内容に使用できる要素

ANY

属性

属性	有効値	デフォルト値
	CDATA	#IMPLIED
	このステータスが関連付けられている操作またはイベントの event-id と同じである必要があります。	
level	fatal error warning success retry	#REQUIRED
	ステータスレベル。	
type	CDATA	#IMPLIED
	レポート目的のために <status> を分類するための識別子。	

内容の規則

ANY

親要素

要素	説明
input	
output	

subscriber-options

driver-config

例

<driver-config> を参照してください。

内容に使用できる要素

ANY

属性

なし

内容の規則

ANY

親要素

要素	説明
driver-config	
パラメータの開始	

subscriber-state

SubscriptionShim のドライバ固有の状態情報を指定するために使用されます。

備考

<subscriber-state> は、ドライバから Identity Manager への任意の <output> または <input> 内の <パラメータの開始> に含めることができ、これにより、eDirectory 内の DirXML-Driver オブジェクトの DirXML-DriverStorage 属性にコンテンツを保存するように Identity Manager に指示されます。

DirXML-Driver オブジェクトの DirXML-DriverStorage に保存されている <subscriber-state> は、ドライバの起動時に SubscriptionShim.init() に送信された <パラメータの開始> に含まれています。

例

<パラメータの開始> を参照してください。

内容に使用できる要素

ANY

属性

なし

内容の規則

ANY

親要素

要素	説明
パラメータの開始	

同期

再同期 (手動または自動) の結果として、または「識別ボールドからの移行」操作の結果として購読者チャンネルに表示されます。

備考

<sync> は一般に、イベント変換ポリシーでのみ表示されます。イベント変換ポリシーを使用してスコープベースのフィルタリングを実行する場合は、<sync> 要素を考慮に入れる必要があります。

<sync> 要素は、アプリケーションシムによって投入されるか、発行者チャンネル上のポリシーによって投入されない限り、発行者チャンネルには表示されません。ただし、「識別ボールドへの移行」操作の場合は、<sync> 要素は発行者チャンネルポリシーによって処理されます。

例

```
<sync class-name="User" src-entry-id="3458909"qualified-src-dn="\T=TREE\O=container\CN=object"src-dn="\TREE\container\object">
<association>67847262</association>          </sync>
```

内容に使用できる要素

要素	説明
関連付け	
operation-data	

属性

属性	有効値	デフォルト値
class-id	CDATA 予約済み。	#IMPLIED
class-name	CDATA	#IMPLIED
	CDATA	#IMPLIED
src-dn	CDATA	#IMPLIED

属性	有効値	デフォルト値
src-entry-id	CDATA	#IMPLIED
タイムスタンプ	CDATA 予約済み。	#IMPLIED

内容の規則

)

親要素

要素	説明
input	

template

Identity Manager 2.0 から廃止されました。代わりに DirXMLScript `<do-set-op-template-dn>` を使用してください。格納元の `<create-rule>` で指定された条件の一部としてオブジェクト作成のために使用されるテンプレートを指定するために使用されます。

例

`<create-rules>` を参照してください。

内容に使用できる要素

なし

属性

属性	有効値
	CDATA
	ターゲットネームスペースで使用されるテンプレートの DN。

内容の規則

なし

親要素

要素	説明
<code>create-rule</code>	

user

<authentication-info> 内のドライバ用のアプリケーションサーバに対して認証するユーザ名を指定するために使用されます。これは、DirXML-Driver オブジェクトの DirXML-ShimAuthID 属性から取得されます。

例

<パラメータの開始> を参照してください。

内容に使用できる要素

#PCDATA

属性

なし

内容の規則

#PCDATA

親要素

要素	説明
authentication-info	

値

格納元の要素で指定された属性の個別の属性値を指定するために使用されます。

備考

タイプが `structured` の場合は、`<value>` には 1 つ以上の `<component>` が含まれます。タイプが `octet` の場合は、`<value>` のコンテンツは `base64` エンコードバイナリデータです。タイプがこれら以外の場合は、コンテンツはテキストです。

次の表では、さまざまな eDirectory™ 構文を表すために使用されるタイプを示しています。

構文	タイプ	コンポーネント (Notes)
SYN_UNKNOWN	octet (オクテット)	(Base64 エンコードデータ)
SYN_DIST_NAME	dn	(参照)
SYN_CE_STRING	文字列	
SYN_CI_STRING	文字列	
SYN_PR_STRING	文字列	
SYN_NU_STRING	文字列	
SYN_CI_LIST	構造	文字列 (1 つ以上)
SYN_BOOLEAN	state	True または False
SYN_INTEGER	int	
SYN_OCTET_STRING	octet (オクテット)	
SYN_TEL_NUMBER	teleNumber	
SYN_FAX_NUMBER	構造	faxNumber faxBitCount faxParameters(Base64 エンコードデータ)
SYN_NET_ADDRESS	構造	netAddrType netAddr(Base64 エンコードデータ)
SYN_OCTET_LIST	構造	octet(Base64 エンコードデータ)(1 つ以上)
SYN_EMAIL_ADDRESS	構造	eMailType (1=SMF70、2=SMF71、3=SMTP、4=x400、5=snads、6=profs、7=Groupwise) eMailAddr

構文	タイプ	コンポーネント (Notes)
SYN_PATH	構造	nameSpace ボリューム (参照) パス
SYN_REPLICA_POINTER	構造	サーバ (参照) replicaType replicaNumber 0 回以上繰り返される {netAddrType }
SYN_OBJECT_ACL	構造	protectedName トラスティ (参照) 特権
SYN_PO_ADDRESS	構造	文字列 (6 個)
SYN_TIMESTAMP	構造	秒 eventId
SYN_CLASS_NAME	classname	
SYN_STREAM	octet (オクテット)	
SYN_COUNTER	counter	
SYN_BACK_LINK	構造	serverDn(参照) remoteld
SYN_TIME	タイム	
SYN_TYPED_NAME	構造	dn(参照) 間隔
SYN_HOLD	構造	holdEntryDn(参照) holdAmount
SYN_INTERVAL	間隔	
SYNTAX_COUNT	count	

例

```
<value type="string">Gulia</value><value type="structured">
<component name="eMailType">3</component>          <component
name="eMailAddr">me@myself.com</component></value>
```

内容に使用できる要素

#PCDATA

要素	説明
----	----

属性

属性	有効値	デフォルト値
	CDATA	#IMPLIED
	この値が参照しているオブジェクトの関連付け値 (アプリケーションオブジェクトの固有キー)。値がドライバからの通知イベントの一部である場合は、これは他のオブジェクトを参照するすべてのコンポーネントが必要です。値が Identity Manager からのコマンドの一部である場合は、これは他のオブジェクトを参照するすべての値で存在しています。ただし、参照先のオブジェクトは eDirectory 内の確立された関連付けを保持していることが条件です。<component> が参照の場合は、 association-ref は <値> 上ではなく <component> 上にあります。	false
タイムスタンプ	予約済み。 CDATA	#IMPLIED
type	予約済み。 値のデータタイプ。	#IMPLIED

内容の規則

(#PCDATA | component) *

親要素

要素	説明
add-attr	属性を追加します。
add-value	
attr	
match-attr	イベントまたはコマンドを入力します。
remove-value	
required-attr	
search-attr	

マップ DTD

Identity Manager マッピングテーブルは、ある値セットを対応する別の値セットにマップするために DirXML スクリプトポリシーで使用されます。

Identity Manager マッピングテーブルは、トップレベルが `<mapping-table>` 要素で、この要素は `<col-def>` 要素のセットと `<row>` 要素のセットで構成されています。`<col-def>` によって、各カラムの名前と各カラムに含まれるタイプが定義されます。`<row>` は、`<col>` 要素のセットで構成されています。DirXML スクリプトでは、`<token-map>` を使用して、指定されたキーカラムの値が、指定された値カラム内の別の 1 つまたは複数の値にマップされます。指定されたキーカラムは複数行にマップされるため、値カラムでは複数の値を指定できます。

マップ DTD のすべての要素のリストについては、「[177 ページの「マップ DTD の要素」](#)」を参照してください。

4.1 マップ DTD の要素

要素	説明
<code>col</code>	行内のマッピングテーブルカラム。
<code>col-def</code>	カラムの定義。
<code>mapping-table</code>	マッピングテーブル。
<code>row</code>	マッピングテーブルの行。

col

マッピングテーブルの行内のカラムの値を定義します。

例

<mapping-table> を参照してください。

内容に使用できる要素

#PCDATA

属性

なし

内容の規則

(#PCDATA)

親要素

要素	説明
----	----

row

col-def

マッピングテーブル内のカラムを定義します。カラム名は名前指定されます。カラムタイプはタイプで指定されます。カラムタイプはデフォルトでは大文字と小文字は区別されず、そのカラムがマッピング操作の際にキーカラムとして使用される場合に比較ルールの指定に使用されます。

備考

タイプ別の有効な値:

値	説明
case (大文字と小文字の区別あり)	1文字ずつ比較する(大文字と小文字の区別あり)。
nocase (大文字と小文字の区別なし)	1文字ずつ比較する(大文字と小文字の区別なし)。
数字	数値で比較する。

例

<mapping-table> を参照してください。

内容に使用できる要素

なし

属性

属性	値	デフォルト値
名前	CDATA カラムの名前	#REQUIRED
type	case nocase numeric カラムのタイプ	nocase (大文字と小文字の区別なし)

内容の宣言

なし

親要素

要素	説明
mapping-table	

mapping-table

マッピングテーブルは、トップレベルが `<mapping-table>` 要素で、この要素は、`<col-def>` 要素のセットと `<row>` 要素のセットで構成されています。`col-def`要素は、`row`要素の `col`属性の値と、`row`要素の `col`属性の値にマップされます。指定されたキーカラムは複数行にマップされるため、値カラムでは複数の値を指定できます。

例

```
<mapping-table><col-def name="dept" type="nocase"/><col-def
name="code" type="nocase"/><col-def name="location" type="nocase"/>
<row> <col>Engineering</col><col>00001</col><col>New York</col></row>
<row> <col>Sales</col><col>00002</col><col>London</col></row> <row>
<col>Accounting</col><col>00003</col><col>Paris</col></row> <row>
<col>Marketing</col><col>00004</col><col>Rome</col></row></mapping-
table>
```

内容に使用できる要素

要素	説明
<code>col-def</code>	
<code>row</code>	

属性

なし

内容の規則

(`col-def` * , `row` *)

親要素

なし

row

<row>によって、マッピングテーブルの行が定義されます。行内のカラムの値は、格納している <col> 要素によって定義され、マッピングテーブルに定義されているカラムに対応します。また、これらの値の順序は、<col-def> 要素の順序と同じです。存在するカラムの数が、テーブルに定義されているカラムの数より少ない場合、存在しないカラムは空白と見なされます。行内のカラム数がテーブルに定義されている数より多い場合、過剰分のカラムは無視されます。

例

<mapping-table> を参照してください。

内容に使用できる要素

要素	説明
col	

属性

なし

内容の規則

(col *)

親要素

要素	説明
mapping-table	

DirXML スクリプト DTD

5

DirXML スクリプトは、Novell Identity Manager メタディレクトリエンジンにポリシーを実装するための主要な方法です。DirXML スクリプトでは、`<policy>` を記述し、順序付けた一連の `<rule>` 要素によって実装します。`<rule>` にはテストする一連の `<conditions>` と、`<conditions>` が満たされたときに順次実行される一連の `<actions>` が含まれています。

DirXML スクリプト DTD の全要素のリストについては、「[183 ページの「DirXML スクリプト DTD の要素」](#)」を参照してください。

5.1 DirXML スクリプト DTD の要素

要素	説明
<code>actions</code>	<code><rule></code> によって実行されるアクションです。
<code>and</code>	論理積です。
<code>arg-actions</code>	アクションの引数です。
<code>arg-association</code>	関連付けの引数です。
<code>arg-component</code>	コンポーネントの引数です。
<code>arg-conditions</code>	条件の引数です。
<code>arg-dn</code>	DN の引数です。
<code>arg-match-attr</code>	一致属性の引数です。
<code>arg-node-set</code>	ノードセットの引数です。
<code>arg-object</code>	Java* オブジェクトの引数です。
<code>arg-password</code>	パスワードの引数です。
<code>arg-string</code>	文字列の引数です。
<code>arg-value</code>	値の引数です。
<code>comment</code>	ルールの詳細説明です。 値コンポーネントです。
<code>conditions</code>	ルールのアクションを実行する条件です。
<code>description</code>	ポリシーまたはルールの説明です。
<code>do-add-association</code>	現在のオブジェクトに関連付けます。
<code>do-add-dest-attr-value</code>	ターゲットデータストア内の属性に値を追加します。
<code>do-add-dest-object</code>	ターゲットデータストア内にオブジェクトを追加します。
<code>do-add-src-attr-value</code>	ソースデータストア内の属性に値を追加します。

要素	説明
<code>do-add-src-object</code>	ソースデータストア内にオブジェクトを追加します。
<code>do-append-xml-element</code>	既存の要素にカスタム XML 要素を追加します。
<code>do-append-xml-text</code>	既存の要素にカスタム XML テキストを追加します。
<code>do-break</code>	このポリシーに対する現在の操作の処理を停止します。
<code>do-clear-dest-attr-value</code>	ターゲットデータストア内の属性のすべての値をクリアします。
<code>do-clear-op-property</code>	操作プロパティをクリアします。
<code>do-clear-src-attr-value</code>	ソースデータストア内の属性のすべての値をクリアします。
<code>do-clear-ssso-credential</code>	SSO 資格情報ストア内の資格情報をクリアします。
<code>do-clone-op-attr</code>	現在の操作の属性に対するすべての操作を別の属性に適用します。
<code>do-clone-xpath</code>	一連のノードのクローンを作り、既存の要素に追加します。
<code>do-delete-dest-object</code>	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。
<code>do-delete-src-object</code>	ソースデータストア内のオブジェクトを削除します。
<code>do-find-matching-object</code>	現在のオブジェクトに自動的に関連付けます。
<code>do-for-each</code>	ノードセット内の各ノードに対し、アクションを繰り返します。
<code>do-generate-event</code>	ユーザ定義イベントを生成します。
<code>do-if</code>	条件付でアクションを実行します。
<code>do-implement-entitlement</code>	エンタイトルメントを実装します。
<code>do-move-dest-object</code>	ターゲットデータストア内のオブジェクトを移動します。
<code>do-move-src-object</code>	ソースデータストア内のオブジェクトを移動します。
<code>do-reformat-op-attr</code>	現在の操作に含まれる特定の属性について、すべての値の形式を変更します。
<code>do-remove-association</code>	アプリケーションオブジェクトの関連付けを解除します。
<code>do-remove-dest-attr-value</code>	ターゲットデータストア内の属性から値を削除します。
<code>do-remove-src-attr-value</code>	ソースデータストア内の属性から値を削除します。

要素	説明
<code>do-rename-dest-object</code>	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。
<code>do-rename-op-attr</code>	現在の操作の属性について、すべての操作の属性名を変更します。
<code>do-rename-src-object</code>	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。
<code>do-send-email</code>	電子メール通知を生成します。
<code>do-send-email-from-template</code>	SMTP の設定と電子メールテンプレートオブジェクトを使用して電子メール通知を生成します。
<code>do-set-default-attr-value</code>	ターゲットデータストア内で作成された属性にデフォルト値を設定します。
<code>do-set-dest-attr-value</code>	ターゲットデータストア内の属性の値を設定します。
<code>do-set-dest-password</code>	ターゲットデータストアにある現在のオブジェクトのパスワードを設定します。
<code>do-set-local-variable</code>	ローカル変数の値を設定します。
<code>do-set-op-association</code>	現在の操作の関連付けの値を設定します。
<code>do-set-op-class-name</code>	現在の操作のオブジェクトクラス名を設定します。
<code>do-set-op-dest-dn</code>	現在の操作のターゲット DN を設定します。
<code>do-set-op-property</code>	操作プロパティを設定します。
<code>do-set-op-src-dn</code>	現在の操作のソース DN を設定します。
<code>do-set-op-template-dn</code>	現在の操作のテンプレート DN を設定します。
<code>do-set-src-attr-value</code>	ソースデータストア内の属性の値を設定します。
<code>do-set-src-password</code>	ソースデータストアにある現在のオブジェクトのパスワードを設定します。
<code>do-set-ssso-credential</code>	SSO 資格情報ストアに資格情報を設定します。
<code>do-set-ssso-passphrase</code>	SSO 資格情報ストアにパスフレーズを設定します。
<code>do-set-xml-attr</code>	既存の要素にカスタム XML 属性を設定します。
<code>do-start-workflow</code>	ワークフローを開始します。
<code>do-status</code>	ステータスをレポートします。
<code>do-strip-op-attr</code>	現在の操作から属性を除去します。
<code>do-strip-xpath</code>	現在の操作から任意のデータを除去します。
<code>do-trace-message</code>	トレースメッセージを送信します。
<code>do-veto</code>	現在の操作を拒否します。

要素	説明
<code>do-veto-if-op-attr-not-available</code>	現在の操作で特定の属性を使用できない場合は、現在の操作を拒否します。
<code>do-while</code>	条件が true の間、アクションを繰り返します。
<code>if-association</code>	関連付けをテストします。
<code>if-attr</code>	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をテストします。
<code>if-class-name</code>	現在の操作のオブジェクトクラスをテストします。
<code>if-dest-attr</code>	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をテストします。
<code>if-dest-dn</code>	現在の操作のターゲット DN をテストします。
<code>if-entitlement</code>	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。
<code>if-global-variable</code>	グローバル変数をテストします。
<code>if-local-variable</code>	ローカル変数をテストします。
<code>if-named-password</code>	指定されたパスワードをテストします。
<code>if-op-attr</code>	現在の操作の属性をテストします。
<code>if-op-property</code>	操作プロパティをテストします。
<code>if-operation</code>	現在の操作の名前をテストします。
<code>if-password</code>	現在の操作のパスワードをテストします。
<code>if-src-attr</code>	ソースデータストアにある現在オブジェクトの属性をテストします。
<code>if-src-dn</code>	現在の操作のソース DN をテストします。
<code>if-xml-attr</code>	現在の操作の XML 属性をテストします。
<code>if-xpath</code>	XPath 式をテストします。
<code>include</code>	他のポリシーからルールをインクルードします。
<code>または</code>	論理和です。
<code>policy</code>	ポリシーです。
<code>rule</code>	ポリシー内のルールです。
<code>token-added-entitlement</code>	現在の操作で許可されたエンタイトルメントの値です。
<code>token-association</code>	現在の操作からの関連付けの値です。
<code>token-attr</code>	現在の操作またはソースデータストア内の現在のオブジェクトに含まれる属性の値です。
<code>token-base64-decode</code>	Base64 データを文字列にデコードします。
<code>token-base64-encode</code>	文字列を Base64 データにエンコードします。

要素	説明
token-char	Unicode 文字です。
token-class-name	現在の操作からのオブジェクトクラス名です。
token-convert-time	日付 / 時刻の形式を別の形式に変換します。
token-dest-attr	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です。
token-dest-dn	現在の操作のターゲット DN から派生した値です。
token-dest-name	現在の操作のターゲット DN から派生した非修飾の RDN です。
token-document	XML ドキュメントを読み込みます。
token-entitlement	現在のオブジェクトに許可されたエンタイトルメントの値です。
token-escape-for-dest-dn	文字列をターゲット DN で使用できるように変換します。
token-escape-for-src-dn	文字列をソース DN で使用できるように変換します。
token-generate-password	ランダムなパスワードを生成します。
token-global-variable	グローバル変数の値です。
token-join	文字列にノードセットを結合します。
token-local-variable	ローカル変数の値です。
token-lower-case	文字列を小文字に変換します。
token-map	マッピングテーブルに従って文字列をマップします。
token-named-password	指定されたパスワードの値です。
token-op-attr	現在の操作に含まれる属性の値です。
token-op-property	操作のプロパティの値です。
token-operation	現在の操作の名前です。
token-parse-dn	DN を解析または変換します。
token-password	現在の操作のパスワード値です。
token-query	ソースまたはターゲットデータストアを問い合わせます。
token-removed-attr	現在の操作で削除された属性の値です。
token-removed-entitlement	現在の操作で取り消されたエンタイトルメントの値です。
token-replace-all	文字列内の一致するすべての部分文字列を置換します。

要素	説明
<code>token-replace-first</code>	文字列内の一致する部分文字列を 1 つ置換します。
<code>token-resolve</code>	DN から関連付けキー、または関連付けキーから DN を解決します。
<code>token-split</code>	文字列をノードセットに分割します。
<code>token-src-attr</code>	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です。
<code>token-src-dn</code>	現在の操作のソース DN から派生した値です。
<code>token-src-name</code>	現在の操作のソース DN から派生した非修飾の RDN です。
<code>token-substring</code>	文字列の部分文字列です。
<code>token-text</code>	定数のテキストです。
<code>token-time</code>	現在の日付 / 時間です。
<code>token-unique-name</code>	固有の名前を生成します。
<code>token-unmatched-src-dn</code>	<code>if-src-dn</code> に一致した DN に関連する DN です。
<code>token-upper-case</code>	文字列を大文字に変換します。
<code>token-xml-parse</code>	XML を解析します。
<code>token-xml-serialize</code>	XML をシリアル化します。
<code>token-xpath</code>	XPath 式の結果です。

actions

<actions> は、取り囲んでいる <rule> の <conditions> が成立したときに実行されます。個々の <action> は、すべて <do-*> の形式の要素で表現します。

備考

ほとんどの actions は、実行するアクションの詳細を記述する引数を取ります。ランタイムに変更されない固定の文字列を取る引数は、アクションの要素の属性によって表現します。実行時に再評価する引数は、<arg-*> の形式の子要素で示します。これらのほとんどの引数の内容には、<token-*> の形式の要素で記述された一連のトークンが含まれます（例外については個々の引数のマニュアルに記載されています）。個々のトークンは、ルールの評価コンテキストと、展開の結果を基にランタイムで展開され、実際の引数を生成するために結合されます。

例

<policy> を参照してください。

内容に使用できる要素

要素	説明
do-add-association	
do-add-dest-attr-value	
do-add-dest-object	
do-add-src-attr-value	
do-add-src-object	
do-append-xml-element	
do-append-xml-text	
do-break	
do-clear-dest-attr-value	
do-clear-op-property	
do-clear-src-attr-value	
do-clear-sso-credential	
do-clone-op-attr	
do-clone-xpath	
do-delete-dest-object	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。
do-delete-src-object	ソースデータストア内のオブジェクトを削除します。

要素	説明
do-find-matching-object	
do-for-each	
do-generate-event	
do-if	
do-implement-entitlement	
do-move-dest-object	ターゲットデータストア内のオブジェクトを移動します。
do-move-src-object	ソースデータストア内のオブジェクトを移動します。
do-reformat-op-attr	
do-remove-association	
do-remove-dest-attr-value	
do-remove-src-attr-value	
do-rename-dest-object	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。
do-rename-op-attr	
do-rename-src-object	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。
do-send-email	
do-send-email-from-template	
do-set-default-attr-value	
do-set-dest-attr-value	
do-set-dest-password	ターゲットデータストアにある現在のオブジェクトのパスワードを設定します。
do-set-local-variable	
do-set-op-association	
do-set-op-class-name	現在の操作のオブジェクトクラス名を設定します。
do-set-op-dest-dn	現在の操作のターゲット DN を設定します。
do-set-op-property	操作プロパティを設定します。
do-set-op-src-dn	現在の操作のソース DN を設定します。
do-set-op-template-dn	
do-set-src-attr-value	
do-set-src-password	ソースデータストアにある現在のオブジェクトのパスワードを設定します。

要素	説明
do-set-sso-credential	
do-set-sso-passphrase	
do-set-xml-attr	
do-start-workflow	
do-status	
do-strip-op-attr	
do-strip-xpath	
do-trace-message	
do-veto	現在の操作を拒否します。
do-veto-if-op-attr-not-available	
do-while	

属性

なし

内容の規則

([do-add-association](#) | [do-add-dest-attr-value](#) | [do-add-dest-object](#) | [do-add-src-attr-value](#) | [do-add-src-object](#) | [do-append-xml-element](#) | [do-append-xml-text](#) | [do-break](#) | [do-clear-dest-attr-value](#) | [do-clear-op-property](#) | [do-clear-src-attr-value](#) | [do-clear-sso-credential](#) | [do-clone-op-attr](#) | [do-clone-xpath](#) | [do-delete-dest-object](#) | [do-delete-src-object](#) | [do-find-matching-object](#) | [do-for-each](#) | [do-generate-event](#) | [do-if](#) | [do-implement-entitlement](#) | [do-move-dest-object](#) | [do-move-src-object](#) | [do-reformat-op-attr](#) | [do-remove-association](#) | [do-remove-dest-attr-value](#) | [do-remove-src-attr-value](#) | [do-rename-dest-object](#) | [do-rename-op-attr](#) | [do-rename-src-object](#) | [do-send-email](#) | [do-send-email-from-template](#) | [do-set-default-attr-value](#) | [do-set-dest-attr-value](#) | [do-set-dest-password](#) | [do-set-local-variable](#) | [do-set-op-association](#) | [do-set-op-class-name](#) | [do-set-op-dest-dn](#) | [do-set-op-property](#) | [do-set-op-src-dn](#) | [do-set-op-template-dn](#) | [do-set-src-attr-value](#) | [do-set-src-password](#) | [do-set-sso-credential](#) | [do-set-sso-passphrase](#) | [do-set-xml-attr](#) | [do-start-workflow](#) | [do-status](#) | [do-strip-op-attr](#) | [do-strip-xpath](#) | [do-trace-message](#) | [do-veto](#) | [do-veto-if-op-attr-not-available](#) | [do-while](#)) *

親要素

要素	説明
rule	ポリシー内のルールです。

and

テストを実行し、すべての結果が論理積で評価される一連のテストを指定します。
<conditions> で囲まれた、一連の <and> 要素は論理和で評価されます。

例

<policy> を参照してください。

内容に使用できる要素

要素	説明
if-association	
if-attr	
if-class-name	
if-dest-attr	
if-dest-dn	
if-entitlement	
if-global-variable	
if-local-variable	
if-named-password	
if-op-attr	
if-op-property	
if-operation	
if-password	
if-src-attr	
if-src-dn	
if-xml-attr	
if-xpath	

属性

属性	有効値	デフォルト値
disabled	true false この要素を無効にする 場合は true を指定します。	false

属性	有効値	デフォルト値
notrace	true false この要素をポリシーの実行時に トレースする必要がない場合は true を指定します。	false

内容の規則

(if-association | if-attr | if-class-name | if-dest-attr | if-dest-dn | if-entitlement | if-global-variable | if-local-variable | if-named-password | if-op-attr | if-op-property | if-operation | if-password | if-src-attr | if-src-dn | if-xml-attr | if-xpath) *

親要素

要素	説明
arg-conditions	
conditions	<rule> のアクションを実行する条件です。

arg-actions

取り囲んでいる `<do-for-each>` によって繰り返し実行されるアクションを指定します。他の引数の型と異なり、トークンではなくアクションを含みます。

例

`<do-for-each>`、`<do-if>`、`<do-while>`、`<do-implement-entitlement>` を参照してください。

内容に使用できる要素

要素	説明
<code>do-add-association</code>	
<code>do-add-dest-attr-value</code>	
<code>do-add-dest-object</code>	
<code>do-add-src-attr-value</code>	
<code>do-add-src-object</code>	
<code>do-append-xml-element</code>	
<code>do-append-xml-text</code>	
<code>do-break</code>	
<code>do-clear-dest-attr-value</code>	
<code>do-clear-op-property</code>	
<code>do-clear-src-attr-value</code>	
<code>do-clear-sso-credential</code>	
<code>do-clone-op-attr</code>	
<code>do-clone-xpath</code>	
<code>do-delete-dest-object</code>	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。
<code>do-delete-src-object</code>	ソースデータストア内のオブジェクトを削除します。
<code>do-find-matching-object</code>	
<code>do-for-each</code>	
<code>do-generate-event</code>	
<code>do-if</code>	
<code>do-implement-entitlement</code>	
<code>do-move-dest-object</code>	ターゲットデータストア内のオブジェクトを移動します。

要素	説明
do-move-src-object	ソースデータストア内のオブジェクトを移動します。
do-reformat-op-attr	
do-remove-association	
do-remove-dest-attr-value	
do-remove-src-attr-value	
do-rename-dest-object	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。
do-rename-op-attr	
do-rename-src-object	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。
do-send-email	
do-send-email-from-template	
do-set-dest-attr-value	
do-set-dest-attr-value	
do-set-dest-password	ターゲットデータストアにある現在のオブジェクトのパスワードを設定します。
do-set-local-variable	
do-set-op-association	
do-set-op-class-name	現在の操作のオブジェクトクラス名を設定します。
do-set-op-dest-dn	現在の操作のターゲット DN を設定します。
do-set-op-property	操作プロパティを設定します。
do-set-op-src-dn	現在の操作のソース DN を設定します。
do-set-op-template-dn	
do-set-src-attr-value	
do-set-src-password	ソースデータストアにある現在のオブジェクトのパスワードを設定します。
do-set-ssso-credential	
do-set-ssso-passphrase	
do-set-xml-attr	
do-start-workflow	
do-status	
do-strip-op-attr	
do-strip-xpath	

要素	説明
<code>do-trace-message</code>	
<code>do-veto</code>	現在の操作を拒否します。
<code>do-veto-if-op-attr-not-available</code>	
<code>do-while</code>	

属性

なし

内容の規則

親要素

要素	説明
<code>do-for-each</code>	
<code>do-if</code>	
<code>do-implement-entitlement</code>	
<code>do-while</code>	条件が <code>true</code> の間、アクションを繰り返します。

arg-association

取り囲んでいるアクションの関連付けの値を指定します。内容のトークンのそれぞれが評価され、結果の文字列値が結合されて関連付けの値になります。

例

`<do-add-association>` を参照してください。

内容に使用できる要素

要素	説明
<code>token-added-entitlement</code>	
<code>token-association</code>	
<code>token-attr</code>	
<code>token-base64-decode</code>	
<code>token-base64-encode</code>	
<code>token-char</code>	
<code>token-class-name</code>	
<code>token-convert-time</code>	
<code>token-dest-attr</code>	
<code>token-dest-dn</code>	
<code>token-dest-name</code>	
<code>token-document</code>	
<code>token-entitlement</code>	
<code>token-escape-for-dest-dn</code>	
<code>token-escape-for-src-dn</code>	
<code>token-generate-password</code>	
<code>token-global-variable</code>	
<code>token-join</code>	
<code>token-local-variable</code>	
<code>token-lower-case</code>	
<code>token-map</code>	
<code>token-named-password</code>	
<code>token-op-attr</code>	
<code>token-op-property</code>	

要素	説明
<code>token-operation</code>	現在の操作の名前です。
<code>token-parse-dn</code>	
<code>token-password</code>	現在の操作のパスワード値です。
<code>token-query</code>	
<code>token-removed-attr</code>	
<code>token-removed-entitlement</code>	
<code>token-replace-all</code>	
<code>token-replace-first</code>	
<code>token-resolve</code>	
<code>token-split</code>	
<code>token-src-attr</code>	
<code>token-src-dn</code>	
<code>token-src-name</code>	
<code>token-substring</code>	文字列の部分文字列です。
<code>token-text</code>	
<code>token-time</code>	
<code>token-unique-name</code>	
<code>token-unmatched-src-dn</code>	<code>if-src-dn</code>
<code>token-upper-case</code>	
<code>token-xml-parse</code>	
<code>token-xml-serialize</code>	
<code>token-xpath</code>	

属性

なし

内容の規則

(`token-added-entitlement` | `token-association` | `token-attr` | `token-base64-decode` | `token-base64-encode` | `token-char` | `token-class-name` | `token-convert-time` | `token-dest-attr` | `token-dest-dn` | `token-dest-name` | `token-document` | `token-entitlement` | `token-escape-for-dest-dn` | `token-escape-for-src-dn` | `token-global-variable` | `token-join` | `token-local-variable` | `token-lower-case` | `token-named-password` | `token-map` | `token-op-attr` | `token-op-property` | `token-operation` | `token-parse-dn` | `token-password` | `token-removed-attr` | `token-removed-entitlement` | `token-replace-all` | `token-replace-first` | `token-resolve` | `token-src-attr` | `token-src-dn` | `token-src-name` | `token-substring` | `token-text` | `token-time` | `token-unique-name` | `token-unmatched-src-dn` | `token-upper-case` | `token-xml-parse` | `token-xml-serialize` | `token-xpath`) *

親要素

要素	説明
do-add-association	
do-add-dest-attr-value	
do-add-src-attr-value	
do-clear-dest-attr-value	
do-clear-src-attr-value	
do-delete-dest-object	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。
do-delete-src-object	ソースデータストア内のオブジェクトを削除します。
do-move-dest-object	ターゲットデータストア内のオブジェクトを移動します。
do-move-src-object	ソースデータストア内のオブジェクトを移動します。
do-remove-association	
do-remove-dest-attr-value	
do-remove-src-attr-value	
do-rename-dest-object	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。
do-rename-src-object	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。
do-set-dest-attr-value	
do-set-dest-password	ターゲットデータストアにある現在のオブジェクトのパスワードを設定します。
do-set-op-association	
do-set-src-attr-value	
do-set-src-password	ソースデータストアにある現在のオブジェクトのパスワードを設定します。
token-dest-attr	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です。
token-query	
token-resolve	
token-src-attr	
token-unique-name	

arg-component

<arg-value> の type 属性が structured の場合に、取り囲んでいる <arg-value> のコンポーネントの値を指定します。内容のトークンのそれぞれが評価され、結果の文字列値が結合されてコンポーネントの値になります。コンポーネントの名前は name 属性で指定します。

例

<arg-value> を参照してください。

内容に使用できる要素

要素	説明
token-added-entitlement	
token-association	
token-attr	
token-base64-decode	
token-base64-encode	
token-char	
token-class-name	
token-convert-time	
token-dest-attr	
token-dest-dn	
token-dest-name	
token-document	
token-entitlement	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-generate-password	
token-global-variable	
token-join	
token-local-variable	
token-lower-case	
token-map	
token-named-password	
token-op-attr	

要素	説明
token-op-property	
token-operation	現在の操作の名前です。
token-parse-dn	
token-password	現在の操作のパスワード値です。
token-query	
token-removed-attr	
token-removed-entitlement	
token-replace-all	
token-replace-first	
token-resolve	
token-split	
token-src-attr	
token-src-dn	
token-src-name	
token-substring	文字列の部分文字列です。
token-text	
token-time	
token-unique-name	
token-unmatched-src-dn	if-src-dn
token-upper-case	
token-xml-parse	
token-xml-serialize	
token-xpath	

属性

属性	有効値	デフォルト値
name	CDATA コンポーネント名です。変数の 拡張がサポートされます。	#REQUIRED

内容の規則

親要素

要素	説明
arg-value	

arg-conditions

取り囲んでいるアクションに関連付ける条件を指定します。他の引数の型と異なり、トークンではなく条件を含みます。

例

<do-if>、<do-while> を参照してください。

内容に使用できる要素

要素	説明
and	論理積です。
または	論理和です。

属性

なし

内容の規則

(and * | or *)

親要素

要素	説明
do-if	
do-while	

arg-dn

取り囲んでいるアクションの DN 値を指定します。内容のトークンのそれぞれが評価され、結果の文字列値が結合されて DN 値になります。

例

<do-add-association> を参照してください。

内容に使用できる要素

要素	説明
token-added-entitlement	
token-association	
token-attr	現在の操作またはソースデータストア内の現在のオブジェクトに含まれる属性の値です。
token-base64-decode	
token-base64-encode	
token-char	
token-class-name	現在の操作からのオブジェクトクラス名です。
token-convert-time	
token-dest-attr	
token-dest-dn	現在のオブジェクトのターゲット DN から派生した値です。
token-dest-name	現在の操作のターゲット DN から派生した非修飾の RDN です。
token-document	
token-entitlement	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-generate-password	
token-global-variable	
token-join	文字列にノードセットを結合します。
token-local-variable	
token-lower-case	
token-map	
token-named-password	

要素	説明
token-op-attr	
token-op-property	
token-operation	現在の操作の名前です。
token-parse-dn	
token-password	現在の操作のパスワード値です。
token-query	ソースまたはターゲットのデータストアでクエリを実行します。
token-removed-attr	
token-removed-entitlement	
token-replace-all	
token-replace-first	
token-resolve	
token-split	
token-src-attr	
token-src-dn	
token-src-name	現在の操作のソース DN から派生した非修飾の RDN です。
token-substring	文字列の部分文字列です。
token-text	
token-time	
token-unique-name	生成された固有名です。
token-unmatched-src-dn	if-src-dn
token-upper-case	
token-xml-parse	
token-xml-serialize	
token-xpath	

属性

なし

内容の規則

親要素

要素	説明
do-add-association	
do-add-dest-attr-value	
do-add-src-attr-value	
do-clear-dest-attr-value	
do-clear-src-attr-value	
do-clear-ssso-credential	
do-delete-dest-object	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。
do-delete-src-object	ソースデータストア内のオブジェクトを削除します。
do-find-matching-object	
do-move-dest-object	ターゲットデータストア内のオブジェクトを移動します。
do-move-src-object	ソースデータストア内のオブジェクトを移動します。
do-remove-association	
do-remove-dest-attr-value	
do-remove-src-attr-value	
do-rename-dest-object	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。
do-rename-src-object	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。
do-set-dest-attr-value	
do-set-dest-password	ターゲットデータストアにある現在のオブジェクトのパスワードを設定します。
do-set-op-dest-dn	現在の操作のターゲット DN を設定します。
do-set-op-src-dn	現在の操作のソース DN を設定します。
do-set-op-template-dn	
do-set-src-attr-value	
do-set-src-password	ソースデータストアにある現在のオブジェクトのパスワードを設定します。
do-set-ssso-credential	
do-set-ssso-passphrase	
do-start-workflow	

要素	説明
----	----

token-dest-attr

token-query

token-resolve

token-src-attr

token-unique-name

arg-match-attr

取り囲んでいる `<do-find-matching-object>` で一致検出に使用する属性を指定します。name 属性は、一致に使用する属性名を指定します。内容に `<arg-value>` がある場合は、その値が一致に使用される属性値となります。ない場合は、現在の操作の使用可能な値になります。

例

`<do-find-matching-object>` を参照してください。

内容に使用できる要素

要素	説明
<code>arg-value</code>	引数値です。

属性

属性	有効値	デフォルト値
名前	CDATA	#REQUIRED
	属性の名前です。変数の拡張がサポートされます。	

内容の規則

(arg-value ?)

親要素

要素	説明
<code>do-find-matching-object</code>	
<code>token-query</code>	

arg-node-set

取り囲んでいるアクションに XPath 1.0 ノードセットを設定します。内容のトークンのそれぞれが評価され、トークンがノードセットを返す場合は、そのセットのノードが結果セットに追加されます。そうでない場合は、テキストノードが作成され、ノードセットに追加されます。

例

<do-for-each> を参照してください。

内容に使用できる要素

要素	説明
token-added-entitlement	
token-association	
token-attr	現在の操作またはソースデータストア内の現在のオブジェクトに含まれる属性の値です。
token-base64-decode	
token-base64-encode	
token-char	
token-class-name	
token-convert-time	
token-dest-attr	
token-dest-dn	
token-dest-name	
token-document	
token-entitlement	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-generate-password	
token-global-variable	
token-join	
token-local-variable	
token-lower-case	
token-map	
token-named-password	

要素	説明
token-op-attr	
token-op-property	
token-operation	現在の操作の名前です。
token-parse-dn	DN を解析および変換します。
token-password	現在の操作のパスワード値です。
token-query	
token-removed-attr	
token-removed-entitlement	
token-replace-all	
token-replace-first	
token-resolve	
token-split	
token-src-attr	
token-src-dn	
token-src-name	
token-substring	文字列の部分文字列です。
token-text	
token-time	
token-unique-name	
token-unmatched-src-dn	if-src-dn
token-upper-case	
token-xml-parse	
token-xml-serialize	
token-xpath	

属性

なし

内容の規則

(token-added-entitlement | token-association | token-attr | token-base64-decode | token-base64-encode | token-char | token-class-name | token-convert-time | token-dest-attr | token-dest-dn | token-dest-name | token-document | token-entitlement | token-escape-for-dest-dn | token-escape-for-src-dn | token-global-variable | token-join | token-local-variable | token-lower-case | token-named-password | token-map | token-op-attr | token-op-property | token-operation | token-parse-dn | token-password | token-removed-attr | token-removed-entitlement | token-replace-all | token-replace-first | token-

resolve | token-src-attr | token-src-dn | token-src-name | token-substring | token-text | token-time | token-unique-name | token-unmatched-src-dn | token-upper-case | token-xml-parse | token-xml-serialize | token-xpath | token-query | token-split) *

親要素

要素	説明
do-for-each	
do-implement-entitlement	
do-set-local-variable	

arg-object

取り囲んでいる `<do-set-local-variable>` アクションで指定されたローカル変数に、格納する Java オブジェクトを指定します。内容のトークンは、Java オブジェクトを返す式を指定する `<token-xpath>` か、Java オブジェクトをすでに含んでいる変数の `<token-local-variable>` である必要があります。

例

`<do-set-local-variable>` を参照してください。

内容に使用できる要素

要素	説明
<code>token-local-variable</code>	
<code>token-xpath</code>	

属性

なし

内容の規則

(`token-local-variable` | `token-xpath`)

親要素

要素	説明
<code>do-set-local-variable</code>	

arg-password

取り囲んでいるアクションで使用されるパスワードを指定します。内容のトークンのそれぞれが評価され、結果の文字列値が結合されて文字列値になります。

例

<do-start-workflow> を参照してください。

内容に使用できる要素

要素	説明
token-added-entitlement	
token-association	
token-attr	
token-base64-decode	Base64 データを文字列にデコードします。
token-base64-encode	
token-char	
token-class-name	
token-convert-time	日付 / 時刻の形式を別の形式に変換します。
token-dest-attr	
token-dest-dn	
token-dest-name	
token-document	
token-entitlement	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-generate-password	
token-global-variable	
token-join	
token-local-variable	
token-lower-case	文字列を小文字に変換します。
token-map	
token-named-password	
token-op-attr	
token-op-property	

要素	説明
token-operation	現在の操作の名前です。
token-parse-dn	
token-password	現在の操作のパスワード値です。
token-query	
token-removed-attr	
token-removed-entitlement	
token-replace-all	
token-replace-first	
token-resolve	
token-split	
token-src-attr	
token-src-dn	
token-src-name	
token-substring	文字列の部分文字列です。
token-text	
token-time	
token-unique-name	
token-unmatched-src-dn	if-src-dn
token-upper-case	文字列を大文字に変換します。
token-xml-parse	
token-xml-serialize	XML をシリアル化します。
token-xpath	

要素	説明
token-added-entitlement	
token-association	
token-attr	
token-base64-decode	
token-base64-encode	
token-char	
token-class-name	
token-convert-time	

要素	説明
token-dest-attr	
token-dest-dn	
token-dest-name	
token-document	
token-entitlement	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-generate-password	
token-global-variable	
token-join	
token-local-variable	
token-lower-case	
token-map	
token-named-password	
token-op-attr	
token-op-property	
token-operation	現在の操作の名前です。
token-parse-dn	
token-password	現在の操作のパスワード値です。
token-query	
token-removed-attr	
token-removed-entitlement	
token-replace-all	
token-replace-first	
token-resolve	
token-split	
token-src-attr	
token-src-dn	
token-src-name	
token-substring	文字列の部分文字列です。
token-text	
token-time	

要素	説明
token-unique-name	
token-unmatched-src-dn	if-src-dn
token-upper-case	
token-xml-parse	
token-xml-serialize	
token-xpath	

属性

なし

内容の規則

親要素

要素	説明
do-send-email	
do-send-email-from-template	
do-start-workflow	

arg-string

取り囲んでいるアクションの文字列値を指定します。

例

`<do-set-op-class-name>` を参照してください。

内容に使用できる要素

要素	説明
<code>token-added-entitlement</code>	
<code>token-association</code>	
<code>token-attr</code>	
<code>token-base64-decode</code>	
<code>token-base64-encode</code>	
<code>token-char</code>	
<code>token-class-name</code>	
<code>token-convert-time</code>	
<code>token-dest-attr</code>	
<code>token-dest-dn</code>	
<code>token-dest-name</code>	
<code>token-document</code>	
<code>token-entitlement</code>	
<code>token-escape-for-dest-dn</code>	
<code>token-escape-for-src-dn</code>	
<code>token-generate-password</code>	
<code>token-global-variable</code>	
<code>token-join</code>	
<code>token-local-variable</code>	
<code>token-lower-case</code>	
<code>token-map</code>	
<code>token-named-password</code>	
<code>token-op-attr</code>	
<code>token-op-property</code>	
<code>token-operation</code>	現在の操作の名前です。

要素	説明
token-parse-dn	
token-password	現在の操作のパスワード値です。
token-query	
token-removed-attr	
token-removed-entitlement	
token-replace-all	
token-replace-first	
token-resolve	
token-split	
token-src-attr	
token-src-dn	
token-src-name	
token-substring	文字列の部分文字列です。
token-text	
token-time	
token-unique-name	
token-unmatched-src-dn	if-src-dn
token-upper-case	
token-xml-parse	
token-xml-serialize	
token-xpath	

属性

属性	有効値	デフォルト値
name	CDATA	#IMPLIED
	引数の名前です。変数の拡張がサポートされます。	

内容の規則

親要素

要素	説明
do-append-xml-text	
do-clear-sso-credential	
do-generate-event	
do-rename-dest-object	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。
do-rename-src-object	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。
do-send-email	
do-send-email-from-template	
do-set-dest-password	ターゲットデータストアにある現在のオブジェクトのパスワードを設定します。
do-set-local-variable	
do-set-op-class-name	現在の操作のオブジェクトクラス名を設定します。
do-set-op-property	操作プロパティを設定します。
do-set-src-password	ソースデータストアにある現在のオブジェクトのパスワードを設定します。
do-set-sso-credential	
do-set-sso-passphrase	
do-set-xml-attr	
do-start-workflow	
do-status	
do-trace-message	
token-document	
token-query	
token-unique-name	

arg-value

取り囲んでいるアクションの属性値を指定します。type 属性が structured の場合、<arg-value> の内容は、一連の <arg-component> 要素である必要があります。type 属性が structured 以外の場合、内容のトークンのそれぞれが評価され、結果の文字列値が結合されて値になります。

例

```
<arg-value> <token-attr name="Surname"/> <token-text>, </token-text>
<token-attr name="Given Name"/></arg-value><arg-value
type="structured"> <arg-component name="string"> <token-text>EN</
token-text> </arg-component> <arg-component name="string"> <token-
text>JP</token-text> </arg-component></arg-value>
```

内容に使用できる要素

要素	説明
arg-component	
token-added-entitlement	
token-association	
token-attr	
token-base64-decode	
token-base64-encode	
token-char	
token-class-name	
token-convert-time	
token-dest-attr	
token-dest-dn	
token-dest-name	
token-document	
token-entitlement	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-generate-password	
token-global-variable	
token-join	
token-local-variable	

要素	説明
token-lower-case	
token-map	
token-named-password	
token-op-attr	
token-op-property	
token-operation	現在の操作の名前です。
token-parse-dn	
token-password	現在の操作のパスワード値です。
token-query	
token-removed-attr	
token-removed-entitlement	
token-replace-all	
token-replace-first	
token-resolve	
token-split	
token-src-attr	
token-src-dn	
token-src-name	
token-substring	文字列の部分文字列です。
token-text	
token-time	
token-unique-name	
token-unmatched-src-dn	if-src-dn
token-upper-case	
token-xml-parse	
token-xml-serialize	
token-xpath	

属性

属性	有効値	デフォルト値
type	CDATA 値の型です。変数の拡張がサポートされます。	文字列

内容の規則

(arg-component + | (token-added-entitlement | token-association | token-attr | token-base64-decode | token-base64-encode | token-char | token-class-name | token-convert-time | token-dest-attr | token-dest-dn | token-dest-name | token-document | token-entitlement | token-escape-for-dest-dn | token-escape-for-src-dn | token-global-variable | token-join | token-local-variable | token-lower-case | token-named-password | token-map | token-op-attr | token-op-property | token-operation | token-parse-dn | token-password | token-removed-attr | token-removed-entitlement | token-replace-all | token-replace-first | token-resolve | token-src-attr | token-src-dn | token-src-name | token-substring | token-text | token-time | token-unique-name | token-unmatched-src-dn | token-upper-case | token-xml-parse | token-xml-serialize | token-xpath) *

親要素

要素	説明
arg-match-attr	属性引数との一致を検出します。
do-add-dest-attr-value	
do-add-src-attr-value	
do-reformat-op-attr	
do-remove-dest-attr-value	
do-remove-src-attr-value	
do-set-default-attr-value	ターゲットデータストア内に作成する属性のデフォルト値を設定します。
do-set-dest-attr-value	
do-set-src-attr-value	

comment

<rule> の内容に関連する詳細説明、またはテキスト情報です。<rule> の実行には影響しません。

備考

コメントには名前があり、ルールを表示または編集するユーザインタフェースエージェントに特定の意味を持たせることができます。現在、ポリシービルダーでは、ルールごとに名前のないコメントが 1 つサポートされています。また、名前付きのコメントは、作成者、バージョン、最終変更日時など、名前ごとに 1 つサポートされています。名前付きおよび名前なしコメントの追加は可能ですが、ポリシービルダーによって無視されます。

例

<policy> を参照してください。

内容に使用できる要素

#PCDATA

属性

属性	有効値	デフォルト値
name	CDATA コメントの名前です。	#IMPLIED

内容の規則

(#PCDATA)

親要素

要素	説明
rule	

mode 属性が structured の場合、取り囲んでいる if 条件のコンポーネントの値を指定します。

例

<if-attr> を参照してください。

内容に使用できる要素

#PCDATA

属性

属性	有効値	デフォルト値
名前	CDATA	#REQUIRED

変数の拡張がサポートされます。

内容の規則

(#PCDATA)

親要素

要素	説明
if-attr	
if-dest-attr	
if-op-attr	
if-src-attr	ソースデータストアにある現在オブジェクトの属性をテストします。

conditions

取り囲んでいる `<rule>` の `<actions>` を実行する条件です。`<conditions>` は常に、論理積正規形 (CNF) または論理和正規形 (DNF) で指定します。このため、`<conditions>` の内容は、一連の `<and>` 要素 (空のままも可能) によって指定された論理積の論理和か、一連の `<または>` 要素 (空のままも可能) によって指定された論理和の論理積のいずれかになります。取り囲んでいる `<rule>` の `<actions>` は、CNF または DNF で記述された論理式が `true` に評価されるか、条件が指定されていないときにのみ実行されます。

備考

条件の評価では短絡的ロジックが使用されるため、`<conditions>` の合成されたブール値が決定可能などときには追加のテストは実行されません。

個々の条件のテストは、すべて `<if-* op= "some operator" >` の形式の要素で表現します。

一部の条件のテストには、比較のアルゴリズムを指定するモードパラメータがあります。次の表に、使用可能なモードの詳細が説明してあります。

要素	説明
case	1文字ずつ比較する (大文字と小文字の区別あり)。
nocase	1文字ずつ比較する (大文字と小文字の区別なし)。
regex	文字列全体を正規表現で比較する。デフォルトでは大文字と小文字は区別されませんが、式でエスケープを使うことで変更できます。 http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html および http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Matcher.html#matches() を参照してください。 [パターン] のオプションには <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反転することができます。
src-dn	ソースデータストアの DN の形式に対する適切なセマンティックを使用して比較します。
dest-dn	ターゲットデータストアの DN の形式に対する適切なセマンティックを使用して比較します。
numeric	数値で比較します。
octet	オクテット値 (Base64 でエンコード) で比較します。
structured	属性の構造構文の比較ルールに従って、 <code>structured</code> 属性を比較します。

例

<policy> を参照してください。

内容に使用できる要素

要素	説明
and	
または	

属性

なし

内容の規則

親要素

要素	説明
rule	

description

<rule> または <policy> に含まれる説明です。<rule> または <policy> の実行には影響しません。

例

<policy> を参照してください。

内容に使用できる要素

#PCDATA

属性

なし

内容の規則

(#PCDATA)

親要素

要素	説明
policy	
rule	

do-add-association

<add-association> コマンドを eDirectory™ へ送信します。このコマンドの **dest-dn** は、<arg-dn> が指定されている場合にはその値から取得され、指定されていない場合には現在のオブジェクトから取得されます。送信される関連付けの値は、<arg-association> で指定します。

例

```
<do-add-association> <arg-dn> <token-src-dn/> </arg-dn> <arg-association> <token-src-name/> </arg-association> </do-add-association>
```

内容に使用できる要素

要素	説明
arg-dn	
arg-association	

属性

属性	有効値	デフォルト値
direct	<p>このアクションを実行するには destCommandProcessor を使用します。</p> <p>非推奨。when= “direct” を使用してください。</p>	false
disabled	<p>この要素を無効にする場合は true を指定します。</p>	false

属性	有効値	デフォルト値
when	auto before after direct このアクションを実行するタイミングを指定します。 auto - 自動的に決定 (現在の操作の間か、操作の後)。 before - 現在の操作の前。 after - 現在の操作の後。 direct - 現在のドキュメントに追加されるのではなく、直接ターゲットデータストアに書き込まれます。	auto

内容の規則

(arg-dn ? , arg-association)

親要素

要素	説明
actions	rule
arg-actions	

do-add-dest-attr-value

<arg-value> で指定された値をターゲットデータストアのオブジェクトの指定された属性に追加します。ターゲットオブジェクトは、<arg-dn> または <arg-association> のいずれかで指定されている場合にはそれらになり、指定されていない場合には現在のオブジェクトになります。

例

```
<do-add-dest-attr-value name="Member"> <arg-dn> <token-text>Users/  
ManagerGroup</token-text> </arg-dn> <arg-value> <token-dest-dn/>  
</arg-value> </do-add-dest-attr-value>
```

内容に使用できる要素

要素	説明
arg-dn	
arg-association	
arg-value	

属性

属性	有効値	デフォルト値
class-name	CDATA 変更するオブジェクトのクラス名です。 オブジェクトが現在のオブジェクトではない場合に、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。	#IMPLIED
direct		false
disabled		false
名前	CDATA 属性の名前です。変数の拡張がサポートされます。	#REQUIRED

属性	有効値	デフォルト値
		auto

内容の規則

((arg-dn | arg-association) ? , arg-value)

親要素

要素	説明
actions	rule
arg-actions	

do-add-dest-object

<arg-dn> で指定された名前と場所のターゲットデータストアに、型が `class-name` のオブジェクトを作成します。オブジェクト作成で追加する属性値は、後続の <do-add-dest-attr-value> アクションで同じ <arg-dn> を使って追加する必要があります。

例

```
<do-add-dest-object class-name="User"> <arg-dn> <token-text>Users/  
Fred Flintstone</token-text> </arg-dn> </do-add-dest-object><do-add-  
dest-attr-value name="Surname"> <arg-dn> <token-text>Users/Fred  
Flintstone</token-text> </arg-dn> <arg-value> <token-  
text>Flintstone</token-text> </arg-value> </do-add-dest-attr-value>
```

内容に使用できる要素

要素	説明
<code>arg-dn</code>	

属性

属性	有効値	デフォルト値
<code>class-name</code>	CDATA 作成するオブジェクトのクラス名です。変数の拡張がサポートされます。	#REQUIRED
<code>direct</code>		false
<code>disabled</code>		false
		auto

before - 現在の操作の前。

内容の規則

(arg-dn)

親要素

要素	説明
actions	rule
arg-actions	

do-add-src-attr-value

<arg-value> で指定された値をソースデータストアのオブジェクトの指定された属性に追加します。 [arg-dnarg-association](#)

例

```
<do-add-src-attr-value name="Member"> <arg-dn> <token-text>Users/
ManagerGroup</token-text> </arg-dn> <arg-value> <token-dest-dn/>
</arg-value> </do-add-src-attr-value>
```

内容に使用できる要素

要素	説明
arg-dn	
arg-association	
arg-value	

属性

属性	有効値	デフォルト値
class-name	CDATA 変更するオブジェクトのクラス名です。 オブジェクトが現在のオブジェクトではない場合に、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。	#IMPLIED
disabled		false
name	CDATA 属性の名前です。変数の拡張がサポートされます。	#REQUIRED
		false この要素をポリシーの実行時にトレースする必要がない場合は true を指定します。

内容の規則

親要素

要素	説明
actions	rule
arg-actions	

do-add-src-object

<arg-dn> で指定された名前と場所のソースデータストアに、型が `class-name` のオブジェクトを作成します。オブジェクトの作成で追加する属性値は、後続の <do-add-src-attr-value> アクションで同じ <arg-dn> を使って追加する必要があります。

例

```
<do-add-src-object class-name="User"> <arg-dn> <token-text>Users/
Fred Flintstone</token-text> </arg-dn> </do-add-src-object><do-add-
src-attr-value name="Surname"> <arg-dn> <token-text>Users/Fred
Flintstone</token-text> </arg-dn> <arg-value> <token-
text>Flintstone</token-text> </arg-value> </do-add-src-attr-value>
```

内容に使用できる要素

要素	説明
<code>arg-dn</code>	

属性

属性	有効値	デフォルト値
<code>class-name</code>	CDATA 作成するオブジェクトのクラス名です。	#REQUIRED
<code>disabled</code>		false
		false

内容の規則

親要素

要素	説明
<code>actions</code>	<code>rule</code>
<code>arg-actions</code>	

do-append-xml-element

`expression` で選択された一連の要素に、`name` 属性で名前が指定されたカスタム要素を追加します。

例

```
<do-append-xml-element name="jdbc:statement" expression=".."/><do-append-xml-element name="jdbc:sql" expression="../jdbc:statement[last()]" /><do-append-xml-text expression="../jdbc:statement[last()]/jdbc:sql"> <arg-string> <token-text> UPDATE dirxml.emp SET fname = '</token-text> <token-op-attr name="Given Name"/> <token-text>' </token-text> </arg-string> </do-append-xml-text>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
before	CDATA XPath 1.0 式です。この式は、 <code>expression</code> で選択した各ノードに対して評価されます。 <code>expression</code> が返すノードセットの子ノードの手前に新しい要素が挿入されます。	#IMPLIED
disabled		false
expression	CDATA 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 の式です。	#REQUIRED
name	NMTOKEN 要素のタグ名です。 <code>policy</code> で名前空間が定義されている場合には、名前空間プレフィックスを含めることができます。	#REQUIRED false

内容の宣言

なし

親要素

要素	説明
<code>actions</code>	<code>rule</code>
<code>arg-actions</code>	

do-append-xml-text

<arg-string> で指定されたテキストを `expression` で選択された一連の要素に追加します。

例

<do-append-xml-element> を参照してください。

内容に使用できる要素

要素	説明
<code>arg-string</code>	文字列の引数です。

属性

属性	有効値	デフォルト値
	CDATA	#REQUIRED
	XPath 1.0 式です。この式は、 <code>expression</code> で選択した各ノードに対して評価されます。 <code>expression</code> が返すノードセットの子ノードの手前にテキストが挿入されます。	
<code>disabled</code>		false
	CDATA	#REQUIRED
		false

内容の規則

(`arg-string`)

親要素

要素	説明
<code>actions</code>	<code>rule</code>
<code>arg-actions</code>	

do-break

現在のポリシー内のアクションまたはルールによる処理がこれ以上行われないように、現在の操作を停止します。

例

```
<do-break/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
disabled		false
		false

内容の宣言

なし

親要素

要素	説明
actions	rule
arg-actions	

do-clear-dest-attr-value

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。[arg-dnarg-association](#)

例

```
<do-clear-dest-attr-value name="Member"> <arg-dn> <token-text>Users/  
ManagerGroup</token-text> </arg-dn></do-clear-dest-attr-value>
```

内容に使用できる要素

要素	説明
arg-dn	
arg-association	

属性

属性	有効値	デフォルト値
class-name	CDATA	#IMPLIED
	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。	
direct		false
disabled		false
name	CDATA	#REQUIRED
	属性の名前です。変数の拡張がサポートされます。	
		false

属性	有効値	デフォルト値
		auto

内容の規則

(arg-dn | arg-association) ?

親要素

要素	説明
actions	rule
arg-actions	

do-clear-op-property

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。

例

```
<do-clear-op-property name="myProperty"/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
disabled		false
名前	NMTOKEN 操作プロパティの名前です。	#REQUIRED false

内容の宣言

なし

親要素

要素	説明
actions	rule
arg-actions	

do-clear-src-attr-value

ソース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。
[arg-dnarg-association](#)

例

```
<do-clear-src-attr-value name="Member"> <arg-dn> <token-text>Users/  
ManagerGroup</token-text> </arg-dn> </do-clear-src-attr-value>
```

内容に使用できる要素

要素	説明
arg-dn	
arg-association	

属性

属性	有効値	デフォルト値
class-name	CDATA 変更するオブジェクトのクラス名です。オブジェクトが現在のオブジェクトではない場合に、スキーママッピングを行うのに必要です。	#IMPLIED
disabled		false
名前	CDATA 属性の名前です。	#REQUIRED
	true false この要素をポリシーの実行時にトレースする必要がない場合は true を指定します。	false

内容の規則

親要素

要素	説明
actions	rule
arg-actions	

do-clear-sso-credential

<store-def-dn> 要素で指定された Single Sign On 資格情報ストアで、<arg-dn> 要素で指定されたオブジェクトから、<app-id> 要素で指定されたアプリケーションに対する資格情報をクリアします。資格情報に関するその他の情報をクリアするには、名前付きの <arg-string> 要素を追加で指定します。使用される文字列と名前の数は、資格情報ストアと資格情報のターゲットアプリケーションによって異なります。

例

```
<do-clear-sso-credential store-def-dn="../Library/SSO1" app-id="AD7">  
<arg-dn> <token-parse-dn src-dn-format="src-dn" dest-dn-format="ldap"  
start="0" length="-1"> <token-src-dn/> </token-parse-dn> </arg-  
dn></do-clear-sso-credential>
```

内容に使用できる要素

要素	説明
arg-dn	
arg-string	

属性

属性	有効値	デフォルト値
app-def-dn	CDATA アプリケーションの資格情報定義オブジェクトの DN です。 UI によってのみ使用されます。各種 UI は使用されている DN の形式に準拠している必要があります。	#IMPLIED
app-id	CDATA 資格情報のアプリケーション ID です。変数の拡張がサポートされます。	#REQUIRED
		false
		false

属性	有効値	デフォルト値
store-def-dn	CDATA 資格情報ストア定義オブジェクトのスラッシュ形式の DN です。 インクルードするポリシーに対する相対アドレスを使用することもできます。変数の拡張がサポートされます。	#REQUIRED

内容の規則

(arg-dn , arg-string *)

親要素

要素	説明
actions	<rule> 要素によって実行されるアクションです。
arg-actions	

do-clone-op-attr

現在の操作の子であるすべての要素のうち、`src-name` で指定した名前と同じ `attr-name` 属性を持つ要素を、`attr-name` を `dest-name` に設定して複製します。

例

```
<do-clone-op-attr src-name="Member" dest-name="Equivalent to Me"/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
<code>dest-name</code>	CDATA クローンに付ける属性名です。	#REQUIRED
<code>disabled</code>		false
<code>notrace t</code>	true false	false
<code>src-name</code>	CDATA クローンに付ける属性名です。	#REQUIRED

内容の規則

なし

親要素

要素	説明
<code>actions</code>	<code>rule</code>
<code>arg-actions</code>	

do-clone-xpath

<src-expression> で選択されたノード (下位のノードを含む) のコピーを <dest-expression> で選択された一連の要素に追加します。

例

```
<do-append-xml-element name="delete" expression=".." /><do-clone-xpath  
src-expression="@*" dest-expression="../modify[last()]" />
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
	CDATA	#REQUIRED
	XPath 1.0 式です。この式は、 dest-expression で選択した各ノードに対して評価されます。 expression が返すノードセットの子ノードの手前に非属性のクローン化されたノードが挿入されます。	
dest-expression	CDATA	#REQUIRED
	XPath 1.0 の式です。この式が返すノードセットに含まれた要素が、クローン化されたノードの追加先になります。	
disabled		false
		false
src-expression	CDATA	#REQUIRED
	クローン化されたノードを含むノードセットを返す XPath 1.0 の式です。	

内容の宣言

なし

親要素

要素	説明
actions	rule
arg-actions	

do-delete-dest-object

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。ターゲットオブジェクトは、`<arg-dn>` または `<arg-association>` のいずれかで指定されている場合にはそれらになり、指定されていない場合には現在のオブジェクトになります。

例

```
<do-delete-dest-object> <arg-dn> <token-text>Users/Fred Flintstone</token-text> </arg-dn> </do-delete-dest-object>
```

内容に使用できる要素

要素	説明
<code>arg-dn</code>	
<code>arg-association</code>	

属性

属性	有効値	デフォルト値
<code>class-name</code>	CDATA ターゲットオブジェクトのクラス名です。 変数の拡張がサポートされます。	#IMPLIED
<code>direct</code>		false
<code>disabled</code>		false
		false
		auto

内容の規則

((arg-dn | arg-association) ?)

親要素

要素	説明
actions	rule
arg-actions	

do-delete-src-object

<do-delete-src-object> アクションは、ソースデータストアにあるオブジェクトを削除します。 [arg-dnarg-association](#)

例

```
<do-delete-src-object> <arg-dn> <token-text>Users/Fred Flintstone</token-text> </arg-dn> </do-delete-src-object>
```

内容に使用できる要素

要素	説明
arg-dn	
arg-association	

属性

属性	有効値	デフォルト値
class-name	CDATA	#IMPLIED
disabled	変数の拡張がサポートされます。	false
		false

内容の規則

)

親要素

要素	説明
actions	rule
arg-actions	

do-find-matching-object

ターゲットデータストア、および現在の操作に追加する `dest-dn` または `dest-dn` `<association>` でクエリを実行します。これは、現在の操作が `<add>` の場合のみ有効です。

備考

`<arg-dn>` は `scope= "entry"` の場合は必須で、そうではない場合は省略可能です。

`scope= "subtree"` または `scope= "subordinates"` の場合は、少なくとも1つの `<arg-match-attr>` が必要です。`scope= "entry"` のときは、`<クエリー>` の実行対象となる `<search-attr>` を定義しないため、`<do-find-matching-object>` の対象も定義しません。

`<クエリー>` は、`<do-find-matching-object>` の `scope` 属性を基に `scope` 属性を生成します。`<arg-dn>` がある場合には、その内容が `dest-dn` 属性に設定されます。`class-name` 属性と `<search-class>` は、現在のオブジェクトの `class-name` 属性が使用されます。`<arg-match-attr>` ごとに、同じ属性に対する `<search-attr>` が存在し、`<arg-match-attr>` (存在する場合) の内容の `<arg-value>`、または現在の操作で利用できる値が割り当てられています。利用できる値がない場合は、クエリは実行されず、このアクションで一致は検出されません。

`<クエリー>` から返されたすべての `<インスタンス>` 要素が一致と見なされます。

ターゲットデータストアがアプリケーションの場合は、`<インスタンス>` 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 `<do-find-matching-object>` 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 eDirectory である場合は、現在の操作の `dest-dn` 属性が設定されます。現在の操作にすでに空でない `dest-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 `<do-find-matching-object>` アクションを複数指定しても問題ありません。`<インスタンス>` が1つだけ返され、`<インスタンス>` がまだ関連付けられていない場合は、現在の操作の `dest-dn` が `<インスタンス>` の `src-dn` に設定されます。`<インスタンス>` が1つだけ返され、`<インスタンス>` がすでに関連付けられている場合は、現在の操作の `dest-dn` が1文字の `￼` に設定されます。複数の `<インスタンス>` が返される場合は、現在の操作の `dest-dn` が1文字の `�` に設定されます。

例

```
<do-find-matching-object scope="subordinates"> <arg-dn> <token-text>Users/</token-text> <token-attr name="OU"/> </arg-dn> <arg-match-attr name="CN"/> <arg-match-attr name="L"/> <arg-value> <token-text>Provo</token-text> </arg-value> </arg-match-attr></do-find-matching-object>
```

内容に使用できる要素

要素	説明
----	----

`arg-dn`

`arg-match-attr`

属性

属性	有効値	デフォルト値
disabled		false
		false
scope		

検索する範囲です。

内容の規則

((arg-dn ? , arg-match-attr +) | (arg-dn , arg-match-attr *))

親要素

要素	説明
actions	rule
arg-actions	

do-for-each

<arg-actions> で指定されたアクションを <arg-node-set> 内のノードに対して 1 回ずつ繰り返します。ローカル変数の **current-node** はそのノードだけを含むノードセットに設定されます。**current-node** が <entitlement-impl> の場合、そのアクションも <do-implement-entitlement> の内容としてマークされます。

例

```
<do-for-each> <arg-node-set> <token-added-entitlement name="Group">
</arg-node-set> <arg-actions> <do-add-dest-attr-value name="Member"
class-name="Group"> <arg-dn> <token-local-variable name="current-
node"/> </arg-dn> <arg-value type="dn"> <token-dest-dn/> </arg-
value> </do-add-dest-attr-value> </arg-actions> </do-for-each>
```

内容に使用できる要素

要素	説明
arg-node-set	
arg-actions	

属性

属性	有効値	デフォルト値
disabled		false
		false

内容の規則

(arg-node-set , arg-actions)

親要素

要素	説明
actions	rule
arg-actions	

do-generate-event

Novell® Auditing サービスへ Identity Manager のユーザ定義イベントを送信します。Identity Manager のユーザ定義イベント ID は、1000 から 1999 の範囲にする必要があります。有効なイベントレベルは、次のテーブルに定義されています。残りのイベントデータフィールドは、名前属性を持つ 4 つの <arg-string> 要素で指定します。Novell Audit イベント構造には、1 つの整数 (値) と汎用フィールド (データ) と共に 2 つの文字列 (text1、text2) が含まれています。2 つのテキストフィールドは 256 バイトに制限され、データフィールドには、3 KB までの情報を格納できます。

備考

レベル	説明
log-emergency	メタディレクトリエンジンまたは Identity Manager ドライバがシャットダウンされるイベントです。
log-alert	早期に注意が必要なイベントです。
log-critical	メタディレクトリエンジンまたは Identity Manager ドライバの一部が正常に動作しなくなるイベントです。
log-error	Metadirectory エンジンまたは Identity Manager 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です。
log-warning	問題ではないネガティブなイベントです。
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ポジティブまたはネガティブなイベントです。
log-info	いずれかの重要度を持つポジティブイベントです。
log-debug	サポートまたはエンジニアが Metadirectory エンジンまたは Identity Manager ドライバの操作をデバッグするためのイベントです。

タグ	説明
text1	ここに入力されるテキストは、text1 イベントフィールドに格納されます。
text2	ここに入力されるテキストは、text2 イベントフィールドに格納されます。
値	ここに入力される任意の数字は、value1 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。

例

```
<do-generate-event id="1000" level="log-info"> <arg-string
name="text1"> <token-text>User defined data for text1field</token-
text> </arg-string> <arg-string name="text2"> <token-text>User defined
data for text2field</token-text> </arg-string> <arg-string
name="value"> <token-text>-602</token-text> </arg-string> <arg-string
name="data"> <token-text>User defined blob data</token-text> </arg-
string></do-generate-event >
```

内容に使用できる要素

要素	説明
arg-string	

属性

属性	有効値	デフォルト値
disabled		false
id	CDATA Identity Manager のユーザ定義イベント ID です (1000-2000)。変数の拡張がサポートされます。	#REQUIRED
	log-emergency log-alert log-critical log-error log-warning log-notice log-info log-debug Novell Audit のログレベル。	log-info false

内容の規則

(arg-string *)

親要素

要素	説明
actions	rule
arg-actions	

do-if

<arg-conditions> で指定された条件が **True** と評価された場合に、最初の <arg-actions> で指定されたアクションが実行され、<arg-conditions> で指定された条件が **False** と判定された場合に、2 番目の <arg-actions>(存在する場合) で指定されたアクションが実行されます。

例

```
<do-if> <arg-conditions> <and> <if-op-attr name="Given Name"
op="equal" mode="nocase">fred</if-op-attr> </and> </arg-conditions>
<arg-actions> <do-set-dest-attr-value name="Surname"> <arg-value
type="string"> <token-text>Flintstone</token-text> </arg-value>
</do-add-dest-attr-value> </arg-actions> <arg-actions> <do-set-
dest-attr-value name="Surname"> <arg-value type="string">
<token-text>Rubble</token-text> </arg-value> </do-add-dest-attr-
value> </arg-actions></do-if>
```

内容に使用できる要素

要素	説明
arg-conditions	
arg-actions	

属性

属性	有効値	デフォルト値
disabled		false
		false

内容の規則

(arg-conditions , arg-actions , arg-actions ?)

親要素

要素	説明
actions	rule
arg-actions	

do-implement-entitlement

<arg-node-set> で指定された <entitlement-impl> 要素を持つ <arg-actions> で指定されたアクションにマークを付けます。これによって、現在のオブジェクトの DirXML-EntitlementResult 属性にアクションの結果をレポートすることをメタディレクトリエンジンが認識できるようにします。

例

```
<do-implement-entitlement> <arg-node-set> <token-removed-entitlement name="Account"/> </arg-node-set> <arg-actions> <do-set-dest-attr-value name="Login Disabled"> <arg-value type="state"> <token-text >true</token-text> </arg-value> </do-set-dest-attr-value> </arg-actions></do-implement-entitlement>
```

内容に使用できる要素

要素	説明
arg-node-set	
arg-actions	

属性

属性	有効値	デフォルト値
disabled		false
		false

内容の規則

親要素

要素	説明
actions	rule
arg-actions	

do-move-dest-object

ターゲットデータストア内のオブジェクトを移動します。2つの引数が指定された場合には、最初の引数で指定されたオブジェクトが、2番目の引数で指定されたコンテナに移動されます。1つの引数が指定された場合には、現在のオブジェクトが1つの引数によって指定されたコンテナに移動されます。

例

```
<do-move-dest-object> <arg-dn> <token-text>Users/Active/  
FredFlintstone</token-text> </arg-dn> <arg-dn> <token-text>Users/  
Inactive</token-text> </arg-dn> </do-move-dest-object>
```

内容に使用できる要素

要素	説明
arg-dn	
arg-association	

属性

属性	有効値	デフォルト値
class-name	CDATA	#IMPLIED
	変数の拡張がサポートされます。	
direct		false
disabled		false
		false

属性	有効値	デフォルト値
		auto

内容の規則

((arg-dn | arg-association) ? , (arg-dn | arg-association))

親要素

要素	説明
actions	rule
arg-actions	

do-move-src-object

ソースデータストア内のオブジェクトを移動します。2つの引数が指定された場合には、最初の引数で指定されたオブジェクトが、2番目の引数で指定されたコンテナに移動されます。1つの引数だけが指定された場合には、現在のオブジェクトが1つの引数によって指定されたコンテナに移動されます。

例

```
<do-move-src-object> <arg-dn> <token-text>Users/Active/  
FredFlintstone</token-text> </arg-dn> <arg-dn> <token-text>Users/  
Inactive</token-text> </arg-dn> </do-move-src-object>
```

内容に使用できる要素

要素	説明
arg-dn	
arg-association	

属性

属性	有効値	デフォルト値
class-name	CDATA	#IMPLIED
disabled		false
	変数の拡張がサポートされます。	false

内容の規則

親要素

要素	説明
actions	rule
arg-actions	

do-reformat-op-attr

現在の操作内にある指定した属性のすべての値を、`<arg-value>` と置き換えます。`<arg-value>` の値は、それぞれ 1 回評価され、元の値に設定されたローカル変数の `current-value` で置換されます。

例

```
<do-reformat-op-attr name="CN"> <arg-value> <token-upper-case>
<token-local-variable name="current-value"/> </token-upper-case> </
arg-value> </do-reformat-op-attr><do-reformat-op-attr name="EMail
Address"> <arg-value> <token-xpathexpression="$current-value/
component[@name='eMailAddr']"/> </arg-value> </do-reformat-op-attr>
```

内容に使用できる要素

要素	説明
<code>arg-value</code>	

属性

属性	有効値	デフォルト値
<code>disabled</code>		<code>false</code>
名前	CDATA 属性の名前です。	<code>#REQUIRED</code> <code>false</code>

内容の規則

(`arg-value`)

親要素

要素	説明
<code>actions</code>	<code>rule</code>
<code>arg-actions</code>	

do-remove-association

< 関連付けの削除 > コマンドを eDirectory へ送信します。 **arg-association**

例

```
<do-remove-association> <arg-association> <token-src-name/> </arg-association> </do-remove-association>
```

内容に使用できる要素

要素	説明
arg-association	

属性

属性	有効値	デフォルト値
direct		false
disabled		false
		false
		auto

内容の規則

(arg-association)

親要素

要素	説明
actions	rule
arg-actions	

do-remove-dest-attr-value

<arg-value> で指定された値をターゲットデータストアのオブジェクトの指定された属性から削除します。arg-dnarg-association

例

```
<do-remove-dest-attr-value name="Member"> <arg-dn> <token-text>Users/ManagerGroup</token-text> </arg-dn> <arg-value> <token-dest-dn/> </arg-value> </do-remove-dest-attr-value>
```

内容に使用できる要素

要素	説明
arg-dn	
arg-association	
arg-value	

属性

属性	有効値	デフォルト値
class-name	CDATA	#IMPLIED
	オブジェクトが現在のオブジェクトではない場合に、スキーママッピングを行うのに必要です。	
direct		false
disabled		false
名前	CDATA	#REQUIRED
	属性の名前です。	
		false

属性	有効値	デフォルト値
	auto before after direct このアクションを実行するタイミングを指定します。 auto - 自動的に決定 (現在の操作の間か、操作の後)。 before - 現在の操作の前。 after - 現在の操作の後。 direct - 現在のドキュメントに追加されるのではなく、直接ターゲットデータストアに書き込まれます。	auto

内容の規則

親要素

要素	説明
actions	rule
arg-actions	

do-remove-src-attr-value

<arg-value> で指定された値をソースデータストアのオブジェクトの指定された属性から削除します。 [arg-dn](#) [arg-association](#)

例

```
<do-remove-src-attr-value name="Member"> <arg-dn> <token-text>Users/  
ManagerGroup</token-text> </arg-dn> <arg-value> <token-src-dn/> </  
arg-value> </do-remove-src-attr-value>
```

内容に使用できる要素

要素	説明
arg-dn	
arg-association	
arg-value	

属性

属性	有効値	デフォルト値
class-name	CDATA	#IMPLIED
disabled		false
名前	CDATA 属性の名前です。	#REQUIRED false

内容の規則

親要素

要素	説明
actions	rule
arg-actions	

do-rename-dest-object

ターゲットデータストア内のオブジェクトの名前を、<arg-string> で指定された名前に変更します。 [arg-dnarg-association](#)

例

```
<do-rename-dest-object> <arg-dn> <token-text>Users/Active/Fred  
Flintstone</token-text> </arg-dn> <arg-string> <token-text>Fat  
Freddy</token-text> </arg-string> </do-rename-dest-object>
```

内容に使用できる要素

要素	説明
arg-dn	
arg-association	
arg-string	

属性

属性	有効値	デフォルト値
class-name	CDATA	#IMPLIED
	変数の拡張がサポートされます。	
direct		false
disabled		false
		false
		auto

内容の規則

((arg-dn | arg-association) ? , arg-string)

親要素

要素	説明
actions	rule
arg-actions	

do-rename-op-attr

現在の操作の子であるすべての要素のうち、`src-name` で指定した名前と同じ `attr-name` 属性を持つ要素で `attr-name` を `dest-name` に設定します。

例

```
<do-rename-op-attr src-name="Surname" dest-name="sn"/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
	CDATA 新しい属性名です。	#REQUIRED
disabled		false
		false
	CDATA 元の属性名です。	#REQUIRED

内容の規則

なし

親要素

要素	説明
<code>actions</code>	<code>rule</code>
<code>arg-actions</code>	

do-rename-src-object

ソースデータストア内のオブジェクトの名前を、`<arg-string>` で指定された名前に変更します。`arg-dnarg-association`

例

```
<do-rename-src-object> <arg-dn> <token-text>Users/Active/Fred  
Flintstone</token-text> </arg-dn> <arg-string> <token-text>Fat  
Freddy</token-text> </arg-string> </do-rename-src-object>
```

内容に使用できる要素

要素	説明
<code>arg-dn</code>	
<code>arg-association</code>	
<code>arg-value</code>	

属性

属性	有効値	デフォルト値
<code>class-name</code>	CDATA	#IMPLIED
	変数の拡張がサポートされます。	
<code>disabled</code>		false
		false

内容の規則

親要素

要素	説明
<code>actions</code>	<code>rule</code>
<code>arg-actions</code>	

do-send-email

指定したサーバに電子メール通知を送信します。SMTP サーバでの認証のために、`id` および `password` 属性によるオプションの資格情報を使用できます。`type` 属性で、電子メールメッセージにプレーンテキストと HTML データのどちらが含まれているかを指定します。`<arg-string>` および対応するタグ名の属性内でさまざまな電子メールアドレス、件名、およびメッセージを指定できます。

備考

タグ名	説明
<code>to</code>	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。
<code>cc</code>	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。
<code>bcc</code>	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。
<code>from</code>	電子メールの発信アドレスとして使用するアドレスを指定します。
<code>reply-to</code>	電子メールメッセージの返信アドレスとして使用するアドレスを指定します。
<code>subject</code>	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。
<code>encoding</code>	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。
<code>custom-smtp-header</code>	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

```
<do-send-email server="smtp.company.com" id="user"
password="emailpwd"type="text"> <arg-string name="to"> <token-
text>to_user1@company.com</token-text> </arg-string> <arg-string
name="to"> <token-text>to_user2@company.com</token-text> </arg-
string> <arg-string name="cc"> <token-text>cc_user@company.com</token-
text> </arg-string> <arg-string name="bcc"> <token-
text>bcc_user@company.com</token-text> </arg-string> <arg-string
name="from"> <token-text>from_user@company.com</token-text> </arg-
string> <arg-string name="subject"> <token-text>This is the email
subject</token-text> </arg-string> <arg-string name="message">
<token-text>This is the email body</token-text> </arg-string></do-
send-email >
```

内容に使用できる要素

要素	説明
arg-string	
arg-password	

属性

属性	有効値	デフォルト値
disabled		false
id	CDATA SMTP サーバのユーザアカウントです。変数の拡張がサポートされます。	#IMPLIED
password	CDATA SMTP サーバのユーザパスワードです。非推奨。<arg-password> を <token-named-password> と共に使用してください。	#IMPLIED
server	CDATA SMTP サーバの DNS 名または IP アドレスです。	#REQUIRED
type	text html 電子メールメッセージにプレーンテキストと HTML データのどちらが含まれているかを指定します。	text

内容の規則

親要素

要素	説明
actions	rule
arg-actions	

do-send-email-from-template

SMTP 通知設定オブジェクトを使用して、電子メールテンプレートオブジェクト、および置換トークンを使用して、電子メール通知を生成します。ターゲット SMTP サーバ、認証のための資格情報、および発信アドレスは、SMTP 通知設定オブジェクトから読み取られます。件名および電子メールメッセージは、テンプレートオブジェクトおよびテンプレート置換トークンを使用して作成されます。置換トークンは `<arg-string>` 要素およびタグ名の属性内で宣言します。予約済みの置換トークンは、さまざまな受信者アドレスを指定します。

備考

予約済みトークン	説明
to	
cc	
bcc	
reply-to	電子メールメッセージの返信アドレスとして使用するアドレスを指定します。
encoding	電子メールメッセージで使用する文字エンコードを指定します。

例

```
<do-send-email-from-template notification-dn="/cn=security/cn=DefaultNotification Collection" template-dn="/cn=security/cn=DefaultNotification Collection/cn=PS-Sync Fail"> <arg-string name="manager"> <token-text>Bill Jones</token-text> </arg-string> <arg-string name="surname"> <token-text>Smith</token-text> </arg-string> <arg-string name="given-name"> <token-text>Joe</token-text> </arg-string> <arg-string name="to"> <token-text>to_user@company.com</token-text> </arg-string> <arg-string name="cc"> <token-text>cc_user@company.com</token-text> </arg-string></do-send-email-from-template >
```

内容に使用できる要素

要素	説明
<code>arg-password</code>	
<code>arg-string</code>	

属性

属性	有効値	デフォルト値
disabled		false
notification-dn	CDATA SMTP 通知設定オブジェクトの スラッシュ形式の DN です。	#REQUIRED
disabled		false
パスワード	CDATA SMTP サーバのユーザパスワード です。arg-passwordtoken- named-password	#IMPLIED
	CDATA 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 です。変数の拡張がサポートさ れます。	#REQUIRED

内容の規則

親要素

要素	説明
actions	rule
arg-actions	

do-set-default-attr-value

指定された属性に値がない場合に、<arg-value> で指定された値を現在の操作のその属性に追加します。これは、現在の操作が<追加>の場合のみ有効です。write-back=true の場合、デフォルト値は、ソースオブジェクトにもライトバックされます。

例

```
<do-set-default-attr-value name="L"> <arg-value> <token-text>Unknown</token-text> </arg-value> </do-set-default-attr-value>
```

内容に使用できる要素

要素	説明
arg-value	

属性

属性	有効値	デフォルト値
disabled		false
name	CDATA 属性の名前です。変数の拡張がサポートされます。	#REQUIRED false false デフォルト値をソースオブジェクトにもライトバックする場合には true を指定します。

内容の規則

(arg-value +)

親要素

要素	説明
actions	rule

要素	説明
----	----

arg-actions	
-------------	--

do-set-dest-attr-value

ターゲットデータストア内のオブジェクトの名前付きの属性に、`<arg-value>` で指定された値を追加し、その属性に設定されている他の値をすべて削除します。`arg-dnarg-association`

例

```
<do-set-dest-attr-value name="OU"> <arg-value> <token-text>Sales</token-text> </arg-value></do-set-dest-attr-value>
```

内容に使用できる要素

要素	説明
<code>arg-dn</code>	
<code>arg-association</code>	
<code>arg-value</code>	

属性

属性	有効値	デフォルト値
<code>class-name</code>	CDATA	#IMPLIED
<code>direct</code>		false
<code>disabled</code>		false
名前	CDATA 属性の名前です。	#REQUIRED false

属性	有効値	デフォルト値
		auto

内容の規則

親要素

要素	説明
actions	rule
arg-actions	

do-set-dest-password

ターゲットデータストア内の現在のオブジェクトのパスワードとして、`<arg-string>` で指定された値を設定します。

例

```
<do-set-dest-password> <arg-string> <token-attr name="Given Name"/>
<token-attr name="Surname"/> </arg-string> </do-set-dest-password>
```

内容に使用できる要素

要素	説明
<code>arg-dn</code>	
<code>arg-association</code>	
<code>arg-string</code>	

属性

属性	有効値	デフォルト値
<code>class-name</code>	CDATA	
	変数の拡張がサポートされます。	
<code>direct</code>		false
<code>disabled</code>		false
		false
		auto

内容の規則

親要素

要素	説明
actions	rule
arg-actions	

do-set-local-variable

指定した名前で、`<arg-string>` で指定された文字列値、`<arg-node-set>` で指定された XPath 1.0 ノードセット、または `<arg-object>` で指定された Java* オブジェクトのローカル変数を設定します。

例

```
<!-- sets variable equal to the string value of the first value of
Surname--><do-set-local-variable name="lastName"> <arg-string>
<token-attr name="Surname"/> </arg-string> </do-set-local-
variable><!-- sets variable equal to all the value elements of Surname
--><do-set-local-variable name="lastName"> <arg-node-set> <token-
attr name="Surname"/> </arg-node-set> </do-set-local-variable><!--
sets variable equal to an instance of java.util.Random --><!-- note
that the prefix jrandom needs to have been mapped to --><!-- the URI
http://www.novell.com/nxsl/java/java.util.Random --><!-- on the
<policy> --><do-set-local-variable name="lastName"> <arg-object>
<token-xpath expression="jrandom:new()"/> </arg-object> </do-set-
local-variable>
```

内容に使用できる要素

要素	説明
<code>arg-string</code>	
<code>arg-node-set</code>	
<code>arg-object</code>	Java オブジェクトの引数です。

属性

属性	有効値	デフォルト値
<code>disabled</code>		false
<code>name</code>	NMTOKEN 変数の名前です。変数の拡張がサポートされます。	#REQUIRED false

属性	有効値	デフォルト値
scope	<p>policy driver</p> <p>変数のスコープです。</p> <p>policy: 変数は、ポリシーの現在の呼び出しで、現在のポリシー内からのみ可視です。</p> <p>driver: 変数は、ドライバーが停止されるまで、現在のドライバー内のすべてのポリシーから可視です。</p> <p>変数の拡張がサポートされます。</p>	ポリシー

内容の規則

(arg-string | arg-node-set | arg-object)

親要素

要素	説明
actions	rule
arg-actions	

do-set-op-association

現在の操作の関連付けの値を、<arg-association> で指定された値に設定します。

例

```
<do-set-op-association> <arg-association> <token-src-name/> </arg-association> </do-set-op-association>
```

内容に使用できる要素

要素	説明
arg-association	

属性

属性	有効値	デフォルト値
disabled		false
		false

内容の規則

親要素

要素	説明
actions	rule
arg-actions	

do-set-op-class-name

現在の操作のオブジェクトクラス名を `<arg-string>` で指定された値に設定します。

例

```
<do-set-op-class-name> <arg-string> <token-text>User</token-text>  
</arg-string> </do-set-op-class-name>
```

内容に使用できる要素

要素	説明
<code>arg-string</code>	

属性

属性	有効値	デフォルト値
<code>disabled</code>		false
		false

内容の規則

親要素

要素	説明
<code>actions</code>	<code>rule</code>
<code>arg-actions</code>	

do-set-op-dest-dn

現在の操作のターゲット DN を、<arg-dn> で指定された値に設定します。

例

```
<do-set-op-dest-dn> <arg-dn> <token-text>Novell\Users\</token-text>  
<token-src-name/> </arg-dn> </do-set-op-dest-dn>
```

内容に使用できる要素

要素	説明
arg-dn	

属性

属性	有効値	デフォルト値
disabled		false
		false

内容の規則

親要素

要素	説明
actions	rule
arg-actions	

do-set-op-property

指定された名前の操作プロパティを現在の操作の <arg-string> によって指定された値に設定します。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、追加のコンテキストを指定するために使用し、操作の結果を処理するポリシーで必要になる場合があります。

例

```
<do-set-op-property name="myProperty"> <arg-string> <token-text>Fred</token-text> </arg-string> </do-set-op-property>
```

内容に使用できる要素

要素	説明
arg-string	

属性

属性	有効値	デフォルト値
disabled		false
名前	NMTOKEN	#REQUIRED
		false

内容の規則

親要素

要素	説明
actions	rule
arg-actions	

do-set-op-src-dn

現在の操作のソース DN を、`<arg-dn>` で指定された値に設定します。

例

```
<do-set-op-src-dn> <arg-dn> <token-text>Novell\Users\</token-text>  
<token-attr name="CN"/> </arg-dn> </do-set-op-src-dn>
```

内容に使用できる要素

要素	説明
<code>arg-dn</code>	

属性

属性	有効値	デフォルト値
<code>disabled</code>		false
		false

内容の規則

親要素

要素	説明
<code>actions</code>	<code>rule</code>
<code>arg-actions</code>	

do-set-op-template-dn

現在の操作のテンプレート DN を、<arg-dn> で指定された値に設定します。これは、現在の操作が <追加> の場合のみ有効です。

例

```
<do-set-op-template-dn> <arg-dn> <token-  
text>Novell\Users\UserTemplate</token-text> </arg-dn> </do-set-op-  
template-dn>
```

内容に使用できる要素

要素	説明
arg-dn	

属性

属性	有効値	デフォルト値
disabled		false
		false

内容の規則

親要素

要素	説明
actions	rule
arg-actions	

do-set-src-attr-value

ソースデータストア内のオブジェクトの名前付きの属性に、<arg-value> で指定された値を追加し、その属性に設定されている他の値をすべて削除します。arg-dnarg-association

例

```
<do-set-src-attr-value name="OU"> <arg-value> <token-text>Sales</token-text> </arg-value></do-set-src-attr-value>
```

内容に使用できる要素

要素	説明
arg-dn	
arg-association	
arg-value	

属性

属性	有効値	デフォルト値
class-name	CDATA	#IMPLIED
disabled	オブジェクトが現在のオブジェクトではない場合に、スキーママッピングを行うのに必要です。	false
名前	CDATA 属性の名前です。	#REQUIRED false

内容の規則

親要素

要素	説明
actions	rule
arg-actions	

do-set-src-password

arg-string

例

```
<do-set-src-password> <arg-string> <token-attr name="Given Name"/>
<token-attr name="Surname"/> </arg-string> </do-set-src-password>
```

内容に使用できる要素

要素	説明
arg-dn	
arg-association	
arg-string	

属性

属性	有効値	デフォルト値
class-name	CDATA	#IMPLIED
	変数の拡張がサポートされます。	
disabled		false
		false

内容の規則

親要素

要素	説明
actions	rule
arg-actions	

do-set-sso-credential

<store-def-dn> 要素で指定された Single Sign On 資格情報ストアで、<arg-dn> 要素で指定されたオブジェクトに、<app-id> 要素で指定されたアプリケーションに対する資格情報を設定します。資格情報は、追加の名前付き <arg-string> 要素によって指定します。

例

```
<do-set-sso-credential store-def-dn="../Library/SSO1" app-id="AD7">
  <arg-dn> <token-parse-dn src-dn-format="src-dn" dest-dn-format="ldap"
  start="0" length="-1"> <token-src-dn/> </token-parse-dn> </arg-dn>
  <arg-string name="username"> <token-src-name/> </arg-string> <arg-
  string name="password"> <token-local-variable
  name="generatedPassword"/> </arg-string> </do-set-sso-credential>
```

内容に使用できる要素

要素	説明
arg-dn	
arg-string	

属性

属性	有効値	デフォルト値
	CDATA	#IMPLIED
	CDATA	#REQUIRED
disabled		false
		false
	CDATA	#REQUIRED
	CDATA	#REQUIRED

内容の規則

親要素

要素	説明
<code>actions</code>	<code>rule</code>
<code>arg-actions</code>	

do-set-sso-passphrase

<store-def-dn> 要素で指定された Single Sign On 資格情報ストアで、<arg-dn> 要素で指定されたオブジェクトに、<arg-string> 要素で指定されたパズフレーズの質問と回答を設定します。

例

```
<do-set-sso-passphrase store-def-dn="../Library/SSO1"> <arg-dn>
<token-parse-dn src-dn-format="src-dn" dest-dn-format="ldap" start="0"
length="-1"> <token-src-dn/> </token-parse-dn> </arg-dn> <arg-
string> <token-text/>What favorite color?<token-text/> </arg-string>
<arg-string> <token-text/>blue<token-text/> </arg-string> </do-set-
sso-passphrase>
```

内容に使用できる要素

要素	説明
arg-dn	
arg-string	

属性

属性	有効値	デフォルト値
disabled		false
		false
	CDATA	#REQUIRED
	{store-def-dn の説明 }	

内容の規則

(arg-dn , arg-string , arg-string)

親要素

要素	説明
actions	rule

要素	説明
----	----

arg-actions	
-------------	--

do-set-xml-attr

式で選択された要素のセットに、名前属性で指定されたカスタムの XML 属性を追加します。

例

```
<do-set-xml-attr name="cert-id" expression="."> <arg-string> <token-text>c:\lotus\domino\data\eng.id</token-text> </arg-string> </do-set-xml-attr><do-set-xml-attr name="cert-pwd" expression="."> <arg-string> <token-text>certify2eng</token-text> </arg-string> </do-set-xml-attr>
```

内容に使用できる要素

要素	説明
<code>arg-string</code>	

属性

属性	有効値	デフォルト値
<code>disabled</code>		false
	CDATA	#REQUIRED
	XPath 1.0 の式です。この式が返すノードセットに含まれた要素が XML 属性の設定先になります。	
<code>name NMTOKEN</code>	XML 属性のタグ名です。	#REQUIRED
	<code><policy></code> で名前スペースが定義されている場合には、名前スペースプレフィックスを含めることができます。変数の拡張がサポートされます。	
		false

内容の規則

親要素

要素	説明
actions	rule
arg-actions	

do-start-workflow

workflow-id で指定されたワークフローを、<arg-password> で指定された受信者に対して、id と <arg-dn> で指定された資格情報を使用して、url で指定されたユーザアプリケーションサーバで開始します。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 形式 DN である必要があります。ワークフローの追加の引数は、名前付きの <arg-string> 要素で指定できます。使用される文字列と名前の数は、資格情報ストアと資格情報のターゲットとなるアプリケーションによって異なります。

特殊な意味を持つ名前、開始するワークフローに関係なく使用できるものがいくつかあります。

- ◆ **:InitiatorOverrideDN:** 認証にユーザ以外を使う場合の、ワークフローのイニシエータに関する LDAP 形式 DN です。
- ◆ **:CorrelationID:** 関連するワークフローを関連付けるために使用する ID です。

ワークフローの開始時に何らかのエラーが発生した場合、エラー文字列を `error.do-start-workflow` という名前のローカル変数に設定することで、取り囲んでいるポリシーで使用することができます。このローカル変数はこれ以外には使用できません。

例

```
<do-start-workflow workflow-  
id="CN=ApproveCellPhone,CN=RequestDefs,CN=AppConfig,CN=UserApplication  
,CN=DriverSet,O=novell" url="http://localhost:8080/IDMProv"  
id="cn=WorkflowAdmin,o=People"> <arg-password> <token-named-password  
name="workflow-admin"/> </arg-password> <arg-dn> <token-parse-dn src-  
dn-format="qualified-slash" dest-dn-format="ldap"> <token-xpath  
expression="@qualified-src-dn"/> </token-parse-dn> </arg-dn> <arg-  
string name="provider"> <token-text>ACMEWireless</token-text> </  
arg-string> <arg-string name="reason"> <token-text>new hire</token-  
text> </arg-string> </do-start-workflow>
```

内容に使用できる要素

要素	説明
----	----

arg-password

arg-dn

arg-string

属性

属性	有効値	デフォルト値
disabled		false
id	CDATA 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認定 されたユーザの LDP 形式 DN で す。 変数の拡張がサポートされます。	#REQUIRED false
url	CDATA ワークフローが実行されるユー ザアプリケーションサーバ URL です。 変数の拡張がサポートされます。	#REQUIRED
workflow-id	CDATA 開始するワークフローの LDAP 形式 DN です。 変数の拡張がサポートされます。	#REQUIRED

内容の規則

(arg-password , arg-dn , arg-string *)

親要素

要素	説明
actions	rule
arg-actions	

do-status

指定されたレベルのステータス通知を、`<arg-string>` で指定されたメッセージと共に生成します。

備考

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

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

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

例

```
<do-status level="warning"> <arg-string> <token-src-dn/> <token-text>: operation vetoed on out-of-scopeobject</token-text> </arg-string> </do-status >
```

内容に使用できる要素

要素	説明
<code>arg-string</code>	

属性

属性	有効値	デフォルト値
<code>disabled</code>		false
	CDATA ステータスのレベルです。	#REQUIRED false

内容の規則

親要素

要素	説明
actions	rule
arg-actions	

do-strip-op-attr

現在の操作の子であるすべての要素のうち、現在の操作の名前で指定された名前と同じ attr-name 属性を持つ要素を、現在の操作から除去します。

例

```
<do-strip-op-attr name="Member"/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
disabled		false
名前	CDATA 属性の名前です。	#REQUIRED false

内容の規則

なし

親要素

要素	説明
actions	rule
arg-actions	

do-strip-xpath

XPath 1.0 式で選択されたノードを現在の操作から削除します。式はノードセットに評価される必要があります。

例

```
<do-strip-xpath expression="*[@attr-name='OU']"/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
disabled		false
	CDATA	#REQUIRED
	XPath 式です。	
		false

内容の宣言

なし

親要素

要素	説明
actions	rule
arg-actions	

do-trace-message

指定されたレベルが、現在設定されているトレースレベルよりも低いか同じ場合に、`<arg-string>` で指定されたトレースメッセージを DSTRACE に指定します。

例

```
<do-trace-level level="0" color="blue"> <arg-string> <token-  
text>placing new object at </token-text> <token-dest-dn/> </arg-  
string> </do-status >
```

内容に使用できる要素

要素	説明
<code>arg-string</code>	

属性

属性	有効値	デフォルト値
color	black blue green cyan red purple brown grey drgrey brblue brgreen brcyan brred brpurple yellow white 送信するテキストの色です。	brpurple
disabled		false
	CDATA メッセージを送信するトレースの最低レベルです。	0
		false

内容の規則

親要素

要素	説明
<code>actions</code>	<code>rule</code>

要素	説明
----	----

arg-actions	
-------------	--

do-veto

現在の操作をキャンセルします。

例

```
<do-veto/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
disabled		false
		false

内容の規則

なし

親要素

要素	説明
actions	rule
arg-actions	

do-veto-if-op-attr-not-available

指定された属性が現在の操作で使用できない場合は、現在の操作をキャンセルします。

例

```
<do-veto-if-op-attr-not-available name="CN"/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
disabled		false
名前	CDATA 属性の名前です。変数の拡張がサポートされます。	#REQUIRED false

内容の規則

なし

親要素

要素	説明
actions	rule
arg-actions	

do-while

<arg-conditions> で指定された条件が true に評価されている間、<arg-actions> で指定されたアクションを繰り返します。

例

```
<do-set-local-variable name="counter"> <arg-string> <token-text>1</token-text> </arg-string></do-set-local-variable><do-while> <arg-conditions> <and> <if-local-variable name="counter" op="not-gt" mode="numeric">10</if-local-variable> </and> </arg-conditions> <arg-actions> <do-trace-message level="0" color="yellow"> <arg-string> <token-text>Counter = </token-text> <token-local-variable name="counter"/> </arg-string> </do-trace-message> <do-set-local-variable name="counter"> <arg-string> <token-xpath expression="$counter + 1"/> </arg-string> </do-set-local-variable> </arg-actions> </do-while>
```

内容に使用できる要素

要素	説明
arg-conditions	
arg-actions	

属性

属性	有効値	デフォルト値
disabled		false
		false

内容の規則

(arg-conditions , arg-actions)

親要素

要素	説明
actions	rule
arg-actions	

if-association

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種類は、op 属性で指定された演算子によって異なります。次の表に、各演算子で実行されるテストの種類を示します。

備考

演算子	次の場合に true が返されます ...
associated	現在のオブジェクトに対して確立された関連付けがある場合。
available	現在の操作で指定された、空ではない関連付けの値がある場合。
equal	現在の操作で指定された関連付けの値が、<if-association> の内容と完全に同じ場合。変数の拡張がサポートされます。
lt	現在の操作で指定された関連付けの値が、指定した比較モードにおける比較で条件の内容より小さい場合。変数の拡張がサポートされます。
gt	現在の操作で指定された関連付けの値が、指定した比較モードにおける比較で条件の内容より大きい場合。変数の拡張がサポートされます。
not-associated	associated が false を返す場合。
not-available	available が false を返す場合。
not-equal	equal が false を返す場合。
not-lt	lt が false を返す場合。
not-gt	gt が false を返す場合。

例

```
<if-association op="associated " /><if-association op="available" /><if-association op="equal" >{07414faa-1b38-40ec-8b7c-c20aa21ddafb}</if-association>
```

内容に使用できる要素

#PCDATA

属性

属性	有効値	デフォルト値
disabled		false
mode	case nocase regex src-dn dest-dn numeric octet op が比較を伴う場合の比較モードです。	nocase false
op	available equal lt gt not-available not-equal not-lt not-gt テスト演算子です。	#REQUIRED

内容の規則

(#PCDATA)

親要素

要素	説明
and	
または	

if-attr

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。

備考

次の操作と論理的に等価と見なすことができます。

```
<or> <if-op-attr/> <if-src-attr/></or>
```

演算子

available	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る場合。
equal	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、指定した比較モードにおける比較で、その値が <code><if-attr></code> の内容と同じ場合。 <code>mode= "structured"</code> の場合は、内容が一連の <code><></code> 要素である必要があります。それ以外の場合はテキストである必要があります。変数の拡張がサポートされます。
lt	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、指定した比較モードにおける比較で、その値が条件の内容より小さい場合。変数の拡張がサポートされます。
gt	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、指定した比較モードにおける比較で、その値が条件の内容より大きい場合。変数の拡張がサポートされます。
not-available	
not-equal	

例

```
<if-attr op="available" name="OU"/><if-attr op="equal" mode="nocase" name="OU">Sales</if-attr><if-attr op="equal" mode="structured" name="Language"> <component name="string">EN</component> <component name="string">JP</component></if-attr >
```

内容に使用できる要素

#PCDATA

要素	説明
----	----

属性

属性	有効値	デフォルト値
disabled		false
モード	case nocase regex src-dn dest-dn numeric octet structured	nocase
名前	CDATA 属性の名前です。	#REQUIRED false
op	available equal lt gt not-available not-equal not-lt not-gt	#REQUIRED

内容の規則

親要素

要素	説明
and	
または	

if-class-name

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。

備考

演算子

available	現在の操作で使用可能なオブジェクトクラス名がある場合。
equal	現在の操作で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における比較で、その名前が <code><if-class-name></code> の内容と同じの場合。
lt	現在の操作で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における比較で、その名前が条件の内容より小さい場合。変数の拡張がサポートされます。
gt	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における比較で、その名前が条件の内容より大きい場合。変数の拡張がサポートされます。
not-available	
not-equal	

例

```
<if-class-name op="available"/><if-class-name op="equal" mode="nocase">User</if-class-name >
```

内容に使用できる要素

#PCDATA

属性

属性	有効値	デフォルト値
disabled		false
モード		nocase

属性	有効値	デフォルト値
		false
op		#REQUIRED

内容の規則

(#PCDATA)

親要素

要素	説明
and	
または	

if-dest-attr

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。次の表に、各演算子で実行されるテストの種類を示します。

備考

演算子

available	ターゲットデータストアに、指定した属性で使用可能な値がある場合。
equal	ターゲットデータストアに、指定した属性で使用可能な値があり、指定した比較モードにおける比較で、その値が <if-dest-attr> の内容と同じ場合。変数の拡張がサポートされます。
lt	ターゲットデータストアに、指定した属性で使用可能な値があり、指定した比較モードにおける比較で、その値が条件の内容より大きい場合。 mode= "structured" の場合は、内容が一連の <> 要素である必要があります。それ以外の場合はテキストである必要があります。変数の拡張がサポートされます。
gt	ターゲットデータストアに、指定した属性で使用可能な値があり、指定した比較モードにおける比較で、その値が条件の内容より小さい場合。変数の拡張がサポートされます。
not-associated	
not-available	
not-equal	

例

```
<if-dest-attr op="available" name="OU"/><if-dest-attr op="equal" mode="nocase" name="OU">Sales</if-dest-attr><if-dest-attr op="equal" mode="structured" name="Language"> <component name="string">EN</component> <component name="string">JP</component></if-dest-attr >
```

内容に使用できる要素

#PCDATA

属性

属性	有効値	デフォルト値
disabled		false
モード		nocase
名前	CDATA 属性の名前です。変数の拡張がサポートされます。	#REQUIRED
op		false
		#REQUIRED

内容の規則

親要素

要素	説明
and	
または	

if-dest-dn

現在の操作のターゲット DN をテストします。

備考

演算子

available	使用可能なターゲット DN がある場合。
equal	使用可能なターゲット DN があり、ターゲット データストアの DN 形式に適したセマンティックによる比較で、その DN が <if-dest-dn> の内容と同じ場合。変数の拡張がサポートされます。
in-container	使用可能なターゲット DN があり、ターゲット データストアの DN 形式に適したセマンティックによる比較で、その DN が <if-dest-dn> の内容で指定したコンテナ内のオブジェクトである場合。変数の拡張がサポートされます。
in-subtree	使用可能なターゲット DN があり、ターゲット データストアの DN 形式に適したセマンティックによる比較で、その DN が <if-dest-dn> の内容で指定されたサブツリー内のオブジェクトである場合。変数の拡張がサポートされます。
not-available	
not-equal	

例

```
<if-dest-dn op="available"/><if-dest-dn op="equal">Novell\Users\Fred</if-dest-dn><if-dest-dn op="in-container">Novell\Users</if-dest-dn><if-dest-dn op="in-subtree">Novell</if-dest-dn >
```

内容に使用できる要素

#PCDATA

属性

属性	有効値	デフォルト値
属性	disabled true false	false

属性	有効値	デフォルト値
		false
op	available equal in-container in-subtree not-available not-equal not-in-container not-in-subtree	#REQUIRED

内容の規則

(#PCDATA)

親要素

要素	説明
and	
または	

if-entitlement

現在の操作または識別ポータルデータストア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。

備考

演算子

available	指定したエンタイトルメントが使用可能で、現在の操作または識別ポータルデータストアで許可されている場合。
changing	現在の操作に、指定したエンタイトルメントの変更 (許可または削除) がある場合。
changing-from	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があり、指定した比較モードにおける比較で、その値が <if-entitlement> の内容と同じ場合。変数の拡張がサポートされます。
changing-to	現在の操作に、指定したエンタイトルメントの値を許可する変更があり、指定した比較モードにおける比較で、その値が <if-entitlement> の内容と同じ場合。変数の拡張がサポートされます。
equal	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タルデータストアで使用可能で許可されており、指定した比較モードにおける比較で、そのエンタイトルメントの値が <if-entitlement> の内容と同じ場合。変数の拡張がサポートされません。
lt	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タルデータストアで使用可能で許可されており、指定した比較モードにおける比較で、そのエンタイトルメントの値が条件の内容より小さい場合。変数の拡張がサポートされます。
gt	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タルデータストアで使用可能で許可されており、指定した比較モードにおける比較で、そのエンタイトルメントの値が条件の内容より大きい場合。変数の拡張がサポートされます。
not-available	
not-changing	changing が false を返す場合。
not-changing-from	changing-from が false を返す場合。
not-changing-to	changing-to が false を返す場合。
not-equal	

例

```
<if-entitlement op="available" name="notes-group"/><if-entitlement  
op="changing" name="notes-group"/><if-entitlement op="changing-  
from" name="notes-group"/>Sales</if-entitlement><if-entitlement  
op="changing-to" name="notes-group"/>Sales</if-entitlement><if-  
entitlement op="equal" mode="nocase" name="notes-group">Sales</if-  
entitlement>
```

内容に使用できる要素

#PCDATA

属性

属性	有効値	デフォルト値
disabled		false
mode		nocase
name	CDATA 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。	#REQUIRED
op	available changing changing- from changing-to equal lt gt not-available not-changing not- changing-from not-changing-to not-equal not-lt not-gt	#REQUIRED

内容の規則

(#PCDATA)

親要素

要素	説明
and	
または	

if-global-variable

グローバル設定変数をテストします。次の表に、各演算子で実行されるテストの種類を示します。

備考

演算子	
available	指定した名前のグローバル設定変数がある場合。
equal	指定した名前のグローバル設定変数があり、指定した比較モードにおける比較で、その値が <code><if-global-variable></code> の内容と同じ場合。変数の拡張がサポートされます。
lt	指定した名前のグローバル構成変数があり、指定した比較モードにおける比較で、その値が条件の内容より小さい場合。変数の拡張がサポートされます。
gt	指定した名前のグローバル構成変数があり、指定した比較モードにおける比較で、その値が条件の内容より大きい場合。変数の拡張がサポートされます。
not-available	
not-equal	

例

```
<if-global-variable op="available" name="myGlobalVariable"/><if-global-variable op="equal" mode="nocase" name="myGlobalVariable">enabled</if-global-variable>
```

内容に使用できる要素

#PCDATA

属性

属性	有効値	デフォルト値
disabled		false

属性	有効値	デフォルト値
mode		nocase
	op が比較を暗黙に定義する場合の比較モードです。	
name	CDATA 変数の名前です。変数の拡張がサポートされます。	#REQUIRED
		false
op		#REQUIRED

内容の規則

(#PCDATA)

親要素

要素	説明
and	
または	

if-local-variable

ローカル変数をテストします。次の表に、各演算子で実行されるテストの種類を示します。ローカル変数にノードセットが含まれている場合、ノードセット内の各ノードの文字列値が比較の目的で使用されます。同じローカル変数がポリシースコープとドライバースコープの両方に存在する場合は、ポリシースコープの変数が優先されます。

備考

演算子

available	指定した名前のローカル変数があり、<policy>内の以前の <rule> のアクションですでに定義されている場合。
equal	指定した名前のローカル変数があり、指定した比較モードにおける比較で、その値が <if-local-variable> の内容と同じ場合。変数の拡張がサポートされます。
lt	指定した名前のローカル変数があり、指定した比較モードにおける比較で、その値が条件の内容より小さい場合。変数の拡張がサポートされます。
gt	指定した名前のローカル変数があり、指定した比較モードにおける比較で、その値が条件の内容より大きい場合。変数の拡張がサポートされます。
not-available	
not-equal	

例

```
<if-local-variable op="available" name="myLocalVariable"/><if-local-variable op="equal" mode="nocase" name="myLocalVariable">enabled</if-local-variable>
```

内容に使用できる要素

```
#PCDATA
```

属性

属性	有効値	デフォルト値
disabled		false
mode	op= “equal”、または op= “not-equal” の場合の比較モードです。	nocase
name	CDATA 変数の名前です。変数の拡張がサポートされます。	#REQUIRED false
op		#REQUIRED

内容の規則

(#PCDATA)

親要素

要素	説明
and	
または	

if-named-password

ドライバの指定されたパスワードをテストします。次の表に、各演算子で実行されるテストの種類を示します。

備考

演算子

available	指定した名前のパスワードが使用可能な場合。
not-available	

例

```
<if-named-password op="available" name="extraPassword"/>
```

内容に使用できる要素

#PCDATA

属性

属性	有効値	デフォルト値
disabled		false
name	CDATA パスワードの名前。	#REQUIRED false
op	available not-available	#REQUIRED

内容の宣言

(#PCDATA)

親要素

要素	説明
and	
または	

if-op-attr

現在の操作の属性値をテストします。

備考

演算子

available	現在の操作 (<add-attr>、<add-value>、または <attr>) に、指定した属性で使用可能な値がある場合。
changing	現在の操作に、指定された属性の変更 (<modify-attr> または <add-attr>) がある場合。
changing-from	現在の操作に、指定した属性の値 (<remove-value>) を削除する変更があり、指定した比較モードにおける比較で、その値が <if-op-attr> の内容と同じ場合。mode= "structured" の場合は、内容が一連の <> 要素である必要があり、それ以外の場合はテキストである必要があります。
changing-to	現在の操作に、指定した属性に値 (<add-value> または <add-attr>) を追加する変更があり、指定した比較モードにおける比較で、その値が <if-op-attr> の内容と同じ場合。mode= "structured" の場合は、内容がテキストである必要があり、それ以外の場合は一連の <> 要素である必要があり。
equal	現在の操作に、指定した属性で使用可能な値 (<remove-value> 以外) があり、指定した比較モードにおける比較で、その値が <if-op-attr> の内容より大きい場合。mode= "structured" の場合は、内容が一連の <> 要素である必要があり、それ以外の場合はテキストである必要があります。変数の拡張がサポートされます。
lt	現在の操作に、指定された属性で使用可能な値 (<remove-value> 以外) があり、指定した比較モードにおける比較で、その値が条件の内容より小さい場合。変数の拡張がサポートされます。
gt	現在の操作に、指定された属性で使用可能な値 (<remove-value> 以外) があり、指定した比較モードにおける比較で、その値が条件の内容より大きい場合。変数の拡張がサポートされます。
not-available	
not-changing	
not-changingfrom	
not-changing-to	
not-equal	

例

```
<if-op-attr op="available" name="OU"/><if-op-attr op="changing"
name="OU"/><if-op-attr op="changing-from" name="OU"/>Sales</if-op-
attr><if-op-attr op="changing-to" name="OU"/>Sales</if-op-attr><if-op-
attr op="equal" mode="nocase" name="OU">Sales</if-op-attr><if-op-attr
op="equal" mode="structured" name="Language"> <component
name="string">EN</component> <component name="string">JP</component></
if-op-attr>
```

内容に使用できる要素

#PCDATA

属性

属性	有効値	デフォルト値
disabled		false
mode	case nocase regex src-dn dest-dn numeric octet structured op= "equal"、op= "not-equal"、op= "changing-from"、または op= "changing-to" の場合の比較モードです。	nocase
name	CDATA 属性の名前です。変数の拡張がサポートされます。	#REQUIRED false
op		#REQUIRED

内容の規則

親要素

要素	説明
and	
または	

if-op-property

現在の操作の操作プロパティをテストします。次の表に、各演算子で実行されるテストの種類が示してあります。

備考

演算子

available	現在の操作に、指定した名前の操作プロパティがある場合。
equal	指定した名前の操作プロパティが現在の操作にあり、指定した比較モードにおける比較で、その値が <if-op-property> の内容と同じ場合。変数の拡張がサポートされます。
lt	指定した名前の操作プロパティが現在の操作にあり、指定した比較モードにおける比較で、その値が条件の内容より小さい場合。変数の拡張がサポートされます。
gt	指定した名前の操作プロパティが現在の操作にあり、指定した比較モードにおける比較で、その値が条件の内容より大きい場合。変数の拡張がサポートされます。
not-available	
not-equal	

例

```
<if-op-property op="available" name="myLocalVariable" /><if-op-property  
op="equal" mode="nocase" name="myProperty">true</if-local-variable>
```

内容に使用できる要素

#PCDATA

属性

属性	有効値	デフォルト値
disabled		false

属性	有効値	デフォルト値
モード		nocase
名前	CDATA 操作プロパティの名前です。変数の拡張がサポートされます。	#REQUIRED false
op		#REQUIRED

内容の規則

(#PCDATA)

親要素

要素	説明
and	
または	

if-operation

現在の操作の名前をテストします。次の表に、各演算子で実行されるテストの種類が示してあります。

備考

演算子

equal	現在の操作の名前が、<if-operation> の内容と完全に同じ場合。変数の拡張がサポートされます。
lt	現在の操作名が、指定した比較モードにおける比較で、条件の内容より小さい場合。変数の拡張がサポートされます。
gt	現在の操作名が、指定した比較モードにおける比較で、条件の内容より大きい場合。変数の拡張がサポートされます。
not-equal	

例

```
<if-operation op="equal">add</if-operation>
```

内容に使用できる要素

#PCDATA

属性

属性	有効値	デフォルト値
disabled		false
モード	case nocase regex scr-dn dest-dn numeric octet	case (大文字と小文字の区別あり) false

属性	有効値	デフォルト値
op	equal lt gt not-equal not-lt not-gt	#REQUIRED

内容の規則

(#PCDATA)

親要素

要素	説明
and	
または	

if-password

現在の操作のパスワードをテストします。次の表に、各演算子で実行されるテストの種類を示します。

備考

演算子

available	現在の操作に使用可能なパスワードがある場合。
equal	現在の操作内に使用可能なパスワードがあり、指定した比較モードにおける比較で、その値が条件の内容と等しい場合。変数の拡張がサポートされます。
lt	現在の操作内に使用可能なパスワードがあり、指定した比較モードにおける比較で、その値が条件の内容より小さい場合。変数の拡張がサポートされます。
gt	現在の操作内に使用可能なパスワードがあり、指定した比較モードにおける比較で、その値が条件の内容より大きい場合。変数の拡張がサポートされます。
not-available (使用不可)	
等しくない	

例

```
<if-password op="available"/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
disabled		false
モード	case nocase regex scr-dn dest-dn numeric octet	case (大文字と小文字の区別あり)

属性	有効値	デフォルト値
		false
op		#REQUIRED

内容の規則

なし

親要素

要素	説明
and	
または	

if-src-attr

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。次の表に、各演算子で実行されるテストの種類を示します。

備考

演算子

available	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場合。
equal	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、指定した比較モードにおける比較で、その値が <if-src-attr> の内容と同じ場合。変数の拡張がサポートされます。
lt	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、指定した比較モードにおける比較で、その値が条件の内容より小さい場合。変数の拡張がサポートされます。
gt	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、指定した比較モードにおける比較で、その値が条件の内容より大きい場合。変数の拡張がサポートされます。
not-available	
not-equal	

例

```
<if-src-attr op="available" name="OU"/><if-src-attr op="equal" mode="nocase" name="OU">Sales</if-src-attr><if-src-attr op="equal" mode="structured" name="Language"> <component name="string">EN</component> <component name="string">JP</component></if-src-attr>
```

内容に使用できる要素

#PCDATA

要素	説明
----	----

属性

属性	有効値	デフォルト値
disabled		false
モード		nocase
名前	CDATA 属性の名前です。変数の拡張がサポートされます。	#REQUIRED false
op		#REQUIRED

内容の規則

親要素

要素	説明
and	
または	

if-src-dn

現在の操作のソース DN をテストします。

備考

演算子

available	使用可能なソース DN がある場合。
equal	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した場合ときに、その DN が <if-src-dn> の内容と同じ場合。変数の拡張がサポートされません。
in-container	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したときに、その DN が <if-src-dn> の内容で指定したコンテナ内のオブジェクトである場合。
in-subtree	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したときに、その DN が <if-src-dn> の内容で指定したサブツリー内のオブジェクトである場合。
not-available	
not-equal	

例

```
<if-src-dn op="available"/><if-src-dn op="equal">Novell\Users\Fred</if-src-dn><if-src-dn op="in-container">Novell\Users</if-src-dn><if-src-dn op="in-subtree">Novell</if-src-dn>
```

内容に使用できる要素

#PCDATA

属性

属性	有効値	デフォルト値
disabled		false

属性	有効値	デフォルト値
		false
op	テスト演算子です。	#REQUIRED

内容の規則

(#PCDATA)

親要素

要素	説明
and	
または	

if-xml-attr

現在の操作の XML 属性に対してテストを実行します。次の表に、各演算子で実行されるテストの種類を示します。

備考

演算子

available	現在の操作に、指定した名前の XML 属性がある場合。
equal	指定した名前の XML 属性が現在の操作にあり、指定した比較モードにおける比較で、その値が条件の内容と等しい場合。変数の拡張がサポートされます。
lt	指定した名前の XML 属性が現在の操作にあり、指定した比較モードにおける比較で、その値が条件の内容より小さい場合。変数の拡張がサポートされます。
gt	指定した名前の XML 属性が現在の操作にあり、指定した比較モードにおける比較で、その値が条件の内容より大きい場合。変数の拡張がサポートされます。
not-available	
not-equal	

例

```
<if-xml-attr op="available" name="from-merge"/><if-xml-attr op="equal" mode="nocase" name="level">error</if-xml-attr>
```

内容に使用できる要素

#PCDATA

属性

属性	有効値	デフォルト値
disabled		false

属性	有効値	デフォルト値
モード		nocase
name	CDATA XML 属性のタグ名です。変数の拡張がサポートされます。 拡張されたものは、XML QName の規則に従う必要があります。 <policy> で名前スペースプレフィックスが定義されている場合、およびこのポリシーでのみ定義されている場合には、名前スペースプレフィックスを含めることができます。	#REQUIRED false
op		#REQUIRED

内容の規則

(#PCDATA)

親要素

要素	説明
and	
または	

if-xpath

XPath 1.0 の式の評価結果をテストします。次の表に、各演算子で実行されるテストの種類を示します。

備考

演算子	
true	XPath 式が true に評価される場合。
not-true	true が false を返す場合。

例

```
<if-xpath op="true">add-attr[@attr-name='OU']/value[string(.)="Sales"]</if-xpath>
```

内容に使用できる要素

#PCDATA

属性

属性	有効値	デフォルト値
disabled		false
		false
op	true not-true	#REQUIRED

内容の規則

(#PCDATA)

親要素

要素	説明
and	
または	

include

指定した属性による参照先ポリシーからインクルード中のポリシーへ、ルールをランタイムにインクルードします。インクルードの時点では、これらのルールが、インクルードしているポリシーの一部のように扱われます。

備考

`name` 属性は、インクルードするポリシーを含む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 はインクルードするポリシーへの相対アドレスを使用することができます。

ポリシーは他のポリシーに含めることができるため、インクルードが再帰的になる場合があります。特定のポリシーを直接または間接的にそれ自体にインクルードするとエラーになります。

例

<policy> を参照してください。

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
<code>disabled</code>		<code>false</code>
<code>name</code>	CDATA インクルードするポリシーの名前です。	#REQUIRED

内容の宣言

なし

親要素

要素	説明
<code>policy</code>	

または

テストを実行し、結果を論理和で評価する一連のテストを指定します。<conditions> で囲まれた、一連の <or> 要素は論理積で評価されます。

例

<policy> を参照してください。

内容に使用できる要素

要素	説明
if-association	
if-attr	
if-class-name	
if-dest-attr	
if-dest-dn	
if-entitlement	
if-global-variable	
if-local-variable	
if-named-password	
if-op-attr	
if-op-property	
if-operation	
if-password	
if-src-attr	
if-src-dn	
if-xml-attr	
if-xpath	

属性

属性	有効値	デフォルト値
disabled		false

属性	有効値	デフォルト値
		false

内容の規則

親要素

要素	説明
arg-conditions	
conditions	rule

policy

ポリシーには、順序が指定された一連の **<rule>** が含まれています。**<rule>** にはテストする一連の **<conditions>** と、その条件を満たしたときに順次実行される一連の **<actions>** が含まれています。

備考

<policy> 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。**<policy>** 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。

次に、**<policy>** の基本的な操作についての概要を示します。

- XDS ドキュメントをその要素の操作に分割します。操作は、**<input>** または **<output>** の子であるすべての要素です。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。
- **<policy>** は、操作ごとに個別に適用されます。**<policy>** が各操作に順番に適用される時、その時点の操作が現在の操作になります。現在の操作である **src-dn**、**src-entry-id**、**dest-dn**、**dest-entry-id** で記述されたオブジェクトや、現在の操作からの関連付けであるオブジェクトは、現在のオブジェクトになります。
- **<rule>** のそれぞれは現在の操作に適用されます。直前の **<rule>** によって実行されたアクションが原因で、それ以降の **<rule>** が適用されなくなる場合を除き、すべての **<rule>** が現在の操作に適用されます。
- **<rule>** の **<conditions>** がテストされ、**<rule>** の **<conditions>** が条件を満たす場合、**<actions>** が適用されます。

変数

DirXML スクリプトでは、グローバルとローカルの2種類の変数がサポートされます。グローバル変数は、グローバル構成値から値を取得する変数で、ドライバまたはドライバセットとして定義されます。グローバル変数は、読み込み専用で定義されます。ローカル変数は、ポリシー単位で設定される変数です。ローカル変数は、ポリシーとドライバの2種類のスコープのいずれかの可視性を持つことができます。ポリシーのスコープの変数は、その変数が設定されたポリシーによる現在の操作の処理中にのみ可視です。ドライバのスコープの変数は、ドライバーが停止されるまで、同じドライバー内で実行されているすべての DirXML スクリプトポリシーから可視です。変数名は XML 名の規則に従う必要があります。

グローバル変数とローカル変数には、次のように自動的に定義されるものがいくつかあります。

名前	型	説明
dirxml.auto.driverdn	グローバル / 文字列	現在のドライバのスラッシュ形式の DN です。
dirxml.auto.driverguid		現在のドライバの GUID です。
dirxml.auto.treename		ローカル eDirectory インスタンスのツリー名です。

名前	型	説明
fromNDS	ポリシーローカル/ブーリアン	ソースデータストアが eDirectory の場合 true です。 ソースデータストアが接続されたアプリケーションの場合 false です。
destQueryProcessor	ポリシーローカル /java オブジェクト	ターゲットデータストアへのクエリに使用する XdsQueryProcessor のインスタンスです。
srcQueryProcessor		
destCommandProcessor		ターゲットデータストアへのクエリに使用する XdsCommandProcessor のインスタンスです。
srcCommandProcessor		
dnConverter		DNConverter のインスタンスです。
current-node	ポリシーローカル/ノードセット	<do-for-each> の各反復処理に対応するループ変数です。
current-value		do-reformat-op-attr
current-op		現在の操作です。 <do-set-local-variable> を使用したこの変数の設定は、<arg-node-set> によって指定された最初の操作を現在の操作に設定します。これは、現在のポリシーにおけるそれ以降の実行の間継続されるか、他の値が設定されるまで継続されます。新しい現在の操作は、元の現在の操作と同じ従属グループの要素で、現在のポリシーによって追加されている必要があります。

変数の拡張

多くの条件、アクション、およびトークンでは、それらの属性または内容での動的な変数の拡張がサポートされています。拡張がサポートされている場合には、`<variable-name>$` の形式の埋め込み参照が、指定された名前のローカルまたはグローバル変数の値と置き換えられます。`<variable-name>$` は変数名の規則に従う必要があります。指定された変数が存在しない場合、参照が空の文字列で置き換えられます。「\$」を使用する必要があります。変数の参照として認識させたくない場合は、「\$」を追加してエスケープする必要があります (例、`$$100.00` の貸し)。変数の拡張表現をサポートする内容と属性は、変数の拡張表現をサポートする句でコメントを記述します。

日付 / 時刻パラメータ

日付と時刻を扱うトークンには、日付と時刻の表現における形式、言語、およびタイムゾーンを扱う引数があります。日付の形式の引数は、2つの方法のいずれかで指定します。形式が「!」記号で始まる場合は、その形式は名前付き形式です。次の表に、規定された名前の定義を示します。

名前	説明
!CTIME	1970年1月1日午前0時からの秒数です。(eDirectoryの時刻の構文に適合します。)
!JTIME	1970年1月1日午前0時からのミリ秒の数です。(Javaの時刻に適合します。)
!FILETIME	1601年1月1日からの100ナノ秒間隔の数です。(Win32 FILETIMEに適合します。)
!FULL.TIME	言語固有のFULLの時刻形式です。
!LONG.TIME	言語固有のLONGの時刻形式です。
!MEDIUM.TIME	言語固有のMEDIUMの時刻形式です。
!SHORT.TIME	言語固有のSHORTの時刻形式です。
!FULL.DATE	言語固有のFULLの日付形式です。
!LONG.DATE	言語固有のLONGの日付形式です。
!MEDIUM.DATE	言語固有のMEDIUMの日付形式です。
!SHORT.DATE	言語固有のSHORTの日付形式です。
!FULL.DATETIME	言語固有のFULLの日付 / 時刻形式です。
!LONG.DATETIME	言語固有のLONGの日付 / 時刻形式です。
!MEDIUM.DATETIME	言語固有のMEDIUMの日付 / 時刻形式です。
!SHORT.DATETIME	言語固有のSHORTの日付 / 時刻形式です。

形式が「!」で開始されていない場合、その形式は `java.text.SimpleDateFormat` として認識されるパターンのカスタム日付 / 時刻形式として解釈されます。

言語の引数は、IETF RFC 3066 に準拠した ID によって指定できます。システムが認識する ID の一覧は、`java.util.Locale.getAvailableLocales()` を呼び出し、結果のアンダースコアをすべてハイフンに置き換えたものになります。言語の引数を省略するか空にすると、デフォルトのシステムの言語が使用されます。

タイムゾーン引数は、`java.util.TimeZone.getTimeZone()` によって認識される任意の ID で指定できます。システムが認識する ID の一覧は、`java.util.TimeZone.getAvailableIDs()` を呼び出して得ることができます。タイムゾーンの引数を省略するか空にすると、デフォルトのシステムのタイムゾーンが使用されます。

XPath の評価

一部の条件およびアクションの引数は、XPath 1.0 式を取ります。この XPath は、次のコンテキストで評価されます。

- ◆ 式の記述で特に指定されていない場合、コンテキストノードは、現在の操作になります。
- ◆ コンテキストの位置とサイズは 1 です。
- ◆ 使用可能な変数
 - ◆ Identity Manager メタディレクトリエンジン内のスタイルシートに対するパラメータとして使用可能なもの (現在のところ、fromNDS、srcQueryProcessor、destQueryProcessor、srcCommandProcessor、destCommandProcessor および dnConverter)。
 - ◆ グローバル設定変数。
 - ◆ ローカルポリシーの変数。
 - ◆ 異なる変数ソース間で名前が衝突している場合は、優先順位はローカル (ポリシースコープ)、ローカル (ドライバスコープ)、グローバルの順になります。
 - ◆ XPath 構文の制限により、名前にコロン記号を含む変数は XPath からアクセスできません。
- ◆ <policy> で宣言された名前空間が使用されます。
- ◆ 使用可能な機能
 - ◆ XPath 1.0 のすべての組み込み機能。
 - ◆ NXSL で提供されている Java 拡張機能。
 - ◆ プリフィックスを Java クラスに関連付けるための名前空間宣言は、<policy> で宣言する必要があります。

例

```
<policy> <description>My policy</description> <include
name="..\..\Library\My shared policy"/> <rule> <description>Rule to
disallow moving a user</description> <comment>This rule was added
because under no circumstances do we ever want to perform a move.</
comment> <conditions> <and> <if-class-name op="equal"
mode="nocase">User</if-class-name> <if-operation op="equal">move</
if-operation> </and> </conditions> <actions> <veto/> </
actions> </rule> <rule> <description>Rule to disallow operations on
a disabled user or group</description> <conditions> <or> <if-
class-name op="equal" mode="nocase">User</if-class-name> <if-
class-name op="equal" mode="nocase">Group</if-class-name> </or>
<or> <if-attr op="equal" mode="nocase" name="Login
Disabled">true</if-attr> </or> </conditions> <actions> <veto/
> </actions> </rule> </policy>
```

内容に使用できる要素

要素	説明
<code>description</code>	<code><policy></code> または <code><rule></code> の説明です。
<code>rule</code>	
<code>include</code>	他のポリシーからルールをインクルードします。

属性

なし

内容の規則

(description ? , (rule | include) *)

親要素

なし

rule

一連の <actions> と、<actions> を実行する一連の <conditions> を指定します。

例

<policy> を参照してください。

内容に使用できる要素

要素	説明
description	<policy> または <rule> の説明です。
comment	<rule> の詳細説明です。
conditions	<rule> のアクションを実行する条件です。
actions	<rule> によって実行されるアクションです。

属性

属性	有効値	デフォルト値
disabled		false
		false

内容の規則

(description ? , comment * , conditions , actions)

親要素

要素	説明
policy	

token-added-entitlement

現在の操作の、指定されたエンタイトルメントの許可された値に展開されます。親要素が `<arg-node-set>` の場合は、使用可能なすべての値が、ノードセット内の `<entitlement-impl>` 要素として返されます。それ以外の場合は、最初の使用可能な値が文字列として返されます。

例

```
<token-added-entitlement name="manager"/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
name	CDATA 変数の名前です。変数の拡張がサポートされます。	#REQUIRED false

内容の宣言

なし

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	

要素	説明
<code>token-base64-encode</code>	
<code>token-convert-time</code>	日付 / 時刻の形式を別の形式に変換します。
<code>token-escape-for-dest-dn</code>	
<code>token-escape-for-src-dn</code>	
<code>token-join</code>	
<code>token-lower-case</code>	
<code>token-map</code>	
<code>token-parse-dn</code>	
<code>token-replace-all</code>	
<code>token-replace-first</code>	
<code>token-split</code>	文字列をノードセットに分割します。
<code>token-substring</code>	文字列の部分文字列です。
<code>token-upper-case</code>	
<code>token-xml-parse</code>	
<code>token-xml-serialize</code>	

token-association

現在の操作で指定された関連付けの値に展開されます。

例

```
<token-association/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
		false

内容の宣言

なし

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	

要素	説明
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-attr

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に展開されます。<token-op-attr> と <token-src-attr> の結合と論理的に等価と見なすことができます。親要素が <arg-node-set> の場合は、使用可能なすべての <値> 要素が、ノードセット内のノードとして返されます。それ以外の場合は、最初の使用可能な値が文字列として返されます。

例

```
<token-attr name="OU"/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
name	CDATA 属性の名前です。変数の拡張がサポートされます。	#REQUIRED false

内容の宣言

なし

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	

要素	説明
<code>token-base64-encode</code>	
<code>token-convert-time</code>	
<code>token-escape-for-dest-dn</code>	
<code>token-escape-for-src-dn</code>	
<code>token-join</code>	
<code>token-lower-case</code>	
<code>token-map</code>	
<code>token-parse-dn</code>	
<code>token-replace-all</code>	文字列内の一致するすべての部分文字列を置換します。
<code>token-replace-first</code>	
<code>token-split</code>	
<code>token-substring</code>	文字列の部分文字列です。
<code>token-upper-case</code>	
<code>token-xml-parse</code>	
<code>token-xml-serialize</code>	

token-base64-decode

内容のトークンの結果を Base64 エンコードデータからバイトにデコードし、charset によって指定された文字セットを使用して、そのバイトを文字列に変換します。

例

```
<token-base64-decode charset="UTF-8"> <token-op-attr name="data"/></token-base64-decode>
```

内容に使用できる要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	日付 / 時刻の形式を別の形式に変換します。
token-escape-for-dest-dn	文字列をターゲット DN で使用できるように変換します。
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

属性

属性	有効値	デフォルト値
charset	CDATA デコードされたバイトを文字列に変換するために使用する文字セットです。指定しない場合、システムプロパティの <code>file.encoding</code> に指定されているエンコーディングが使用されます。変数の拡張がサポートされます。	#IMPLIED false

内容の規則

(token-added-entitlement | token-association | token-attr | token-base64-decode | token-base64-encode | token-char | token-class-name | token-convert-time | token-dest-attr | token-dest-dn | token-dest-name | token-document | token-entitlement | token-escape-for-dest-dn | token-escape-for-src-dn | token-generate-password | token-global-variable | token-join | token-local-variable | token-lower-case | token-named-password | token-map | token-op-attr | token-op-property | token-operation | token-parse-dn | token-password | token-removed-attr | token-removed-entitlement | token-replace-all | token-replace-first | token-resolve | token-src-attr | token-src-dn | token-src-name | token-substring | token-text | token-time | token-unique-name | token-unmatched-src-dn | token-upper-case | token-xml-parse | token-xml-serialize | token-xpath) +

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	

要素	説明
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-base64-encode

内容のトークンの結果を `charset` で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。

例

```
<token-base64-encode charset="UTF-8"> <token-op-attr name="Surname" /></token-base64-encode>
```

内容に使用できる要素

要素	説明
<code>token-added-entitlement</code>	
<code>token-association</code>	
<code>token-attr</code>	
<code>token-base64-decode</code>	
<code>token-base64-encode</code>	
<code>token-char</code>	
<code>token-class-name</code>	現在の操作のオブジェクトクラス名です。
<code>token-convert-time</code>	
<code>token-dest-attr</code>	
<code>token-dest-dn</code>	
<code>token-dest-name</code>	
<code>token-document</code>	
<code>token-entitlement</code>	
<code>token-escape-for-dest-dn</code>	
<code>token-escape-for-src-dn</code>	
<code>token-generate-password</code>	
<code>token-global-variable</code>	
<code>token-join</code>	
<code>token-local-variable</code>	
<code>token-lower-case</code>	
<code>token-map</code>	
<code>token-named-password</code>	
<code>token-op-attr</code>	

要素	説明
token-op-property	
token-operation	現在の操作の名前です。
token-parse-dn	
token-password	現在の操作のパスワード値です。
token-query	
token-removed-attr	
token-removed-entitlement	
token-replace-all	
token-replace-first	
token-resolve	
token-split	
token-src-attr	
token-substring	文字列の部分文字列です。
token-text	
token-time	
token-unique-name	
token-unmatched-src-dn	if-src-dn
token-upper-case	
token-xml-parse	
token-xml-serialize	
token-xpath	

属性

属性	有効値	デフォルト値
	CDATA	#REQUIRED
	文字列をバイトに変換するために使用する文字セットです。変数の拡張がサポートされます。	false

内容の規則

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-char

value で指定した Unicode コードポイントの文字に展開されます。

例

```
<token-char value="10"/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
		false
value	CDATA 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。	#REQUIRED

内容の宣言

なし

親要素

要素	説明
arg-actions	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	

要素	説明
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-class-name

現在の操作で指定されたオブジェクトクラスの名前に展開されます。

例

```
<token-class-name/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
		false

内容の宣言

なし

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	

要素	説明
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-convert-time

内容のトークンの結果に含まれる日付/時刻の表現を、`src-format`、`src-lang`、および `src-tz` で指定された形式、言語、およびタイムゾーンから、`dest-format`、`dest-lang`、および `dest-tz` で指定された形式、言語、およびタイムゾーンへ変換します。形式、言語、およびタイムゾーンの指定については、「[352 ページの「日付/時刻パラメータ」](#)」を参照してください。

例

```
<token-convert-time src-format="MM/dd/YYYY" src-lang="en-US" src-tz="MST" dest-format="dd/MM/YYYY" src-lang="en-US" src-tz="MST"/>
<token-op-attr name="birthdate"/></token-convert-time>
```

内容に使用できる要素

要素	説明
token-added-entitlement	
token-association	
token-attr	
token-base64-decode	
token-base64-encode	
token-char	
token-class-name	
token-convert-time	
token-dest-attr	
token-dest-dn	
token-dest-name	
token-document	
token-entitlement	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-generate-password	
token-global-variable	
token-join	
token-local-variable	
token-lower-case	
token-map	

要素	説明
token-named-password	
token-op-attr	
token-op-property	
token-operation	現在の操作の名前です。
token-parse-dn	
token-password	現在の操作のパスワード値です。
token-query	
token-removed-attr	
token-removed-entitlement	
token-replace-all	
token-replace-first	
token-resolve	
token-split	
token-src-attr	
token-src-dn	
token-src-name	
token-substring	文字列の部分文字列です。
token-text	
token-time	
token-unique-name	
token-unmatched-src-dn	if-src-dn
token-upper-case	
token-xml-parse	
token-xml-serialize	
token-xpath	

属性

属性	有効値	デフォルト値
dest-format	CDATA	#REQUIRED
	ターゲットの日付 / 時刻の形式です。変数の拡張がサポートされます。	

属性	有効値	デフォルト値
dest-lang	CDATA ターゲットの言語です (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。	#IMPLIED
dest-tz	CDATA ターゲットのタイムゾーンです (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。	#IMPLIED false
src-format	CDATA ソースの日付 / 時刻の形式です。変数の拡張がサポートされます。	#REQUIRED
src-lang	CDATA ソースの言語です (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。	#IMPLIED
src-tz	CDATA ソースのタイムゾーンです (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。	#IMPLIED

内容の規則

(token-added-entitlement | token-association | token-attr | token-base64-decode | token-base64-encode | token-char | token-class-name | token-convert-time | token-dest-attr | token-dest-dn | token-dest-name | token-document | token-entitlement | token-escape-for-dest-dn | token-escape-for-src-dn | token-generate-password | token-global-variable | token-join | token-local-variable | token-lower-case | token-named-password | token-map | token-op-attr | token-op-property | token-operation | token-parse-dn | token-password | token-removed-attr | token-removed-entitlement | token-replace-all | token-replace-first | token-resolve | token-src-attr | token-src-dn | token-src-name | token-substring | token-text | token-time | token-unique-name | token-unmatched-src-dn | token-upper-case | token-xml-parse | token-xml-serialize | token-xpath)

親要素

要素	説明
arg-association	

要素	説明
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-dest-attr

ターゲットデータストアにある現在オブジェクトの属性値に展開されます。親要素が `<arg-node-set>` の場合は、使用可能なすべての `<value>` 要素が、ノードセット内のノードとして返されます。

例

```
<token-dest-attr name="OU"/>
```

内容に使用できる要素

要素	説明
<code>arg-dn</code>	
<code>arg-association</code>	

属性

属性	有効値	デフォルト値
<code>class-name</code>	CDATA 読み取るオブジェクトのクラス名。	#IMPLIED
<code>name</code>	CDATA 属性の名前です。	#REQUIRED
		false

内容の規則

親要素

要素	説明
<code>arg-association</code>	
<code>arg-component</code>	
<code>arg-dn</code>	

要素	説明
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-dest-dn

現在の操作で指定されたターゲット DN、またはその一部分に展開されます。start または length が指定されていないか、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、start または length で指定された DN の一部分が使用されます。convert 属性が true に設定されている場合は、DN の形式がソースデータストアの形式に自動的に設定されます。

例

```
<token-dest-dn/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
convert		false
	ソースデータストアの DN 形式に変換します。	
length	CDATA	-1
	負の数を含む DN セグメントの数は、(セグメントの合計数 + length) + 1 で解釈されます。たとえば、5 セグメントの DN で長さが -1 の場合は $(5 + (-1)) + 1 = 5$ 、-2 の場合は $(5 + (-2)) + 1 = 4$ のようになります。	
		false
start	CDATA	0
	開始するセグメントのインデックスです。0 は、ルートに最も近いセグメントです。>0 はルートに最も近いセグメントからのオフセットです。-1 はリーフに最も近いセグメントです。<-1 はリーフに最も近いセグメントからルートに最も近いセグメントの方向へのオフセットです。	

内容の宣言

なし

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-dest-name

現在の操作で指定されたターゲット DN の非修飾の RDN に展開されます。

例

```
<token-dest-name/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
		false

内容の規則

(#PCDATA)

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	

要素	説明
<code>token-lower-case</code>	
<code>token-map</code>	
<code>token-parse-dn</code>	
<code>token-replace-all</code>	
<code>token-replace-first</code>	
<code>token-split</code>	
<code>token-substring</code>	文字列の部分文字列です。
<code>token-upper-case</code>	
<code>token-xml-parse</code>	
<code>token-xml-serialize</code>	

token-document

arg-string で指定された 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 には、インクルードするポリシーの URI への相対アドレスを使用することができます。URI または DN を適切な形式の XML ドキュメントへ解決できない場合には、結果が空のノードセットになります。

例

```
<token-document> <arg-string> <token-text>../MyDriver#DirXML-DriverFilter</token-text> </arg-string></token-document>
```

内容に使用できる要素

要素	説明
arg-string	

属性

属性	有効値	デフォルト値
		false

内容の規則

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	

要素	説明
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-entitlement

現在のオブジェクトに対する、指定されたエンタイトルメントの許可された値に展開されます。 [arg-node-setentitlement-impl](#)

例

```
<token-entitlement name="manager"/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
name	CDATA 属性の名前です。変数の拡張がサポートされます。	#REQUIRED false

内容の宣言

なし

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	

要素	説明
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-escape-for-dest-dn

内容のトークンを連結して展開された文字列が、ターゲットの DN 形式のルールに従った DN で使用されるようにエスケープされて展開されます。

例

```
<token-escape-for-dest-dn> <token-attr name="Surname"/></token-escape-for-dest-dn>
```

内容に使用できる要素

要素	説明
token-added-entitlement	
token-association	
token-attr	
token-base64-decode	
token-base64-encode	
token-char	
token-class-name	
token-convert-time	
token-dest-attr	
token-dest-dn	
token-dest-name	
token-document	
token-entitlement	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-generate-password	
token-global-variable	
token-join	
token-local-variable	
token-lower-case	
token-map	
token-named-password	
token-op-attr	

要素	説明
token-op-property	
token-operation	現在の操作の名前です。
token-parse-dn	
token-password	現在の操作のパスワード値です。
token-query	
token-removed-attr	
token-removed-entitlement	
token-replace-all	
token-replace-first	
token-resolve	
token-split	
token-src-attr	
token-src-dn	
token-src-name	
token-substring	文字列の部分文字列です。
token-text	
token-time	
token-unique-name	
token-unmatched-src-dn	if-src-dn
token-upper-case	
token-xml-parse	
token-xml-serialize	
token-xpath	

属性

属性	有効値	デフォルト値
		false

内容の規則

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-escape-for-src-dn

内容のトークンが連結されて展開された文字列が、ソースの DN 形式のルールに従った DN で使用できるようにエスケープされて展開されます。

例

```
<token-escape-for-src-dn> <token-attr name="Surname" /></token-escape-for-src-dn>
```

内容に使用できる要素

要素	説明
token-added-entitlement	
token-association	
token-attr	
token-base64-decode	
token-base64-encode	
token-char	
token-class-name	
token-convert-time	
token-dest-attr	
token-dest-dn	
token-dest-name	
token-document	
token-entitlement	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-generate-password	
token-global-variable	
token-join	
token-local-variable	
token-lower-case	
token-map	
token-named-password	
token-op-attr	

要素	説明
token-op-property	
token-operation	現在の操作の名前です。
token-parse-dn	
token-password	現在の操作のパスワード値です。
token-query	
token-removed-attr	
token-removed-entitlement	
token-replace-all	
token-replace-first	
token-resolve	
token-split	
token-src-attr	
token-src-dn	
token-src-name	
token-substring	文字列の部分文字列です。
token-text	
token-time	
token-unique-name	
token-unmatched-src-dn	if-src-dn
token-upper-case	
token-xml-parse	
token-xml-serialize	
token-xpath	

属性

属性	有効値	デフォルト値
		false

内容の規則

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-generate-password

policy-dn で指定されたパスワードポリシーに従った、ランダムに生成されたパスワードに展開されます。

例

```
<token-generate-password policy-dn="..\my password policy"/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
		false
policy-dn	CDATA nspmPasswordPolicy オブジェクトのスラッシュ形式の DN です。 変数の拡張がサポートされます。	#REQUIRED

内容の宣言

なし

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	Base64 データを文字列にデコードします。
token-base64-encode	文字列を Base64 データにエンコードします。

要素	説明
<code>token-convert-time</code>	日付 / 時刻の形式を別の形式に変換します。
<code>token-escape-for-dest-dn</code>	文字列をターゲット DN で使用できるように変換します。
<code>token-escape-for-src-dn</code>	文字列をソース DN で使用できるように変換します。
<code>token-join</code>	文字列にノードセットを結合します。
<code>token-lower-case</code>	文字列を小文字に変換します。
<code>token-map</code>	マッピングテーブルに従って文字列をマップします。
<code>token-parse-dn</code>	
<code>token-replace-all</code>	文字列内の一致するすべての部分文字列を置換します。
<code>token-replace-first</code>	
<code>token-split</code>	
<code>token-substring</code>	文字列の部分文字列です。
<code>token-upper-case</code>	
<code>token-xml-parse</code>	
<code>token-xml-serialize</code>	

token-global-variable

指定されたグローバル設定変数の値に展開されます。

例

```
<token-global-variable name="Fred"/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
name	CDATA 変数の名前です。変数の拡張がサポートされます。	#REQUIRED false

内容の宣言

なし

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	

要素	説明
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-join

内容のトークンの結果であるノードセット内のノードの値を `delimiter` で指定された文字で区切ります。 `csv` が `true` の場合、値に CSV の引用符を付けるルールが適用されます。

例

```
<!-- combine all of the members of the group into a CSV record -->  
><token-join delimiter="," csv="true"> <token-op-attr name="Member"/>  
></token-join>
```

内容に使用できる要素

要素	説明
<code>token-added-entitlement</code>	
<code>token-association</code>	
<code>token-attr</code>	
<code>token-base64-decode</code>	
<code>token-base64-encode</code>	
<code>token-char</code>	
<code>token-class-name</code>	
<code>token-convert-time</code>	
<code>token-dest-attr</code>	
<code>token-dest-dn</code>	
<code>token-dest-name</code>	
<code>token-document</code>	
<code>token-entitlement</code>	
<code>token-escape-for-dest-dn</code>	
<code>token-escape-for-src-dn</code>	
<code>token-generate-password</code>	
<code>token-global-variable</code>	
<code>token-join</code>	
<code>token-local-variable</code>	
<code>token-lower-case</code>	
<code>token-map</code>	
<code>token-named-password</code>	
<code>token-op-attr</code>	

要素	説明
token-op-property	
token-operation	現在の操作の名前です。
token-parse-dn	
token-password	現在の操作のパスワード値です。
token-query	
token-removed-attr	
token-removed-entitlement	
token-replace-all	
token-replace-first	
token-resolve	
token-split	
token-src-attr	
token-src-dn	
token-src-name	
token-substring	文字列の部分文字列です。
token-text	
token-time	
token-unique-name	
token-unmatched-src-dn	if-src-dn
token-upper-case	
token-xml-parse	
token-xml-serialize	
token-xpath	

属性

属性	有効値	デフォルト値
csv		false
	値に CSV の引用符を付けます。	
delimiter	CDATA	#REQUIRED
	結合された値を区切るために使用する文字列です。変数の拡張がサポートされます。	

属性	有効値	デフォルト値
		false

内容の規則

(token-added-entitlement | token-association | token-attr | token-base64-decode | token-base64-encode | token-char | token-class-name | token-convert-time | token-dest-attr | token-dest-dn | token-dest-name | token-document | token-entitlement | token-escape-for-dest-dn | token-escape-for-src-dn | token-generate-password | token-global-variable | token-join | token-local-variable | token-lower-case | token-named-password | token-map | token-op-attr | token-op-property | token-operation | token-parse-dn | token-password | token-removed-attr | token-removed-entitlement | token-replace-all | token-replace-first | token-resolve | token-src-attr | token-src-dn | token-src-name | token-substring | token-text | token-time | token-unique-name | token-unmatched-src-dn | token-upper-case | token-xml-parse | token-xml-serialize | token-xpath | token-query | token-split) +

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	

要素	説明
<code>token-split</code>	
<code>token-substring</code>	文字列の部分文字列です。
<code>token-upper-case</code>	
<code>token-xml-parse</code>	
<code>token-xml-serialize</code>	

token-local-variable

指定したローカル変数の値に展開されます。親要素が `<arg-node-set>` ではなく、値にノードセットが保持されている場合、ノードセットの文字列値が返されます。同じローカル変数がポリシースコープとドライバスコープの両方に存在する場合は、ポリシースコープの変数が優先されます。

例

```
<token-local-variable name="myVariable"/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
name	CDATA 変数の名前です。変数の拡張がサポートされます。	#REQUIRED false

内容の宣言

なし

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	

要素	説明
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-lower-case

内容のトークンの結果の各文字を小文字に変換します。

例

```
<token-lower-case> <token-attr name="Surname"/></token-lower-case>
```

内容に使用できる要素

要素	説明
token-added-entitlement	
token-association	
token-attr	
token-base64-decode	
token-base64-encode	
token-char	
token-class-name	
token-convert-time	
token-dest-attr	
token-dest-dn	
token-dest-name	
token-document	
token-entitlement	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-generate-password	
token-global-variable	
token-join	
token-local-variable	
token-lower-case	
token-map	
token-named-password	
token-op-attr	
token-op-property	
token-operation	現在の操作の名前です。

要素	説明
token-parse-dn	
token-password	現在の操作のパスワード値です。
token-query	
token-removed-attr	
token-removed-entitlement	
token-replace-all	
token-replace-first	
token-resolve	
token-split	
token-src-attr	
token-src-dn	
token-src-name	
token-substring	文字列の部分文字列です。
token-text	
token-time	
token-unique-name	
token-unmatched-src-dn	if-src-dn
token-upper-case	
token-xml-parse	
token-xml-serialize	
token-xpath	

属性

属性	有効値	デフォルト値
name	CDATA 属性の名前です。変数の拡張がサポートされます。	#REQUIRED false

内容の規則

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-map

table で指定されたマッピングテーブル内の、src 列で指定された値から、dest 列で指定された値へ、内容のトークンの結果をマップします。

備考

テーブル属性は、使用するマッピングテーブルを含む DirXML 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 にはインクルードするポリシーへの相対アドレスを使用することができます。

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、一致する各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。

例

```
<token-map table="./Department Table" src="dept" dest="code"> <token-op-attr name="OU"/></token-map>
```

内容に使用できる要素

要素	説明
token-added-entitlement	
token-association	
token-attr	
token-base64-decode	
token-base64-encode	
token-char	
token-class-name	
token-convert-time	
token-dest-attr	
token-dest-dn	
token-dest-name	
token-document	
token-entitlement	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-generate-password	
token-global-variable	

要素	説明
token-join	
token-local-variable	
token-lower-case	
token-map	
token-named-password	
token-op-attr	
token-op-property	
token-operation	現在の操作の名前です。
token-parse-dn	
token-password	現在の操作のパスワード値です。
token-query	
token-removed-attr	
token-removed-entitlement	
token-replace-all	
token-replace-first	
token-resolve	
token-split	
token-src-attr	
token-src-dn	
token-src-name	
token-substring	文字列の部分文字列です。
token-text	
token-time	
token-unique-name	
token-unmatched-src-dn	if-src-dn
token-upper-case	
token-xml-parse	
token-xml-serialize	
token-xpath	

属性

属性	有効値	デフォルト値
dest	CDATA ターゲット列の名前です。変数の拡張がサポートされます。	#REQUIRED false
src	CDATA ソース列の名前です。変数の拡張がサポートされます。	#REQUIRED
table	CDATA マッピングテーブルを含む DirXML リソースオブジェクトのスラッシュ形式の DN です。変数の拡張がサポートされます。	#REQUIRED

内容の規則

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	

要素	説明
<code>token-lower-case</code>	
<code>token-map</code>	
<code>token-parse-dn</code>	
<code>token-replace-all</code>	
<code>token-replace-first</code>	
<code>token-split</code>	
<code>token-substring</code>	文字列の部分文字列です。
<code>token-upper-case</code>	
<code>token-xml-parse</code>	
<code>token-xml-serialize</code>	

token-named-password

ドライバの名前付きパスワードに展開されます。

例

```
<token-named-password name="extraPassword"/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
name	CDATA 属性の名前です。変数の拡張がサポートされます。	#REQUIRED false

内容の宣言

なし

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	

要素	説明
<code>token-escape-for-dest-dn</code>	
<code>token-escape-for-src-dn</code>	
<code>token-join</code>	
<code>token-lower-case</code>	
<code>token-map</code>	
<code>token-parse-dn</code>	
<code>token-replace-all</code>	
<code>token-replace-first</code>	
<code>token-split</code>	
<code>token-substring</code>	文字列の部分文字列です。
<code>token-upper-case</code>	
<code>token-xml-parse</code>	
<code>token-xml-serialize</code>	

token-op-attr

現在の操作 (<add-attr>、<add-value>、または <attr>) の関連付けの値に展開されます。親要素が <arg-node-set> の場合は、使用可能なすべての <値> 要素が、ノードセット内のノードとして返されます。

例

```
<token-op-attr name="OU"/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
name	CDATA 属性の名前です。変数の拡張がサポートされます。	#REQUIRED false

内容の宣言

なし

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	

要素	説明
<code>token-convert-time</code>	
<code>token-escape-for-dest-dn</code>	
<code>token-escape-for-src-dn</code>	
<code>token-join</code>	
<code>token-lower-case</code>	
<code>token-map</code>	
<code>token-parse-dn</code>	
<code>token-replace-all</code>	
<code>token-replace-first</code>	
<code>token-split</code>	
<code>token-substring</code>	文字列の部分文字列です。
<code>token-upper-case</code>	
<code>token-xml-parse</code>	
<code>token-xml-serialize</code>	

token-op-property

現在の操作の指定された操作プロパティの値に展開されます。

例

```
<token-op-property name="myProperty"/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
name	CDATA 属性の名前です。変数の拡張がサポートされます。	#REQUIRED false

内容の宣言

なし

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	

要素	説明
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-operation

現在の操作の名前に展開されます。

例

```
<token-operation/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
		false

内容の宣言

なし

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	

要素	説明
<code>token-lower-case</code>	
<code>token-map</code>	
<code>token-parse-dn</code>	
<code>token-replace-all</code>	
<code>token-replace-first</code>	
<code>token-split</code>	
<code>token-substring</code>	文字列の部分文字列です。
<code>token-upper-case</code>	
<code>token-xml-parse</code>	
<code>token-xml-serialize</code>	

token-parse-dn

内容のトークンの連結を展開したもので指定された DN に展開されます。DN は、<src-dn-format> で指定された形式に従って解析されます。次に、start および length で指定された DN の一部分が、<dest-dn-format> で指定された形式に変換されます。

備考

<src-dn-delims> と <dest-dn-delims> は、DN 形式を指定するために使用します。区切り記号セットを構成する 8 文字は、次のように定義されます。

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

区切り記号セットが 8 文字を超える場合、超過した文字はすべて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

```
<token-parse-dn src-dn-format="src-dn" dest-dn-format="dest-dn"
start="0" length="-1"> <token-op-attr name="Group Membership"/></token-
parse-dn>
```

内容に使用できる要素

要素	説明
----	----

[token-added-entitlement](#)

[token-association](#)

[token-attr](#)

[token-base64-decode](#)

[token-base64-encode](#)

[token-char](#)

[token-class-name](#)

[token-convert-time](#)

要素	説明
token-dest-attr	
token-dest-dn	
token-dest-name	
token-document	
token-entitlement	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-generate-password	
token-global-variable	
token-join	
token-local-variable	
token-lower-case	
token-map	
token-named-password	
token-op-attr	
token-op-property	
token-operation	現在の操作の名前です。
token-parse-dn	
token-password	現在の操作のパスワード値です。
token-query	
token-removed-attr	
token-removed-entitlement	
token-replace-all	
token-replace-first	
token-resolve	
token-split	
token-src-attr	
token-src-dn	
token-src-name	
token-substring	文字列の部分文字列です。
token-text	
token-time	

要素	説明
token-unique-name	
token-unmatched-src-dn	if-src-dn
token-upper-case	
token-xml-parse	
token-xml-serialize	
token-xpath	

属性

属性	有効値	デフォルト値
	CDATA	#IMPLIED
	dest-dn-format= “custom” のとき、カスタム DN 区切り記号セットを指定します。	
	src-dn dest-dn dot qualified-dot slash qualified-slash ldap custom	dest-dn
	解析された DN の出力に使用される形式です。	
	CDATA	-1
	たとえば、5 セグメントの DN で長さが -1 の場合は $(5 + (-1)) + 1 = 5$ 、-2 の場合は $(5 + (-2)) + 1 = 4$ のようになります。	
		false
	CDATA	#IMPLIED
	src-dn-format= “custom” のとき、カスタム DN 区切り記号セットを指定します。	
		src-dn
	囲まれた DN の解析に使用される形式です。	

属性	有効値	デフォルト値
start	CDATA 開始するセグメントのインデックスです。0 は、ルートに最も近いセグメントです。>0 はルートに最も近いセグメントからのオフセットです。-1 はリーフに最も近いセグメントです。<-1 はリーフに最も近いセグメントからルートに最も近いセグメント方向へのオフセットです。	0

内容の規則

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	

要素	説明
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-password

現在の操作の指定されたパスワードに展開されます。

例

```
<token-password/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
		false

内容の宣言

なし

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	

要素	説明
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-query

<クエリー> 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。

備考

検索するデータストアは、`datastore` で指定します。

クエリのベースは、<arg-dn> または <arg-association> のいずれかで指定します。いずれも指定されていない場合、ベースはデータストアのルートになります。

クエリのスコープは、`scope` で指定します。

クエリのクラスは、`class-name` で指定します。省略すると、クエリですべてのクラスが検索されます。

検索する一連の属性は、<arg-match-attr> 要素で指定します。

返す一連の属性は、<arg-string> 要素で指定します。<arg-string> 要素が指定されていない場合は、属性の読み取りは行われません。<arg-string> のいずれかが、アスタリスク記号の場合、すべての属性が読み取られます。

`max-result-count` が指定された場合、<クエリー> の代わりに <query-ex> が発行され、結果がバッチで返されます。<do-for-each> のコンテキストで使用する場合、後続のバッチは自動的に取得されます。

例

```
<token-query scope="subordinates" class-name="User" datastore="dest">
<arg-dn> <token-text>Users/</token-text> <token-attr name="OU"/> </
arg-dn> <arg-match-attr name="CN"/> <arg-match-attr name="L"/> <arg-
value> <token-text>Provo</token-text> </arg-value> </arg-match-
attr> <arg-string> <token-text>Surname</token-text> </arg-string>
<arg-string> <token-text>Given Name</token-text> </arg-string></
token-query>
```

内容に使用できる要素

要素	説明
<code>arg-association</code>	
<code>arg-dn</code>	
<code>arg-match-attr</code>	attribute 引数との一致を検出します。
<code>arg-string</code>	

属性

属性	有効値	デフォルト値
class-name	CDATA クエリのクラス名です。変数の拡張がサポートされます。	#IMPLIED
datastore	src dest クエリの対象のデータストアです。	
	CDATA 1つのバッチで返す結果の最大数です。	#IMPLIED
		false
scope		

内容の規則

((arg-dn | arg-association) ? , arg-match-attr * , arg-string *)

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	

要素	説明
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-removed-attr

現在の操作で削除されている属性の値に展開されます (<remove-attr>)。親要素が <arg-node-set> の場合は、使用可能なすべての <value> 要素が、ノードセット内のノードとして返されます。

例

```
<token-removed-attr name="OU" />
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
name	CDATA 属性の名前です。	#REQUIRED false

内容の宣言

なし

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	

要素	説明
<code>token-escape-for-dest-dn</code>	
<code>token-escape-for-src-dn</code>	
<code>token-join</code>	
<code>token-lower-case</code>	
<code>token-map</code>	
<code>token-parse-dn</code>	
<code>token-replace-all</code>	
<code>token-replace-first</code>	
<code>token-split</code>	
<code>token-substring</code>	文字列の部分文字列です。
<code>token-upper-case</code>	
<code>token-xml-parse</code>	
<code>token-xml-serialize</code>	

token-removed-entitlement

現在の操作で削除された、指定されたエンタイトルメントの値に展開されます。[arg-node-setentitlement-impl](#)

例

```
<token-removed-entitlement name="manager"/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
name	CDATA 属性の名前です。	#REQUIRED false

内容の宣言

なし

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	

要素	説明
<code>token-escape-for-dest-dn</code>	
<code>token-escape-for-src-dn</code>	
<code>token-join</code>	
<code>token-lower-case</code>	
<code>token-map</code>	
<code>token-parse-dn</code>	
<code>token-replace-all</code>	
<code>token-replace-first</code>	
<code>token-split</code>	
<code>token-substring</code>	文字列の部分文字列です。
<code>token-upper-case</code>	
<code>token-xml-parse</code>	
<code>token-xml-serialize</code>	

token-replace-all

内容のトークンが連結されて展開された文字列において、`regex` で指定された正規表現によるすべての一致が、`replace-with` で指定された文字列で置換されて展開されます。

備考

<http://java.sun.com/j2se/1.4/docs/api/java/util/regex/Pattern.html> および [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)) 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。

例

```
<!-- remove escaping from DN in slash format --><token-replace-all  
regex="'(.)" replace-with="$1"> <token-dest-dn/></token-replace-all>
```

内容に使用できる要素

要素	説明
<code>token-added-entitlement</code>	
<code>token-association</code>	
<code>token-attr</code>	
<code>token-base64-decode</code>	
<code>token-base64-encode</code>	
<code>token-char</code>	
<code>token-class-name</code>	
<code>token-convert-time</code>	
<code>token-dest-attr</code>	
<code>token-dest-dn</code>	
<code>token-dest-name</code>	
<code>token-document</code>	
<code>token-entitlement</code>	
<code>token-escape-for-dest-dn</code>	
<code>token-escape-for-src-dn</code>	
<code>token-generate-password</code>	
<code>token-global-variable</code>	

要素	説明
token-join	
token-local-variable	
token-lower-case	
token-map	
token-named-password	
token-op-attr	
token-op-property	
token-operation	現在の操作の名前です。
token-parse-dn	
token-password	現在の操作のパスワード値です。
token-query	
token-removed-attr	
token-removed-entitlement	
token-replace-all	
token-replace-first	
token-resolve	
token-split	
token-src-attr	
token-src-dn	
token-src-name	
token-substring	文字列の部分文字列です。
token-text	
token-time	
token-unique-name	
token-unmatched-src-dn	if-src-dn
token-upper-case	
token-xml-parse	
token-xml-serialize	
token-xpath	

属性

属性	有効値	デフォルト値
		false
regex	CDATA 置換する部分文字列と一致検出する正規表現です。	#REQUIRED
replace-with	CDATA 置換文字列を指定する正規表現です。	#REQUIRED

内容の規則

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	

要素	説明
<code>token-replace-first</code>	
<code>token-split</code>	
<code>token-substring</code>	文字列の部分文字列です。
<code>token-upper-case</code>	
<code>token-xml-parse</code>	
<code>token-xml-serialize</code>	

token-replace-first

内容のトークンが連結されて展開された文字列において、`regex` で指定された正規表現による最初の一致が、`replace-with` で指定された文字列で置換されたものに展開されます。

備考

[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

```
<!-- change Full Name for "Surname, Given-Names" to "Given-Names Surname" --><token-replace-first regex="^(.*), (.*)$" replace-with="$2 $1"> <token-attr name="Full Name"/></token-replace-first>
```

内容に使用できる要素

要素	説明
token-added-entitlement	
token-association	
token-attr	
token-base64-decode	
token-base64-encode	
token-char	
token-class-name	
token-convert-time	
token-dest-attr	
token-dest-dn	
token-dest-name	
token-document	
token-entitlement	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-generate-password	
token-global-variable	

要素	説明
token-join	
token-local-variable	
token-lower-case	
token-map	
token-named-password	
token-op-attr	
token-op-property	
token-operation	現在の操作の名前です。
token-parse-dn	
token-password	現在の操作のパスワード値です。
token-query	
token-removed-attr	
token-removed-entitlement	
token-replace-all	
token-replace-first	
token-resolve	
token-split	
token-src-attr	
token-src-dn	
token-src-name	
token-substring	文字列の部分文字列です。
token-text	
token-time	
token-unique-name	
token-unmatched-src-dn	if-src-dn
token-upper-case	
token-xml-parse	
token-xml-serialize	
token-xpath	

属性

属性	有効値	デフォルト値
		false
regex	CDATA 置換する部分文字列と一致検出する正規表現です。	#REQUIRED
	CDATA 置換文字列を指定する正規表現です。	#REQUIRED

内容の規則

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	

要素	説明
<code>token-replace-first</code>	
<code>token-split</code>	
<code>token-substring</code>	文字列の部分文字列です。
<code>token-upper-case</code>	
<code>token-xml-parse</code>	
<code>token-xml-serialize</code>	

token-resolve

<token-split> が、<arg-dn> で指定された DN を関連付けキーに解決するか、<arg-association> で指定された関連付けキーを datastore で指定されたデータストアの DN に解決します。

例

```
<token-resolve datastore="src"> <arg-dn> <token-op-attr  
name="manager" /> </arg-dn></token-resolve>
```

内容に使用できる要素

要素	説明
arg-association	
arg-dn	

属性

属性	有効値	デフォルト値
		false

内容の規則

(arg-dn | arg-association)

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	

要素	説明
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-split

内容のトークンの結果を、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。csv が true の場合、文字列の解析時に CSV の引用符を付けるルールが適用されます。

例

```
<token-split delimiter="," csv="true"> <token-text>Doe,John,"Doe,John"</token-text></token-join>
```

内容に使用できる要素

要素	説明
----	----

token-added-entitlement

token-association

token-attr

token-base64-decode

token-base64-encode

token-char

token-class-name

token-convert-time

token-dest-attr

token-dest-dn

token-dest-name

token-document

token-entitlement

token-escape-for-dest-dn

token-escape-for-src-dn

token-generate-password

token-global-variable

token-join

token-local-variable

token-lower-case

token-map

token-named-password

token-op-attr

要素	説明
token-op-property	
token-operation	現在の操作の名前です。
token-parse-dn	
token-password	現在の操作のパスワード値です。
token-query	
token-removed-attr	
token-removed-entitlement	
token-replace-all	
token-replace-first	
token-resolve	
token-split	
token-src-attr	
token-src-dn	
token-src-name	
token-substring	文字列の部分文字列です。
token-text	
token-time	
token-unique-name	
token-unmatched-src-dn	if-src-dn
token-upper-case	
token-xml-parse	
token-xml-serialize	
token-xpath	

属性

属性	有効値	デフォルト値
csv		false
	CVS の引用符付けが適用されま す。	
	CDATA	#REQUIRED
	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。	

属性	有効値	デフォルト値
		false

内容の規則

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-src-attr

ソースデータストアにある現在オブジェクトの属性値に展開されます。親要素が <arg-node-set> の場合は、使用可能なすべての <値> 要素が、ノードセット内のノードとして返されます。

例

```
<token-src-attr name="OU"/>
```

内容に使用できる要素

要素	説明
arg-association	
arg-dn	

属性

属性	有効値	デフォルト値
class-name	CDATA 読み取るオブジェクトのクラス名。	#IMPLIED
name	CDATA 属性の名前です。	#REQUIRED
		false

内容の規則

親要素

要素	説明
arg-association	
arg-component	
arg-dn	

要素	説明
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-src-dn

現在の操作で指定されたソース DN、またはその一部分に展開されます。start または length が指定されていないか、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、start または length で指定された DN の一部分が使用されます。convert 属性が true に設定されている場合は、DN の形式がターゲットデータストアの形式に変換されます。

例

```
<token-src-dn/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
convert	ターゲットデータストアの DN 形式に変換します。 CDATA	false -1
start	CDATA >0 はルートに最も近いセグメントからのオフセットです。-1 はリーフに最も近いセグメントです。<-1 はリーフに最も近いセグメントからルートに最も近いセグメント方向へのオフセットです。	false 0

内容の宣言

なし

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-src-name

現在の操作で指定されたソース DN の非修飾の RDN に展開されます。

例

```
<token-src-name/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
		false

内容の宣言

なし

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	

要素	説明
<code>token-lower-case</code>	
<code>token-map</code>	
<code>token-parse-dn</code>	
<code>token-replace-all</code>	
<code>token-replace-first</code>	
<code>token-split</code>	
<code>token-substring</code>	文字列の部分文字列です。
<code>token-upper-case</code>	
<code>token-xml-parse</code>	
<code>token-xml-serialize</code>	

token-substring

内容のトークンを連結して展開された文字列の部分文字列に展開されます。

例

```
<token-substring start="0" length="1"> <token-attr name="Given Name"/></token-substring>
```

内容に使用できる要素

要素	説明
token-added-entitlement	
token-association	
token-attr	
token-base64-decode	
token-base64-encode	
token-char	
token-class-name	
token-convert-time	
token-dest-attr	
token-dest-dn	
token-dest-name	
token-document	
token-entitlement	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-generate-password	
token-global-variable	
token-join	
token-local-variable	
token-lower-case	
token-map	
token-named-password	
token-op-attr	
token-op-property	

要素	説明
token-operation	現在の操作の名前です。
token-parse-dn	
token-password	現在の操作のパスワード値です。
token-query	
token-removed-attr	
token-removed-entitlement	
token-replace-all	
token-replace-first	
token-resolve	
token-split	
token-src-attr	
token-src-dn	
token-src-name	
token-substring	文字列の部分文字列です。
token-text	
token-time	
token-unique-name	
token-unmatched-src-dn	if-src-dn
token-upper-case	
token-xml-parse	
token-xml-serialize	
token-xpath	

属性

属性	有効値	デフォルト値
	CDATA	-1
	負の数を含む文字の数は、(文字の合計数 + length) + 1 で解釈されます。たとえば、5 文字の文字列で長さが -1 の場合は (5 + (-1)) + 1 = 5、-2 の場合は (5 + (-2)) + 1 = 4 のようになります。	false

属性	有効値	デフォルト値
start	CDATA	0
	<p>開始する文字のインデックスです。0 は最初の文字です。>0 は文字列の開始からのオフセットです。-1 は最後の文字です。<-1 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。</p>	

内容の規則

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	

要素	説明
----	----

token-xml-parse

token-xml-serialize

token-text

内容のテキストに展開されます。

例

```
<token-text>Fred</token-text>
```

内容に使用できる要素

#PCDATA

属性

属性	有効値	デフォルト値
		false

内容の規則

(#PCDATA)

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	

要素	説明
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-time

format、lang、およびtzで指定された形式、言語、およびタイムゾーンで現在の日付に展開されます。[352 ページの「日付/時刻パラメータ」](#)

例

```
<token-time src-format="!CTIME" tz="UTC"/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
format	CDATA 日付/時刻の形式です。変数の拡張がサポートされます。	#REQUIRED
lang	CDATA 言語です (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。	#IMPLIED false
tz	CDATA タイムゾーンです (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。	#IMPLIED

内容の宣言

なし

親要素

要素	説明
arg-association	
arg-component	
arg-dn	

要素	説明
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-unique-name

指定した条件に従って、ターゲットデータストアで一意的なパターンに基づいた名前に展開されます。

備考

<arg-string> 要素はそれぞれ、指定した名前の作成に使用されるパターンを含んでいます。

指定した名前は、ターゲットデータストアに対する **name** 属性の値についてクエリを実行することでテストされます。クエリでは、クエリのベースとして <arg-dn> 要素または <arg-association> 要素が使用され、クエリのスコープとして **scope** が使用されます。ターゲットデータストアが識別ポールドであり、**name** が省略された場合は、擬似属性の「[Entry].rdn」に対して検索が実行されます。これは、名前の属性が何であるかにかかわらず、オブジェクトの **RDN** を指定します。ターゲットデータストアがアプリケーションの場合、**name** は必須です。

パターンは、**counter-use** と **counter-pattern** で示されるカウンタを使用して、または使用しないでテストできます。パターンをカウンタを使用してテストする場合、パターンは名前が見つかるまで付加されたカウンタで繰り返しテストされます。インスタンスが返されない場合、カウンタを使い果たします。カウンタの開始値は、**counter-start** で指定され、カウンタの最大値は、**counter-digits** の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パディングされます。カウンタを指定した桁数で表すことができない場合、カウンタが最大の値までカウントされたことと見なされます。

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。

指定した名前は次の順番でテストされます。

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

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

例

```
<token-unique-name name="CN" scope="subtree"
counter-start="1" counter-digits="2" counter-pad="true"
counter-pattern="first" counter-use="fallback" on-unavailable="error">
<arg-string> <token-upper-case> <token-substring start="0"
length="1"> <token-attr name="Given Name"/> </token-substring>
<token-attr name="Surname"/> </token-upper-case> </arg-string> <arg-
string> <token-upper-case> <token-substring start="0" length="1">
<token-attr name="Given Name"/> </token-substring> <token-
substring start="0" length="1"> <token-attr name="MI"/> </token-
substring> <token-attr name="Surname"/> </token-upper-case> </arg-
```

```
string> <arg-string> <token-upper-case> <token-attr name="Given
Name"/> <token-attr name="Surname"/> </token-upper-case> </arg-
string></token-substring>
```

内容に使用できる要素

要素	説明
arg-association	
arg-dn	
arg-string	

属性

属性	有効値	デフォルト値
counter-digits	CDATA カウンタの桁数です。	#IMPLIED
counter-pad	カウンタの右側の桁へのゼロパディングを有効または無効にします。	true
counter-pattern	first last all カウンタで使用するパターンを選択します。 first: カウンタで最初のパターンだけを使用します。 last: カウンタで最後のパターンだけを使用します。 all: カウンタですべてのパターンを使用します。	last
counter-start	CDATA カウンタを開始する数です。	1

属性	有効値	デフォルト値
counter-use	<p>always never fallback カウンタを使用するタイミングを選択します。</p> <p>never: カウンタを使用しません。</p> <p>always: counter-pattern に示されているパターンでカウンタを常に使用します。</p> <p>fallback: カウンタなしのすべてのパターンで見つからなかった後に、counter-pattern に示されているパターンでカウンタを使用します。</p>	fallback
name	<p>CDATA</p> <p>一意性をチェックする属性の名前です。</p>	<p>#IMPLIED</p> <p>false</p>
on-unavailable	<p>ignore warning error fatal</p> <p>固有の名前を作成できない場合に実行するアクションです。</p> <p>ignore: 無視して空の名前を返します。</p> <p>warning: 警告を発行して空の名前を返します。</p> <p>error: エラーを生成し、現在のトランザクションを中止します。</p> <p>fatal: 致命的エラーを生成し、ドライバをシャットダウンします。</p>	error
scope	<p>subordinates subtree</p> <p>一意性をチェックするスコープです。</p>	

内容の規則

((arg-dn | arg-association) ? , arg-string +)

親要素

要素	説明
arg-association	

要素	説明
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-unmatched-src-dn

(短絡評価を考慮した)このルールの中で、<if-src-dn>の最後の検索で一致しなかったDNの部分に対応する、現在の操作のソースDNの部分に展開されます。一致するものがなかった場合は、DN全体が使用されます。

例

```
<token-unmatched-src-dn/>
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
convert		false
		false

内容の宣言

なし

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	

要素	説明
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-upper-case

内容のトークンが連結され展開された文字列で、すべての文字が大文字に変換された文字列に展開されます。

例

```
<token-upper-case> <token-attr name="Surname" /></token-upper-case>
```

内容に使用できる要素

要素	説明
token-added-entitlement	
token-association	
token-attr	
token-base64-decode	
token-base64-encode	
token-char	
token-class-name	
token-convert-time	
token-dest-attr	
token-dest-dn	
token-dest-name	
token-document	
token-entitlement	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-generate-password	
token-global-variable	
token-join	
token-local-variable	
token-lower-case	
token-map	
token-named-password	
token-op-attr	
token-op-property	

要素	説明
token-operation	現在の操作の名前です。
token-parse-dn	
token-password	現在の操作のパスワード値です。
token-query	
token-removed-attr	
token-removed-entitlement	
token-replace-all	
token-replace-first	
token-resolve	
token-split	
token-src-attr	
token-src-dn	
token-src-name	
token-substring	文字列の部分文字列です。
token-text	
token-time	
token-unique-name	
token-unmatched-src-dn	if-src-dn
token-upper-case	
token-xml-parse	
token-xml-serialize	
token-xpath	

属性

属性	有効値	デフォルト値
		false

内容の規則

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-xml-parse

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

例

```
<token-xml-parse> <token-base64-decode charset="UTF-8"> <token-op-attr name="data"/> </token-base64-decode></token-xml-parse>
```

内容に使用できる要素

要素	説明
token-added-entitlement	
token-association	
token-attr	
token-base64-decode	
token-base64-encode	
token-char	
token-class-name	
token-convert-time	
token-dest-attr	
token-dest-dn	
token-dest-name	
token-document	
token-entitlement	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-generate-password	
token-global-variable	
token-join	
token-local-variable	
token-lower-case	
token-map	
token-named-password	
token-op-attr	

要素	説明
token-op-property	
token-operation	現在の操作の名前です。
token-parse-dn	
token-password	現在の操作のパスワード値です。
token-query	
token-removed-attr	
token-removed-entitlement	
token-replace-all	
token-replace-first	
token-resolve	
token-split	
token-src-attr	
token-src-dn	
token-src-name	
token-substring	文字列の部分文字列です。
token-text	
token-time	
token-unique-name	
token-unmatched-src-dn	if-src-dn
token-upper-case	
token-xml-parse	
token-xml-serialize	
token-xpath	

属性

属性	有効値	デフォルト値
		false

内容の規則

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-xml-serialize

内容のトークンのノードセットの結果を 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、適切な形式の XML ドキュメントか解析済みの適切な形式の標準エンティティになります。解析済みの標準エンティティは、スタンドアロン XML ドキュメントとして解析できません。

例

```
<token-xml-serialize> <token-xpath expression="."/></token-xml-serialize>
```

内容に使用できる要素

要素	説明
token-added-entitlement	
token-association	
token-attr	
token-base64-decode	
token-base64-encode	
token-char	
token-class-name	
token-convert-time	
token-dest-attr	
token-dest-dn	
token-dest-name	
token-document	
token-entitlement	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-generate-password	
token-global-variable	
token-join	
token-local-variable	
token-lower-case	
token-map	
token-named-password	

要素	説明
token-op-attr	
token-op-property	
token-operation	現在の操作の名前です。
token-parse-dn	
token-password	現在の操作のパスワード値です。
token-query	
token-removed-attr	
token-removed-entitlement	
token-replace-all	
token-replace-first	
token-resolve	
token-split	
token-src-attr	
token-src-dn	
token-src-name	
token-substring	文字列の部分文字列です。
token-text	
token-time	
token-unique-name	
token-unmatched-src-dn	if-src-dn
token-upper-case	
token-xml-parse	
token-xml-serialize	
token-xpath	

属性

属性	有効値	デフォルト値
		false

内容の規則

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-password	
arg-string	
arg-value	
token-base64-decode	
token-base64-encode	
token-convert-time	
token-escape-for-dest-dn	
token-escape-for-src-dn	
token-join	
token-lower-case	
token-map	
token-parse-dn	
token-replace-all	
token-replace-first	
token-split	
token-substring	文字列の部分文字列です。
token-upper-case	
token-xml-parse	
token-xml-serialize	

token-xpath

XPath 1.0 の式の評価結果に展開されます。親要素が `<arg-node-set>` で、式がノードセットを返す場合、ノードセットはそのままの状態が返されます。親要素が `<arg-node-set>` で、式がノードセット以外のデータ型を返す場合、結果の文字列値を含むテキストノードが返されます。これら以外の場合、結果の文字列値が返されます。

例

```
<token-xpathexpression="*[@attr-name='OU']//value[starts-with(string(.),'xxx')]" />
```

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
	CDATA	#REQUIRED
	XPath 式です。	false

内容の宣言

なし

親要素

要素	説明
arg-association	
arg-component	
arg-dn	
arg-node-set	
arg-object	Java オブジェクトの引数です。
arg-password	
arg-string	
arg-value	

要素	説明
<code>token-base64-decode</code>	
<code>token-base64-encode</code>	
<code>token-convert-time</code>	
<code>token-escape-for-dest-dn</code>	
<code>token-escape-for-src-dn</code>	
<code>token-join</code>	
<code>token-lower-case</code>	
<code>token-map</code>	
<code>token-parse-dn</code>	
<code>token-replace-all</code>	
<code>token-replace-first</code>	文字列内の一致する部分文字列を 1 つ置換します。
<code>token-split</code>	
<code>token-substring</code>	文字列の部分文字列です。
<code>token-upper-case</code>	
<code>token-xml-parse</code>	
<code>token-xml-serialize</code>	

DirXML エンタイトルメント DTD

6

DirXML エンタイトルメントは、概念的には名前付きフラグであり、特定のアクションが実行されるように Identity Manager ドライバの環境設定を指定します。通常、このアクションは、接続システムのリソースに対するアクセスの許可に関連するものになります。このため、エンタイトルメントは、(役割ベースエンタイトルメントを実現するために)、次の3つの基本的なアクションに使用されてきました。

- ◆ 接続システムのアカунトの作成、削除、および無効化。
- ◆ 接続システムのアカунトグループに対するメンバーシップの追加と削除。
- ◆ 接続システムのアカунトに対する属性値の追加と設定。

エンタイトルメントは、DirXML-Driver オブジェクトに含まれる、eDirectory DirXML-Entitlement オブジェクトで実現されます。DirXML-Entitlement オブジェクトを同じコンテナで包含することによって、エンタイトルメントと実装する Identity Manager ドライバの環境設定との対応関係を確立できます。DirXML-Entitlement オブジェクトの名前が、エンタイトルメントの名前になります。DirXML-Entitlement オブジェクトの XmlData 属性には、ルート要素が <entitlement> の XML ドキュメントが格納されます。

エンタイトルメントで、eDirectory オブジェクトへアクセスを許可したり取り消したりするには、補助クラスの DirXML-EntitlementRecipient と関連する DirXML-EntitlementRef 属性を eDirectory オブジェクトに追加します。DirXML-EntitlementRef 属性は、SYN_PATH 構文で記述され、「書き込み管理」が行われています。このパス構文の値の「volume」(DN) 部分は、DirXML-Entitlement オブジェクトを参照します。この属性は書き込み管理されているため、eDirectory オブジェクトの DirXML-EntitlementRef 属性の値を設定するエージェントには、書き込み対象オブジェクトの DirXML-EntitlementRef 属性に対する書き込みアクセス権が必要です。さらに、DirXML-EntitlementRef 値の DN の部分から参照される、DirXML-Entitlement オブジェクトの ACL 属性に対する書き込みアクセス権が必要です。DirXML-EntitlementRef 属性の「path」(文字列) 部分には、ルート要素が <ref> の XML ドキュメントが格納されます。DirXML-EntitlementRef 属性の「nameSpace」(整数) 部分は、フラグを保持するためのビットマスクとして使用されます。32 ビット整数のビット 0 は、このフラグ値として使用され、状態ビットとして認識されます。0 は取り消し、1 は許可を意味します。ビット 1 は、アップグレードプロセスによって許可されたエンタイトルメントを示すフラグとして使用され、アップグレードビットとして認識されます。1 は、エンタイトルメントが以前の形式で前に許可されており、エンタイトルメントの状態に変化がないことを意味します。ビット 2～31 は、将来の使用のために予約されています。

Identity Manager ドライバの環境設定によって、エンタイトルメントの許可または取り消しアクションが(正常または異常に)終了すると、結果が DirXML-EntitlementResult 属性によって eDirectory オブジェクトに書き込まれます。DirXML-EntitlementResult は、複数値の SYN_OCTET_STRING であり、ルート要素が <result> の XML ドキュメントを格納します。

エンタイトルメントは、エンタイトルメントの許可または取り消しを実現されるように、Identity Manager ドライバに特定のリソースを許可するよう指定する単なるフラグであるため、接続アプリケーションでリソースへのアクセス権の実際の許可または取り消しを処理するポリシーがドライバに必要です。DirXML スクリプトでは、エンタイトルメントのポリシーの実装が明確にサポートされています。<if-entitlement> 条件は、指定したエンタ

イトルメントが許可または変更されているかを判定するために使用します。<token-entitlement>、<token-added-entitlement>、および<token-removed-entitlement> トークンは、許可または取り消されたエンタイトルメントのリストの取得に使用します。<do-implement-entitlement> アクションは、エンタイトルメントを実装するポリシーアクションの指定に使用します。これにより、エンタイトルメントの結果を DirXML-EntitlementResult に自動的に記録させることができます。エンタイトルメントのトークンは、空のノードセットか、1 つ以上の<entitlement-impl> 要素を含むノードセットを返します。エンタイトルメントの情報の取得や、<do-implement-entitlement> に引数を渡すために使用することができます。

DirXML 要素 DTD のすべての要素のリストについては、「[476 ページの「DirXML エンタイトルメント DTD の要素」](#)」を参照してください。

6.1 DirXML エンタイトルメント DTD の要素

要素	説明
description	記述の結果を問い合わせます。
display-name	表示名の結果を問い合わせます。
dn	エンタイトルメントオブジェクトの DN です。
ent-value	結果の値を問い合わせます。
entitlement	エンタイトルメントの定義です。
entitlement-impl	エンタイトルメントの実装マーカーです。
id	エンタイトルメントの許可エージェントの関連 ID です。
item	エンタイトルメントのキャッシュされたクエリ結果項目です。
item-description	エンタイトルメントのキャッシュされたクエリ結果項目の説明です。
item-display-name	エンタイトルメントのキャッシュされたクエリ結果項目の表示名です。
item-value	エンタイトルメントのキャッシュされたクエリ結果項目の値です。
項目	エンタイトルメントのキャッシュされたクエリ結果です。
msg	エンタイトルメントの結果のステータスメッセージです。
param	エンタイトルメントのパラメータ値です。
query-app	エンタイトルメントの有効な値の定義を問い合わせます。
query-xml	XDS クエリです。
ref	エンタイトルメントの参照です。

要素	説明
result	エンタイトルメントの結果です。
result-set	クエリ結果の解釈です。
src	エンタイトルメントの許可エージェントのタイプです。
state	エンタイトルメントの状態です。
status	エンタイトルメントの結果のステータスレベルです。
timestamp	エンタイトルメントの結果のタイムスタンプです。
token-association	クエリ結果の関連付けの値を使用します。
token-attr	クエリ結果の属性値を使用します。
token-src-dn	クエリ結果の src-dn 値を使用します。
value	列挙値です。
values	エンタイトルメントの有効な値です。

description

説明の取得方法を指定します。この指定は、上位要素の <entitlement> の接続アプリケーションから動的に有効値のリストを取得する際に、XDS クエリが返す XDS インスタンス要素ごとに適用されます。

例

<entitlement> を参照してください。

内容に使用できる要素

要素	説明
token-association	クエリ結果の関連付けの値を使用します。
token-src-dn	クエリ結果の src-dn 値を使用します。
token-attr	クエリ結果の属性値を使用します。

内容の規則

(token-association | token-src-dn | token-attr)

親要素

要素	説明
result-set	クエリ結果の翻訳です。

display-name

表示名の取得方法を指定します。この指定は、上位要素の <entitlement> の接続アプリケーションから動的に有効値のリストを取得する際に、XDS クエリが返す XDS インスタンス要素ごとに適用されます。

例

<entitlement> を参照してください。

内容に使用できる要素

要素	説明
token-association	クエリ結果の関連付けの値を使用します。
token-src-dn	クエリ結果の src-dn 値を使用します。
token-attr	クエリ結果の属性値を使用します。

属性

なし

内容の規則

(token-attr | token-src-dn | token-association)

親要素

要素	説明
result-set	クエリ結果の翻訳です。

dn

上位要素の **<result>** が参照する DirXML-Entitlement オブジェクトの LDAP 形式の DN が挿入されます。

例

<result> を参照してください。

内容に使用できる要素

#PCDATA

属性

なし

内容の規則

(#PCDATA)

親要素

要素	説明
result-set	クエリ結果の翻訳です。

ent-value

実パラメータの値の取得方法を指定します。この指定は、上位要素の <entitlement> の接続アプリケーションから動的に有効値のリストを取得する際に、XDS クエリが返す XDS インスタンス要素ごとに適用されます。

例

<entitlement> を参照してください。

内容に使用できる要素

要素	説明
token-association	クエリ結果の関連付けの値を使用します。
token-src-dn	クエリ結果の src-dn 値を使用します。
token-attr	クエリ結果の属性値を使用します。

属性

なし

内容の規則

(token-association | token-src-dn | token-attr)

親要素

要素	説明
result-set	クエリ結果の翻訳です。

entitlement

DirXML-Entitlement オブジェクトの XmlData 属性に格納されるエンタイトルメント定義です。DirXML-Entitlement オブジェクトの RDN が、エンタイトルメントの実際の名前になります。エンタイトルメント定義は、UI エージェントで使用するエンタイトルメントの表示名と説明を定義します。また、エンタイトルメントに1つのパラメータ値が必要かどうかを定義することもできます。パラメータ値が必要な場合は、パラメータの有効値のリストを指定したり、関連アプリケーションに発行するクエリを指定して、パラメータの有効値のリストを取得したりできます。

例

```
<entitlement conflict-resolution="union"    description="Recipient is
entitled to an account in the connected application"    display-
name="User Account"/><entitlement conflict-resolution="union"
description="Groups that the recipient is entitled to belong to in the
connected application"    display-name="Group Membership"> <values>
<query-app>        <query-xml>            <nds dtd-version="2.0">
<input>            <query class-name="Group" scope="subtree">
<search-class class-name="Group"/>            <read-attr attr-
name="Description"/>            </query>            </input>            </nds>
</query-xml>        <result-set>            <display-name>            <token-
src-dn/>            </display-name>            <description>            <token-
attr attr-name="Description"/>            </description>            <ent-value>
<token-association/>            </ent-value>            </result-set>    </query-
app> </values></entitlement><entitlement conflict-
resolution="priority"    description="The Musical Instrument played by
the recipient"    display-name="Musical Instrument"> <values multi-
valued="false">    <value>Trumpet</value>    <value>Clarinet</value>
<value>Tuba</value>    <value>Trombone</value>    <value>Flute</value>
<value>Violin</value>    </values></entitlement>
```

内容に使用できる要素

要素	説明
<code>values</code>	エンタイトルメントの有効な値です。

属性

属性	有効値	デフォルト値
conflict-resolution	priority union エンタイトルメントドライバで使用する矛盾の解決方法です。エンタイトルメントと役割ベースのエンタイトルメントを併用しているときに、エンタイトルメントが複数の役割を通じて許可される場合に使用されます。 union: 許可される役割のすべてのパラメータ値が、対象者に許可されます。 priority: 許可される役割のうち優先度が最も高い役割のパラメータ値だけが対象者に許可されます。	priority
description	CDATA UI エージェントに表示されるエンタイトルメントの説明です。	#REQUIRED
display-name	CDATA UI エージェントに表示されるエンタイトルメントの名前です。	#REQUIRED

内容の規則

(values ?)

親要素

なし

entitlement-impl

DirXML スクリプト内の許可または取り消されたエンタイトルメントを表します。

例

```
<entitlement-impl id="xxx" src="AF" state="1" name="Group"
src-dn="\MYTREE\Novell\Users\Fred" src-entry-id="65535"
>cn=Managers,o=People</entitlement-impl>
```

内容に使用できる要素

#PCDATA

属性

属性	有効値	デフォルト値
id	CDATA 許可エージェントによって指定される ID です。	#IMPLIED
名前	CDATA エンタイトルメントの名前です。	#REQUIRED
src	CDATA 許可エージェントです。	#REQUIRED
src-dn	CDATA エンタイトルメント対象者のスラッシュ形式の eDirectory™ DN です。	#REQUIRED
src-entry-id	CDATA エンタイトルメント対象者のスラッシュ形式の eDirectory エントリ ID です。	#IMPLIED
state	0 1 許可または取り消しの状態です。 1- 許可 0- 取り消し	#REQUIRED

内容の規則

(#PCDATA)

親要素

なし

id

許可または取り消されたエンタイトルメントに指定される **ID** です。許可または取り消しを実行したエージェントによって指定され、許可または取り消されたエンタイトルメントの結果でも使用されます。id の意味はエンタイトルメントのシステムでは認識されません。id は、許可または取り消しを実行したエージェントが、エンタイトルメントの結果と元の許可または取り消しとの相関を確立するものです。

例

<ref> を参照してください。

内容に使用できる要素

#PCDATA

属性

なし

内容の規則

(#PCDATA)

親要素

要素	説明
ref	エンタイトルメントの参照です。
result	エンタイトルメントの結果です。

item

エンタイトルメントの有効パラメータ値を動的に検出するクエリでキャッシュされた、クエリ結果項目です。

例

<項目>を参照してください。

内容に使用できる要素

要素	説明
<code>item-display-name</code>	エンタイトルメントのキャッシュされたクエリ結果項目の表示名です。
<code>item-description</code>	エンタイトルメントのキャッシュされたクエリ結果項目の説明です。
<code>item-value</code>	エンタイトルメントのキャッシュされたクエリ結果項目の値です。

属性

なし

内容の規則

(`item-display-name ?` , `item-description ?` , `item-value`)

親要素

要素	説明
<code>項目</code>	エンタイトルメントクエリのキャッシュされた結果です。

item-description

エンタイトルメントの有効パラメータ値を動的に検出するクエリでキャッシュされた、クエリ結果項目の説明です。

例

<項目>を参照してください。

内容に使用できる要素

#PCDATA

属性

なし

内容の規則

(#PCDATA)

親要素

要素	説明
item	エンタイトルメントのキャッシュされたクエリ結果項目です。

item-display-name

エンタイトルメントの有効パラメータ値を動的に検出するクエリでキャッシュされた、クエリ結果項目の表示名です。

例

<項目>を参照してください。

内容に使用できる要素

#PCDATA

属性

なし

内容の規則

(#PCDATA)

親要素

要素	説明
<code>item</code>	エンタイトルメントのキャッシュされたクエリ結果項目です。

item-value

エンタイトルメントの有効パラメータ値を動的に検出するクエリでキャッシュされた、クエリ結果項目のパラメータ値です。

例

<項目>を参照してください。

内容に使用できる要素

#PCDATA

属性

なし

内容の規則

(#PCDATA)

親要素

要素	説明
item	エンタイトルメントのキャッシュされたクエリ結果項目です。

項目

エンタイトルメントの有効パラメータ値を動的に検出するクエリでキャッシュされた、クエリ結果項目を保持します。DirXML-Entitlement オブジェクトの DirXML-SPCacheExternalQuery 属性に保存されます。

例

```
<items> <item> <item-display-name>cn=Managers,o=People</display-name>
<item-description>Managers</description> <item-
value>cn=Managers,o=People</value > </item> <item> <item-display-
name>cn=Contractors,o=People</display-name> <item-
description>Contractors</description> <item-
value>cn=Contractors,o=People</value > </item></items>
```

内容に使用できる要素

要素	説明
item	エンタイトルメントのキャッシュされたクエリ結果項目です。

属性

なし

内容の規則

(item *)

親要素

なし

msg

上位要素の <result> が参照する、エンタイトルメントの許可または取り消しのステータスメッセージが挿入されます。

例

<result> を参照してください。

内容に使用できる要素

ANY

属性

なし

内容の規則

ANY

親要素

要素	説明
result	エンタイトルメントの結果です。

param

エンタイトルメントのパラメータ値です。許可または取り消されたエンタイトルメントおよび RBE ポリシーと関連付けられているエンタイトルメントで使用されます。また、エンタイトルメントの許可や取り消しの結果で使用されます。

例

<ref> を参照してください。

内容に使用できる要素

#PCDATA

属性

なし

内容の規則

(#PCDATA)

親要素

要素	説明
ref	エンタイトルメントの参照です。
result	エンタイトルメントの結果です。

query-app

query-app は、存在する場合、XDS クエリを指定します。XDS クエリは、上位要素の `<entitlement>` の接続アプリケーションから有効値のリストを動的に取得するために使用します。また、クエリから返される各 XDS インスタンス要素の表示名、記述、および実パラメータ値を取得する方法も指定できます。

例

`<entitlement>` を参照してください。

内容に使用できる要素

要素	説明
<code>query-xml</code>	XDS クエリです。
<code>result-set</code>	クエリ結果の翻訳です。

属性

なし

内容の規則

(query-xml , result-set)

親要素

要素	説明
<code>values</code>	エンタイトルメントの有効な値です。

query-xml

XDS クエリを指定します。XDS クエリは、上位要素の <entitlement> の接続アプリケーションから有効値のリストを動的に取得するために使用します。

例

<entitlement> を参照してください。

内容に使用できる要素

ANY

属性

なし

内容の規則

ANY

親要素

要素	説明
query-app	エンタイトルメントの有効な値の定義を問い合わせます。

ref

DirXML-EntitlementRef 属性の「path」(文字列)部分の値です。DirXML-EntitlementRecipient で使用される場合、ref は、許可または取り消されたエンタイトルメントを表します。ref は、許可または取り消しを実行するエージェントの情報とパラメータ値(エンタイトルメントに必要な場合)を含みます。DirXML-SharedProfile (RBE ポリシーまたは役割)で使用される場合、ref は役割によって許可されるパラメータ値を指定するためだけに使用されます。

例

```
<ref> <src>RBE</src> <id>{26dfb70f-0371-4fe2-a67f-bc101101e5d7}</id>  
<param>cn=Managers,o=People</param></ref>
```

内容に使用できる要素

要素	説明
src	エンタイトルメントの許可エージェントのタイプです。
id	エンタイトルメントの許可エージェントの関連 ID です。
param	エンタイトルメントのパラメータ値です。

属性

なし

内容の規則

(src ?, id ?, param ?)

親要素

なし

result

DirXML-EntitlementRecipient の DirXML-EntitlementResult 属性の値が出力されます。result には、ポリシーの実装に従って、エンタイトルメントの許可または取り消しの結果が挿入されます。

例

```
<result> <dn>cn=Group,cn=LDAP Driver,cn=DriverSet,o=novell</dn>
<src>RBE</src> <id>{26dfb70f-0371-4fe2-a67f-bc101101e5d7}</id>
<param>cn=Managers,o=People</param> <state>1</state> <status>error</
status> <msg>Access denied</msg> <timestamp>1112101901523</
timestamp></result>
```

内容に使用できる要素

要素	説明
dn	エンタイトルメントオブジェクトの DN です。
src	エンタイトルメントの許可エージェントのタイプです。
id	エンタイトルメントの許可エージェントの関連 ID です。
param	エンタイトルメントのパラメータ値です。
state	エンタイトルメントの状態です。
status	エンタイトルメントの結果のステータスレベルです。
msg	エンタイトルメントの結果のステータスメッセージです。
timestamp	エンタイトルメントの結果のタイムスタンプです。

属性

なし

内容の規則

(dn , src , id ? , param ? , state , status , msg ? , timestamp)

親要素

なし

result-set

表示名、説明、実パラメータの値の取得方法を指定します。この指定は、上位要素の `<entitlement>` の接続アプリケーションから動的に有効値のリストを取得する際に、XDS クエリが返す XDS インスタンス要素ごとに適用されます。

例

`<entitlement>` を参照してください。

内容に使用できる要素

要素	説明
<code>display-name</code>	表示名の結果を問い合わせます。
<code>description</code>	記述の結果を問い合わせます。
<code>ent-value</code>	結果の値を問い合わせます。

属性

なし

内容の規則

(`display-name` , `description` , `ent-value`)

親要素

要素	説明
<code>query-app</code>	エンタイトルメントの有効な値の定義を問い合わせます。

src

許可または取り消されたエンタイトルメントに対して許可または取り消しを実行したエージェントの識別に使用します。エージェントが役割ベースエンタイトルメントシステムの場合の一般的な値は **RBE** で、許可エージェントが承認フローシステムの場合の一般的な値は **AF** です。エンタイトルメントを許可するその他のエージェントには、固有の識別子が必要です。

例

<ref> を参照してください。

内容に使用できる要素

#PCDATA

属性

なし

内容の規則

(#PCDATA)

親要素

要素	説明
ref	エンタイトルメントの参照です。
result	エンタイトルメントの結果です。

state

上位要素の **<result>** が参照するエンタイトルメントの状態 (1=許可、0=取り消し) を指定します。

例

<result> を参照してください。

内容に使用できる要素

#PCDATA

属性

なし

内容の規則

(#PCDATA)

親要素

要素	説明
result	エンタイトルメントの結果です。

status

上位要素の `<result>` が参照するエンタイトルメントの許可または取り消しのステータス (success、error、warning、retry、fatal) を指定します。

例

`<result>` を参照してください。

内容に使用できる要素

#PCDATA

属性

なし

内容の規則

(#PCDATA)

親要素

要素	説明
<code>result</code>	エンタイトルメントの結果です。

timestamp

上位要素の <result> が参照する、エンタイトルメントの許可または取り消しのタイムスタンプが挿入されます。タイムスタンプは、1970年1月1日午前0時からミリ秒の数値を表す整数です。

例

<result> を参照してください。

内容に使用できる要素

#PCDATA

属性

なし

内容の規則

(#PCDATA)

親要素

要素	説明
result	エンタイトルメントの結果です。

token-association

上位要素の <display-name>、<description>、または <ent-value> の値として、XDS インスタンス要素の関連付けキーの値を使用することを指定します。

例

<entitlement> を参照してください。

内容に使用できる要素

なし

属性

なし

内容の規則

なし

親要素

要素	説明
description	記述の結果を問い合わせます。
display-name	結果の表示名を問い合わせます。
ent-value	結果の値を問い合わせます。

token-attr

上位要素の <display-name>、<description>、または <ent-value> の値として、XDS インスタンス要素の指定された属性の最初の値を使用することを指定します。

例

<entitlement> を参照してください。

内容に使用できる要素

なし

属性

属性	有効値	デフォルト値
attr-name	CDATA 属性の名前です。	#REQUIRED

内容の規則

(#PCDATA)

親要素

要素	説明
description	記述の結果を問い合わせます。
display-name	結果の表示名を問い合わせます。
ent-value	結果の値を問い合わせます。

token-src-dn

上位要素の <display-name>、<description>、または <ent-value> の値として、XDS インスタンス要素の src-dn 属性の値を使用することを指定します。

例

<entitlement> を参照してください。

内容に使用できる要素

なし

属性

なし

内容の宣言

なし

親要素

要素	説明
description	記述の結果を問い合わせます。
display-name	結果の表示名を問い合わせます。
ent-value	結果の値を問い合わせます。

value

<value> 要素は、存在する場合、上位要素の <entitlement> で有効値のスタティックリストを指定します。

例

<entitlement> を参照してください。

内容に使用できる要素

#PCDATA

属性

なし

内容の規則

(#PCDATA)

親要素

要素	説明
values	エンタイトルメントの有効な値です。

values

<values> は、存在する場合、上位要素の <entitlement> がパラメータ値を要求することを示します。multi-valued が「false」の場合、エンタイトルメントは指定された対象者に一度に 1 つの値だけを許可できます。そうでない場合、エンタイトルメントは同一の対象者に毎回異なる値を指定して複数回許可できます。<values> には、パラメータの有効値のリストを指定するか、関連アプリケーションからパラメータ有効値のリストを取得するクエリを指定することもできます。

例

<entitlement> を参照してください。

内容に使用できる要素

要素	説明
query-app	エンタイトルメントの有効な値の定義を問い合わせます。
value	列挙値です。

属性

属性	有効値	デフォルト値
multi-valued	True False 複数值のフラグです。	True

内容の規則

(query-app | value +) ?

親要素

要素	説明
entitlement	エンタイトルメントの定義です。