

# SiteMinder

## C 用プログラミング ガイド

12.52 SP1



このドキュメント（組み込みヘルプシステムおよび電子的に配布される資料を含む、以下「本ドキュメント」）は、お客様への情報提供のみを目的としたもので、日本 CA 株式会社（以下「CA」）により随時、変更または撤回されることがあります。本ドキュメントは、CA が知的財産権を有する機密情報であり、CA の事前の書面による承諾を受けずに本書の全部または一部を複製、譲渡、変更、開示、修正、複製することはできません。

本ドキュメントで言及されている CA ソフトウェア製品のライセンスを受けたユーザは、社内でユーザおよび従業員が使用する場合に限り、当該ソフトウェアに関連する本ドキュメントのコピーを妥当な部数だけ作成できます。ただし、CA のすべての著作権表示およびその説明を当該複製に添付することを条件とします。

本ドキュメントを印刷するまたはコピーを作成する上記の権利は、当該ソフトウェアのライセンスが完全に有効となっている期間内に限定されます。いかなる理由であれ、上記のライセンスが終了した場合には、お客様は本ドキュメントの全部または一部と、それらを複製したコピーのすべてを破棄したことを、CA に文書で証明する責任を負いません。

準拠法により認められる限り、CA は本ドキュメントを現状有姿のまま提供し、商品性、特定の使用目的に対する適合性、他者の権利に対して侵害のないことについて、黙示の保証も含めいかなる保証もしません。また、本ドキュメントの使用に起因して、逸失利益、投資損失、業務の中断、営業権の喪失、情報の喪失等、いかなる損害（直接損害か間接損害かを問いません）が発生しても、CA はお客様または第三者に対し責任を負いません。CA がかかる損害の発生の可能性について事前に明示に通告されていた場合も同様とします。

本ドキュメントで参照されているすべてのソフトウェア製品の使用には、該当するライセンス契約が適用され、当該ライセンス契約はこの通知の条件によっていかなる変更も行われません。

本書の制作者は CA および CA Inc. です。

「制限された権利」のもとでの提供：アメリカ合衆国政府が使用、複製、開示する場合は、FAR Sections 12.212、52.227-14 及び 52.227-19(c)(1)及び(2)、ならびに DFARS Section 252.227-7014(b)(3) または、これらの後継の条項に規定される該当する制限に従うものとします。

Copyright © 2014 CA. All rights reserved. 本書に記載されたすべての商標、商号、サービス・マークおよびロゴは、それぞれの各社に帰属します。

## マニュアルの変更点

SiteMinder の旧リリースで発見された問題の結果として、**12.52 SP1** のドキュメントに以下の更新が行われました。

- セッション トークン への新しい属性の追加を反映するために、以下のトピックが更新されました (CQ 1752506、Star 21313153-1)
  - [Sm AgentApi Attribute t](#) (P. 55)
  - [Sm AgentApi CreateSSOToken\(\)](#) (P. 80)
  - [Sm AgentApi DecodeSSOToken\(\)](#) (P. 82)

## CA Technologies 製品リファレンス

このマニュアルが参照している CA Technologies の製品は以下のとおりです。

SiteMinder

## マニュアルの変更点

以下のドキュメントの更新は、本書の最新のリリース以降に行われたものです。

## CA への連絡先

テクニカルサポートの詳細については、弊社テクニカルサポートの Web サイト (<http://www.ca.com/jp/support/>) をご覧ください。



# 目次

---

<b>第 1 章: API の概要</b>	<b>25</b>
SiteMinder SDK の概要.....	25
SDK インストール.....	27
カスタム アプリケーションおよびポリシー サーバ拡張.....	27
コード サンプル.....	28
カスタム コードのサポート.....	28
<b>第 2 章: C 用のエージェント API ガイド</b>	<b>29</b>
エージェント API の概要.....	29
SiteMinder エージェントについて.....	30
エージェント初期化.....	31
エージェント ディスカバリ.....	31
エージェント API を使用してリソースにアクセスする方法.....	35
カスタム エージェントのコンパイルおよびリンク.....	37
中央ホスト設定.....	38
設定要件.....	39
コード要件.....	40
エージェントのアップグレード.....	41
エージェント コール シーケンス.....	41
サンプル カスタム エージェント.....	42
エージェント API サービス.....	42
セッション サービス.....	42
認可 サービス.....	46
管理 サービス.....	47
トンネル サービス.....	48
レスポンス 属性.....	48
カスタム エージェントおよびシングル サインオン.....	49
標準 エージェント サポート.....	50
カスタム エージェント 経由の ログイン.....	51
標準 エージェント 経由の ログイン.....	52
メモリの解放.....	52
エージェント API のデータ 構造体 (C).....	53
Sm_AgentApi_AgentDiscovery_t.....	53
Sm_AgentApi_Attribute_t.....	55

---

Sm_AgentApi_Init_t .....	59
Sm_AgentApi_ManagementContext_t .....	61
Sm_AgentApi_Realm_t .....	63
Sm_AgentApi_ResourceContext_t .....	65
Sm_AgentApi_Server_t .....	66
Sm_AgentApi_Session_t .....	67
Sm_AgentApi_TunnelServiceRequest_t .....	69
Sm_AgentApi_UserCredentials_t .....	70
エージェント API の関数宣言 (C) .....	71
関数リターンコード .....	73
Sm_AgentApi_Audit() .....	73
Sm_AgentApi_Authorize() .....	75
Sm_AgentApi_AuthorizeDLP() .....	76
Sm_AgentApi_ChangePassword() .....	78
Sm_AgentApi_CreateSSOToken() .....	80
Sm_AgentApi_DecodeSSOToken() .....	82
Sm_AgentApi_DelSessionVariables() .....	85
Sm_AgentApi_DoManagement() .....	87
Sm_AgentApi_FreeAttributes() .....	89
Sm_AgentApi_FreeServers() .....	89
Sm_AgentApi_GetAgentApiUpdateVersion() .....	90
Sm_AgentApi_GetAllowedTunnelBufSize() .....	90
Sm_AgentApi_GetConfig() .....	91
Sm_AgentApi_GetMaxTunnelBufSize() .....	94
Sm_AgentApi_GetSessionVariables() .....	94
Sm_AgentApi_Init() .....	97
Sm_AgentApi_IsProtected() .....	99
Sm_AgentApi_IsProtectedDLP() .....	100
Sm_AgentApi_Login() .....	101
Sm_AgentApi_Logout() .....	104
Sm_AgentApi_MakeCertificateHash() .....	105
Sm_AgentApi_SetAgentInstanceInfo() .....	106
Sm_AgentApi_SetDefaultAgentId() .....	107
Sm_AgentApi_SetSessionVariables() .....	108
Sm_AgentApi_Tunnel() .....	111
Sm_AgentApi_UnInit() .....	113
Sm_AgentApi_UpdateAttributes() .....	115

### 第 3 章: カスタム エージェント タイプの設定 117

カスタム エージェント タイプの概要 .....	117
エージェント タイプ ワークシート .....	117

エージェントタイプの設定 .....	119
エージェントタイプ属性 .....	120
エージェントタイプの変更 .....	123

## 第 4 章: ポリシー管理 API ガイド 125

ポリシー管理 API の概要 .....	125
ポリシー管理のセットアップ .....	125
オブジェクト取得関数 .....	126
オブジェクト作成関数 .....	127
オブジェクト削除関数 .....	127
オブジェクトの関連付け .....	128
オブジェクト識別子 .....	128
ディレクトリ検索順序関数 .....	129
パフォーマンス拡張 .....	129
メモリ、キャッシュ、エージェント キーの管理 .....	130
オブジェクトスコープ .....	130
フェデレーション API .....	132
SAML アサーション .....	132
SAML 1.x .....	133
SAML 2.0 .....	135
WS-フェデレーション .....	143
ポリシー管理 API データ構造体 .....	144
Sm_PolicyApi_Admin_t .....	145
Sm_PolicyApi_Affiliate_t .....	146
Sm_PolicyApi_AffiliateAttr_t .....	149
Sm_PolicyApi_AffiliateDomain_t .....	150
Sm_PolicyApi_Agent_t .....	151
Sm_PolicyApi_AgentConfig_t .....	152
Sm_PolicyApi_AgentType_t .....	153
Sm_PolicyApi_AgentTypeAttr_t .....	154
Sm_PolicyApi_Association_t .....	155
Sm_PolicyApi_AuthAzMap_t .....	156
Sm_PolicyApi_CertMap_t .....	157
Sm_PolicyApi_Domain_t .....	160
Sm_PolicyApi_Group_t .....	161
Sm_PolicyApi_HostConfig_t .....	162
Sm_PolicyApi_IPAddress_t .....	164
Sm_PolicyApi_ManagementCommand_t .....	165
Sm_PolicyApi_ODBCQueryScheme_t .....	166
Sm_PolicyApi_Oid_t .....	168

---

Sm_PolicyApi_PasswordMsgField_t .....	169
Sm_PolicyApi_PasswordPolicy_t.....	170
Sm_PolicyApi_Policy_t.....	176
Sm_PolicyApi_PolicyLink_t.....	179
Sm_PolicyApi_Realm_t .....	180
Sm_PolicyApi_RegistrationScheme_t .....	182
Sm_PolicyApi_RegularExpression_t.....	183
Sm_PolicyApi_Response_t.....	184
Sm_PolicyApi_ResponseAttr_t.....	185
Sm_PolicyApi_Rule_t .....	188
Sm_PolicyApi_SAMLAffiliation_t.....	190
Sm_PolicyApi_SAMLProviderProp_t.....	191
Sm_PolicyApi_SAMLRequesterAttr_t .....	203
Sm_PolicyApi_SAMLSP_t .....	204
Sm_PolicyApi_SAMLSPAssertionConsumerService_t .....	205
Sm_PolicyApi_SAMLSPAttr_t .....	206
Sm_PolicyApi_Scheme_t.....	207
Sm_PolicyApi_SharedSecretPolicy_t.....	209
Sm_PolicyApi_Server_t .....	210
Sm_PolicyApi_TrustedHost_t.....	211
Sm_PolicyApi_User_t .....	212
Sm_PolicyApi_UserContext_t .....	213
Sm_PolicyApi_UserDir_t .....	214
Sm_PolicyApi_UserPasswordState_t .....	218
Sm_PolicyApi_Variable_t .....	219
Sm_PolicyApi_VariableType_t .....	225
Sm_PolicyApi_WSFEDProviderProp_t.....	227
Sm_PolicyApi_WSFEDResourcePartner_t .....	236
エクスポートされたタイプ .....	237
管理者権限.....	237
アフィリエイト属性タイプ.....	240
属性モードタイプ.....	240
認証と認可のマッピングタイプ .....	241
証明書マッピング属性タイプ.....	241
証明書マッピングディレクトリタイプ .....	241
証明書マッピングフラグ定義.....	242
ディレクトリ機能.....	243
ドメインフラグ .....	244
グループタイプ.....	245
IPアドレスタイプ .....	245
管理コマンド.....	247

---

パスワード メッセージ .....	248
パスワード メッセージ フィールド .....	250
パスワード メッセージ フィールド タイプ .....	250
パスワード ポリシー 動作フラグ .....	251
ポリシー フラグ .....	254
ポリシー 管理 API 初期化フラグ .....	254
ポリシー オブジェクト ID .....	256
ポリシー 解決 .....	258
リターン コード .....	258
SAML1x リダイレクト URL タイプ .....	263
SAML アサーション コンシューマ サービス バインディング .....	263
SAML 属性名の形式識別子 .....	264
SAML プロファイル .....	264
方式タイプ .....	264
共有秘密キーのロールオーバー パラメータ .....	266
構造体 ID .....	266
ポリシー アプリケーションの構造体 .....	268
ポリシー 管理 API のカテゴリ別の関数 .....	269
必須関数 .....	270
管理者関数 .....	271
エージェント関数 .....	271
エージェント設定関数 .....	273
認証/許可マップ関数 .....	275
認証方式関数 .....	275
証明書マッピング関数 .....	275
ドメイン関数 .....	276
フェデレーション関数 .....	276
一般オブジェクト関数 .....	279
グループ関数 .....	279
ODBC クエリ方式関数 .....	280
パスワード ポリシー関数 .....	280
ポリシー関数 .....	281
レルム関数 .....	282
登録方式関数 .....	282
正規表現関数 .....	283
レスポンス関数 .....	283
ルール関数 .....	284
SAML1.x 設定関数 .....	285
SAML 2.0 属性権限関数 .....	285
SAML 2.0 設定関数 .....	286

---

SAML 2.0 インデックス エンドポイント関数 .....	288
ユーザ ディレクトリ関数 .....	289
ユーザおよびユーザ状態関数 .....	290
ユーザ パスワード状態関数 .....	291
ユーティリティ関数 .....	292
変数関数 .....	292
WS-Federation 関数 .....	293
ポリシー管理 API の関数宣言 .....	295
Sm_PolicyApi_AddAdmin() .....	295
Sm_PolicyApi_AddAdminToAffiliateDomain() .....	296
Sm_PolicyApi_AddAdminToDomain() .....	298
Sm_PolicyApi_AddAffiliate() .....	299
Sm_PolicyApi_AddAffiliateDomain() .....	301
Sm_PolicyApi_AddAgent() .....	303
Sm_PolicyApi_AddAgentConfig() .....	305
Sm_PolicyApi_AddAgentConfigAssociation() .....	306
Sm_PolicyApi_AddAssertionConsumerServiceToSAMLSP .....	307
Sm_PolicyApi_AddAttributeToAffiliate() .....	309
Sm_PolicyApi_AddAttributeToSAMLScheme() .....	310
Sm_PolicyApi_AddAttributeToSAMLSP() .....	311
Sm_PolicyApi_AddDomain() .....	312
Sm_PolicyApi_AddGlobalPolicy() .....	314
Sm_PolicyApi_AddGlobalResponse() .....	315
Sm_PolicyApi_AddGlobalRule() .....	317
Sm_PolicyApi_AddGroup() .....	318
Sm_PolicyApi_AddHostConfig() .....	320
Sm_PolicyApi_AddMessageConsumerPluginToSAML1xScheme() .....	321
Sm_PolicyApi_AddOneTimeUsePropToAffiliate() .....	322
Sm_PolicyApi_AddPasswordPolicy() .....	323
Sm_PolicyApi_AddPolicy() .....	325
Sm_PolicyApi_AddPolicyLink() .....	327
Sm_PolicyApi_AddRealm() .....	329
Sm_PolicyApi_AddRedirectURLToSAML1xScheme() .....	331
Sm_PolicyApi_AddRegistrationScheme() .....	332
Sm_PolicyApi_AddRegularExpressionToPasswordPolicy() .....	333
Sm_PolicyApi_AddResponse() .....	335
Sm_PolicyApi_AddResponseAttr() .....	336
Sm_PolicyApi_AddRule() .....	338
Sm_PolicyApi_AddSAMLAffiliation() .....	339
Sm_PolicyApi_AddSAMLScheme() .....	341
Sm_PolicyApi_AddSAMLServiceProvider() .....	343

---

Sm_PolicyApi_AddScheme()	345
Sm_PolicyAPI_AddTargetConfigToSAML1xScheme	347
Sm_PolicyApi_AddToGroup()	348
Sm_PolicyApi_AddTrustedHost()	350
Sm_PolicyApi_AddUserDirToAffiliateDomain()	352
Sm_PolicyApi_AddUserDirToDomain()	354
Sm_PolicyApi_AddUseSecureAuthPropToAffiliate()	356
Sm_PolicyApi_AddUsersToAffiliate()	357
Sm_PolicyApi_AddUsersToPolicy()	358
Sm_PolicyApi_AddUsersToSAMLServiceProvider()	360
Sm_PolicyApi_AddUsersToWSFEDResourcePartner()	362
Sm_PolicyApi_AddVariable()	363
Sm_PolicyApi_AddWSFEDResourcePartner()	364
Sm_PolicyApi_AddWSFEDScheme()	367
Sm_PolicyApi_ConvertFromLegacyAgent()	369
Sm_PolicyApi_ConvertToLegacyAgent()	370
Sm_PolicyApi_CreateAuthAzMap()	371
Sm_PolicyApi_CreateCertMap()	372
Sm_PolicyApi_CreateODBCQueryScheme()	374
Sm_PolicyApi_CreateUserDir()	376
Sm_PolicyApi_DeleteAdmin()	380
Sm_PolicyApi_DeleteAffiliate()	381
Sm_PolicyApi_DeleteAffiliateDomain()	383
Sm_PolicyApi_DeleteAgent()	384
Sm_PolicyApi_DeleteAgentConfig()	385
Sm_PolicyApi_DeleteAuthAzMap()	386
Sm_PolicyApi_DeleteCertMap()	387
Sm_PolicyApi_DeleteDomain()	388
Sm_PolicyApi_DeleteHostConfig()	389
Sm_PolicyApi_DeleteGroup()	390
Sm_PolicyApi_DeleteODBCQueryScheme()	391
Sm_PolicyApi_DeletePasswordPolicy()	392
Sm_PolicyApi_DeletePolicy()	393
Sm_PolicyApi_DeleteRealm()	394
Sm_PolicyApi_DeleteRegistrationScheme()	395
Sm_PolicyApi_DeleteResponse()	396
Sm_PolicyApi_DeleteRule()	397
Sm_PolicyApi_DeleteSAMLAffiliation()	398
Sm_PolicyApi_DeleteSAMLServiceProvider()	399
Sm_PolicyApi_DeleteScheme()	400
Sm_PolicyApi_DeleteTrustedHost()	401
Sm_PolicyApi_DeleteUserDir()	402

---

Sm_PolicyApi_DeleteVariable()	403
Sm_PolicyApi_DeleteWSFEDResourcePartner()	405
Sm_PolicyApi_DisableUser()	406
Sm_PolicyApi_EnableUser()	407
Sm_PolicyApi_FlushRealm()	409
Sm_PolicyApi_FlushUser()	410
Sm_PolicyApi_FreeMemory()	411
Sm_PolicyApi_FreeMemoryEx()	412
Sm_PolicyApi_FreeString()	413
Sm_PolicyApi_FreeStringArray()	414
Sm_PolicyApi_GetAdmin()	415
Sm_PolicyApi_GetAdminByName()	416
Sm_PolicyApi_GetAffiliate()	418
Sm_PolicyApi_GetAffiliateByName()	420
Sm_PolicyApi_GetAffiliateDomain()	422
Sm_PolicyApi_GetAffiliateDomainByName()	423
Sm_PolicyApi_GetAffiliateDomainObjects()	425
Sm_PolicyApi_GetAffiliateDomainUserDirSearchOrder()	426
Sm_PolicyApi_GetAffiliatedSAMLAuthSchemes()	427
Sm_PolicyApi_GetAffiliatedSAMLServiceProviders()	428
Sm_PolicyApi_GetAffiliateUsers()	430
Sm_PolicyApi_GetAgent()	431
Sm_PolicyApi_GetAgentByName()	433
Sm_PolicyApi_GetAgentConfig()	434
Sm_PolicyApi_GetAgentConfigByName()	435
Sm_PolicyApi_GetAgentConfigAssociations()	437
Sm_PolicyApi_GetAgentType()	438
Sm_PolicyApi_GetAgentTypeByName()	439
Sm_PolicyApi_GetAgentTypeAttr()	441
Sm_PolicyApi_GetAgentTypeAttrByName()	442
Sm_PolicyApi_GetAllAffiliateAttributes()	443
Sm_PolicyApi_GetAllAffiliates()	445
Sm_PolicyApi_GetAllSAMLAffiliations()	446
Sm_PolicyApi_GetAllSAMLSchemeAttributes()	447
Sm_PolicyApi_GetAllSAMLServiceProviders()	448
Sm_PolicyApi_GetAllSAMLSPAssertionConsumerService()	450
Sm_PolicyApi_GetAllSAMLSPAttributes()	451
Sm_PolicyApi_GetAllWSFEDResourcePartners()	453
Sm_PolicyApi_GetAuthAzMap()	454
Sm_PolicyApi_GetCertMap()	455
Sm_PolicyApi_GetChildren()	456
Sm_PolicyApi_GetDirectoryContents()	457

---

Sm_PolicyApi_GetDisabledUserState()	459
Sm_PolicyApi_GetDomain()	460
Sm_PolicyApi_GetDomainByName()	462
Sm_PolicyApi_GetDomainObjects()	463
Sm_PolicyApi_GetGlobalObjects()	465
Sm_PolicyApi_GetGlobalPolicyByName()	466
Sm_PolicyApi_GetGlobalResponseByName()	468
Sm_PolicyApi_GetGlobalRuleByName()	469
Sm_PolicyApi_GetGroup()	471
Sm_PolicyApi_GetGroupByName()	473
Sm_PolicyApi_GetGroupOids()	475
Sm_PolicyApi_GetHostConfig()	477
Sm_PolicyApi_GetHostConfigByName()	478
Sm_PolicyAPI_GetMessageConsumerPluginFromSAML1xScheme()	479
Sm_PolicyApi_GetODBCQueryScheme()	480
Sm_PolicyApi_GetODBCQuerySchemeByName()	481
Sm_PolicyApi_GetOneTimeUsePropFromAffiliate()	483
Sm_PolicyApi_GetPasswordMsg()	484
Sm_PolicyApi_GetPasswordPolicy()	487
Sm_PolicyApi_GetPasswordPolicyByName()	488
Sm_PolicyApi_GetPolicy()	489
Sm_PolicyApi_GetPolicyByName()	490
Sm_PolicyApi_GetPolicyLinks()	492
Sm_PolicyApi_GetPolicyUsers()	493
Sm_PolicyApi_GetRealm()	495
Sm_PolicyApi_GetRealmByName()	496
Sm_PolicyAPI_GetRedirectURLFromSAML1xScheme()	498
Sm_PolicyApi_GetRegistrationScheme()	499
Sm_PolicyApi_GetRegistrationSchemeByName()	500
Sm_PolicyApi_GetRegularExpressions()	501
Sm_PolicyApi_GetResponse()	502
Sm_PolicyApi_GetResponseByName()	504
Sm_PolicyApi_GetResponseAttrs()	505
Sm_PolicyApi_GetRule()	507
Sm_PolicyApi_GetRuleByName()	508
Sm_PolicyApi_GetSAMLAffiliation()	510
Sm_PolicyApi_GetSAMLAffiliationById()	511
Sm_PolicyApi_GetSAMLScheme()	513
Sm_PolicyApi_GetSAMLServiceProvider()	514
Sm_PolicyApi_GetSAMLServiceProviderById()	515
Sm_PolicyApi_GetSAMLServiceProviderUsers()	517
Sm_PolicyApi_GetScheme()	518

---

Sm_PolicyApi_GetSchemeByName()	519
Sm_PolicyApi_GetSharedSecretPolicy()	520
Sm_PolicyApi_GetTargetConfigFromSAML1xScheme	522
Sm_PolicyApi_GetTrustedHost()	523
Sm_PolicyApi_GetTrustedHostByName()	524
Sm_PolicyApi_GetUseSecureAuthPropFromAffiliate()	526
Sm_PolicyApi_GetUserContext()	527
Sm_PolicyApi_GetUserDir()	529
Sm_PolicyApi_GetUserDirByName()	530
Sm_PolicyApi_GetUserDirCapabilities()	532
Sm_PolicyApi_GetUserDirSearchOrder()	533
Sm_PolicyApi_GetUserGroups()	534
Sm_PolicyApi_GetUserPasswordState()	536
Sm_PolicyApi_GetUsersFromWSFEDResourcePartner()	538
Sm_PolicyApi_GetVariable()	539
Sm_PolicyApi_GetVariableByName()	541
Sm_PolicyApi_GetVariableType()	543
Sm_PolicyApi_GetVariableTypeByName()	545
Sm_PolicyApi_GetWSFEDResourcePartner()	547
Sm_PolicyApi_GetWSFEDScheme()	549
Sm_PolicyApi_Init()	550
Sm_PolicyApi_InitEx()	551
Sm_PolicyApi_IsGroup()	552
Sm_PolicyApi_Login()	554
Sm_PolicyApi_Logout()	556
Sm_PolicyApi_LookupDirectoryEntry()	557
Sm_PolicyApi_ManagementCommand()	560
Sm_PolicyApi_Release()	561
Sm_PolicyApi_RemoveAdminFromAffiliateDomain()	562
Sm_PolicyApi_RemoveAdminFromDomain()	563
Sm_PolicyApi_RemoveAgentConfigAssociation()	565
Sm_PolicyApi_RemoveAssertionConsumerServiceFromSAMLSP()	566
Sm_PolicyApi_RemoveAttributeFromAffiliate()	567
Sm_PolicyApi_RemoveAttributeFromSAMLScheme()	568
Sm_PolicyApi_RemoveAttributeFromSAMLSP()	570
Sm_PolicyApi_RemoveFromGroup()	571
Sm_PolicyApi_RemovePolicyLinkFromPolicy()	573
Sm_PolicyApi_RemoveRegularExpressionFromPasswordPolicy()	574
Sm_PolicyApi_RemoveResponseAttr()	575
Sm_PolicyApi_RemoveUserDirFromAffiliateDomain()	576
Sm_PolicyApi_RemoveUserDirFromDomain()	578
Sm_PolicyApi_RemoveUsersFromAffiliate()	579

---

Sm_PolicyApi_RemoveUsersFromPolicy()	580
Sm_PolicyApi_RemoveUsersFromSAMLServiceProvider()	582
Sm_PolicyApi_RemoveUsersFromWSFEDResourcePartner()	584
Sm_PolicyApi_RenameObject()	585
Sm_PolicyApi_SetAffiliateDomainUserDirSearchOrder()	586
Sm_PolicyApi_SetDisabledUserState()	588
Sm_PolicyApi_SetPassword()	592
Sm_PolicyApi_SetPath()	595
Sm_PolicyApi_SetResponseInPolicyLink()	597
Sm_PolicyApi_SetSharedSecretPolicy()	599
Sm_PolicyApi_SetUserDirSearchOrder()	600
Sm_PolicyApi_SetUserPasswordState()	601
Sm_PolicyApi_ValidateDirectoryEntry()	603
認証方式の設定	604
匿名テンプレート	606
SSL を介した基本テンプレート	607
基本テンプレート	609
カスタム テンプレート	610
HTML フォーム テンプレート	611
インパーソネーション テンプレート	613
RADIUS CHAP/PAP テンプレート	614
RADIUS サーバ テンプレート	615
SafeWord HTML フォーム テンプレート	617
SafeWord テンプレート	618
SAML Artifact テンプレート	619
SAML POST テンプレート	622
SAML 2.0 テンプレート	625
SecurID HTML フォーム テンプレート	627
SecurID テンプレート	628
smauthetsso 認証方式	630
TeleID テンプレート	632
Windows 認証テンプレート	634
WS-Federation テンプレート	637
X.509 クライアント証明書および基本テンプレート	639
X.509 クライアント証明書およびフォーム テンプレート	640
X.509 クライアント証明書または基本テンプレート	642
X.509 クライアント証明書またはフォーム テンプレート	643
X.509 クライアント証明書テンプレート	645

---

## 第 5 章: ポリシー サーバのカスタマイズ 647

認証および認可 API での動作.....	647
認証フック.....	648
認可フック.....	648
カスタム データ フック.....	648
SiteMinder イベント.....	648
モジュール呼び出しの間のデータ保存.....	649
汎用ユーザ属性.....	650
パスワードの取得.....	656
Web エージェントとの統合.....	657

## 第 6 章: コンテキスト構造 659

Sm_Api_AppSpecificContext_t.....	659
Sm_Api_Context_t.....	662
Sm_Api_RequestContext_t.....	664
Sm_Api_TunnelContext_t.....	665
Sm_Api_UserContext_t.....	666
LDAP の複数値属性.....	672
Sm_Api_UserCredentials_t.....	673

## 第 7 章: C 用の認証 API 675

認証 API の概要.....	675
認証方式ライブラリのインストール.....	676
認証方式のロード.....	677
ユーザ コンテキスト.....	677
認証イベント.....	677
リダイレクト.....	678
サポートされた認証情報.....	678
カスタム認証方式ライブラリの作成.....	679
SmAuthenticate().....	680
SmAuthInit().....	683
SmAuthQuery().....	684
SmAuthRelease().....	688

## 第 8 章: C 用の許可 API 689

認可 API の概要.....	689
ファイルを含める.....	690

アクティブな式.....	691
SiteMinder がアクティブな式を解釈する方法.....	691
アクティブ ルールの定義.....	693
アクティブ レスポンスの定義.....	693
アクティブ ポリシーの定義.....	693
ポリシー サーバへの HTTP ヘッダおよび Cookie のパス.....	694
認可関数宣言.....	695
ユーザ定義関数.....	695
SmQueryVersion().....	697
アクティブな式の例.....	697
アクティブ ルールの例.....	698
アクティブ レスポンスの例.....	700

## 第 9 章: トンネル サービス API ガイド 703

トンネル サービス API の概要.....	703
カスタム トンネル サービスの開発.....	704
ファイルを含める.....	704
トンネル サービス API リファレンス.....	705
SmQueryVersion().....	705
SmTunnelInit().....	706
SmTunnelRelease().....	707
ユーザ定義関数.....	707

## 第 10 章: イベント API ガイド 709

イベント API 概要.....	709
イベント API セットアップ.....	710
イベント プロバイダ構造.....	710
SmLog_Access_t.....	711
SmLog_EMS_t.....	713
SmLog_Obj_t.....	714
SmLog_System_t.....	715
イベント API リファレンス.....	715
アクセス イベント タイプ.....	716
EMS イベント タイプ.....	717
オブジェクト イベント タイプ.....	721
システム イベント タイプ.....	723
SmLogAccessEvent_AuthAccept.....	723
SmLogAccessEvent_AuthReject.....	725
SmLogAccessEvent_AuthAttempt.....	726

---

SmLogAccessEvent_AuthChallenge .....	726
SmLogAccessEvent_AzAccept .....	727
SmLogAccessEvent_AzReject .....	728
SmLogAccessEvent_AdminLogin .....	729
SmLogAccessEvent_AdminLogout .....	729
SmLogAccessEvent_AdminReject .....	730
SmLogAccessEvent_AuthLogout .....	730
SmLogAccessEvent_ValidateAccept .....	731
SmLogAccessEvent_ValidateReject .....	731
SmLogEmsEvent_CreateUser、SmLogEmsEvent_DeleteUser、SmLogEmsEvent_ModifyUser .....	732
SmLogEmsEvent_PasswordModify .....	733
SmLogEmsEvent_AssignUserRole、SmLogEmsEvent_RemoveUserRole .....	734
SmLogEmsEvent_EnableUser、SmLogEmsEvent_DisableUser .....	735
SmLogEmsEvent_CreateOrg、SmLogEmsEvent_DeleteOrg、SmLogEmsEvent_ModifyOrg .....	736
SmLogEmsEvent_CreateRole、SmLogEmsEvent_DeleteRole、SmLogEmsEvent_ModifyRole .....	737
SmLogEmsEvent_CreateObject、SmLogEmsEvent_DeleteObject、SmLogEmsEvent_ModifyObject .....	738
SmLogEmsEvent_Login、SmLogEmsEvent_Logout .....	739
SmLogEmsEvent_AuthFail .....	739
SmLogEmsEvent_SessionTimeout .....	740
SmLogObjEvent_Create .....	741
SmLogObjEvent_Update .....	741
SmLogObjEvent_Delete .....	742
SmLogObjEvent_Login .....	743
SmLogObjEvent_FailedLoginAttemptsCount .....	743
SmLogObjEvent_Logout .....	744
SmLogObjEvent_LoginReject .....	744
SmLogObjEvent_FlushAll .....	745
SmLogObjEvent_FlushUser .....	745
SmLogObjEvent_FlushUsers .....	746
SmLogObjEvent_FlushRealms .....	746
SmLogObjEvent_ChangeDynamicKeys .....	747
SmLogObjEvent_ChangePersistentKey .....	747
SmLogObjEvent_ChangeSessionKey .....	748
SmLogObjEvent_ChangeUserPassword .....	748
SmLogObjEvent_CreateUserSuccess .....	749
SmLogObjEvent_DeleteUserSuccess .....	749
SmLogObjEvent_ChangeDisabledUserState .....	750
SmLogObjEvent_ModifyUserSuccess .....	751
SmLogObjEvent_CreateUserFail .....	751
SmLogObjEvent_DeleteUserFail .....	752
SmLogObjEvent_ModifyUserFail .....	752
SmLogSystemEvent_ServerInit .....	753

---

SmLogSystemEvent_ServerInitFail .....	754
SmLogSystemEvent_ServerUp .....	754
SmLogSystemEvent_ServerDown .....	755
SmLogSystemEvent_LogFileOpenFail .....	755
SmLogSystemEvent_ServerHeartbeat .....	755
SmLogSystemEvent_AgentInfo .....	756
SmLogSystemEvent_AgentConnectionStart .....	757
SmLogSystemEvent_AgentConnectionFail .....	757
SmLogSystemEvent_AgentConnectionEnd .....	758
SmLogSystemEvent_DbConnect .....	758
SmLogSystemEvent_DbConnectFail .....	759
SmLogSystemEvent_LdapConnect .....	759
SmLogSystemEvent_LdapConnectFail .....	760
SmLogSystemEvent_AmbiguousResourceMatch .....	760
SmLogSystemEvent_AmbiguousRadiusMatch .....	761
SmLogSystemEvent_AgentHeartbeat .....	761
イベント関数宣言 .....	762
SmEventInit() .....	762
SmEventRecord() .....	763
SmEventRelease() .....	764
アクティブポリシーの例 .....	765
イベントハンドラに対するポリシーサーバの設定 .....	767

## 第 11 章: DMS ワークフロー API 769

DMS ワークフロー API ガイド .....	769
DMS セッション .....	770
DMS のワークフロー .....	771
ワークフロー イベント .....	772
プリプロセス イベント .....	772
ポストプロセス イベント .....	773
DMS ワークフロー API リファレンス .....	773
Sm_Api_DmsContext_t .....	775
Sm_DmsWorkflow_Attribute_t .....	777
SmDmsWorkflowInit() .....	778
SmDmsWorkflowPostprocess() .....	779
SmDmsWorkflowPreprocess() .....	780
SmDmsWorkflowRelease() .....	783

## 第 12 章: ディレクトリ API ガイド 785

ディレクトリ API の目的 .....	785
----------------------	-----

---

ディレクトリ API を使用する前に.....	786
ディレクトリ API を使用する.....	787
ディレクトリ アプリケーションの構築.....	787
エクスポートされた列挙.....	788
一般的なデータ型と構造.....	791
初期化およびリリース関数.....	792
ユーティリティ関数.....	795
ディレクトリ上の操作.....	796
ディレクトリ エントリ (ユーザ) 上の操作.....	798
ディレクトリ API リファレンス.....	799
SmDirAddEntry().....	802
SmDirAddMemberToGroup().....	804
SmDirAddMemberToRole().....	805
SmDirAuthenticateUser().....	806
SmDirChangeUserPassword().....	809
SmDirEnumerate().....	811
SmDirFreeString().....	812
SmDirFreeStringArray().....	812
SmDirGetDirConnection().....	813
SmDirGetDirObjInfo().....	814
SmDirGetGroupMembers().....	816
SmDirGetLastErrMsg().....	817
SmDirGetRoleMembers().....	818
SmDirGetUserAttr().....	819
SmDirGetUserAttrMulti().....	820
SmDirGetUserClasses().....	822
SmDirGetUserDisabledState().....	823
SmDirGetUserGroups().....	826
SmDirGetUserProperties().....	827
SmDirGetUserRoles().....	828
SmDirInit().....	830
SmDirInitDirInstance().....	832
SmDirInitUserInstance().....	835
SmDirLookup().....	837
SmDirQueryVersion().....	839
SmDirRelease().....	840
SmDirReleaseInstance().....	840
SmDirRemoveEntry().....	842
SmDirRemoveMemberFromGroup().....	844
SmDirRemoveMemberFromRole().....	845
SmDirSearch().....	846

---

SmDirSearchCount().....	848
SmDirSetUserAttr() .....	850
SmDirSetUserAttrMulti().....	851
SmDirSetUserDisabledState() .....	852
SmDirValidateInstance() .....	855
SmDirValidateUserDN().....	856
SmDirValidateUsername() .....	857
SmDirValidateUserPolicyRelationship() .....	858
サンプルディレクトリ アプリケーションで使用される構造.....	860
ディレクトリ インスタンス ハンドル.....	861
ディレクトリ プロバイダ ハンドル.....	862
ディレクトリ エントリ (ユーザ) インスタンス ハンドル .....	863
ハンドル タイプを区別する方法.....	864
<b>第 13 章: 共通データ型および構造</b> .....	<b>865</b>
エクスポートされた列挙.....	865
Sm_Api_DisabledReason_t.....	865
Sm_Api_Reason_t.....	868
Sm_PolicyResolution_t.....	870
共通構造.....	873
<b>第 14 章: イベント ログ形式</b> .....	<b>873</b>
アクセス イベント.....	873
認証形式と許可形式.....	874
管理形式.....	877
オブジェクト イベント.....	878
AdminChange 形式.....	879
管理コマンド形式.....	880
EMS イベント.....	881
EMS ログ形式.....	881
<b>付録 A: SAML 2.0 プロパティ参照</b> .....	<b>885</b>
SAML 2.0 プロパティについて.....	885
SAML 2.0 プロパティ参照.....	886



# 第 1 章: API の概要

---

このセクションには、以下のトピックが含まれています。

[SiteMinder SDK の概要](#) (P. 25)

[SDK インストール](#) (P. 27)

[カスタム アプリケーションおよびポリシー サーバ拡張](#) (P. 27)

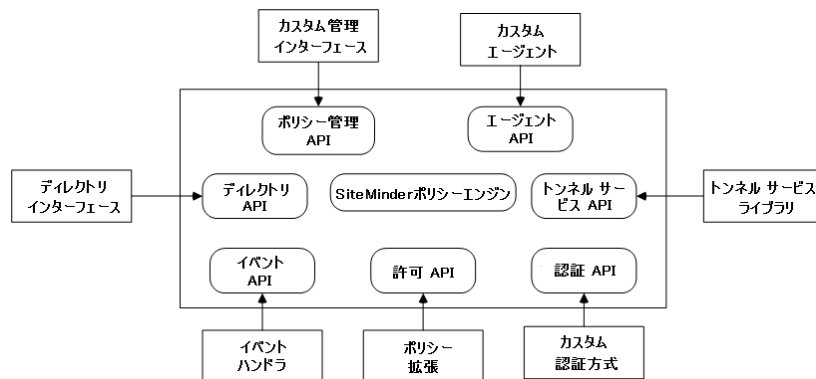
[コード サンプル](#) (P. 28)

[カスタム コードのサポート](#) (P. 28)

## SiteMinder SDK の概要

CA SiteMinder Software Development Kit (SDK) には、特定の環境内で SiteMinder の機能の統合や拡張に使用可能な一連のアプリケーション プログラミング インターフェース (API) が文書化されて含まれています。

以下の図は、SiteMinder で API がどのように実装されるかを示しています。



SiteMinder SDK には、以下の API が含まれています。

- **ポリシー管理 API** -- カスタム **Administrative UI** アプリケーションの開発、または選択したポリシー オブジェクト コンポーネント（アプリケーション内のルール、ポリシー、レスポンスなど）のカスタマイズに使用します。この API のユーザは、ネイティブ **Administrative UI** のユーザが実行可能なデータ操作ほとんどを実行できます。
- **エージェント API** -- ポリシー サーバの認証および認可の機能を活用するためのカスタム エージェント アプリケーションの作成に使用します。また、この API を使用して、アプリケーション固有データを送信するためのセキュアな通信トンネルを構築することもできます。
- **認証 API** -- カスタム認証方式の開発およびカスタム スキームの統合に使用します。ポリシー サーバプラグインです。
- **認可 API** -- カスタム認可機能を実行するためのモジュールの開発に使用します。この API を使用して開発するモジュールは、**Administrative UI** でアクティブルール、アクティブ ポリシー、およびアクティブ レスポンスの設定ができます。ポリシー サーバプラグインです。
- **イベント API** -- **SiteMinder** のイベントを処理するカスタム イベント ハンドラの構築に使用します。ポリシー サーバプラグインです。
- **トンネルサービス API** -- エージェントと通信してデータを安全に転送するプラグイン トンネルサービスの開発に使用します。ポリシー サーバプラグインです。
- **DMS ワークフロー API** -- **DMS** イベントの前処理および後処理のワークフローの作成に使用します。ポリシー サーバプラグインです。
- **ディレクトリ API** -- カスタム ネームスペースを使用するユーザ ディレクトリの管理に使用します。ポリシー サーバプラグインです。

## SDK インストール

SDK は、次の場所にインストールされています。

- UNIX プラットフォーム : `<install_path>/sdk`
- Windows プラットフォーム : `<install_path>%sdk`

`<install_path>` は、SDK ソフトウェアのインストール時に指定したインストールパスです。

注: インストール済みディレクトリ構造体の詳細については、「SDK の概要」を参照してください。SDK のインストール手順については、「SDK リリースノート」を参照してください。

## カスタム アプリケーションおよびポリシー サーバ拡張

カスタム アプリケーションおよびポリシー サーバ拡張を構築する際には、以下の点に注意してください。

- カスタム アプリケーションおよびポリシー サーバ拡張を構築するマシン上で SiteMinder プロセスが実行されている必要はありません。
- ほとんどの場合、カスタム アプリケーションおよび拡張の構築時、ポリシー サーバと同じマシンに SDK をインストールする必要はありません。
- ポリシー サーバは、SiteMinder SDK で構築したアプリケーションおよび拡張の実行に必要となります。
- 一部のアプリケーションおよび拡張のランタイム ファイルは、ポリシー サーバに対してローカルであることが必要となります。それ以外は、リモートでもかまいません。
- ポリシー管理 API で構築されたアプリケーションは、ポリシー サーバ（たとえば UNIX プラットフォーム上の `smuser`）をインストールしたユーザと同じユーザとして実行される必要があります。
- CAPI は JNI をコールできません。ポリシー サーバには、このようなコールをサポートするための同期はありません。ただし、ソケットを使用して Java を呼び出し、メインプロセスに通知を返す個別のプロセスを生成することは可能です。

## コード サンプル

SiteMinder SDK には、SiteMinder クライアント アプリケーションおよび拡張用の C コードの試験済みサンプルが含まれます。これらのサンプルのソース ファイルは以下の場所にあります。

- Windows プラットフォーム：  
`<install_path>%sdk%samples%<api-name>`
- UNIX プラットフォーム：  
`<install_path>/sdk/samples/<api-name>`

## カスタム コードのサポート

CA は標準的な提供物の一部としてソフトウェア開発キット (SDK) をサポートします。ただし、カスタマやパートナーによって記述されたコードはサポートされていません。ユーザが自分の記述するコードに責任を持ちます。SDK ベースのコードの設計や実装にサポートが必要な場合は、CA カスタマ アカウント チームにご連絡ください。

# 第 2 章: C 用のエージェント API ガイド

---

このセクションには、以下のトピックが含まれています。

- [エージェント API の概要 \(P. 29\)](#)
- [SiteMinder エージェントについて \(P. 30\)](#)
- [エージェント初期化 \(P. 31\)](#)
- [エージェントディスカバリ \(P. 31\)](#)
- [エージェント API を使用してリソースにアクセスする方法 \(P. 35\)](#)
- [カスタムエージェントのコンパイルおよびリンク \(P. 37\)](#)
- [中央ホスト設定 \(P. 38\)](#)
- [エージェントコールシーケンス \(P. 41\)](#)
- [サンプルカスタムエージェント \(P. 42\)](#)
- [エージェント API サービス \(P. 42\)](#)
- [レスポンス属性 \(P. 48\)](#)
- [カスタムエージェントおよびシングルサインオン \(P. 49\)](#)
- [メモリの解放 \(P. 52\)](#)
- [エージェント API のデータ構造体 \(C\) \(P. 53\)](#)
- [エージェント API の関数宣言 \(C\) \(P. 71\)](#)

## エージェント API の概要

エージェント API は、ポリシー サーバと連携して動作するため、アプリケーション開発が簡略化され、アプリケーションの拡張性も向上します。エージェント API を直接または（別のエージェントを介して）間接的に使用して構築されるアプリケーションを作成する開発者は、以下に示す実装特有の詳細を意識する必要がありません。

- LDAP ディレクトリ、SQL データベース、Windows ドメインなどのユーザネームスペース
- ユーザ名とパスワードによる単純な認証方式や、PKI システムの複雑な認証方式
- グループメンバシップまたは個別のプロファイルデータに基づく認可

エージェント API のその他の利点としては、全面的なセッション管理サポート、自動暗号化キーロールオーバー、リアルタイムポリシー更新などがあります。

## SiteMinder エージェントについて

SiteMinder エージェントはエージェント API のクライアントです。このエージェントは、ポリシー サーバによって提供されるアクセス制御ポリシーを適用します。ポリシー サーバは、リソースに関する情報を持たない汎用のポリシー エンジンです。エージェントは、リソースの解釈を定義し、許可のないユーザからリソースを保護するゲートキーパーとして機能します。

各種のリソースを保護するためにさまざまなエージェントタイプが用意されています。一部のエージェントタイプは、SiteMinder 製品の一部として出荷される事前定義された標準エージェントです。たとえば、Web サーバに HTTP アクセス制御を提供する Web エージェントなどです。また、エージェント API を使用して、カスタムエージェントを実装することもできます。

エージェント API を使用してアプリケーションを作成する際に、コンテキストに応じたさまざまな方法でユーザの認証および認可を行うカスタムエージェントを作成できます。たとえば、以下の機能を実装する FTP 転送のエージェントを作成できます。

- （基本的な名前とパスワードの認証ではなく）証明書ベースの認証
- 個々のユーザの認可レベルに基づいたアップロードおよびダウンロード

SDK を使用してカスタムエージェントを構築する場合は、その SDK のバージョンに対応したバージョンの SiteMinder ポリシー サーバでカスタムエージェントを実行する必要があります。

## エージェント初期化

エージェントがユーザに代わって作業を実行できるようになるには、先に `Sm_AgentApi_Init()` をコールすることにより 1 つまたは複数のポリシーサーバへの接続を初期化する必要があります。 `Sm_AgentApi_Init()` をコールすることにより、フェールオーバーモードや接続プールサイズなどの接続パラメータを指定できます。このコールは TCP 接続を作成するものであり、通常、エージェントインスタンス 1 つにつき 2 回以上実行する必要はありません。

複数の API インスタンスを初期化できます (個別のポリシーストアを複数使用するポリシーサーバと関係して操作する場合など)。

初期化直後、エージェントは、構造体 `Sm_AgentApi_ManagementContext_t` で設定されたコマンド `SM_AGENTAPI_MANAGEMENT_SET_AGENT_INFO` を使用して `Sm_AgentApi_DoManagement()` をコールすることにより、ポリシーサーバにそのバージョン情報を通知する必要があります。実際の情報は、エージェントに関する十分な情報 (ビルド番号、バージョン番号など) が含まれる任意の文字列を使用できます。文字列はポリシーサーバログに記録されます。

## エージェント ディスカバリ

SiteMinder 管理者は、エージェント ディスカバリを使用して、数年にわたって展開されているエージェントを含め、さまざまなタイプのエージェントのインスタンスを追跡できます。エージェント インスタンスは、任意のタイプのエージェント (たとえば Web エージェント、カスタム エージェント、ERP エージェント) が可能です。エージェント ディスカバリの認識範囲に含めるためには、エージェントがアクティブで、ポリシーサーバと通信中であることが必要です。

5.x 以降のエージェントのみ追跡できます。r12.5 以前に作成されたエージェントの場合、エージェントの識別に IP アドレスとトラステッドホストの組み合わせが使用されます。同じエージェントでも、この組み合わせが変わると、1 つエージェントに対して複数のエントリが生じることになります。

各 r12.5 エージェント インスタンスは固有の **GUID** により識別されます。**GUID** は設定ファイルに保存されています。

**Sm\_AgentApi\_SetAgentInstanceInfo()** 関数は、エージェント ディスカバリのプロセスにおいて中心的な役割を果たします。

**Sm\_AgentApi\_SetAgentInstanceInfo()** 関数は、そのエージェント インスタンスに対して指定された **GUID** が設定にあることを判別します。**GUID** が検出された場合、そのエージェントはすでに検出済みであり、追跡が可能です。**GUID** が検出されない場合、この関数は、そのエージェント インスタンスに対して **GUID** を作成し、後で呼び出すことができるように、この **GUID** を設定ファイルに書き込みます。複数のエージェント インスタンスで1つの設定ファイルを共有することはできません。

エージェント API は、エージェント情報を登録するために以下を提供します。

```
int SM_EXTERN Sm_AgentApi_SetAgentInstanceInfo (  
    const void* pHandle,  
    Sm_AgentApi_AgentDiscovery_t arrParams[], int nCount);
```

- `pHandle` は、`Sm_AgentApi_Init` によって生成されるハンドルであり、すべてのエージェント API 呼び出しで使用されます。
- `arrParams[]` は、`m_AgentApi_AgentDiscovery_t` データタイプによって定義される名前/値ペアの配列です。
- `nCount` は、`arrParams` で渡される要素数です。

`SmAgentApi.h` の以下のマクロは、`arrParams` で渡すことができる、サポートされる名前付き要素を定義します。これらはすべてオプションです。また、エージェントは、関連する情報を必要に応じて提供します。

```
#define SM_AGENT_INSTANCE_AGENTPRODUCTTYPE "AGENTPRODUCTTYPE"
```

例：WebAgent

```
#define SM_AGENT_INSTANCE_AGENTPRODUCTVERSION "AGENTPRODUCTVERSION"
```

例：1.0

```
#define SM_AGENT_INSTANCE_AGENTPRODUCTSUBTYPE "AGENTPRODUCTSUBTYPE"
```

例：IIS

```
#define SM_AGENT_INSTANCE_AGENTPRODUCTOSTYPE "AGENTPRODUCTOSTYPE"
```

例：Linux

```
#define SM_AGENT_INSTANCE_AGENTIDFILE "AGENTIDFILE"
```

一意の GUID を格納するために使用される、書き込み権限が付与されたテキストファイルへの有効なパス。空のファイルへのパスが渡された場合、エージェント API は新しい GUID を生成し、このファイルに書き込みます。今後、エージェントインスタンスを一意に識別するために、この GUID を読み取りおよび使用できます。

```
#define SM_AGENT_INSTANCE_ACONAME          "ACONAME"
```

エージェントによって使用される有効な ACO 名。

```
#define SM_AGENT_INSTANCE_HCONAME          "HCONAME"
```

エージェントによって使用される有効な HCO 名

```
#define SM_AGENT_INSTANCE_FIPSMODE          "FIPSMODE"
```

ホスト登録がエージェントによって使用される場合、SmHost.conf 内の値ストア。

Sm\_AgentApi\_SetAgentInstanceInfo が正常に呼び出された後、AgentAPI は、ハートビートなど、バックグラウンドでポリシー サーバへの定期的な呼び出しを行います。エージェントからこの API を初めて呼び出すと、ポリシー サーバによって使用されるエージェント ディスカバリ テーブルに、新しい一意のレコードが作成されます。 GUID が利用可能な場合、このレコードは GUID によってインデックス付けされます。 有効な GUID が使用されるようにするため、エージェント ID ファイルを提供することをお勧めします。 この値がない場合、エージェント インスタンスを SiteMinder エージェント システムにある一意のエンティティとして追跡および保持できません。

正しい呼び出しシーケンスは、以下のとおりです。

```
Sm_AgentApi_Init  
Sm_AgentApi_SetAgentInstanceInfo  
..  
PS その他の呼び出し  
..  
Sm_AgentApi_Uninit
```

pAgentApiHandle を作成するために Sm\_AgentApi\_Init が先に呼び出されていることを前提とするソースの例を以下に示します。

```
Sm_AgentApi_AgentDiscovery_t arrParams[8] = {  
    {SM_AGENT_INSTANCE_AGENTPRODUCTTYPE,  
    "CustomAgent"},  
    {SM_AGENT_INSTANCE_AGENTPRODUCTVERSION, "1.0"},  
    {SM_AGENT_INSTANCE_AGENTPRODUCTSUBTYPE, "IIS"},  
    {SM_AGENT_INSTANCE_AGENTPRODUCTOSTYPE,  
    "Windows"},  
    {SM_AGENT_INSTANCE_AGENTIDFILE, "c:%idfile.txt"},  
    {SM_AGENT_INSTANCE_ACONAME, "MyACO"},
```

```
        {SM_AGENT_INSTANCE_HCONAME, "MyHCO"},
        {SM_AGENT_INSTANCE_FIPSMODE, "COMPAT"}}};

    int nCount = 8;

    // Call the Agent API to send the data
    int nResult = Sm_AgentApi_SetAgentInstanceInfo(pAgentApiHandle, arrParams,
nCount);

    switch (nResult)
    {
        case SM_AGENTAPI_NOCONNECTION:
        case SM_AGENTAPI_FAILURE:
        case SM_AGENTAPI_TIMEOUT:
        {
            // put error handling here
            break;
        }
        case SM_AGENTAPI_NO:
        {
            // The call was refused - probably not properly formatted. This return
indicates
            // that the call ran on the PS and returned an explicit NO response, as
opposed to
            // some other connection or protocol error that might generate the
conditions above.
            break;
        }
        case SM_AGENTAPI_YES:
        {
            // Success
            break;
        }
    } // switch (nResult)
```

(新規関連グループ 1)

[Sm\\_AgentApi\\_AgentDiscovery\\_t](#) (P. 53)

[Sm\\_AgentApi\\_SetAgentInstanceInfo\(\)](#) (P. 106)

## エージェント API を使用してリソースにアクセスする方法

エージェント API が初期化された後、エージェントは、そのユーザからの要求の受け付け (URL に対する GET 要求を受信するなど) を開始できます。

以下の手順は、エージェントがリソースへのアクセスを許可するために必要な操作を説明しています。エージェントのパフォーマンスを向上するために、ほとんどの手順は結果をキャッシュできるようになっています。エージェントは、最小限のキャッシュをするか最大限のキャッシュするかを選択できます。

### ユーザにリソースへのアクセスを許可する方法

1. リソースにアクセスするためのユーザの要求を受理します。これはアプリケーション固有の要求です。たとえば、Web エージェントは、特定 URL に対する GET 要求を受理できます。
2. `Sm_AgentApi_IsProtected()` をコールすることにより、要求されたリソースが保護されているかどうかを判別します。リソースが保護されている場合、ポリシー サーバは、ユーザの身元を検証するためにユーザから取得される必要がある必須の認証情報を返します。リソースが保護されていない場合、要求されたリソースへのアクセスが許可されます。この手順の結果はキャッシュできます。

`Sm_AgentApi_Login()` をコールすることにより、ユーザからの必要な認証情報を収集し、ユーザを認証します。認証が成功した後、ポリシーサーバは、セッションを作成し、固有のセッション ID とセッション仕様が含まれるレスポンス属性を返します。これらのレスポンス属性は、ポリシーによって制御され、ユーザプロファイルデータ、静的または動的な権限、多数の事前定義済み認証状態属性のほか、ポリシー管理者によって指定された他のデータを含めることができます。エージェントは、ユーザセッション情報をキャッシュし、セッション有効期限を追跡することにより、セッション管理を実行できるようになります。

3. `Sm_AgentApi_Authorize()` をコールすることにより、要求されたリソースへのアクセス権がユーザに付与されていることを検証します。認可が成功した後、ポリシーサーバは、リソース固有の権限が含まれるレスポンス属性を返します。これらのレスポンス属性は、ポリシーによって制御され、ユーザプロファイルデータ、静的または動的な権限、ポリシー管理者によって指定された他のデータを含めることができます。要求されたリソースに対するユーザの認可情報が利用可能になり、これをキャッシュしておくことで今後の要求の速度を向上できます。
4. (オプション) エージェントがそのキャッシュから認可を実行する場合は、`Sm_AgentApi_Audit()` をコールすることにより、認証および認可のトランザクションをログ記録します。
5. ユーザの身元が既知であり、認可が検証され、必要な資格が取得されている場合に、リソースへのアクセスをユーザに許可します。

6. (オプション) `Sm_AgentApi_DoManagement()` をコールすることにより、管理要求を発行します。これは、ポリシー サーバで更新コマンドをポーリングするために必要に応じて実行される手順です。エージェントでは、コマンドに応じて暗号化キーを更新したり、キャッシュをクリアすることができます。
7. API インスタンスごとに `Sm_AgentApi_UnInit()` をコールすることにより、すべての API インスタンスを解放します。これにより、すべてのポリシー サーバへの TCP 接続が閉じられます。

エージェント API で提供されるキャッシュ機能では、セッション有効期間は適用されません。ユーザセッションやリソース固有権限をキャッシュするように選択すると、エージェントには各ユーザ要求中に独自のセッション管理を実行する必要性が生じることになります。これは、エージェント上でキャッシュすることにより、セッションの検証やリソース認可またはその両方を実行するためにポリシー サーバとやり取りする必要がなくなるためです。

## カスタム エージェントのコンパイルおよびリンク

カスタム エージェントを有効にして SiteMinder とやり取りするには、`SmAgentAPI.h` を使用してエージェント アプリケーションをコンパイルし、以下に示すプラットフォーム固有のライブラリにリンクします。UNIX プラットフォームでは、ライブラリ検索パスを指定する環境変数にライブラリを追加する必要があります。変数には、複数のディレクトリをコロンで区切って指定できます。これらの変数も以下に示します。

プラットフォーム	ライブラリ	ディレクトリ	変数名
Windows	SmAgentAPI.lib	<install_path>%sdk%lib%win32%	
Solaris	libsmagentapi.so	<install_path>/sdk/bin	LD_LIBRARY_PATH
HP-UX	libsmagentapi.so	<install_path>/sdk/bin	SHLIB_PATH:
AIX	libsmagentapi.so	<install_path>/sdk/bin	LIBPATH
Linux	libsmagentapi.so	<install_path>/sdk/bin	LD_LIBRARY_PATH

注: Microsoft Visual C++ を使用してエージェント API アプリケーションを構築する場合は、必ず `ws2_32.lib` とリンクしてください。

## 中央ホスト設定

カスタム エージェントも含め、SiteMinder エージェントは、エージェント API を介してポリシー サーバに接続されます。SiteMinder では、接続パラメータが提供される方法に基づいて、以下の 2 種類のエージェントを認識します。

- **v4.x エージェント**。ポリシー サーバに接続するために必要な接続パラメータはエージェント固有です。接続パラメータとしては、エージェントの名前、エージェントが配置されたホスト マシンの名前または IP アドレス、共有秘密キーなどがあります。これらのパラメータは、ポリシー サーバでエージェント オブジェクトを定義するときに提供されます。v4.x エージェント用に定義された接続パラメータは、そのエージェントのみに適用されます。
- **v5.x 以降のエージェント**。ポリシー サーバに接続するために必要な接続パラメータは、エージェントが配置されたホスト マシン上の複数のエージェントに適用可能です。v5.x 以降のエージェントを使用する場合、ホスト マシンはトラステッドホストと呼ばれます。

ポリシー サーバへの接続を初期化するために必要な情報は、ホスト マシンの `SmHost.conf` ファイルに保存されています。ポリシー サーバがそのエージェントとやり取りする方法を決定するためのその他の情報は、ポリシー サーバでホスト設定オブジェクトに定義されています。

ホスト設定オブジェクトパラメータ（フェールオーバーやクラスタ化の手順など）は、そのホスト設定オブジェクトに関連付けられたすべてのエージェントに適用されます。それらは **Administrative UI** で一元管理されます。

v5.x 以降のカスタム エージェントは、中央ホスト設定（ポリシー サーバとそのエージェントがやり取りする方法を決定する）をサポートしますが、中央エージェント設定（エージェントの作動方法を決定する）はサポートしません。ポリシー サーバでカスタム エージェントに関するエージェント設定オブジェクトを定義することはできません。カスタム エージェントの操作に関する設定パラメータは、`WebAgent.conf` ファイル（IIS 6.0 エージェントの場合には `LocalConfig.conf` ファイル）で定義されています。

## 設定要件

ポリシー サーバでホスト設定オブジェクトを使用してカスタム エージェントを設定するには、以下の手順を実行する必要があります。

1. エージェントがトラステッド ホストとして配置されているクライアント マシンを登録します。

トラステッド ホストを `smregghost` ツールに登録します。このツールは、`<install_path>/sdk/bin` ディレクトリにインストールされています。

トラステッド ホストを登録すると、以下のアイテムが作成されます。

- ポリシー サーバ上のホスト設定オブジェクト。ホスト設定オブジェクトは、**Administrative UI** でいつでも変更できます。
- クライアント上の `SmHost.conf` ファイル。このファイルには、ポリシー サーバ上のホスト設定オブジェクトを参照する `hostconfigobject` パラメータが含まれます。

`SmHost.conf` 内の情報は、エージェント API を介して、ポリシー サーバとそのエージェントの間の接続を初期化するために使用されます。接続が初期化されると、ホスト設定オブジェクト内の情報によりポリシー サーバとそのエージェントがやり取りする方法が決定されます。

2. ポリシー サーバ上のエージェント オブジェクトを定義します。

エージェント オブジェクトにより、カスタム エージェントに固有の名前やその他の情報が定義されるため、カスタム エージェント固有のアイデンティティが確立されます。

カスタム エージェントに割り当てられる名前は、プログラム上でカスタム エージェントが `SiteMinder` に渡す名前と一致する必要があります。

### コード要件

カスタム エージェントがポリシー サーバ上の中央ホスト設定オブジェクトによって設定されるようにするには、カスタム エージェントが以下の手順を実行する必要があります。

- `Sm_AgentApi_GetConfig()` を使用してエージェント設定を初期化します。

この関数は、トラステッド ホスト上に存在する `SmHost.conf` ファイルの名前とパスを `SiteMinder` に渡すことができます。このファイルは、ポリシー サーバ上のホスト設定オブジェクトを参照します。

また、`WebAgent.conf` ファイルに `SmHost.conf` への参照が含まれる場合には、`WebAgent.conf` ファイルの名前とパスを渡すこともできます。

トラステッド ホストがポリシー サーバに接続した後、トラステッド ホスト上のポリシー サーバとエージェントの間のやり取りは、ポリシー サーバ上のホスト設定オブジェクトによって決定されます。

**注:** `Sm_AgentApi_GetConfig()` をコールすると、`SiteMinder` によって `Sm_AgentApi_Init_t` 構造体に入力されます。中央ホスト設定では、手動で構造体に直接入力することはありません。

- `Sm_AgentApi_SetDefaultAgentId()` を使用してエージェントの名前を渡します。

カスタム エージェントがこの関数で `SiteMinder` に渡す名前は、ポリシー サーバでカスタム エージェント固有のアイデンティティを確立するエージェント オブジェクトの名前と一致する必要があります。

`Sm_AgentApi_Init()` をコールした後、他のエージェント API をコールする前に、`Sm_AgentApi_SetDefaultAgentId()` をコールします。これにより、トランザクションごとに名前を送信しなくても、カスタム エージェント名を `SiteMinder` に指定することができます。

## エージェントのアップグレード

### 既存の v4.x エージェントをアップグレードする方法

1. v4.x エージェントが配置されたホストマシンがトラステッドホストとしてポリシーサーバに現在登録されていない場合は、`smreghost` を実行してこのホストマシンを登録します。  
  
登録すると、トラステッドホスト上に `SmHost.conf` ファイルが作成され、ポリシーサーバ上にホスト設定オブジェクトが作成されます。
2. `Sm_AgentApi_GetConfig()` をコールして、`SmHost.conf` 内の情報で構造体 `Sm_AgentApi_Init_t` を初期化します。
3. `Sm_AgentApi_Init()` をコールして、ポリシーサーバに接続します。
4. `Sm_AgentApi_SetDefaultAgentId()` をコールして、カスタムエージェントのデフォルトの名前を設定します。

## エージェントコールシーケンス

通常、カスタムエージェントは、以下の順序でエージェント API 関数をコールします。

```
Sm_AgentApi_GetConfig() // Required with central host configuration
Sm_AgentApi_Init()
Sm_AgentApi_SetAgentInstanceInfo() エージェント ディスカバリの場合 (12.5 エージェント以降)
Sm_AgentApi_SetDefaultAgentId() // Central host configuration only
Sm_AgentApi_DoManagement()
Sm_AgentApi_IsProtected()
Sm_AgentApi_Login()
Sm_AgentApi_Authorize()
// . . .

// Call other Agent API functions here, including
// periodic calling of Sm_AgentApi_DoManagement()

Sm_AgentApi_Logout()
Sm_AgentApi_Uninit()
```

## サンプル カスタム エージェント

エージェント API のサンプル ソース コードが `smagentexample.cpp` で提供されています。このファイルは、`<install_path>/sdk/samples/smagentapi` にインストールされます。

このエージェント API サンプルは、`v5.x` またはそれ以降のエージェントとしてポリシー サーバに接続できます（中央ホスト設定を使用）。サンプルを実行すると、接続を初期化する前に、エージェント インターフェース（`v4.x` タイプまたは `v5.x` タイプ）の選択が求められます。

## エージェント API サービス

エージェント API が提供するサービスを使用して、高機能かつ安全で堅牢なエージェントを開発できます。エージェント構築の一環で、以下のサービスを使用します。

- セッション サービス
- 許可 サービス
- 監査 サービス
- 管理 サービス（キーの暗号化、キャッシュの更新）
- トンネル サービス

## セッション サービス

セッションは、ユーザ ログインが成功した後に作成されます。作成後は終了されるまでユーザセッションが継続します。多層アプリケーション環境内で整合性のあるユーザセッションを維持するために、（ポリシーサーバではなく）Web エージェントによってユーザセッション仕様が維持されます。セッション仕様は、セッションチケットと呼ばれます。セッション仕様は、ユーザセッションを表し、SiteMinder セッション管理の鍵となります。ユーザセッションが作成された環境によって、セッション仕様の永続ストレージが決定されます。たとえば、Web エージェント（HTTP 環境）はセッション仕様を HTTP Cookie に保存します。

SiteMinder のユニバーサル ID は、セッション メカニズムに統合されます。ユニバーサル ID は、カスタムアカウント番号などの一意の識別子を使用して、SiteMinder 環境内のアプリケーションのユーザを識別します。ユニバーサル ID は、アプリケーションに関係なく、ユーザの識別を自動的に実行するとにより、新旧アプリケーション間のユーザ識別を簡素化します。ユーザのユニバーサル ID をポリシー サーバ上で設定すると、このユニバーサル ID は、セッション仕様の一部となり、セッション期間を通じてエージェントに利用可能になります。

エージェントは、`Sm_AgentApi_Login()` を使用してセッションを作成します。この関数は、ユーザ認証情報を認証し、セッション仕様および一意のセッション ID を `Sm_AgentApi_Session_t` で返します。セッション仕様は、次のエージェント API コール時に更新され、これらのコールでも更新された有効回数が返されます。エージェントは、この情報を使用してカスタムセッション管理を実行し、セッションタイムアウトを追跡できます。

使用している Web サーバのユーザ追跡機能が有効な場合、SiteMinder から、セッション仕様と共に識別チケットが発行されます。識別チケットは、匿名認証方式によって保護されたリソースにユーザがアクセスしている場合のアイデンティティ ベースのパーソナライゼーションに使用できません。識別チケットに有効期限はありません。

アプリケーションのロジック フローがアプリケーション層を横切る場合、2つのエージェント間でセッション仕様を渡すことによりセッションを委任できます。各エージェントは、セッション指定の検証を選択することができます。

セッション仕様の検証は、ユーザセッションが期限切れではなく、終了や取り消しもされていないことを確認する場合に実行されます。これは、セッション期間中いつでも発生する可能性があります。エージェントは、`Sm_AgentApi_Login()` をコールしてセッション仕様を検証します。

ユーザがログアウトした後や、セッションの期限切れまたはセッションの取り消しによりエージェントがセッション仕様を破棄した後に、セッションが終了されます。セッションが終了された場合、ユーザが新しいセッションを確立するには再度ログインする必要があります。

セッションの開始後にユーザが無効化された場合、セッションは終了される必要があります。ユーザの無効状態を調べるには、`Sm_AgentApi_Login()` をコールしてセッションを検証します。

セッションを終了する場合、エージェントは `Sm_AgetnApi_Logout()` をコールします。このとき、そのセッション仕様 (`Sm_AgentApi_Session_t`) に対して割り当てられたメモリがすべて解放される必要があります。

### アプリケーション セッション情報

セッション情報の構成には、セッション仕様以外の情報も含まれる場合があります。セッション情報には、クライアントアプリケーションがユーザのセッションとの関連付けに必要とする情報を含めることができます。

アプリケーションに定義されるセッション情報は、セッション変数と呼ばれる名前/値のペアで構成されます。たとえば、アフィリエイト操作のビジネスロジック、証明書情報および SAML アサーションはすべて、セッション変数として保存し、セッション ID にバインドできます。

エージェント API では、セッション変数の設定、取得、削除用に以下の関数を提供します。

- `Sm_AgentApi_SetSessionVariables()`
- `Sm_AgentApi_GetSessionVariables()`
- `Sm_AgentApi_DelSessionVariables()`

セッション変数は、セッションストアと呼ばれるサーバ側のデータベースに保存されます。セッションストアはポリシーサーバによって管理されます。

## セッション変数の利点

クライアントアプリケーションがセッション変数を使用する場合、

- 各セッション変数値に対して最大 4K のデータを保存できます。
- セッション情報が複数のポリシー サーバにわたって維持されます。サーバ上のセッション情報の一元化により、異なるドメインにわたるログアウトおよびアイドルタイムアウトの適用を含むクロスドメインセッション管理などの機能が可能になります。

## セッション変数を使用するための要件

クライアントアプリケーションでセッション変数を使用するには、以下の前提条件の両方に適合する必要があります。

- セッションストアをポリシーサーバ管理コンソールで有効にする必要があります。
- **Administrative UI** でのレلم設定時に、セッション中にアクセスされるレلمの少なくとも 1 つに対して [永続セッション] が選択されていること。永続セッションに対して設定されたレلمにユーザがアクセスすると即座に、それ以降のセッション全体を通じてセッション変数が使用可能になります。

## セッション終了時のクリーンアップ

ユーザがログアウトし、エージェントがセッション指定を破棄すると、セッションは終了します。永続セッションの場合、SiteMinder は、すべてのセッション変数を含めたすべてのセッション情報をセッションストアから削除します。

## タイムアウト

エージェントでは、セッションタイムアウトを自身で適用したり、各要求の検証をポリシーサーバに依存することができます。通常、エージェントによるユーザセッションや権限のキャッシュには、エージェント側での何らかの形式のタイムアウト適用が必要となります。この場合、最終アクセス時間を追跡し、セッションの期限を把握する役割がエージェントに課せられます。

キャッシュしないエージェントは、ポリシー サーバのタイムアウトの適用を利用できます。以下のエージェント API メソッドは、それぞれのコール後に更新されたタイムアウト情報を返します。

- `Sm_AgentApi_Login()`
- `Sm_AgentApi_Authorize()`
- `Sm_AgentApi_Audit()`

## 認可サービス

アクセス制御機能を実行するエージェントは、エージェント API の認可サービスを使用します。これらのサービスを使用することにより、クライアントでは、どのアクセス制御がリソースに対して実行されているかを判別したり、リソースにアクセスするためのユーザ権限を確認したり、特定のリソースに対するユーザ権限を取得することができます。

リソースが保護されているかどうかの判別は、`Sm_AgentApi_IsProtected()` メソッドをコールすることにより行えます。このメソッドは、要求元のエージェントによってサービスが提供されているリソースに対応して、認証に必要な認証情報に関する情報を返します。

ユーザの身元が確認された後、エージェントは、`Sm_AgentApi_Authorize()` メソッドをコールして、要求元ユーザが要求対象リソースのアクセス権を持つかどうかを判別します。エージェントでは、このメソッドによって返されたレスポンス属性の値をテストすることにより、きめ細かなアクセス制御を実行できます。

## トランザクションの追跡

エージェントがセッション中のすべてのユーザ アクティビティを追跡するための機能が用意されています。アクティビティの多くはポリシーサーバによってログ記録されますが、エージェント キャッシュから実行された認可のログ記録が必要となる場合があります。エージェントは、`Sm_AgentApi_Audit()` メソッドをコールして、リソースに対する要求をログ記録することができます。

エージェントは、一意のトランザクション ID を生成することによって、アクセス制御アクティビティアプリケーションアクティビティに関連させることができます。トランザクション ID は、認可方式と監査方式の両方に与えることができるため、ポリシーサーバではトランザクションに固有の ID をアプリケーションアクティビティと関連付けて記録できます。これは非否認に使用できます。

## 管理サービス

管理プロトコルはエージェントとポリシーサーバの間に存在します。エージェントでは、このプロトコルを使用して、ポリシーサーバのポリシーおよび管理上の変更に合致した方式でキャッシュと暗号化キーを管理することができます。

## キャッシュコマンド

エージェントは、SM\_AGENTAPI\_MANAGEMENT\_GET\_AGENT\_COMMANDS コマンドと共に Sm\_AgentApi\_DoManagement() コールを発行して、最新のエージェントコマンドを要求します。通常、これは、バックグラウンドで実行されているスレッド別に *N* 秒ごとに実行されます。受信可能なエージェントコマンドのタイプは、キャッシュコマンドおよび暗号化コマンドです。

キャッシュコマンドは、ポリシーサーバの管理上の更新の結果として実行されることが必要となる可能性のあるキャッシュ変更をエージェントに通知します。これらのエージェントコマンドは次のとおりです。

- SM\_AGENTAPI\_CACHE\_FLUSH\_ALL
- SM\_AGENTAPI\_CACHE\_FLUSH\_ALL\_USERS
- SM\_AGENTAPI\_CACHE\_FLUSH\_THIS\_USER
- SM\_AGENTAPI\_CACHE\_FLUSH\_ALL\_REALMS
- SM\_AGENTAPI\_CACHE\_FLUSH\_THIS\_REALM

### 暗号化コマンド

暗号化コマンドは、ポリシー サーバによって自動的に生成された新しい暗号化キーや管理上で生成された新しい暗号化キーをエージェントに通知します。セキュア状態を保存する必要があるエージェントは、このプロトコルを活用して、最新の暗号化キーを常に把握することができます。これらのエージェント コマンドは次のとおりです。

- SM\_AGENTAPI\_AGENT\_KEY\_UPDATE\_NEXT
- SM\_AGENTAPI\_AGENT\_KEY\_UPDATE\_LAST
- SM\_AGENTAPI\_AGENT\_KEY\_UPDATE\_CURRENT
- SM\_AGENTAPI\_AGENT\_KEY\_UPDATE\_PERSISTENT

### トンネル サービス

トンネル サービスは、ポリシー サーバ上に置かれた呼び出し可能なサービスとのセキュアな通信を確立するためにエージェントを有効化します。これにより、VPN と同様なセキュアなチャンネルでカスタム アクションを実行することができ、暗号化やキー管理など問題を処理する必要がなくなります。

## レスポンス属性

レスポンス属性は、エージェントに情報を配信するためにポリシー サーバを有効化します。以下に示す 2 種類の属性があります。

- 既知の属性
- ポリシー ベースの属性

既知の属性は、`Sm_AgentApi_Login()` などの特定のコールの後、常にポリシー サーバから返されます。これらの属性は、ユーザ DN やユニバーサル ID などの静的な固定データを表します。

ポリシー ベースの属性は、`Sm_AgentApi_Login()` および `Sm_AgentApi_Authorize()` から返されます。これらの属性は、ポリシーに基づくものであり、ポリシー サーバからエージェントに静的なデータおよび動的なデータを提供するための伝達手段となります。その結果、エージェントでの認証の属性と認可の属性の区別が可能になります。データの実際のソースは、各種のソースからのデータを配信するように設定できるレスポンス機能を使用してポリシー サーバに定義されます。データには、静的な情報、すなわちディレクトリ プロファイルやカスタム ポリシー サーバ プラグインからの情報が含まれることがあります。レスポンスが正しく設定されると、エージェントでは、きめの細かいアクセス制御やプロファイル方式のパーソナライゼーションを実行できます。

ポリシー定義に基づき、レスポンス属性はタイムアウトする場合や、ユーザセッションの期間にキャッシュされる場合があります。ポリシー サーバは、秒単位で計算された TTL (有効期間) 値と共に属性を配信します。エージェントは、ユーザセッションや認可またはその両方をキャッシュする場合、関連する属性を最新の状態に維持する役割を持ちます。エージェントでは、`Sm_AgentApi_UpdateAttributes()` コールを発行して、古い属性を更新します。

## カスタム エージェントおよびシングル サインオン

シングルサインオン環境では、所定のエージェントで正しく認証されたユーザが別のエージェントによって保護されたレルムにアクセスする場合、再認証は不要です。カスタム エージェントがシングルサインオン環境に関与する場合、2 つのエージェントは同じ Cookie ドメイン (たとえば、`xxx.domainname.com` など) に配置される必要があります。

シングルサインオンは、`SMSESSION` というシングルサインオン Cookie によって実現されます。この Cookie は、`SiteMinder` またはカスタム エージェントによって作成され、ユーザのブラウザに書き込まれます。

エージェント API には、標準の SiteMinder Web エージェントによるシングルサインオン環境にカスタム エージェントが参加する場合に使用可能な以下の 2 つの関数が含まれています。

- **Sm\_AgentApi\_DecodeSSOToken()**。カスタム エージェントは、既存の SMSESSION Cookie から Cookie の内容（トークンと呼ばれます）を抽出し、トークンをこの関数に渡します。関数では、トークンを復号し、指定された情報を抽出します。この関数は、トークンの最終アクセスのタイムスタンプを更新するためにも使用できます。
- **Sm\_AgentApi\_CreateSSOToken()**。ユーザがカスタム エージェントを介して正常にログインした後、カスタム エージェントは、ユーザに関する情報をこの関数に渡します。関数では、このユーザ情報、およびログインコールから返されたセッション情報から、暗号化されたトークンを作成します。カスタム エージェントは、トークンを SMSESSION Cookie に書き込みます。

シングルサインオン関数のパラメータをセットアップし、結果を解析する例については、サンプルカスタム エージェント コードを参照してください。サンプル カスタム エージェント コードは、`<install_path>%sdk%samples` の `smagentapi` ディレクトリにあります。

## 標準エージェント サポート

SiteMinder SDK v5.5 SPx 以降で作成されたカスタム エージェントは、標準の SiteMinder Web エージェントによって作成された SMSESSION Cookie を使用できます。

ただし、標準エージェントが適切な SiteMinder Agent Quarterly Maintenance Release (QMR) でアップグレードされている場合には、標準 SiteMinder Web エージェントはカスタム エージェントによって作成された Cookie しか使用できません。各標準エージェントのバージョンに必要な QMR バージョンの詳細については、付属の SDK リリース ノートを参照してください。

また、適切な QMR でアップグレードされた SiteMinder エージェントがカスタム エージェントによって作成された SMSESSION Cookie を使用できるようにするには、標準エージェントのエージェント設定ファイル (IIS サーバの場合は `LocalConfig.conf`、他のサーバの場合は `WebAgent.conf`) または中央設定オブジェクト (v5.x またはそれ以降に対応) に以下のエントリが含まれる必要があります。

```
AcceptTPCookie="yes"
```

AcceptTPCookie を以下のとおりに設定します。

- 4.xQMR4 エージェント以上の場合、AcceptTPCookie="yes" を標準エージェントのエージェント設定ファイルに直接追加します。
- 5.xQMR1 エージェント以降において、エージェント設定オブジェクトの AllowLocalConfig パラメータが no に設定されている場合は、エントリをエージェント設定オブジェクトに追加します。AllowLocalConfig が yes に設定されている場合には、標準エージェントのエージェント設定ファイルに AcceptTPCookie を設定できます。

## カスタム エージェント経由のログイン

ここでは、初期ログインがカスタム エージェントを通して実行される場合のシングルサインオン環境のイベントの典型的なシーケンスを示します。

1. ユーザはカスタム エージェント経由でログインします。
2. カスタム エージェントは、Sm\_AgentApi\_Login() をコールしてユーザを認証します。ユーザは認証情報を要求されます。
3. カスタム エージェントは Sm\_AgentApi\_CreateSSOToken() をコールし、これにユーザに関する情報（ユーザ名、ユーザ DN、要求元クライアントの IP アドレス）を渡します。SiteMinder は、この情報を、ログインコールから返されたセッション情報と共にトークンに追加します。SiteMinder はトークン内の情報の暗号化も実行します。
4. カスタム エージェントは、ユーザのブラウザに SMSESSION Cookie を作成し、この Cookie にトークンを書き込みます。
5. ユーザは標準的な SiteMinder エージェントによって保護されたリソースを要求します。
6. 標準エージェントはログイン操作を実行します。この操作により、シングルサインオン Cookie 内の情報に基づいてユーザが検証されます。ユーザは、認証情報を要求されません。

### 標準エージェント経由のログイン

ここでは、シングルサインオン環境において、初期ログインが標準 SiteMinder Web エージェント経由の場合のイベントの標準的なシーケンスを示します。

1. ユーザは標準エージェント経由でログインします。
2. 標準エージェントは、ログインコールを通じて認証情報をユーザに要求して、ユーザを認証します。
3. SiteMinder は、ユーザのブラウザに **SMSESSION Cookie** を作成し、セッション情報が含まれる暗号化トークンを挿入します。
4. ユーザはカスタム エージェントによって保護されたリソースを要求します。
5. カスタム エージェントは、ユーザの要求から **SMSESSION Cookie** を取得し、トークンを抽出します。
6. カスタム エージェントは、トークンを `Sm_AgentApi_DecodeSSOToken()` 関数に渡します。関数は、トークンをデコードし、トークンの属性のサブセットをカスタム エージェントに返します。
7. カスタム エージェントは、セッション仕様をトークンから取得し、このセッション仕様を `Sm_AgentApi_Login()` に渡します。ログインコールにより、ユーザへ認証情報を要求することなく、ユーザが検証されます。
8. ユーザは標準的な SiteMinder エージェントによって保護されたリソースを要求します。
9. 標準エージェントはログイン操作を実行し、**SMSESSION Cookie** の内容に基づいてユーザを検証します。ユーザは、認証情報を要求されません。

### メモリの解放

カスタム エージェント用に割り当てたすべてのメモリを明示的に解放する必要があります。 `Sm_AgentApi_Attribute_t` 構造体内のレスポンス属性を解放するには、`Sm_AgentApi_FreeAttributes()` をコールします。

詳細情報:

[Sm\\_AgentApi\\_FreeAttributes\(\)](#) (P. 89)

## エージェント API のデータ構造体 (C)

構造体を使用する前に、すべての未使用のフィールドをゼロに設定する必要があります。その最良の方法は、構造体の全体をゼロに設定することです。

### Sm\_AgentApi\_AgentDiscovery\_t

この構造体は、エージェント インスタンス パラメータに関する情報を定義します。

#### 構文

この構造体の形式は以下のとおりです。

```
typedef struct Sm_AgentApi_AgentDiscovery_s
{
    char paramname[SM_AGENTAPI_SIZE_NAME];
    char paramvalue[SM_AGENTAPI_SIZE_NAME];
} Sm_AgentApi_AgentDiscovery_t;
```

### パラメータ

この構造体には以下のパラメータがありました。

#### paramname

エージェント インスタンス属性を指定するには、以下の名前を使用します。

- `SM_AGENT_INSTANCE_AGENTPRODUCTTYPE` -- エージェント インスタンスのタイプ
- `SM_AGENT_INSTANCE_AGENTPRODUCTVERSION` -- エージェント インスタンスのバージョン
- `SM_AGENT_INSTANCE_AGENTPRODUCTSUBTYPE` -- Web サーバのタイプ
- `SM_AGENT_INSTANCE_AGENTPRODUCTOSTYPE` -- ホストしているエージェントの OS のタイプ
- `SM_AGENT_INSTANCE_AGENTIDFILE` -- エージェント GUID が含まれるファイルのパス
- `SM_AGENT_INSTANCE_ACONAME` -- エージェント設定オブジェクトの名前
- `SM_AGENT_INSTANCE_HCONAME` -- ホスト設定オブジェクトの名前
- `SM_AGENT_INSTANCE_FIPSMODE` -- エージェントと PS の接続に使用されている FIPS モード

#### paramvalue

指定されたユーザ属性の値。

### 注釈

すべてのエージェント インスタンス属性が「unknown (不明)」の値に初期化されます。 エージェント API を使用する開発者が

`Sm_AgentApi_SetAgentInstanceInfo()` をコールすることによって適切な値を指定します。 このコールが実行されない場合、エージェントはエージェント ディスカバリの範囲に含まれません。

## Sm\_AgentApi\_Attribute\_t

この構造体は、レスポンス属性に関する情報を定義します。

### 構文

この構造体の形式は以下のとおりです。

```
typedef struct Sm_AgentApi_Attribute_s
{
    long nAttributeId;
    long nAttributeTTL;
    long nAttributeFlags;
    char lpszAttributeOid[SM_AGENTAPI_SIZE_OID];
    long nAttributeLen;
    char* lpszAttributeValue;
} Sm_AgentApi_Attribute_t;
```

### パラメータ

この構造体には以下のパラメータがあります。

#### nAttributeId

レスポンス属性の ID。

#### nAttributeTTL

レスポンス属性の存続期間 (TTL) の値 (秒)。属性は TTL 値の期間中キャッシュに保持されます。

#### nAttributeFlags

レスポンス属性フラグ。このフラグは、以下のセッションストア関数で使用されます。

- Sm\_AgentApi\_DelSessionVariables()
- Sm\_AgentApi\_GetSessionVariables()
- Sm\_AgentApi\_SetSessionVariables()

詳細については、これらの関数の ppRespAttributes パラメータを参照してください。

#### lpszAttributeOid

レスポンス属性オブジェクト識別子。

**nAttributeLen**

レスポンス属性の長さ。

**lpszAttributeValue**

レスポンス属性の Null 終端属性値。

**注釈**

以下の既知の認証属性は、**Sm\_AgentApi\_Login()** によって返され、**Sm\_AgentApi\_Attribute\_t** 構造体の **nAttributeId** フィールドで参照されます。

**SM\_AGENTAPI\_ATTR\_AUTH\_DIR\_OID**

ユーザが認証されたディレクトリの **SiteMinder** オブジェクト ID。これは、**SiteMinder** ユーザディレクトリに割り当てられた内部オブジェクト ID です。

**SM\_AGENTAPI\_ATTR\_AUTH\_DIR\_NAME**

ユーザが認証されたディレクトリに関する **SiteMinder** の「name」指定。これは、**SiteMinder** ユーザディレクトリ ダイアログで指定されたディレクトリ名です。

**SM\_AGENTAPI\_ATTR\_AUTH\_DIR\_SERVER**

ユーザが認証されたディレクトリに関する **SiteMinder** の「server」指定。これは、**SiteMinder** ユーザディレクトリ ダイアログで指定されたディレクトリ サーバです。

**SM\_AGENTAPI\_ATTR\_AUTH\_DIR\_NAMESPACE**

ユーザが認証されたディレクトリに関する **SiteMinder** の「namespace」指定。これは、**SiteMinder** ユーザディレクトリ ダイアログで指定されたディレクトリ ネームスペース (LDAP:、ODBC:、WinNT:、AD:) です。

**SM\_AGENTAPI\_ATTR\_USERMSG**

認証の結果としてユーザに示されたテキスト。認証方式によっては、認証リクエスト時のテキストや認証に失敗した理由が表示されます。この属性に対する値は、**SmAuthenticate()** の **lpszUserMsg** パラメータによって提供できます。

**SM\_AGENTAPI\_ATTR\_USERDN**

**SiteMinder** によって認識されているユーザ識別名。  
この属性は、シングルサインオン操作でも使用されます。

**SM\_AGENTAPI\_ATTR\_USERUNIVERSALID**

ユーザディレクトリに設定されているユーザユニバーサル ID。

**SM\_AGENTAPI\_ATTR\_IDENTITYSPEC**

ユーザの識別チケットです。これは、ユーザ追跡機能が有効化されている場合に SiteMinder から返されます。

以下の既知の属性がシングルサインオン操作で使用され、`Sm_AgentApi_Attribute_t` 構造体の `nAttributeId` フィールドで参照されます。

**SM\_AGENTAPI\_ATTR\_USERDN**

ユーザの識別名。

**SM\_AGENTAPI\_ATTR\_SESSIONSPEC**

ログインコールから返されたセッション仕様。

**SM\_AGENTAPI\_ATTR\_SESSIONID**

ログインコールから返されたセッション ID。

**SM\_AGENTAPI\_ATTR\_USERNAME**

ユーザの名前。

**SM\_AGENTAPI\_ATTR\_CLIENTIP**

ユーザが保護されたリソースのリクエストを開始したマシンの IP アドレス。

**SM\_AGENTAPI\_ATTR\_DEVICENAME**

トークンをデコードしているエージェントの名前です。

**SM\_AGENTAPI\_ATTR\_InnnnnDLESSIONTIMEOUT**

セッションの最大アイドル時間。

### SM\_AGENTAPI\_ATTR\_STARTSESSIONTIME

セッションが成功した後、ログインが開始された時刻。

### SM\_AGENTAPI\_ATTR\_LASTSESSIONTIME

セッション期間中にポリシー サーバが最後にアクセスされた時刻。

### SM\_AGENTAPI\_ATTR\_SSOZONE

SSO ゾーン名を指定します。これは、Sm\_AgentApi\_CreateSSOToken メソッドをコールする場合に提供します。ゾーン名を指定しない場合、デフォルトは "SM" です。この値は、Sm\_AgentApi\_DecodeSSOToken メソッドによって返される属性リストで読み取ることができます。

以下の既知の管理属性が Sm\_AgentApi\_DoManagement() によって返され、Sm\_AgentApi\_Attribute\_t 構造体の nAttributeld フィールドで参照されます。

### SM\_AGENTAPI\_AFFILIATE\_KEY\_UPDATE

アフィリエイト エージェントの名前を更新するようにエージェントに指示します。

### SM\_AGENTAPI\_AGENT\_KEY\_UPDATE\_NEXT

「next」エージェント キーを更新するようにエージェントに指示します。値には、バイナリ データの 24 バイトが含まれます。

### SM\_AGENTAPI\_AGENT\_KEY\_UPDATE\_LAST

「last」エージェント キーを更新するようにエージェントに指示します。値には、バイナリ データの 24 バイトが含まれます。

### SM\_AGENTAPI\_AGENT\_KEY\_UPDATE\_CURRENT

「current」エージェント キーを更新するようにエージェントに指示します。値には、バイナリ データの 24 バイトが含まれます。

### SM\_AGENTAPI\_AGENT\_KEY\_UPDATE\_PERSISTENT

スタティック (永続) エージェント キーを更新するようにエージェントに指示します。値には、バイナリ データの 24 バイトが含まれます。

### SM\_AGENTAPI\_CACHE\_FLUSH\_ALL

キャッシュ内のすべての情報をクリアするようにエージェントに指示します。

### SM\_AGENTAPI\_CACHE\_FLUSH\_ALL\_USERS

キャッシュに保存されたすべてのユーザ情報をクリアするようにエージェントに指示します。

**SM\_AGENTAPI\_CACHE\_FLUSH\_THIS\_USER**

指定されたユーザに関連するすべてのキャッシュ情報をクリアするようにエージェントに指示します。値には、<user dir oid> / <user dn> が含まれます。

**SM\_AGENTAPI\_CACHE\_FLUSH\_ALL\_REALMS**

キャッシュに保存されたすべてのリソース情報をクリアするようにエージェントに指示します。

**SM\_AGENTAPI\_CACHE\_FLUSH\_THIS\_REALM**

指定されたレルムに関連するすべてのリソース情報をクリアするようにエージェントに指示します。値は、レルム **OID** です。

## Sm\_AgentApi\_Init\_t

この構造体は、サーバ情報を含めたエージェントの初期化を定義し、さらにフェールオーバーしきい値も指定します。

### 構文

この構造体の形式は以下のとおりです。

```
typedef struct Sm_AgentApi_Init_s
{
    long nVersion;
    char lpszHostName[SM_AGENTAPI_SIZE_NAME];
    char lpszSharedSecret[SM_AGENTAPI_SIZE_NAME];
    long nFailover;
    long nNumServers;
    Sm_AgentApi_Server_t* pServers;
} Sm_AgentApi_Init_t;
```

### パラメータ

この構造体には以下のパラメータがありました。

#### nVersion

エージェント API のバージョン。SM\_AGENTAPI\_VERSION に設定します。

#### lpszHostName

エージェント名。この名前は、ポリシー サーバに提供されたエージェント名に一致する必要があります。エージェント名は、大文字と小文字が区別されません。

#### lpszSharedSecret

それぞれのエージェントには、ポリシー サーバに登録された共有秘密キーがあります。その秘密キーをここで入力します。このフィールドは、大文字と小文字が区別されます。

#### nFailover

これを **0** に設定すると、エージェントは、ラウンドロビン設定内の指定されたポリシー サーバにアクセス可能になります。これを **1** に設定すると、ラウンドロビン設定が無効になり、その場合には、エージェントがフェールオーバーモードで動作します。

#### nNumServers

next パラメータで定義されたポリシー サーバの数。

#### pServers

Sm\_AgentApi\_Server\_t 型の nNumServers 構造体の配列。この構造体は、エージェントが通信する同一設定のポリシー サーバのそれぞれを定義します。これらのサーバは、共通のポリシー ストアを共有する必要があります。指定されたエージェント API バージョンに応じて、配列にはクラスタ化されたサーバまたはクラスタ化されていないサーバが含まれます。

### 注釈

この構造体は、Sm\_AgentApi\_GetConfig() をコールすることにより入力されます。

詳細情報:

[Sm\\_AgentApi\\_GetConfig\(\)](#) (P. 91)

[中央ホスト設定](#) (P. 38)

## Sm\_AgentApi\_ManagementContext\_t

この構造体は、管理コマンドに関する情報を定義します。

### 構文

この構造体の形式は以下のとおりです。

```
typedef struct Sm_AgentApi_ManagementContext_t
{
    long nCommand;
    char lpszData[SM_AGENTAPI_SIZE_NAME];
} Sm_AgentApi_ManagementContext_t;
```

### パラメータ

この構造体には以下のパラメータがあります。

#### nCommand

これらの管理コマンドのいずれかです。

- **SM\_AGENTAPI\_MANAGEMENT\_GET\_AGENT\_COMMANDS**  
最新のエージェント コマンドを要求します。このコマンドでは、`lpszData` を空のままにする必要があります。
- **SM\_AGENTAPI\_MANAGEMENT\_SET\_AGENT\_INFO**  
エージェントに関する情報をポリシー サーバに提供します。情報は `lpszData` に含まれています。

#### lpszData

**SM\_AGENTAPI\_MANAGEMENT\_SET\_AGENT\_INFO** が設定されている場合、以下のエージェントに関する情報の Null 終端文字列が含まれています。

```
"Product=XXX,Platform=XXX,Version=XXX,Label=XXX"
```

不明な句は省略可能です。

この情報は、ログ記録とトラブルシューティングの目的でポリシーサーバによって記録されます。

例：

```
"Product=WebAgent,Platform=NT/ISAPI,  
Version=5.0,Update=SP1,Label=C134"
```

エージェント API は、暗号強度およびエージェント時間 (GMT) とタイムゾーンをこの文字列に追加します。

カスタム エージェントに関する情報をログに記録するには、次のコード行をカスタム エージェント定義に追加します。

```
ManagementContext.nCommand = SM_AGENTAPI_MANAGEMENT_SET_AGENT_INFO;  
strncpy (ManagementContext.lpszData, version info goes here, 255);
```

## Sm\_AgentApi\_Realm\_t

この構造体は、リソースが保護されているレルムに関する情報を定義します。

### 構文

この構造体の形式は以下のとおりです。

```
typedef struct Sm_AgentApi_Realm_s
{
    char lpszDomainOid[SM_AGENTAPI_SIZE_OID];
    char lpszRealmOid[SM_AGENTAPI_SIZE_OID];
    char lpszRealmName[SM_AGENTAPI_SIZE_NAME];
    long nRealmCredentials;
    char lpszFormLocation[SM_AGENTAPI_SIZE_URL];
} Sm_AgentApi_Realm_t;
```

### パラメータ

この構造体には以下のパラメータがあります。

#### lpszDomainOid

ドメインの一意の識別子。

#### lpszRealmOid

レルムの一意の識別子。

#### lpszRealmName

リソースが保護されているレルムの名前。

#### nRealmCredentials

必要な認証情報を示す値のビットマスク。値は、`Sm_Api_Credentials_t` 列挙型で定義されています (`SmApi.h` で定義)。値 0 は、認証情報が不要であることを意味します。型は以下のとおりです。

- `Sm_AuthApi_Cred_None`  
認証情報は不要です。
- `Sm_AuthApi_Cred_Basic`  
ユーザ名およびパスワードが必要です。
- `Sm_AuthApi_Cred_Digest`  
ダイジェストが必要です。
- `Sm_AuthApi_Cred_X509Cert`  
X.509 証明書が必要です。

- **Sm\_AuthApi\_Cred\_X509CertUserDN**  
X.509 証明書およびユーザ DN が必要です。
- **Sm\_AuthApi\_Cred\_X509CertIssuerDN**  
X.509 証明書の発行者 DN が必要です。
- **Sm\_AuthApi\_Cred\_CertOrBasic**  
X.509 証明書またはユーザ名/パスワードのいずれかが必要です。
- **Sm\_AuthApi\_Cred\_CertOrForm**  
X.509 証明書またはフォーム ベースの認証方式のいずれかが必要です。
- **Sm\_AuthApi\_Cred\_NTChalResp**  
NT チャレンジ/レスポンス プロトコルを使用します。
- **Sm\_AuthApi\_Cred\_SSLRequired**  
SSL が必要です。
- **Sm\_AuthApi\_Cred\_FormRequired**  
HTML フォームへのリダイレクトが必要です。
- **Sm\_AuthApi\_Cred\_AllowSaveCreds**  
認証情報のヒントを保存します。
- **Sm\_AuthApi\_Cred\_PreserveSessionId**  
現在のセッションが引き続き有効な場合は、セッション ID が保持される必要があります。
- **Sm\_AuthApi\_Cred\_DoNotChallenge**  
認証情報は要求されません。

### lpszFormLocation

フォーム認証情報プロバイダの URL (<http://...>) 。

## Sm\_AgentApi\_ResourceContext\_t

この構造体は、保護および認可の対象のリソースを定義します。

### 構文

この構造体の形式は以下のとおりです。

```
typedef struct Sm_AgentApi_ResourceContext_s
{
    char lpszAgent[SM_AGENTAPI_SIZE_NAME];
    char lpszServer[SM_AGENTAPI_SIZE_NAME];
    char lpszAction[SM_AGENTAPI_SIZE_NAME];
    char lpszResource[SM_AGENTAPI_SIZE_URL];
} Sm_AgentApi_ResourceContext_t;
```

### パラメータ

この構造体には以下のパラメータがあります。

#### lpszAgent

リソースを保持するアフィリエイト エージェントの名前です。このフラグは、アフィリエイト エージェントでの操作用に予約されています。このフィールドは空のままにします。

#### lpszServer

オプションフィールド。**Web** リソースをホストする **Web** サーバの名前を指定するために使用されます（例：**www.myorg.org**）。

#### lpszResoure

要求されている **Web** リソースの名前（例：**/inventory/**）。

#### lpszAction

リソース上で実行されるアクションのタイプ（例：**GET**）。

### Sm\_AgentApi\_Server\_t

この構造体は、各サーバの接続設定を定義します。

#### 構文

この構造体の形式は以下のとおりです。

```
typedef struct Sm_AgentApi_Server_s
{
    char lpszIpAddr[SM_AGENTAPI_SIZE_NAME];
    long nConnMin;
    long nConnMax;
    long nConnStep;
    long nTimeout;
    long nPort[3];
    void* pHandle[3];
    long nClusterSeq;
} Sm_AgentApi_Server_t;
```

#### パラメータ

この構造体には以下のパラメータがあります。

##### pszIpAddr

ポリシーサーバの IP アドレス。ポリシーサーバの内の 3 つのサービス（認可、認証、およびアカウントिंग）のそれぞれに対して TCP 接続のパイプが形成されます。

##### nConnMin

TCP 接続のパイプを記述する場合、これが接続の初期数です。

##### nConnMax

TCP 接続のパイプを記述する場合、これがパイプ内の許可されている接続の最大数です。

##### nConnStep

必要に応じて、この値を大きくすることにより、TCP 接続のパイプ内の接続の数が増加されます。

##### nTimeout

エージェントがポリシーサーバに到達できないと判定されるまで秒数。このパラメータは、SiteMinder インストール全体の総合的なスループットおよび遅延条件に基づいて設定可能です。

**nPort**

ポリシー サーバが単一のアクセス制御 TCP ポートに対して設定される場合、結合されたポートを指すには、SM\_AGENTAPI\_POLICYSERVER 定数を使用します。以下の定数は、下位互換性が維持されています。

- SM\_AGENTAPI\_AZ\_SERVER
- SM\_AGENTAPI\_AUTH\_SERVER
- SM\_AGENTAPI\_ACCT\_SERVER

**pHandle**

予約されています。Null に設定します。

**nClusterSeq**

クラスタ シーケンス番号。シーケンス番号は 1 から始まります。非クラスタ サーバの場合は、このパラメータを省略します。

## Sm\_AgentApi\_Session\_t

この構造体は、ユーザのセッションに関する情報を定義します。

**構文**

この構造体の形式は以下のとおりです。

```
typedef struct Sm_AgentApi_Session_s
{
    long nReason;
    long nIdleTimeout;
    long nMaxTimeout;
    long nCurrentServerTime;
    long nSessionStartTime;
    long nSessionLastTime;
    char lpszSessionId[SM_AGENTAPI_SIZE_OID];
    char lpszSessionSpec[SM_AGENTAPI_SIZE_SESSIONSPEC];
} Sm_AgentApi_Session_t;
```

### パラメータ

この構造体には以下のパラメータがあります。

#### nReason

追加のステータス コード：失敗した認証や認可の理由を説明します。または、`Sm_AgentApi_Logout()` の呼び出しに成功した場合に渡されます。SmApi Designer で定義されます。

#### nIdleTimeout

ユーザがリソースにアクセスしない場合に、エージェントがユーザに再認証を要求するまで、セッションが有効な状態を継続できる最大時間（秒で定義）。

#### nMaxTimeout

エージェントがユーザに再認証を要求するまで、ユーザセッションがアクティブ状態を継続できる最大時間（秒で定義）。

#### nCurrentServerTime

ポリシー サーバに設定された現在時刻（GMT）。

#### nSessionStartTime

セッションが開始したサーバ時間（GMT）。

#### nSessionLastTime

ポリシー サーバがセッションを最後に確認したサーバ時間（GMT）。

#### lpszSessionId

セッションを識別するため返される opaque 値。

#### lpszSessionSpec

セッションを識別するため返される opaque 値。opaque 値はセッション仕様を表します。

## Sm\_AgentApi\_TunnelServiceRequest\_t

この構造体は、リモート サービス ライブラリに関する情報を定義します。

### 構文

この構造体の構文は以下のとおりです。

```
typedef struct Sm_AgentApi_TunnelServiceRequest_s
{
    char lpszLibrary[SM_AGENTAPI_SIZE_NAME];
    char lpszFunction[SM_AGENTAPI_SIZE_NAME];
    char lpszParameter[SM_AGENTAPI_SIZE_USERINFO];
    long nLength;
    void* pData;
} Sm_AgentApi_TunnelServiceRequest_t;
```

### パラメータ

この構造体には以下のパラメータがあります。

#### lpszLibrary

ポリシー サーバによって呼び出されるサービスの名前。

#### lpszFunction

サービス内でコールするメソッドの名前。

#### lpszParameter

メソッドに渡される任意の文字列パラメータ。

#### nLength

メソッドに渡されるデータの長さ。最大長は、`Sm_AgentApi_GetMaxTunnelBufSize()` をコールすることにより決定されます。

#### pData

データへのポインタ。

### Sm\_AgentApi\_UserCredentials\_t

この構造体は、認証情報をサーバに渡すために使用されます。

**注:** エージェントは、ポリシー サーバによって要求された関連情報のみを提供します。

#### 構文

この構造体の形式は以下のとおりです。

```
typedef struct Sm_AgentApi_UserCredentials_s
{
    long nChallengeReason;
    char lpszUsername[SM_AGENTAPI_SIZE_USERINFO];
    char lpszPassword[SM_AGENTAPI_SIZE_USERINFO];
    char lpszCertUserDN[SM_AGENTAPI_SIZE_USERINFO];
    char lpszCertIssuerDN[SM_AGENTAPI_SIZE_USERINFO];
    long nCertBinaryLen;
    char* lpszCertBinary;
} Sm_AgentApi_UserCredentials_t;
```

#### パラメータ

この構造体には以下のパラメータがあります。

##### nChallengeReason

失敗したまたは認証情報要求済みの以前の認証の最初の理由コード。

##### lpszUsername

認証中のユーザの名前。

##### lpszPassword

認証中のユーザのパスワード。

##### lpszCertUserDN

このフィールドは、Null に設定される必要があります。ユーザ DN を含めた完全な証明書データを **lpszCertBinary** に指定します。このフィールドでユーザ DN を指定する既存エージェントアプリケーションは、値を Null に変更する必要はありません。ユーザ DN 値は、下位互換性の目的でサポートされています。

**lpszCertIssuerDN**

このフィールドは、Null に設定される必要があります。発行者 DN を含めた完全な証明書データを **lpszCertBinary** に指定します。このフィールドで発行者 DN を指定する既存エージェントアプリケーションは、値を Null に変更する必要はありません。発行者 DN 値は、下位互換性の目的でサポートされています。lpszCertUserDN が Null の場合、lpszCertIssuerDN は無視されます。

**nCertBinaryLen**

バイナリ証明書内の文字数。

**lpszCertBinary**

証明書データへのポインタ。

## エージェント API の関数宣言 (C)

以下の表に、エージェント API で使用される C 言語関数の概要を示します。関数はアルファベット順に並んでいます。

関数	説明
<a href="#">Sm_AgentApi_Audit()</a> (P. 73)	トランザクションを監査します。
<a href="#">Sm_AgentApi_Authorize()</a> (P. 75)	ユーザがリソースにアクセス権があるかどうかを判別します。
<a href="#">Sm_AgentApi_ChangePassword()</a> (P. 78)	ユーザのパスワードを変更します。
<a href="#">Sm_AgentApi_CreateSSOToken()</a> (P. 80)	標準の SiteMinder Web エージェントとカスタムエージェントの間で共有可能な暗号化済みのシングルサインオン トークンを作成します。
<a href="#">Sm_AgentApi_DecodeSSOToken()</a> (P. 82)	シングルサインオン トークンをデコードします。
<a href="#">Sm_AgentApi_DelSessionVariables()</a> (P. 85)	指定されたセッション変数をセッションストアから削除します。
<a href="#">Sm_AgentApi_DoManagement()</a> (P. 87)	エージェント コマンドをリクエストします。
<a href="#">Sm_AgentApi_FreeAttributes()</a> (P. 89)	レスポンス属性のバッファを解放します。

関数	説明
<a href="#">Sm_AgentApi_FreeServers()</a> (P. 89)	<a href="#">Sm_AgentApi_GetConfig()</a> コールの後にサーバ構造体の配列を解放します。
<a href="#">Sm_AgentApi_GetAgentApiUpdateVersion()</a> (P. 90)	現在の API 更新バージョンを取得します。
<a href="#">Sm_AgentApi_GetAllowedTunnelBufferSize()</a> (P. 90)	<a href="#">Sm_AgentApi_Tunnel()</a> のコールで転送可能な最大のデータバッファサイズを取得します。
<a href="#">Sm_AgentApi_GetConfig()</a> (P. 91)	レジストリ (Microsoft Windows の場合のみ) またはエージェント設定ファイルで定義されたエージェント設定を取得します。
<a href="#">Sm_AgentApi_GetSessionVariables()</a> (P. 94)	既存のセッション変数の値を取得します。
<a href="#">Sm_AgentApi_Init()</a> (P. 97)	ポリシーサーバへの接続をセットアップするためにエージェント API を初期化します。
<a href="#">Sm_AgentApi_IsProtected()</a> (P. 99)	リソースが保護されているかどうかを判別します。
<a href="#">Sm_AgentApi_Login()</a> (P. 101)	セッション ログインおよび検証を実行します。
<a href="#">Sm_AgentApi_Logout()</a> (P. 104)	ユーザをセッションからログアウトします。
<a href="#">Sm_AgentApi_MakeCertificateHash()</a> (P. 105)	証明書のハッシュを生成します。
<a href="#">Sm_AgentApi_SetDefaultAgentId()</a> (P. 107)	中央ホスト設定によって設定されたカスタム エージェントの名前を設定します。
<a href="#">Sm_AgentApi_SetSessionVariables()</a> (P. 108)	新しいセッション変数を作成するか、または既存のセッション変数を更新します。
<a href="#">Sm_AgentApi_Tunnel()</a> (P. 111)	ポリシーサーバ側のリモートサービスとエージェントの間でデータを転送します。
<a href="#">Sm_AgentApi_Uninit()</a> (P. 113)	エージェント API を初期化解除します。
<a href="#">Sm_AgentApi_UpdateAttributes()</a> (P. 115)	レスポンス属性を更新します。

詳細情報:

[エージェントコールシーケンス](#) (P. 41)

## 関数リターンコード

エージェント API 関数は、以下のリターンコードの一部またはすべてを使用できます。

名前	値
SM_AGENTAPI_NOCONNECTION	-3
SM_AGENTAPI_TIMEOUT	-2
SM_AGENTAPI_FAILURE	-1
SM_AGENTAPI_SUCCESS	0
SM_AGENTAPI_YES	1
SM_AGENTAPI_NO	2
SM_AGENTAPI_CHALLENGE	3

## Sm\_AgentApi\_Audit()

認可をエージェント キャッシュから実行する場合にこの関数をコールします。

### 構文

```
int Sm_AgentApi_Audit (
    const void*                pHandle,
    const char*                lpszClientIpAddr,
    const char*                lpszTransactionId,
    const Sm_AgentApi_ResourceContext_t* pResourceContext,
    const Sm_AgentApi_Realm_t* pRealm,
    Sm_AgentApi_Session_t*     pSession
);
```

パラメータ	I/O	説明
<i>pHandle</i>	I	Sm_AgentApi_Init() の <i>ppHandle</i> パラメータで返されるエージェント API セッションハンドル。

パラメータ	I/O	説明
<i>lpszClientIpAddr</i>	I	ユーザがログインしているクライアントの IP アドレス。これはオプションのパラメータです。 クライアント IP がアスタリスク (*) で始まる場合、ポリシー サーバはこの IP アドレスをログ記録しますが、セッション仕様に照らした検証は行いません。
<i>lpszTransactionId</i>	I	エージェントがセキュリティ アクティビティをアプリケーション アクティビティに関連付けるために使用する ID。ポリシー サーバはこの ID をログ記録します。これはオプションのパラメータです。
<i>pResourceContext</i>	I	定義済みのリソース定義構造体。
<i>pRealm</i>	I	レルム定義構造体。
<i>pSession</i>	I	セッション定義構造体。

#### ReturnValues

- SM\_AGENTAPI\_YES。操作は成功しました。
- SM\_AGENTAPI\_NO。操作は失敗しました。
- SM\_AGENTAPI\_NOCONNECTION。初期化できませんでした。
- SM\_AGENTAPI\_FAILURE。サーバにアクセスできませんでした。
- SM\_AGENTAPI\_TIMEOUT。関数がタイムアウトしました。

## Sm\_AgentApi\_Authorize()

定義済みユーザが定義済みリソースで定義済みのアクションを実行する許可を SiteMinder によって与えられているかどうかを判別し、そのリソースに対するそのユーザに関するレスポンス属性を返します。

### 構文

```
int SM_EXTERN Sm_AgentApi_Authorize
    const void*                pHandle,
    const char*                lpszClientIpAddr,
    const char*                lpszTransactionId,
    const Sm_AgentApi_ResourceContext_t* pResourceContext,
    const Sm_AgentApi_Realm_t* pRealm,
    Sm_AgentApi_Session_t*     pSession,
    long*                      pNumAttributes,
    Sm_AgentApi_Attribute_t** ppAttributes
);
```

パラメータ	I/O	説明
<i>pHandle</i>	I	Sm_AgentApi_Init() の <i>ppHandle</i> パラメータで返されるエージェント API セッションハンドル。
<i>lpszClientIpAddr</i>	I	リソースを要求しているクライアントの IP アドレス。このパラメータは任意です。クライアント IP がアスタリスク (*) で始まる場合、ポリシーサーバはこの IP アドレスをログ記録しますが、セッション仕様に照らした検証は行いません。
<i>lpszTransactionId</i>	I	エージェントがセキュリティアクティビティをアプリケーションアクティビティに関連付けるために使用する ID。ポリシーサーバはこの ID をログ記録します。これはオプションのパラメータです。
<i>pResourceContext</i>	I	リソース定義構造体。
<i>pRealm</i>	I	レルム定義構造体。
<i>pSession</i>	I	セッション定義構造体。
<i>pNumAttributes</i>	O	返された属性の数。
<i>ppAttributes</i>	O	レスポンス属性定義構造体の配列へのポインタ。

### 戻り値

- SM\_AGENTAPI\_YES。ユーザは許可されています。
- SM\_AGENTAPI\_NO。ユーザは認可されていません。
- SM\_AGENTAPI\_NOCONNECTION。初期化できませんでした。
- SM\_AGENTAPI\_FAILURE。サーバにアクセスできませんでした。
- SM\_AGENTAPI\_TIMEOUT。関数がタイムアウトしました。

### 例

この関数の例については、サンプル `smagentexample.cpp` を参照してください。

## Sm\_AgentApi\_AuthorizeDLP()

定義済みユーザがリソースにアクセスする許可を SiteMinder DLP 統合によって与えられているかどうかを判別し、そのリソースに対するそのユーザに関するレスポンス属性を返します。

### 構文

```
int SM_EXTERN Sm_AgentApi_AuthorizeDLP (  
    const void*                pHandle,  
    const char*                lpszClientIpAddr,  
    const char*                lpszTransactionId,  
    const Sm_AgentApi_ResourceContext_t* pResourceContext,  
    const Sm_AgentApi_Realm_t*  pRealm,  
    Sm_AgentApi_Session_t*     pSession,  
    long*                      pNumAttributes,  
    Sm_AgentApi_Attribute_t**  ppAttributes  
);
```

---

パラメータ	I/O	説明
<i>pHandle</i>	I	Sm_AgentApi_Init() の <i>ppHandle</i> パラメータで返されるエージェント API セッションハンドル。

---

パラメータ	I/O	説明
<i>lpszClientIpAddr</i>	I	リソースを要求しているクライアントの IP アドレス。このパラメータは任意です。クライアント IP がアスタリスク (*) で始まる場合、ポリシー サーバはこの IP アドレスをログ記録しますが、セッション仕様に照らした検証は行いません。
<i>lpszTransactionId</i>	I	エージェントがセキュリティ アクティビティをアプリケーション アクティビティに関連付けるために使用する ID。ポリシー サーバはこの ID をログ記録します。これはオプションのパラメータです。
<i>pResourceContext</i>	I	リソース定義構造体。
<i>pRealm</i>	I	レルム定義構造体。
<i>pSession</i>	I	セッション定義構造体。
<i>pNumAttributes</i>	O	返された属性の数。
<i>ppAttributes</i>	O	レスポンス属性定義構造体の配列へのポインタ。

### 戻り値

- SM\_AGENTAPI\_YES。ユーザは許可されています。
- SM\_AGENTAPI\_NO。ユーザは認可されていません。
- SM\_AGENTAPI\_NOCONNECTION。初期化できませんでした。
- SM\_AGENTAPI\_FAILURE。サーバにアクセスできませんでした。
- SM\_AGENTAPI\_TIMEOUT。関数がタイムアウトしました。

## Sm\_AgentApi\_ChangePassword()

ユーザのパスワードを変更します。結果として得られた属性およびセッションオブジェクトからの理由コードは、正しいパスワードサービスリダイレクトの作成に使用されます。

### 構文

```
int SM_EXTERN Sm_AgentApi_ChangePassword (
    const void* pHandle,
    const char* lpszClientIpAddr,
    const char* lpszNewPassword,
    const char* pszTokenValue,
    const Sm_AgentApiResourceContext_t* pResourceContext,
    const Sm_AgentApi_Realm_t* pRealm,
    const Sm_AgentApi_UserCredentials_t* pUserCredentials,
    Sm_AgentApi_Session_t* pSession,
    long * numAttributes,
    Sm_AgentApi_Attribute_t** ppAttributes
);
```

パラメータ	I/O	説明
<i>pHandle</i>	I	Sm_AgentApi_Init() の <i>ppHandle</i> パラメータで返されるエージェント API セッションハンドル。
<i>lpszClientIpAddr</i>	I	リソースを要求しているクライアントの IP アドレス。このパラメータは任意です。クライアント IP がアスタリスク (*) で始まる場合、ポリシーサーバはこの IP アドレスをログ記録しますが、セッション仕様に照らした検証は行いません。
<i>lpszNewPassword</i>	I	ユーザが変更を予定している新しいパスワード (文字列)。
<i>pszTokenValue</i>	I	パスワードサービスリダイレクトの場合にポリシーサーバと Web エージェントの間で交換されるトークン。(Sm_AgentApi_ChangePassword() によって返された) <i>ppAttributes</i> から抽出された SMTOKEN をポリシーサーバに送信する場合にこのパラメータを使用します。 <i>pUserCredentials</i> パラメータの <i>nChallengeReason</i> 値が Sm_Api_Reason_PWSelfChange (ユーザによって開始されたパスワード変更を示します) に設定されている場合、この値は Null の可能性があります。

---

パラメータ	I/O	説明
<i>pResourceContext</i>	I	リソース定義構造体へのポインタ。
<i>pRealm</i>	I	レルム定義構造体。
<i>pUserCredentials</i>	I	ユーザ認証情報定義構造体。
<i>pSession</i>	O	ユーザセッション定義構造体。
<i>pNumAttributes</i>	O	<i>ppAttributes</i> 内の属性の数。
<i>ppAttributes</i>	O	レスポンス属性定義構造体の配列へのポインタ。

---

### 戻り値

- SM\_AGENTAPI\_YES。ユーザパスワードが変更されました。
- SM\_AGENTAPI\_NO。ユーザパスワードは変更されませんでした。
- SM\_AGENTAPI\_NOCONNECTION。初期化できませんでした。
- SM\_AGENTAPI\_TIMEOUT。関数がタイムアウトしました。
- SM\_AGENTAPI\_FAILURE。パスワードは変更されませんでした。

## Sm\_AgentApi\_CreateSSOToken()

セッションおよび他の情報の暗号化済みトークンを作成します。このトークンは、標準の SiteMinder Web エージェントとカスタムエージェントの間で共有可能です。この情報に相互アクセスすることにより、標準の SiteMinder Web エージェントによるシングルサインオン環境にカスタムエージェントが参加可能となります。

### 構文

```
int SM_EXTERN Sm_AgentApi_CreateSSOToken (
    const void*                pHandle,
    Sm_AgentApi_Session_t*     pSession,
    long                        nNumAttributes,
    Sm_AgentApi_Attribute_t*   pTokenAttributes,
    long*                       pNumSSOTokenLength,
    char*                       lpszSSOToken
);
```

パラメータ	I/O	説明
<i>pHandle</i>	I	Sm_AgentApi_Init() の <i>ppHandle</i> パラメータで返されるエージェント API セッションハンドル。
<i>pSession</i>	I	Sm_AgentApi_Login() コールによって返されるセッション情報。
<i>nNum Attributes</i>	I	トークンに含める属性の数。属性は <i>pTokenAttributes</i> パラメータで指定されます。

パラメータ	I/O	説明
<i>pTokenAttributes</i>	I	<p>トークンに含めるユーザ属性。有効な値は以下のとおりです。</p> <ul style="list-style-type: none"> <li>■ SM_AGENTAPI_ATTR_USERDN。ユーザの識別名。</li> <li>■ SM_AGENTAPI_ATTR_USERNAME。ユーザの名前。</li> <li>■ SM_AGENTAPI_ATTR_CLIENTIP。ユーザが保護されたリソースのリクエストを開始したマシンの IP アドレス。</li> <li>■ SM_AGENTAPI_ATTR_SSOZONE</li> </ul> <p>SSO ゾーン名の指定。ゾーン名を指定しない場合、デフォルトは "SM" です。この値は、<i>SmAgentApi_DecodeSSOToken</i> メソッドによって返される属性リストで読み取ります。</p> <p>他の属性はすべて無視されます。</p> <p>この関数に適用される <i>SmAgentApi_Attribute_t</i> 構造体のフィールドは以下のとおりです。</p> <ul style="list-style-type: none"> <li>■ <i>nAttributeId</i> (上記の値の 1 つ)</li> <li>■ <i>nAttributeLen</i></li> <li>■ <i>lpszAttributeValue</i></li> </ul>
<i>pNumSSOTokenLength</i>	I、O	<p>トークンを受信するために渡された <i>lpszSSOToken</i> バッファの長さ。最大サイズは、<i>SmAgentAPI.h</i> で定義される <i>SSO_TOKEN_MAX_SIZE</i> によって指定されます。Null 終端文字のためのスペースも含まれます。</p> <p>出力時、このパラメータは、Null 終端文字の文字も含めて、返されたトークンの実際の長さに設定されます。</p>
<i>lpszSSOToken</i>	O	<p>この関数から返されたトークン。このトークンを <i>SMSESSION Cookie</i> に書き込んでください。</p>

**戻り値**

- SM\_AGENTAPI\_SUCCESS。操作は成功しました。
- SM\_AGENTAPI\_FAILURE。トークンは作成されませんでした。
- SM\_AGENTAPI\_NOCONNECTION。初期化できませんでした。

### 注釈

この関数は、*pTokenAttributes* パラメータで指定されたユーザ属性情報を、*Sm\_AgentApi\_Login()* コールから返されたセッションおよび他の属性情報に関連付けます。結果として得られるトークン内の情報は、標準の SiteMinder Web エージェントとカスタム エージェントの間で共有することができ、標準エージェントとカスタム エージェント間のシングルサインオン操作が可能になります。

このコールはメモリを割り当てません。

トークン情報をデコードするには、*Sm\_AgentApi\_DecodeSSOToken()* をコールします。

## Sm\_AgentApi\_DecodeSSOToken()

シングルサインオン トークンをデコードし、その属性のサブセットを返します。オプションで、トークンの最終アクセスのタイム スタンプを更新し、新しいトークンで *SMSSESSION Cookie* を更新することができます。

### 構文

```
int SM_EXTERN Sm_AgentApi_DecodeSSOToken (  
    const void*          pHandle,  
    const char*         lpszSSOToken,  
    long*                nTokenVersion,  
    long*                pThirdPartyToken,  
    long*                pNumAttributes,  
    Sm_AgentApi_Attribute_t** ppTokenAttributes,  
    long                 nUpdateToken,  
    long*                pNumUpdatedSSOTokenLength,  
    char*                lpszUpdatedSSOToken  
);
```

---

パラメータ	I/O	説明
<i>pHandle</i>	I	<i>Sm_AgentApi_Init()</i> の <i>ppHandle</i> パラメータで返されるエージェント API セッションハンドル。

---

パラメータ	I/O	説明
<i>lpszSSOToken</i>	I	<p>デコードされるトークンが含まれる <b>Null</b> 終端文字配列。                      カスタム エージェントは、以下の場所のいずれかでトークンを検索します。</p> <ul style="list-style-type: none"> <li>■ トークンがカスタム エージェントによって作成されている場合、トークンは、<b>Sm_AgentApi_CreateSSOToken()</b> のコールから <i>lpszSSOToken</i> 出力パラメータに返されます。</li> <li>■ トークンが標準 <b>SiteMinder Web</b> エージェントによって作成されている場合、トークンは <b>SMSESSION Cookie</b> に含まれています。カスタム エージェントは、<b>Cookie</b> の内容を抽出し、このパラメータに割り当てる役割を持ちます。</li> </ul>
<i>nTokenVersion</i>	O	トークンの <b>SiteMinder</b> バージョン。
<i>pThirdParty Token</i>	O	値がゼロ以外の場合、トークンがもともとカスタム (サードパーティ) エージェントによって作成されたものであり、まだ標準 <b>SiteMinder</b> エージェントによって更新されていないことを示します。
<i>pNumAttributes</i>	O	トークンから取得した属性の数。属性は、 <i>ppTokenAttributes</i> パラメータで指定されます。
<i>ppToken Attributes</i>	O	<p>トークンから抽出された属性。有効な値は以下のとおりです。</p> <ul style="list-style-type: none"> <li>■ <b>SM_AGENTAPI_ATTR_USERDN</b></li> <li>■ <b>SM_AGENTAPI_ATTR_SESSIONSPEC</b></li> <li>■ <b>SM_AGENTAPI_ATTR_SESSIONID</b></li> <li>■ <b>SM_AGENTAPI_ATTR_USERNAME</b></li> <li>■ <b>SM_AGENTAPI_ATTR_CLIENTIP</b></li> <li>■ <b>SM_AGENTAPI_ATTR_DEVICENAME</b></li> <li>■ <b>SM_AGENTAPI_ATTR_IDLESESSIONTIMEOUT</b></li> <li>■ <b>SM_AGENTAPI_ATTR_MAXSESSIONTIMEOUT</b></li> <li>■ <b>SM_AGENTAPI_ATTR_STARTSESSIONTIME</b></li> <li>■ <b>SM_AGENTAPI_ATTR_LASTSESSIONTIME</b></li> <li>■ <b>SM_AGENTAPI_ATTR_SSOZONE</b></li> </ul>

パラメータ	I/O	説明
<i>nUpdateToken</i>	I	値がゼロ以外の場合、更新されたトークンがリクエストされていることを示します。更新されたトークンは、 <i>lpszUpdatedSSOToken</i> に書き込まれます。 属性 <code>SM_AGENTAPI_ATTR_LASTSESSIONTIME</code> を更新する場合は、 <i>nUpdateToken</i> フラグをゼロ以外に設定してください。
<i>pNumUpdatedSSOTokenLength</i>	I、O	トークンを受信するための <i>lpszUpdatedSSOToken</i> バッファの長さ。最大サイズは、 <code>SmAgentAPI.h</code> で定義される <code>SSO_TOKEN_MAX_SIZE</code> によって指定されます。Null 終端文字のためのスペースも含まれます。 出力時、このパラメータは、Null 終端文字の文字も含めて、返されたトークンの実際の長さに設定されます。
<i>lpszUpdatedSSOToken</i>	O	この関数から返された更新済みトークン。このトークンを <code>SMSESSION Cookie</code> に書き込んでください。 <i>nUpdateToken</i> がゼロ以外の値に設定されている場合にのみ、トークンが返されます。

### 戻り値

- `SM_AGENTAPI_SUCCESS`。操作は成功しました。
- `SM_AGENTAPI_FAILURE`。トークンはデコードされませんでした。
- `SM_AGENTAPI_NOCONNECTION`。初期化できませんでした。

### 注釈

この関数は、シングルサインオン トークンを入力として受け取り、トークンの属性のサブセットを返します。

トークンの最終アクセスのタイム スタンプを更新できます。更新する場合には、*nUpdateToken* パラメータにゼロ以外の値を割り当ててください。更新されたタイム スタンプが含まれるトークンが *lpszUpdatedSSOToken* で返されます。更新済みのトークンを **SMSESSION Cookie** に書き込んでください。

この関数は、属性リスト用にメモリを割り当てます。このメモリを解放するには、*Sm\_AgentApi\_FreeAttributes()* をコールします。

シングルサインオン トークンを作成するには、*Sm\_AgentApi\_CreateSSOToken()* をコールします。

## Sm\_AgentApi\_DelSessionVariables()

指定されたセッション変数をセッションストアから削除します。

### 構文

```
int Sm_AgentApi_DelSessionVariables (
    const void*                pHandle,
    const Sm_AgentApi_ResourceContext_t* pResourceContext,
    const char*                lpszSessionId,
    long                        nNumReqAttributes,
    Sm_AgentApi_Attribute_t*    pReqAttributes,
    long*                       pRespNumAttributes,
    Sm_AgentApi_Attribute_t**  ppRespAttributes
);
```

パラメータ	I/O	説明
<i>pHandle</i>	I	<i>Sm_AgentApi_Init()</i> の <i>ppHandle</i> パラメータで返されるエージェント API セッションハンドル。
<i>pResourceContext</i>	I	将来使用のために予約済みです。フィールドをすべて 0 に設定してください。

パラメータ	I/O	説明
<i>lpszSessionId</i>	I	変数が削除される予定のセッションの一意の識別子。変数は、アクティブセッションに対してのみ削除できます。 ログインに成功した後、セッション ID が <b>Sm_AgentApi_Session_t</b> 構造体の <i>lpszSessionId</i> フィールドに返されます。
<i>nNumReqAttributes</i>	I	<i>pReqAttributes</i> 内のセッション変数の配列のサイズ。
<i>pReqAttributes</i>	I	削除されるセッション変数の名前を表す属性の配列。変数名を <b>Sm_AgentApi_Attribute_t</b> 構造体の <i>lpszAttributeOid</i> フィールドに設定してください。 <b>Sm_AgentApi_Attribute_t</b> 構造体の <i>nAttributeFlags</i> フィールドを <b>SM_AGENTAPI_REQATTR_FLAGS_NONE</b> に設定してください。 この構造体の <i>nAttributeId</i> フィールドおよび <i>nAttributeTTL</i> フィールドは無視されます。
<i>pRespNumAttributes</i>	O	<i>ppRespAttributes</i> 内のレスポンスの配列のサイズ (存在する場合)。
<i>ppRespAttributes</i>	O	指定された変数がすべて削除されている場合、レスポンス属性はこのパラメータに返されません。 この関数が <b>SM_AGENTAPI_UNRESOLVED</b> を返す場合、レスポンス属性の結果セットには、削除できなかった変数が含まれます。また、それぞれのレスポンス属性で、 <b>Sm_AgentApi_Attribute_t</b> 構造体の <i>nAttributeFlags</i> フィールドが <b>SM_AGENTAPI_RESPATTR_FLAGS_UNRESOLVED</b> に設定されます。 この構造体の <i>nAttributeId</i> フィールドおよび <i>nAttributeTTL</i> フィールドは無視されます。

### 戻り値

- **SM\_AGENTAPI\_YES**。操作は成功しました。
- **SM\_AGENTAPI\_NO**。コールは拒否されました。
- **SM\_AGENTAPI\_UNRESOLVED**。いくつかの変数を削除できませんでした。  
*ppRespAttributes* パラメータの説明を参照してください。

- SM\_AGENTAPI\_NOCONNECTION。初期化できませんでした。
- SM\_AGENTAPI\_FAILURE。サーバにアクセスできませんでした。
- SM\_AGENTAPI\_TIMEOUT。関数がタイムアウトしました。

### 注釈

*lpszSessionId* によって参照されるセッションがアクティブでない場合やセッションストアへの接続が失われている場合、この関数は失敗します。

*ppRespAttributes* パラメータで返された変数に対して割り当てられたメモリを解放するには、`Sm_AgentApi_FreeAttributes()` をコールします。

詳細情報:

[Sm\\_AgentApi\\_FreeAttributes\(\)](#) (P. 89)

## Sm\_AgentApi\_DoManagement()

ポリシーサーバからのエージェント コマンドを要求します。エージェント コマンドは、エージェントによって実行される作業を示します。

### 構文

```
int SM_EXTERN Sm_AgentApi_DoManagement (
    const void*                pHandle,
    Sm_AgentApi_ManagementContext_t* pManagementContext,
    long*                       pNumAttributes,
    Sm_AgentApi_Attribute_t**   ppAttributes
);
```

パラメータ	I/O	説明
<i>pHandle</i>	I	<code>Sm_AgentApi_Init()</code> の <i>ppHandle</i> パラメータで返されるエージェント API セッションハンドル。
<i>pManagement Context</i>	I	管理定義構造体。エージェントは、この構造体を使用して管理コマンドを定義します。
<i>pNumAttributes</i>	I	<i>ppAttributes</i> 内の属性の数。

---

パラメータ	I/O	説明
<i>ppAttributes</i>	O	要求された属性定義構造体の配列へのポインタ。以下の属性の 1 つまたは複数返されます。 <ul style="list-style-type: none"><li>■ SM_AGENTAPI_AGENT_KEY_UPDATE_NEXT</li><li>■ SM_AGENTAPI_AGENT_KEY_UPDATE_LAST</li><li>■ SM_AGENTAPI_AGENT_KEY_UPDATE_CURRENT</li><li>■ SM_AGENTAPI_AGENT_KEY_UPDATE_PERSISTENT</li><li>■ SM_AGENTAPI_CACHE_FLUSH_ALL</li><li>■ SM_AGENTAPI_CACHE_FLUSH_ALL_USERS</li><li>■ SM_AGENTAPI_CACHE_FLUSH_THIS_USER</li><li>■ SM_AGENTAPI_CACHE_FLUSH_ALL_REALMS</li><li>■ SM_AGENTAPI_CACHE_FLUSH_THIS_REALM</li></ul>

---

#### 戻り値

- SM\_AGENTAPI\_YES。操作は成功しました。
- SM\_AGENTAPI\_NOCONNECTION。初期化できませんでした。
- SM\_AGENTAPI\_FAILURE。サーバにアクセスできませんでした。
- SM\_AGENTAPI\_TIMEOUT。関数がタイムアウトしました。

#### 例

サンプルアプリケーション `smagentexample.cpp` の `SmAgentExample::Sm_DoManagement()` 関数を参照してください。

## Sm\_AgentApi\_FreeAttributes()

レスポンス属性のバッファを解放します。

### 構文

```
void Sm_AgentApi_FreeAttributes (
    const long          nNumAttributes,
    const Sm_AgentApi_Attribute_t* pAttributes
);
```

パラメータ	I/O	説明
<i>nNumAttributes</i>	I	<i>pAttributes</i> 内の属性の数。
<i>pAttributes</i>	I	レスポンス属性定義構造体の配列へのポインタ。

### 例

サンプルアプリケーション `smagentexample.cpp` の `SmAgentExample::Sm_Login()` 関数を参照してください。

## Sm\_AgentApi\_FreeServers()

`Sm_AgentApi_GetConfig()` のコールの後、サーバ構造体の配列を解放します。

### 構文

```
void SM_EXTERN Sm_AgentApi_FreeServers (
    Sm_AgentApi_Server_t* pServers
);
```

パラメータ	I/O	説明
<i>pServers</i>	I	解放するサーバ構造体。

## Sm\_AgentApi\_GetAgentApiUpdateVersion()

SM\_AGENTAPI\_UPDATE\_VERSION 内の現在の API 更新バージョンを取得します。

たとえば、エージェント API の現在のバージョンが v6.0 SP3 の場合、Sm\_AgentApi\_GetAgentApiUpdateVersion() をコールすることにより、カスタムプログラムがエージェント API v6.0 SP3 に対してコンパイルされたことを検証できます。

### 構文

```
int SM_EXTERN Sm_AgentApi_GetAgentApiUpdateVersion();
```

## Sm\_AgentApi\_GetAllowedTunnelBufSize()

Sm\_AgentApi\_Tunnel() 関数のコールで転送可能な最大のデータ バッファ サイズを取得します。

### 構文

```
long SM_EXTERN Sm_AgentApi_GetAllowedTunnelBufSize (
    void* pHandle,
    int nServer
);
```

パラメータ	I/O	説明
<i>pHandle</i>	I	Sm_AgentApi_Init() の <i>ppHandle</i> パラメータで返されるエージェント API セッションハンドル。
<i>nServer</i>	I	トンネル コールの時に要求を処理するサーバ。 smAgentAPI.h で定義された以下の値のいずれかです。 /* server ports */ /* authorization server */ #define SM_AGENTAPI_AZ_SERVER 0 /* authentication server */ #define SM_AGENTAPI_AUTH_SERVER 1 /* accounting server */ #define SM_AGENTAPI_ACCT_SERVER 2

## 戻り値

バッファの最大サイズ。

## 注釈

この関数は SDK v5.5 で導入されました。そのリリース以降、最大許容バッファサイズは前のリリースよりも拡大されています。

## Sm\_AgentApi\_GetConfig()

エージェントの設定情報を取得します。

この関数では、v5.0 以上のエージェント API バージョンが必要です。

この関数は、設定ファイルまたは Microsoft Windows プラットフォームの Windows レジストリから設定情報を読み取ることができます。

## 構文

```
int SM_EXTERN Sm_AgentApi_GetConfig (
    Sm_AgentApi_Init_t*  pInit,
    const char*          lpszAgentName,
    const char*          lpszPath
);
```

パラメータ	I/O	説明
<i>pInit</i>	O	Sm_AgentApi_Init_t 構造体へのポインタ。 Sm_AgentApi_GetConfig() は、エージェント設定情報を取得し、それぞれの設定を <i>pInit</i> のフィールドにコピーします。 Sm_AgentApi_Init_t に配置されたサーバ構造体の配列を解放するには、Sm_AgentApi_FreeServers() をコールします。

パラメータ	I/O	説明
<i>lpszAgentName</i>	I	<p>ターゲットエージェントの名前です。関数 <code>Sm_AgentApi_GetConfig()</code> は、<i>lpszAgentName</i> で指定されたエージェント名の <code>agentname</code> リストを検索します。<i>lpszAgentName</i> が空の場合、<code>Sm_AgentApi_GetConfig()</code> はデフォルトのエージェント名を検索します。</p> <p>中央ホスト設定によって設定された <code>v5.x</code> または <code>v6.x</code> カスタムエージェントの場合、このパラメータは無視されます。</p>
<i>lpszPath</i>	I	<p>エージェント設定情報が含まれる設定ファイルの完全パスおよび名前。IIS 上の <code>4QMRx Web</code> エージェントの場合のみ、このパラメータを空の文字列 ("") として設定してください。</p> <p><i>lpszPath</i> が空の文字列である場合、<code>Sm_AgentApi_GetConfig()</code> は、エージェントが WIN32 プラットフォーム向けに記述されているかどうかを確認します。そうである場合、関数は、Windows レジストリで設定情報を検索します。そうでない場合、関数は、<code>SM_AGENTAPI_FAILURE</code> を返します。</p>

### 戻り値

- `SM_AGENTAPI_SUCCESS`。設定の取得は成功しました。
- `SM_AGENTAPI_FAILURE`。設定の取得は失敗しました。
- `SM_AGENTAPI_FAILURE` は、以下の条件のいずれかの場合に返されます。
  - エージェント名が *lpszAgentName* で指定されておらず、さらに `Sm_AgentApi_GetConfig()` がデフォルトエージェント名を取得できない。
  - エージェント名が *lpszAgentName* で指定されているが、`Sm_AgentApi_GetConfig()` がそのエージェント名を検索できない。
  - `Sm_AgentApi_GetConfig()` が少なくとも 1 つの SiteMinder ポリシーサーバ設定を検索できない。

- *lpszPath* でパスが指定されておらず、さらに以下のいずれかが発生する。
  - プラットフォームが WIN32 ではないため、取得するレジストリ設定が存在しない。
  - `Sm_AgentApi_GetConfig()` が必要な設定の 1 つをレジストリから取得できない。たとえば、共有秘密キーを保持するレジストリ キーを開くことができない。
  - *lpszPath* で指定されたパスは存在するが、*lpszPath* で名前を指定された設定ファイルを `Sm_AgentApi_GetConfig()` が開いて読み取ることができない。

### 注釈

この関数は、中央ホスト設定によって設定されているカスタム エージェントによってコールされる必要があります。

v5.x 以降のエージェントの接続の場合

- 関数の *lpszPath* パラメータは、トラステッドホストがポリシー サーバに登録されたときに作成された `SmHost.conf` ファイル、または `SmHost.conf` を参照する `WebAgent.conf` ファイルを参照します。
- *lpszAgentName* パラメータは無視されます。

4.x のエージェントの接続の場合

- エージェント名は *lpszAgentName* パラメータで指定されます。
- v4.x の `WebAgent.conf` ファイルは *lpszPath* で参照されます。

`Sm_AgentApi_GetConfig()` は、*lpszPath* の値を確認して、設定ファイルのパスおよび名前を検索します。

- *lpszPath* が空で、エージェントが Microsoft Windows プラットフォーム向けに記述されている場合、`Sm_AgentApi_GetConfig()` は Windows レジストリで設定情報を検索します。
- *lpszPath* が空で、エージェントが UNIX システム上で実行されている場合、`Sm_AgentApi_GetConfig()` は `SM_AGENTAPI_FAILURE` を返します。

`Sm_AgentApi_GetConfig()` は、エージェント名を検索できない場合、デフォルトのエージェント名を使用します。デフォルト エージェント名を取得できない場合、関数は `SM_AGENTAPI_FAILURE` を返します。

`Sm_AgentApi_GetConfig()` では、正しいエージェントの設定情報が検索できると、その情報を初期化構造体 (`Sm_AgentApi_Init_t`) のフィールドにコピーします。パラメータ `pInit` は、この初期化構造体を指します。たとえば、エージェント名パラメータに `Agent1` という文字列が含まれ、設定ファイルの `agentname` リストが以下のように設定されているとします。

```
agentname="Agent1,123.112.12.12"
```

この場合、`Sm_AgentApi_GetConfig()` は、初期化構造体 `pInit` の `lpszHostName` フィールドを `Agent1` に設定します。IP アドレスは無視されます。その後、`Sm_AgentApi_GetConfig()` は、初期化構造体の他のフィールドの情報を取得します。

### Sm\_AgentApi\_GetMaxTunnelBufSize()

SDK v5.5 で廃止されました。 [Sm\\_AgentApi\\_GetAllowedTunnelBufSize\(\)](#) (P. 90) に置き換えられています。

### Sm\_AgentApi\_GetSessionVariables()

既存のセッション変数の値を取得します。

#### 構文

```
int Sm_AgentApi_GetSessionVariables (  
    const void*                pHandle,  
    const Sm_AgentApi_ResourceContext_t* pResourceContext,  
    const char*                lpszSessionId,  
    long                       nNumReqAttributes,  
    Sm_AgentApi_Attribute_t*   pReqAttributes,  
    long*                      pRespNumAttributes,  
    Sm_AgentApi_Attribute_t** ppRespAttributes  
);
```

---

パラメータ	I/O	説明
<code>pHandle</code>	I	<code>Sm_AgentApi_Init()</code> の <code>ppHandle</code> パラメータで返されるエージェント API セッションハンドル。
<code>pResourceContext</code>	I	将来使用のために予約済みです。フィールドをすべて 0 に設定してください。

---

パラメータ	I/O	説明
<i>lpszSessionId</i>	I	変数が取得されるセッションの一意的識別子。変数は、アクティブセッションに対してのみ取得できます。 ログインに成功した後、セッション ID が <b>Sm_AgentApi_Session_t</b> 構造体の <i>lpszSessionId</i> フィールドに返されます。
<i>nNumReqAttributes</i>	I	<i>pReqAttributes</i> 内の属性の配列のサイズ。
<i>pReqAttributes</i>		取得されるセッション変数の名前を表す属性の配列。 変数名を <b>Sm_AgentApi_Attribute_t</b> 構造体の <i>lpszAttributeOid</i> フィールドに設定してください。 <b>Sm_AgentApi_Attribute_t</b> 構造体の <i>nAttributeFlags</i> フィールドを以下の値のいずれかに設定してください。 <ul style="list-style-type: none"> <li>■ <b>SM_AGENTAPI_REQATTR_FLAGS_NONE</b> 指定された変数を取得しますが、削除はしません。</li> <li>■ <b>SM_AGENTAPI_REQATTR_FLAGS_DELETE</b> 変数を取得した後、セッションストアから削除します。 この構造体の <i>nAttributeId</i> フィールドおよび <i>nAttributeTTL</i> フィールドは無視されます。 <b>Sm_AgentApi_Attribute_t</b> 構造体の <i>nAttributeFlags</i> フィールドを以下の値のいずれかに設定してください。</li> <li>■ <b>SM_AGENTAPI_REQATTR_FLAGS_NONE</b> 指定された変数を取得しますが、削除はしません。</li> <li>■ <b>SM_AGENTAPI_REQATTR_FLAGS_DELETE</b> 変数を取得した後、セッションストアから削除します。 この構造体の <i>nAttributeId</i> フィールドおよび <i>nAttributeTTL</i> フィールドは無視されます。</li> </ul>
<i>pRespNumAttributes</i>	O	<i>ppRespAttributes</i> 内のレスポンスの配列のサイズ。

パラメータ	I/O	説明
<i>ppRespAttributes</i>	O	<p>セッション変数およびそれらの値を表すレスポンス属性の配列。</p> <p>この関数から返された値は、レスポンス属性の結果セットの内容を以下のように示します。</p> <ul style="list-style-type: none"> <li>■ SM_AGENTAPI_UNRESOLVED                     <p>いくつかの変数をフェッチできませんでした。特定の変数をフェッチできない場合、<code>Sm_AgentApi_Attribute_t</code> 構造体の関連する <code>nAttributeFlags</code> フィールドが <code>SM_AGENTAPI_RESPATTR_FLAGS_UNRESOLVED</code> に設定されます。</p> <p>取得される変数に対応する <code>nAttributeFlags</code> フィールドは <code>SM_AGENTAPI_RESPATTR_FLAGS_NONE</code> に設定されます。</p> </li> <li>■ SM_AGENTAPI_YES                     <p>すべての要求された変数それらの値がセッションストアから取得され、<i>ppRespAttributes</i> で返されました。</p> <p>この構造体の <code>nAttributeId</code> フィールドおよび <code>nAttributeTTL</code> フィールドは無視されます。</p> </li> </ul>

### 戻り値

- SM\_AGENTAPI\_YES。操作は成功しました。
- SM\_AGENTAPI\_NO。コールは拒否されました。
- SM\_AGENTAPI\_UNRESOLVED。いくつかの変数をフェッチできませんでした。*ppRespAttributes* パラメータの説明を参照してください。
- SM\_AGENTAPI\_NOCONNECTION。初期化できませんでした。
- SM\_AGENTAPI\_FAILURE。サーバにアクセスできませんでした。
- SM\_AGENTAPI\_TIMEOUT。関数がタイムアウトしました。
- SM\_AGENTAPI\_SUCCESS。操作は成功しました。

## 注釈

*lpszSessionId* によって参照されるセッションがアクティブでない場合、変数が検索されない場合、またはセッションストアへの接続が失われている場合、この関数は失敗します。

変数をフェッチした後にセッションストアから削除するには、*pReqAttributes* パラメータの `SM_AGENTAPI_REQATTR_FLAGS_DELETE` フラグを設定します。変数をフェッチせずにセッションストアから削除するには、`Sm_AgentApi_DelSessionVariables()` 関数をコールします。

*ppRespAttributes* パラメータで返された変数に対して割り当てられたメモリを解放するには、[Sm\\_AgentApi\\_FreeAttributes\(\)](#) (P. 89) をコールします。

## Sm\_AgentApi\_Init()

エージェント API を初期化し、ポリシー サーバへの接続をセットアップします。この関数は、エージェントごとに 1 回コールされます。

注: ポリシー サーバへの接続が即座に確立できなくても、このコールは成功します。エージェント API は、引き続き再接続を試行します。詳細については、「注釈」を参照してください。

## 構文

```
int SM_EXTERN Sm_AgentApi_Init (
    const Sm_AgentApi_Init_t* pInitStruct,
    void** ppHandle
);
```

パラメータ	I/O	説明
<i>pInitStruct</i>	I	サーバに関する情報へのポインタ。
<i>ppHandle</i>	O	この API セッションに対して返されたハンドルを保持するポインタのアドレス。これは、 <code>opaque</code> 型です。

## 戻り値

- `SM_AGENTAPI_SUCCESS`。初期化は成功しました。
- `SM_AGENTAPI_FAILURE`。初期化は失敗しました。

### 注釈

すべてのエージェントは、スタートアップ時に 1 度 `DoManagement()` コールを発行し、`SM_AGENTAPI_MANAGEMENT_SET_AGENT_INFO` コマンドを指定する必要があります。

この関数は、ポリシー サーバへの接続が確立されたが、共有秘密キーやエージェント名が正しくないときに限りに失敗するように設計されています。他のすべての状況において、この関数は `SM_AGENTAPI_SUCCESS` を返します。たとえば、以下のような状況も含まれます。

- 初期化動作中に、間違ったポリシー サーバ IP アドレスやポート番号が提供されている。
- 正しいポリシー サーバ IP アドレスおよびポート番号が提供されているが、ポリシー サーバがダウンしている。

これらの場合、エージェント API は、成功のステータスを返し、ポリシー サーバへの接続を確立するための試行を続行します（接続レイヤでは、提供された情報が正しいか正しくないかは認識されません）。

`Sm_AgentApi_Init()` 関数が成功しても、ポリシー サーバへの接続が確立されたと断定できません。

カスタム エージェントのメモリは手動で解放する必要があります。

`Sm_AgentApi_Init()` を使用してエージェント API を初期化すると、`Sm_AgentApi_Init_t` 構造体内のすべての情報がコピーされ、初期化後に構造体のメモリの解放が可能になります。

### 例

サンプルアプリケーション `smagentexample.cpp` の `SmAgentExample::Sm_Init()` 関数を参照してください。

## Sm\_AgentApi\_IsProtected()

定義済みリソースが SiteMinder によって保護されているかどうかを確認します。

### 構文

```
int SM_EXTERN Sm_AgentApi_IsProtected (
    const void                pHandle,
    const char*               lpszClientIpAddr,
    const Sm_AgentApi_ResourceContext_t* pResourceContext,
    Sm_AgentApi_Realm_t*      pRealm
);
```

パラメータ	I/O	説明
<i>pHandle</i>	I	Sm_AgentApi_Init() の <i>ppHandle</i> パラメータで返されるエージェント API セッションハンドル。
<i>lpszClientIpAddr</i>	I	リソースを要求しているクライアントの IP アドレス。これはオプションのパラメータです。
<i>pResourceContext</i>	I	リソース定義構造体。
<i>pRealm</i>	O	レルム定義構造体。返されたレルムによってリソースは保護されています。

### 戻り値

- SM\_AGENTAPI\_YES。リソースは保護されています。
- SM\_AGENTAPI\_NO。リソースは保護されていません。
- SM\_AGENTAPI\_NOCONNECTION。初期化できませんでした。
- SM\_AGENTAPI\_FAILURE。サーバにアクセスできませんでした。
- SM\_AGENTAPI\_TIMEOUT。関数がタイムアウトしました。

### 例

サンプルアプリケーション smagentexample.cpp の SmAgentExample::Sm\_IsProtected() 関数を参照してください。

## Sm\_AgentApi\_IsProtectedDLP()

定義済みリソースが SiteMinder DLP 統合によって保護されているかどうかを確認します。

### 構文

```
int SM_EXTERN Sm_AgentApi_IsProtectedDLP (
    const void                pHandle,
    const char*               lpszClientIpAddr,
    const Sm_AgentApi_ResourceContext_t* pResourceContext,
    Sm_AgentApi_Realm_t*      pRealm
);
```

パラメータ	I/O	説明
<i>pHandle</i>	I	Sm_AgentApi_Init() の <i>ppHandle</i> パラメータで返されるエージェント API セッションハンドル。
<i>lpszClientIpAddr</i>	I	リソースを要求しているクライアントの IP アドレス。これはオプションのパラメータです。
<i>pResourceContext</i>	I	リソース定義構造体。
<i>pRealm</i>	O	レルム定義構造体。返されたレルムによってリソースは保護されています。

### 戻り値

- SM\_AGENTAPI\_YES。リソースは DLP 統合によって保護されています。
- SM\_AGENTAPI\_NO。リソースは DLP 統合によって保護されていません。
- SM\_AGENTAPI\_NOCONNECTION。初期化できませんでした。
- SM\_AGENTAPI\_FAILURE。サーバにアクセスできませんでした。
- SM\_AGENTAPI\_TIMEOUT。関数がタイムアウトしました。

## Sm\_AgentApi\_Login()

この関数は、セッション ログインおよびセッション 検証を実行します。

ポリシー サーバは、セッション ログイン中にユーザ認証情報を認証し、セッション 検証中にセッション 仕様を検証します。ポリシー サーバがセッション ログインを実行するかセッション 検証を実行するかは、以下のように、セッション 仕様 `Sm_AgentApi_Session_t` 構造体の `IpszSessionSpec` フィールドに定義されているかどうかによります。

- セッション 仕様が存在する場合、ポリシー サーバはセッション 検証を実行します。セッション 検証中、`IpszSessionSpec` フィールドに特定の長さがある場合、ポリシー サーバは以下のアクションを実行します。
  - `Sm_AgentApi_Session_t` の `nMaxTimeout` フィールドに基づき、セッションが期限切れになっていないことを確認します。
  - IP アドレスを確認します。
  - ユーザがユーザ ディレクトリ存在して、無効化されていないことを確認します。
- セッション 仕様が存在しない場合、ポリシー サーバはセッション ログインを実行します。セッション ログイン中、セッション ID が指定されている場合には、認証成功時にこれがセッション ID として使用されます。

### 構文

```
int SM_EXTERN Sm_AgentApi_Login (
    const void*                pHandle,
    const char*                IpszClientIpAddr,
    const Sm_AgentApi_ResourceContext_t* pResourceContext,
    const Sm_AgentApi_Realm_t*   pRealm,
    const Sm_AgentApi_UserCredentials_t* pUserCredentials,
    Sm_AgentApi_Session_t*       pSession,
    long*                       pNumAttributes,
    Sm_AgentApi_Attribute_t**    ppAttributes
);
```

パラメータ	I/O	説明
<code>pHandle</code>	I	<code>Sm_AgentApi_Init()</code> の <code>ppHandle</code> パラメータで返されるエージェント API セッション ハンドル。

パラメータ	I/O	説明
<i>lpszClientIpAddr</i>	I	ユーザのログイン元のクライアントの IP アドレス。これはオプションのパラメータです。クライアント IP がアスタリスク (*) で始まる場合、ポリシー サーバはこの IP アドレスをログ記録しますが、セッション仕様に照らした検証は行いません。
<i>pResourceContext</i>	I	リソース定義構造体へのポインタ。
<i>pRealm</i>	I	レルム定義構造体。
<i>pUserCredentials</i>	I	ユーザ認証情報定義構造体。
<i>pSession</i>	O	ユーザセッション定義構造体。
<i>pNumAttributes</i>	O	<i>ppAttributes</i> 内の属性の数。
<i>ppAttributes</i>	O	レスポンス属性定義構造体の配列へのポインタ。 この関数は、以下の属性を返します (利用可能な場合)。 <ul style="list-style-type: none"> <li>■ SM_AGENTAPI_ATTR_AUTH_DIR_OID</li> <li>■ SM_AGENTAPI_ATTR_AUTH_DIR_NAME</li> <li>■ SM_AGENTAPI_ATTR_AUTH_DIR_SERVER</li> <li>■ SM_AGENTAPI_ATTR_AUTH_DIR_NAMESPACE</li> <li>■ SM_AGENTAPI_ATTR_USERMSG</li> <li>■ SM_AGENTAPI_ATTR_USERDN</li> <li>■ SM_AGENTAPI_ATTR_USERUNIVERSALID</li> <li>■ SM_AGENTAPI_ATTR_IDENTITYSPEC</li> </ul> 匿名認証方式によってリソースが保護される場合に設定される属性の詳細については、「注釈」を参照してください。

### 戻り値

- SM\_AGENTAPI\_YES。ユーザが認証されました。
- SM\_AGENTAPI\_NO。ユーザは認証されませんでした。
- SM\_AGENTAPI\_CHALLENGE。認証にはチャレンジが必要です。
- SM\_AGENTAPI\_NOCONNECTION。初期化できませんでした。
- SM\_AGENTAPI\_FAILURE。サーバにアクセスできませんでした。
- SM\_AGENTAPI\_TIMEOUT。関数がタイムアウトしました。

## 注釈

認証イベントが発生すると、レスポンス属性が返されることがあります。「レスポンス属性」に述べられているように、既知の属性とポリシーベースの属性の両方が返される可能性があります。たとえば、認証の成功時、レスポンスはユーザの DN を返すことがあります。

リソースが匿名認証方式によって保護されている場合、以下の属性のみが設定されます。

- `SM_AGENTAPI_ATTR_USERDN`。匿名セッションの `SessionID` を設定します。
- `SM_AGENTAPI_ATTR_IDENTITYSPEC`。匿名セッションのグローバルに一意の識別チケットを設定します。

必須の認証情報のみを提供してください（必要な認証情報は `Sm_AgentApi_IsProtected()` のコールによって決定されます。この関数は `Sm_AgentApi_Login()` の前にコールされる必要があります）。ユーザ認証情報構造体の未使用のフィールドは、ゼロ初期化される必要があります。

`Sm_AgentApi_Login()` は、`Sm_AgentApi_Attribute_t` 構造体内の属性を返します。属性を解放するには、`Sm_AgentApi_FreeAttributes()` をコールします。

ログインの成功時、`Sm_AgentApi_Session_t` 構造体には、セッション仕様が入力されます。この構造体に対してメモリを割り当てている場合には、メモリを手動で解放する必要があります。

## 例

この関数の例については、サンプルアプリケーション `smagentexample.cpp` を参照してください。

## Sm\_AgentApi\_Logout()

ユーザセッションからユーザをログアウトし、イベントを発行します。データベースは更新されません。

ユーザがログアウトしたら、セッション仕様を破棄することにより明示的にセッションを終了する必要があります。

この関数ではメモリは解放されません。カスタムエージェントに割り当てたすべてのメモリを手動で解放する必要があります。

### 構文

```
int SM_EXTERN Sm_AgentApi_Logout (
    const void*          pHandle,
    const char*          lpszClientIpAddr,
    const Sm_AgentApi_Session_t* pSession
);
```

パラメータ	I/O	説明
<i>pHandle</i>	I	Sm_AgentApi_Init() の <i>pHandle</i> パラメータで返されるエージェント API セッションハンドル。
<i>lpszClientIpAddr</i>	I	ユーザがログアウトするクライアントの IP アドレス。これはオプションのパラメータです。クライアント IP がアスタリスク (*) で始まる場合、ポリシーサーバはこの IP アドレスをログ記録しますが、セッション仕様に照らした検証は行いません。
<i>pSession</i>	I	ユーザのセッションのセッション定義構造体 (Sm_AgentApi_Session_t)。nreason フィールドは、ポリシーサーバによって発行されたイベントに渡されます。Sm_Api_Reason_t を参照してください。

### 戻り値

- SM\_AGENTAPI\_YES。ユーザは正常にログアウトしました。
- SM\_AGENTAPI\_NO。ユーザはログアウトされませんでした。
- SM\_AGENTAPI\_NOCONNECTION。初期化できませんでした。

- SM\_AGENTAPI\_FAILURE。サーバにアクセスできませんでした。
- SM\_AGENTAPI\_TIMEOUT。関数がタイムアウトしました。

### 注釈

ユーザがログアウトしたときにユーザのセッションを終了するには、セッション仕様を破棄する必要があります。その操作は、**Sm\_AgentApi\_Session\_t** 構造体内のセッション仕様を再初期化することにより行えます。以下のコードを参照してください。

```
iResult = Sm_AgentApi_Logout (
    pHandle,
    SMAPI_SAMPLE_AGENTIP,
    &pSession);

if (SM_AGENTAPI_YES==iResult)
    memset(&pSession,0,sizeof(Sm_AgentApi_Session_t));
```

## Sm\_AgentApi\_MakeCertificateHash()

この関数は、バイナリ証明書の SHA1 ハッシュを生成する場合に使用します。ハッシュは、ユーザ認証情報構造体内のバイナリ証明書に配置される必要があります。この関数が正しく操作するためには、*iCertHashLen* パラメータが 20 バイト以上であることが必要です。この関数は、任意のバッファの SHA1 の生成に使用できます。

### 構文

```
int Sm_AgentApi_MakeCertificateHash (
    const unsigned char* pCertificateData,
    const int nCertLen,
    unsigned char* pCertHash,
    const int nCertHashLen
);
```

パラメータ	I/O	説明
<i>pCertificateData</i>	I	バッファへのポインタ。
<i>nCertLen</i>	I	バッファの長さ (バイト)。
<i>pCertHash</i>	O	ハッシュがされるバッファ。

パラメータ	I/O	説明
<i>nCertHashLen</i>	I	ハッシュファイルのサイズ (バイト)。サイズは、20 以上である必要があります。

### 戻り値

- SM\_AGENTAPI\_SUCCESS。操作は成功しました。
- SM\_AGENTAPI\_FAILURE。サイズが小さすぎます。

## Sm\_AgentApi\_SetAgentInstanceInfo()

エージェント ディスカバリ データ構造体で指定されたエージェントインスタンス情報を設定します。Sm\_AgentApi\_AgentDiscovery\_t 内の値はすべて「不明」に初期化されます。エージェントをエージェント ディスカバリの範囲に含める場合にこれらの属性の値を設定するには、Sm\_AgentApi\_Init() の後にこの関数をコールします。

### 構文

```
int SM_EXTERN Sm_AgentApi_SetAgentInstnaceInfo (
    const void*                                pHandle,
    Sm_AgentApi_AgentDiscovery_t*             arrParams[]
    int                                         nCount
);
```

パラメータ	I/O	説明
<i>pHandle</i>	I	Sm_AgentApi_Init() の <i>ppHandle</i> パラメータで返されるエージェント API セッションハンドル。
<i>arrParams[]</i>	I	エージェントの属性の名前/値ペアが含まれる配列。
<i>nCount</i>	I	属性の配列内のエレメント数。

### 戻り値

- SM\_AGENTAPI\_YES。リソースは保護されています。
- SM\_AGENTAPI\_NO。リソースは保護されていません。
- SM\_AGENTAPI\_NOCONNECTION。初期化できませんでした。

- SM\_AGENTAPI\_FAILURE。サーバにアクセスできませんでした。
- SM\_AGENTAPI\_TIMEOUT。関数がタイムアウトしました。

## Sm\_AgentApi\_SetDefaultAgentId()

中央ホスト設定によって設定される v5.x 以降のエージェントの名前を設定します。

### 構文

```
int SM_EXTERN Sm_AgentApi_SetDefaultAgentId(
    const char *pszAgentIdentity,
    void* pHandle
);
```

パラメータ	I/O	説明
pszAgentIdentity	I	デフォルト エージェント名として設定する名前を指定します。この名前は、ポリシー サーバ 上の対応する エージェント オブジェクトの名前に一致する必要があります。
pHandle	I	Sm_AgentApi_Init() の ppHandle パラメータで返される エージェント API セッション ハンドル。

### 戻り値

- SM\_AGENTAPI\_SUCCESS。デフォルト エージェント名は正常に設定されました。
- SM\_AGENTAPI\_FAILURE。パラメータ pszAgentIdentity が Null であったか、または SM\_AGENTAPI\_SIZE\_NAME を超えていました。
- SM\_AGENTAPI\_NOCONNECTION。パラメータ pHandle が Null でした。

### 注釈

この関数は、Sm\_AgentApi\_Init() をコールした後、エージェント API 内の他の関数をコールする前にコールします。そうすることで、エージェント API 要求ごとにエージェントの名前を渡す必要がなくなります。

この関数は、中央ホスト設定によって設定される v5 以降のカスタム エージェントのみで使用されます。

## Sm\_AgentApi\_SetSessionVariables()

新しいセッション変数を作成するか、または既存のセッション変数を更新します。

### 構文

```
int Sm_AgentApi_SetSessionVariables (
    const void*                pHandle,
    const Sm_AgentApi_ResourceContext_t* pResourceContext,
    const char*                lpszSessionId,
    long                        nNumReqAttributes,
    Sm_AgentApi_Attribute_t*    pReqAttributes,
    long*                       pRespNumAttributes,
    Sm_AgentApi_Attribute_t**   ppRespAttributes
);
```

パラメータ	I/O	説明
<i>pHandle</i>	I	Sm_AgentApi_Init() の <i>ppHandle</i> パラメータで返されるエージェント API セッションハンドル。
<i>pResourceContext</i>	I	将来使用のために予約済みです。フィールドをすべて 0 に設定してください。
<i>lpszSessionId</i>	I	変数が設定されるセッションの一意の識別子。変数は、アクティブセッションに対してのみ設定できます。 ログインに成功した後、セッション ID が Sm_AgentApi_Session_t 構造体の <i>lpszSessionId</i> フィールドに返されます。
<i>nNumReqAttributes</i>	I	<i>pReqAttributes</i> 内のセッション変数の配列のサイズ。

パラメータ	I/O	説明
<i>pReqAttributes</i>	I	<p>セッション変数の値を表す属性の配列。 変数名を構造体 <code>Sm_AgentApi_Attribute_t</code> の <code>lpszAttributeOid</code> フィールドに設定してください。</p> <ul style="list-style-type: none"> <li>■ 変数名がすでに存在する場合、関連する変数値が <i>pReqAttributes</i> の値によって上書きされます。</li> <li>■ 変数名が存在しない場合、新しい変数が作成されます。</li> </ul> <p><code>Sm_AgentApi_Attribute_t</code> 構造体の <code>nAttributeFlags</code> フィールドを <code>SM_AGENTAPI_REQATTR_FLAGS_NONE</code> に設定してください。 この構造体の <code>nAttributeId</code> フィールドおよび <code>nAttributeTTL</code> フィールドは無視されます。</p>
<i>pRespNumAttributes</i>	O	<p><i>ppRespAttributes</i> 内のレスポンスの配列のサイズ。</p>
<i>ppRespAttributes</i>	O	<p>設定できなかった変数を表すレスポンス属性の配列。 この関数が <code>SM_AGENTAPI_UNRESOLVED</code> を返す場合、レスポンス属性の結果セットには未解決の変数が含まれます。また、返された未解決の変数ごとに、<code>Sm_AgentApi_Attribute_t</code> 構造体の <code>nAttributeFlags</code> フィールドが <code>SM_AGENTAPI_RESPATTR_UNRESOLVED</code> に設定されます。 この構造体の <code>nAttributeId</code> フィールドおよび <code>nAttributeTTL</code> フィールドは無視されます。</p>

### 戻り値

- SM\_AGENTAPI\_YES。操作は成功しました。
- SM\_AGENTAPI\_NO。操作は失敗しました。
- SM\_AGENTAPI\_UNRESOLVED。いくつかの変数を設定できませんでした。未解決の変数のリストが *ppRespAttributes* パラメータで返されます。
- SM\_AGENTAPI\_NOCONNECTION。初期化できませんでした。
- SM\_AGENTAPI\_FAILURE。サーバにアクセスできませんでした。
- SM\_AGENTAPI\_TIMEOUT。関数がタイムアウトしました。
- SM\_AGENTAPI\_SUCCESS。操作は成功しました。

### 注釈

*lpszSessionId* によって参照されるセッションがアクティブでない場合やセッションストアへの接続が失われている場合、この関数は失敗します。

*ppRespAttributes* パラメータで返された未解決の変数値に対して割り当てられたメモリを解放するには、[Sm AgentApi FreeAttributes\(\)](#) (P. 89) をコールします。

## Sm\_AgentApi\_Tunnel()

この関数は、ポリシーサーバ側のリモートサービスとエージェント間でデータを転送する場合にコールします。

`Sm_AgentApi_GetMaxTunnelBufSize()` 関数のコールでは、転送可能な最大のデータサイズが提供されます。現時点では、この関数は、各コールにつき 1 つのバッファのみをサポートします。`pServiceRequest` は、ポリシーサーバによって呼び出されるリモートサービスに関する情報を保持します。

**注:** `SMTUNNEL` は、事前定義済みトンネルエージェント名であり、その共有秘密キーも `SMTUNNEL` です。事前にエージェントを作成しなくても、これらの名前を使用して、トンネルエージェントを初期化できます。事前定義済みの `SMTUNNEL` エージェントは、`Sm_AgentApi_Tunnel()` のみをコールできます。

名前と共有秘密キーが `SMTUNNEL` のトンネルエージェントを明示的に作成する場合も、このトンネルエージェントは `Sm_AgentApi_Tunnel()` のコールに制限されます。

`SMTUNNEL` という名前のエージェントが `Sm_AgentApi_Tunnel()` 以外の関数をコールした場合、ポリシーサーバはエラーを返します。

### 構文

```
int SM_EXTERN Sm_AgentApi_Tunnel (
    const void*                pHandle,
    const int                  nServer,
    const char*                 lpszClientIpAddr,
    const char*                 lpszTransactionId,
    const Sm_AgentApi_ResourceContext_t* pResourceContext,
    const Sm_AgentApi_TunnelServiceRequest_t* pServiceRequest,
    long*                       pRespNumAttributes,
    Sm_AgentApi_Attribute_t**   ppRespAttributes
);
```

パラメータ	I/O	説明
<code>pHandle</code>	I	<code>Sm_AgentApi_Init()</code> の <code>ppHandle</code> パラメータで返されるエージェント API セッションハンドル。

パラメータ	I/O	説明
<i>nServer</i>	I	トンネル コールの時に要求を処理するサーバ。 smAgentAPI.h で定義された以下の値のいずれかです。 /* server ports */ /* authorization server */ #define SM_AGENTAPI_AZ_SERVER 0 /* authentication server */ #define SM_AGENTAPI_AUTH_SERVER 1 /* accounting server */ #define SM_AGENTAPI_ACCT_SERVER 2
<i>lpzClientIpAddr</i>	I	(オプション) ユーザのログイン元のクライアントの IP アドレス。
<i>lpzTransactionId</i>	I	(オプション) エージェントがセキュリティ アクティビティをアプリケーション アクティビティに関連付けるために使用する ID です。ポリシー サーバはこの ID をログ記録します。
<i>pResourceContext</i>	I	リソース定義構造体。
<i>pServiceRequest</i>	I	サービス要求定義構造体。
<i>pRespNumAttributes</i>	O	<i>ppRespAttributes</i> 内の属性の数。
<i>ppRespAttributes</i>	O	レスポンス属性定義構造体の配列へのポインタ。属性識別子 SM_AGENTAPI_ATTR_SERVICE_DATA は、レスポンス構造体にリモート サービスによって返されたデータが含まれることを示します。属性識別子 SM_AGENTAPI_ATTR_STATUS_MESSAGE には、リモート サービスからのステータス メッセージが含まれています。

### 戻り値

- SM\_AGENTAPI\_YES。操作は成功しました。
- SM\_AGENTAPI\_NO。操作は失敗しました。
- SM\_AGENTAPI\_NOCONNECTION。初期化できませんでした。
- SM\_AGENTAPI\_FAILURE。サーバにアクセスできませんでした。
- SM\_AGENTAPI\_TIMEOUT。関数がタイムアウトしました。

### 例

Sm\_AgentApi\_Tunnel() の例を次に示します。

```
<install_path>%sdk%samples%smtunnelagent%smtunnelexample.cpp
```

## Sm\_AgentApi\_UnInit()

エージェントが不要になった場合、API インスタンスごとに Sm\_AgentApi\_UnInit() を発行することにより、すべての API インスタンスを未初期化してください。これにより、すべてのポリシーサーバへの TCP 接続が閉じられます。

この関数ではメモリは解放されません。カスタムエージェントに割り当てたすべてのメモリを手動で解放する必要があります。

### 構文

```
int SM_EXTERN Sm_AgentApi_UnInit (
    void** ppHandle
);
```

パラメータ	I/O	説明
<i>ppHandle</i>	I	この API セッションのハンドルへのポインタのアドレス。これは、opaque 型です。

### 戻り値

- SM\_AGENTAPI\_SUCCESS。未初期化は成功しました。
- SM\_AGENTAPI\_NOCONNECTION。未初期化されませんでした。

### 例

サンプルアプリケーション `smagentexample.cpp` の `SmAgentExample::Sm_UnInit()` 関数を参照してください。

## Sm\_AgentApi\_UpdateAttributes()

この関数は、存続期間 (TTL) 値が期限切れになり、レスポンス属性を更新する場合にコールします。要求属性を指定する際、それぞれの属性構造体がゼロに初期化される必要があります。その後、**Sm\_AgentApi\_Attribute\_t** 構造体の *lpszAttributeOid* フィールドを設定してください。レスポンス属性を要求された属性にマップするためのフィールドと同じフィールドを使用します。すべての属性が更新されるとは限りません。

この関数は、当初データを保持していなかったアクティブレスポンスを再計算します。その戻り値がゼロ以外の長さであるか、その TTL 値が設定されている場合、アクティブレスポンス属性が有効であると見なされます。

### 構文

```
int SM_EXTERN Sm_AgentApi_UpdateAttributes (
    const void*                pHandle,
    const char*                lpszClientIpAddr,
    const char*                lpszTransactionId,
    const Sm_AgentApi_ResourceContext_t* pResourceContext,
    const Sm_AgentApi_Realm_t* pRealm,
    const Sm_AgentApi_Session_t* pSession,
    long                       nNumReqAttributes,
    Sm_AgentApi_Attribute_t*   pReqAttributes,
    long*                      pRespNumAttributes,
    Sm_AgentApi_Attribute_t** ppRespAttributes
);
```

パラメータ	I/O	説明
<i>pHandle</i>	I	<b>Sm_AgentApi_Init()</b> の <i>ppHandle</i> パラメータで返されるエージェント API セッションハンドル。
<i>lpszClientIpAddr</i>	I	ユーザのログイン元のクライアントの IP アドレス。これはオプションのパラメータです。クライアント IP がアスタリスク (*) で始まる場合、ポリシーサーバはこの IP アドレスをログ記録しますが、セッション仕様に照らした検証は行いません。

パラメータ	I/O	説明
<i>lpszTransactionId</i>	I	エージェントがセキュリティ アクティビティをアプリケーション アクティビティに関連付けるために使用する ID。ポリシー サーバはこの ID をログ記録します。これはオプションのパラメータです。
<i>pResourceContext</i>	I	リソース定義構造体。
<i>pRealm</i>	O	レルム定義構造体。
<i>pSession</i>	I	セッション定義構造体。
<i>nNumReqAttributes</i>	I	<i>pReqAttributes</i> 内の属性の数。
<i>pReqAttributes</i>	I	要求された属性定義構造体の配列。
<i>pRespNumAttributes</i>	O	<i>ppRespAttributes</i> 内の属性の数。
<i>ppRespAttributes</i>	O	レスポンス属性定義構造体の配列。

#### 戻り値

- SM\_AGENTAPI\_YES。操作は成功しました。
- SM\_AGENTAPI\_NO。操作は失敗しました。
- SM\_AGENTAPI\_NOCONNECTION。初期化できませんでした。
- SM\_AGENTAPI\_FAILURE。サーバにアクセスできませんでした。
- SM\_AGENTAPI\_TIMEOUT。関数がタイムアウトしました。

# 第 3 章: カスタム エージェント タイプの設定

---

このセクションには、以下のトピックが含まれています。

[カスタム エージェント タイプの概要 \(P. 117\)](#)

[エージェント タイプの設定 \(P. 119\)](#)

[エージェント タイプの変更 \(P. 123\)](#)

## カスタム エージェント タイプの概要

エージェント API を使用することにより、固有の環境の特定のニーズに合わせたカスタム エージェントを開発できます。 エージェントを開発した後、そのエージェントの新規エージェントタイプを **Administrative UI** で設定する必要があります。

エージェントタイプは、エージェントの動作を定義します。たとえば、カスタム FTP エージェントを開発した場合、その FTP エージェントのエージェントタイプを **Administrative UI** で定義する必要があります。

## エージェントタイプ ワークシート

カスタム エージェント タイプを定義するプロセスを開始する前に、そのエージェントがサポートする環境について十分に理解する必要があります。 エージェントタイプを設定するための手順を進める前に、以下の質問に答えてください。

- このエージェントによって保護されるプロトコルは何か -- たとえば、HTTP?
- これらのリソース上で保護するアクションまたはコマンドは何か? プロトコルに基づいて、許可するアクションを一覧にしてください。たとえば、FTP サイトの場合、GET と PUT を保護するなど。

- エージェントが受け取る必要のあるレスポンスの内容は何か? レスポンスに渡すデータ型は何か -- 文字列、数値、IP アドレス?
  - レスポンスして、名前の値のペアを使用できます。以下に例を示します。  
`time-restriction=yes`
  - ユーザに表示する文字列 (名前入りの挨拶文) も使用できます -- たとえば、`Hello John Smith, where name=John Smith.`

これらの質問に答えた後に、カスタム エージェント タイプの設定を開始することが可能になります。

## エージェントタイプの設定

### エージェントタイプを設定する方法

1. Administrative UI にログインします。
2. [インフラストラクチャ]-[エージェントタイプ]-[エージェントタイプの作成] をクリックします。  
  
SiteMinder の [エージェントタイプの作成] ダイアログボックスが表示されます。
3. [名前] フィールドに新しいエージェントタイプの名前を入力します。
4. 簡単な説明を [説明] フィールドに入力します。
5. 新しいエージェントタイプが RADIUS エージェントである場合は、  
[RADIUS デバイス] チェックボックスをオンにします。  
  
[RADIUS デバイス] チェックボックスをオンにすると、[エージェントタイプの定義] タブが以下のように変化します。
  - [ベンダー指定オフセット] -- RADIUS ベンダーによって提供されるオフセット値を入力します。これは必須フィールドではありません。
  - [IETF ベンダー ID] -- IANA (Internet Assigned Numbers Authority) によってベンダーに割り当てられた整数値を入力します。これは必須フィールドではありません。2552 の値を使用しないでください。この値は、SiteMinder エージェントに予約済みです。
6. 新しいエージェントタイプが RADIUS エージェントではない場合、  
[アクション] ボックスの [作成] をクリックして、このエージェントによって保護されたリソース上で許可するアクションを指定します。
7. 新しいエージェントタイプが認識するアクション (認証など) を説明するテキスト文字列を入力した後、[OK] をクリックします。  
  
[エージェントアクション] ダイアログボックスが閉じ、新しいアクションが [エージェントタイプ] ダイアログボックスのアクションのリストに追加されます。
8. このエージェントタイプの属性を設定するには、[エージェントタイプの作成] をクリックします。
9. [サブミット] をクリックします。

注: [リソースタイプ] は URL です。このタイプのリソースは文字列と同様の方法で処理されます。すなわち、ワイルドカードおよび正規表現に基づいてリソースの一致を実行できます。

### エージェントタイプ属性

エージェントタイプの設定時、以下のエージェントタイプ属性を入力できます。

- プロパティ
- 値

## プロパティ

[プロパティ] グループ ボックス内のフィールドを使用して、SiteMinder レスポンスの作成時に使用可能なレスポンス属性の操作を定義することができます。

- 名前 -- エージェント タイプ プロパティの名前。
- 説明 -- エージェント タイプ プロパティの簡単な説明
- データ型 -- ポリシー サーバがエージェントに返す値のデータ型として以下のいずれか。
  - 文字列 -- 文字列値。
  - 数値 -- 特定のネットワーク バイト オーダーの 4 バイトの数値。
  - IP アドレス -- IP アドレスとして解釈可能な特定ネットワーク バイト オーダーの 4 バイトの数値。
  - 暗号化文字列 -- RADIUS と同様な暗号化を使用して暗号化された文字列。暗号化文字列は、RADIUS エージェントに使用されます。
- 識別子 -- 属性に割り当てられた 1 ~ 255 の範囲の数値 ID。カスタム エージェントの場合、使用可能な値は 224 ~ 255 および 1 ~ 150 です。

レスポンス動作の割り当て時には、以下を考慮してください。

- このレスポンス タイプの目的： 認証または認可?
- このレスポンスが必要とするレスポンス動作のタイプ： 成功または失敗?
- このタイプの属性のインスタンスを 1 つのレスポンス内で許可する数： [ゼロまたは1] または [ゼロまたは複数] ?
- 認可レスポンス タイプをセットアップしている場合は、次の 2 つの属性選択のいずれかを使用できます： [アクセスの受け付け] または [アクセスの拒否] 。
- 認証レスポンス タイプをセットアップしている場合は、表示されている 3 つの属性選択のいずれかを任意に使用できます： [アクセスの受け付け]、[アクセスの拒否]、または [アクセスのチャレンジ]

### 値

このグループボックスは、[エージェントタイプ属性]で[データ型]が[数値]となる属性を選択している場合のみ表示されます。このようなデータ型のレスポンスには、[記号名]と[数値]を割り当てることができます。これにより、このレスポンスタイプのレスポンスを作成する際に、それぞれの数値が表わす内容の識別が容易になります。

### エージェントタイプ属性([プロパティ]および[値])の設定

#### エージェントタイプ属性を設定する方法

1. [エージェントタイプ属性]グループボックスの[エージェントタイプ属性の作成]をクリックします。
2. 属性の名前および説明を[一般]グループボックスに入力します。
3. データ型および識別子を[プロパティ]グループボックスに入力します。
4. データ型が数値の場合、[値]グループボックスが表示されます。記号名および数値を入力します。
5. 該当するレスポンス操作を入力します。
6. [サブミット]をクリックして、選択内容を保存します。

## エージェントタイプの変更

### エージェントタイププロパティを編集する方法

1. Administrative UI にログインします。
2. [インフラストラクチャ]-[エージェントタイプ]-[エージェントタイプの変更] をクリックします。
3. 変更するエージェントタイプの名前を入力し、[検索] をクリックします。または、何もせずに [検索] をクリックすると、エージェントタイプの全リストが表示されます。
4. エージェントタイプを選択します。  
[エージェントタイプの変更] ダイアログボックスが表示されます。
5. 変更する [アクション] または [属性] の左側の [編集] ボタンをクリックします。また、アクションや属性を削除することもできます (リストの右側の [削除] ボタン)。
6. 必要な変更を行います。
7. [サブミット] をクリックして変更内容を保存します。  
[エージェントタイプの変更] ダイアログボックスが表示されます。
8. 再度 [サブミット] をクリックして、エージェントタイプの変更を保存します。



# 第 4 章: ポリシー管理 API ガイド

---

このセクションには、以下のトピックが含まれています。

- [ポリシー管理 API の概要 \(P. 125\)](#)
- [フェデレーション API \(P. 132\)](#)
- [ポリシー管理 API データ構造体 \(P. 144\)](#)
- [エクスポートされたタイプ \(P. 237\)](#)
- [ポリシーアプリケーションの構造体 \(P. 268\)](#)
- [ポリシー管理 API のカテゴリ別の関数 \(P. 269\)](#)
- [ポリシー管理 API の関数宣言 \(P. 295\)](#)
- [認証方式の設定 \(P. 604\)](#)

## ポリシー管理 API の概要

ポリシー管理 API を使用して、SiteMinder インストール内のポリシー オブジェクトを操作できます。ポリシー管理 API を使用することで、Administrative UI によって提供されるデータ操作のほとんどを実行できます。また、SiteMinder に接続するための独自のカスタム インターフェースを開発することもできます。

**注:** ポリシー管理 API で作業を行う前に、SiteMinder の概念に精通しておく必要があります。

## ポリシー管理のセットアップ

### ポリシー管理 API を使用する方法

1. ポリシー サーバおよび SiteMinder Software Development Kit を同じマシンにインストールします。Windows 環境では、ポリシー管理アプリケーションの実行にポリシー サーバが必要です。UNIX 環境では、ポリシー管理アプリケーションの構築と実行の両方にポリシー サーバが必要です。

**注:** ポリシー サーバサービスを実行しなくても、ポリシー管理アプリケーションを構築できます。

2. ポリシー サーバ管理コンソールを使用して、アクセスするポリシー ストアをポリシー サーバが指すように、ポリシー サーバを設定します。

3. ポリシー サーバがインストールされ、ポリシー ストアを指すように設定されたマシンでポリシー サーバ管理アプリケーションを実行します。ポリシー ストアは、ポリシー サーバとは異なるマシン上に配置することもできます。

ポリシー管理アプリケーションを実行するには、以下のファイルが必要です。

**Windows プラットフォーム** : SmPolicyApi45.dll

**UNIX プラットフォーム** : libsmpolicyapi45.so および libsmutilities.so (以下の場所にあります)

<sitefinder\_install\_location>%Netegrity%SiteMinder%lib

UNIX ビルドを実行する前に、サンプル makefile を参照してください。

ポリシー アプリケーションを構築するには、SmPolicyAPI45.h を含め、必要な共有ライブラリへのリンクを設定してください。

- **Windows プラットフォーム** : SmPolicyAPI45.lib へのリンク (以下の場所にあります) <install\_path>%sdk%lib%win32%
- **UNIX プラットフォーム** : libsmpolicyapi45.so ライブラリおよび libsmutilities.so ライブラリへのリンク (場所は上述のとおり)。

**注:** UNIX 向けのポリシー管理アプリケーションを構築する前に、SiteMinder SDK をポリシー サーバと同じマシンにインストールする必要があります。

## オブジェクト取得関数

これらの関数は、オブジェクトに関する情報を SiteMinder ポリシー ストアから取得します。

リターンコードが成功を示す場合、要求に一致するオブジェクトのリンクの一覧が返されます。ほとんどの場合、API は、一意のオブジェクト識別子に一致する単一のアイテムを返します。一致するオブジェクトが見つからない場合、リターンコードは失敗を示し、返されたリンク一覧のポインタは Null を指します。

オブジェクト取得関数には、Sm\_PolicyApi\_Get が前に付けられています。特定のオブジェクトの情報を取得する関数を検索するには、そのオブジェクトに対応する関数の表を調べてください。

詳細情報:

[ポリシー管理 API のカテゴリ別の関数 \(P. 269\)](#)

## オブジェクト作成関数

SiteMinder オブジェクトを作成するには、適切なデータ構造体に入力し、正しく初期化されたハンドルを使用して適切な関数をコールする必要があります。呼び出しが成功すると、

- 関数が `Sm_PolicyApi_Success` を返します。
- オブジェクトが SiteMinder ポリシー ストアに追加されます。
- 対応するオブジェクト構造体内の `pszOid` フィールドが、オブジェクトのオブジェクト識別子に設定されます。

オブジェクト作成関数には、`Sm_PolicyApi_Add` または `Sm_PolicyApi_Create` のいずれかが前に付けられています。

詳細情報:

[ポリシー管理 API のカテゴリ別の関数 \(P. 269\)](#)

## オブジェクト削除関数

これらの関数は、SiteMinder ポリシー ストアからオブジェクトを削除します。一度に 1 つのオブジェクトのみを削除できます。

オブジェクト削除関数には、`Sm_PolicyApi_Delete` または `Sm_PolicyApi_Remove` が前に付けられています。

詳細情報:

[ポリシー管理 API のカテゴリ別の関数 \(P. 269\)](#)

## オブジェクトの関連付け

いくつかのオブジェクトは、相互に関連付けたり関連付けを解除することができます。たとえば `Sm_PolicyApi_AddAdminToDomain()` は、管理者オブジェクトをドメインへの追加し、`Sm_PolicyApi_RemoveAdminFromDomain()` は、管理者オブジェクトをドメインから削除します。

「add-to」操作では、両方のオブジェクトがコールより前に存在し、関連付けが確立されていることが必要となります。「remove-from」操作の後、両方のオブジェクトは引き続き存在しますが、これら相互の関連付けは解除されています。

2つのオブジェクトの関連付けを設定または解除する関数を検索するときは、追加または削除するメソッドのカテゴリを調べてください。たとえば、関数 `Sm_PolicyApi_AddAdminToDomain()` および `Sm_PolicyApi_RemoveAdminFromDomain()` はいずれも管理者関数で検索できます。

## オブジェクト識別子

ネストされたレルムの導入により、オブジェクトの一意の識別は、レルム名に依存できなくなりました。`SiteMinder` オブジェクトが作成されると、一意のオブジェクト識別子 (OID) がオブジェクトの定義構造体の `pszOid` フィールドに書き込まれます。

これらの関数は、`SiteMinder` オブジェクトを返しません。代わりに、`SiteMinder` オブジェクトの OID が含まれる文字列ポインタの配列を返します。これらの OID を `SiteMinder` オブジェクト取得関数

(`Sm_PolicyApi_Get...`) に渡すことにより、取得するオブジェクトを指定できます。

OID の配列を返す関数は以下のとおりです。

- `Sm_PolicyApi_GetDomainObjects()`
- `Sm_PolicyApi_GetGlobalObjects()`
- `Sm_PolicyApi_GetUserDirSearchOrder()`

このグループ関数によって割り当てられたメモリは、`Sm_PolicyApi_FreeMemoryEx()` をコールすることにより解放してください。

## ディレクトリ検索順序関数

以下の関数は、ユーザディレクトリの検索順序の設定および取得に役立ちます。

- `Sm_PolicyApi_GetUserDirSearchOrder()` は、指定されたドメインに関連付けられたユーザディレクトリ オブジェクトのオブジェクト識別子 (OID) を取得します。
- `Sm_PolicyApi_SetUserDirSearchOrder()` は、ドメイン内のユーザディレクトリの検索順序を設定します。OID の順序リストは、*pszArray* 文字列配列で指定されます。この配列内のユーザディレクトリは、`Sm_PolicyApi_GetUserDirSearchOrder()` のコールによって取得されたユーザディレクトリ OID のリストと OID および数 (順序ではない) が一致する必要があります。

## パフォーマンス拡張

以下のアクションのいずれかを実行することにより、カスタム ポリシー管理アプリケーションがポリシーストア オブジェクトを更新するために必要な時間を短縮できます。

- `Sm_PolicyApi_Init()` のコールで `Sm_PolicyApi_InitFlags_PreLoadCache` フラグを省略します。
- 終了前のキャッシュ処理のための適切な時間を保証するために、カスタム アプリケーションにごく微小の時間遅延を導入します。

## メモリ、キャッシュ、エージェント キーの管理

以下の関数は、ポリシー管理 API によって割り当てられたメモリを解放します。

- `Sm_PolicyApi_FreeMemory()` および `Sm_PolicyApi_FreeMemoryEx()` は、API によって返されたすべてのリンク リスト ポインタを解放します。
- `Sm_AgentApi_FreeServers()` は、`Sm_AgentApi_GetConfig()` によって割り当てられたサーバ構造体の配列を解放します。
- `Sm_PolicyApi_FreeString()` は、1 つの文字列 (`pszUseMsg` や `pszErrMsg` などの文字列) に割り当てられたメモリを解放します。
- `Sm_PolicyApi_FreeStringArray()` は、返された文字列の配列を解放します。

また、管理コマンド `Sm_PolicyApi_ManagementCommand()` では、以下のよう  
なキャッシュおよびエージェント暗号化キーの管理を実行します。

- すべてのキャッシュのクリア
- ユーザ キャッシュのクリア
- リソース キャッシュからのレルムのクリア
- ダイナミック キーの変更
- 永続キーの変更

実行する管理操作のタイプは、`Sm_PolicyApi_ManagementCommand()` に渡す管理コマンドによって決定されます。

## オブジェクト スコープ

SiteMinder オブジェクトは、スコープに従って分類できます。

- **ドメイン オブジェクト**はドメイン内のみで表示されます。これらをドメイン間で共有することはできません。
- **グローバル オブジェクト**は、すべてのドメインで表示されます。グローバル オブジェクトはシステム オブジェクトと呼ばれる場合もあります。

SiteMinder オブジェクトのスコープは以下のとおりです。

- グローバル オブジェクトには以下のものが含まれます。
  - 管理者
  - エージェント タイプ
  - エージェント および エージェント グループ
  - 認証方式
  - 認証/認可マップ
  - 証明書マップ
  - ドメイン
  - ODBC クエリ方式
  - パスワード ポリシー
  - ポリシー
  - 登録方式
  - レスポンス
  - ルール
  - ユーザ ディレクトリ
- ドメイン オブジェクトには以下が含まれます。
  - ポリシー
  - レルム
  - レスポンス および レスポンス グループ
  - レスポンス 属性
  - ルール および ルール グループ
  - ユーザ ポリシー

## フェデレーション API

フェデレーション API は、アフィリエイト ドメイン オブジェクト (アフィリエイト、サービス プロバイダ、リソース パートナーなど) に関連するポリシー ストア データの操作をサポートします。

SiteMinder フェデレーション製品は、SAML 1.x、SAML 2.0、および WS フェデレーション プロファイルをサポートします。パートナーは、セキュアな方式でユーザ プロファイル情報を交換することができます。

詳細情報:

[フェデレーション関数](#) (P. 276)

[SAML 2.0 設定関数](#) (P. 286)

## SAML アサーション

SAML アサーションには次の項目が含まれます。

- アフィリエイト属性。以下に例を示します。
  - ユーザ ディレクトリからのユーザ プロファイル情報 (ユーザの電子メールアドレス、肩書きなど)。
  - ユーザ権限 (ユーザのアフィリエイト サイトにおける信用限度など)。
- セッション情報 (SAML 1.x アサーション) -- たとえば、アサーション プロデューサおよびコンシューマが個別のセッションを維持できるかどうかなど。

注: ポリシー サーバによって生成されたデフォルト アサーションを変更できます。その場合は、SiteMinder SDK の Java API で作成するカスタム Java クラスを使用してください。

## SAML 1.x

SAML 1.x のサポートを使用することで、ユーザは 2 回以上認証情報を提供する必要なしに、コンシューマ サイトに直接またはアサーションプロデューサ サイトからアクセスできます。

ユーザがアフィリエイト サイトで保護されたリソースへのアクセスを要求すると、プロデューサ サイトのポリシー サーバに通知されます。ユーザを認証した後（ユーザがまだ認証されていない場合）、ポリシー サーバは、コンシューマ サイトに関連付けられたアフィリエイト オブジェクトから SAML アサーションを生成します。

すると、アフィリエイト サイトのアプリケーションは、ポリシー サーバから SAML アサーションを取得し、認可の目的およびその他の必要な目的のための情報を使用します。

たとえば、銀行のサイト（プロデューサ サイト）にユーザがログインするとします。プロデューサ には、ポリシー サーバ ソフトウェアが含まれています。ポリシー サーバ には、クレジットカード サービスを提供するサイトを表すアフィリエイト オブジェクトのほか、銀行と提携した他のサイトを表す他のアフィリエイト オブジェクトも含まれます。ユーザがプロデューサ において認証される場合、そのユーザは、クレジットカード サイトのリンクをクリックし、認証情報を再入力せずに、サイトにアクセスできます。

### SAML 1.x 擬似コード例

このセクション内の擬似コードは、以下の操作を示しています。

1. API を初期化します。
2. アフィリエイト ドメインを追加します。
3. ユーザ ディレクトリをアフィリエイト ドメインに追加します。
4. アフィリエイト ドメインでアフィリエイトを作成します。
5. アフィリエイトにユーザを追加します。
6. アフィリエイトに属性を追加します。
7. 既存のアフィリエイト ドメインを取得します。
8. アフィリエイト ドメイン内のすべてのアフィリエイトを取得します。

9. アフィリエイト内のすべての属性を取得します。
10. アフィリエイト ドメインを削除します。

注: <> を使用して表記されたコメントは、分かりやすくするためにコードが省略されていることを表します。リターンコードの確認は、分かりやすくするために省略されています。

```
# 1. Initialize the API
use Netegrity::PolicyMgtAPI;
$policyapi = Netegrity::PolicyMgtAPI->New();
$session = $policyapi->CreateSession("adminid", "adminpwd");

# 2. アフィリエイト ドメインを追加
$affdomain = $session->CreateAffDomain("name", "description");

# 3. Add a previously obtained user directory to the affiliate domain
# <Obtain $userdir via $session->GetAllUserDirs>
$affdomain->AddUserDir($userdir);

# 4. Create an affiliate in the affiliate domain
$affiliate = $affdomain->CreateAffiliate("affname", "password",
                                         http://authurl, 60, 30);

# 5. Add users from a previously obtained user table to the affiliate
# <Obtain $user via $userdir->GetContents>
$affdomain->AddUser($user);

# 6. Add an attribute for the affiliate
$affdomain->AddAttribute(1, "staticAttrName=StaticAttrValue");
# 7. Get an existing affiliate domain
$affiliate = $affdomain->GetAffiliate("affname");

# 8. Get all the affiliates in an affiliate domain
@affiliates = $affdomain->GetAllAffiliate();

# 9. Get all the attributes in an affiliate
@affiliateAttrs = $affiliate->GetAllAttributes();

# 10. Remove an affiliate domain
$session->DeleteAffDomain($affiliate);
```

## SAML 2.0

SAML 2.0 を使用する場合、セキュリティ アサーションはフェデレーション内の以下のエンティティ間で共有されます。

### アイデンティティプロバイダ

アイデンティティプロバイダは、SAML 2.0 フェデレーション内のプリンシパルのためのアサーションを生成します。アイデンティティプロバイダは、プリンシパルがリソースへのアクセスを試行するサービスプロバイダに SAML アサーションを送信します。

### サービスプロバイダ

サービスプロバイダは、アサーションで提供される ID 情報を使用して、フェデレーション内のプリンシパルがアプリケーションおよび他のリソースを使用できるようにします。プリンシパルとは、ユーザまたは他のフェデレーションエンティティです。

サービスプロバイダは、SAML 2.0 認証方式を使用して、SAML 2.0 アサーションの情報に基づいてユーザを検証します。

アイデンティティプロバイダおよびサービスプロバイダは、SAML アフィリエーションに属することができます。SAML アフィリエーションは、単一のプリンシパルの名前識別子を共有する SAML エンティティのグループです。

サービスプロバイダおよびアイデンティティプロバイダはアフィリエーションに属することができますが、1つのエンティティが属することができるのは、1つのアフィリエーションのみです。各種サービスプロバイダは、アフィリエーションの全域で名前 ID 定義を共有します。アイデンティティプロバイダは、アフィリエーションにわたってユーザの不明瞭解消を共有します。

アフィリエーションを使用すると、各サービスプロバイダで必要とされる設定が少なくなります。さらに、1つのプリンシパルに対して1つの名前 ID を使用することにより、アイデンティティプロバイダでのストレージ容量が節約されます。

## シングル サインオンの例

プリンシパルは、セキュリティ アサーションを共有することによって、1 つのサイト（アイデンティティ プロバイダとして機能するサイト）にログインした後、別のサイト（サービス プロバイダ）のリソースにアクセスでき、2 番目のサイトでは明示的に認証情報を提供する必要がなくなります。

例：

1. ユーザは住宅購入者で、不動産業者の **Web** サイトで認証します。  
ユーザの認証には、任意の認証方式を使用できます。
2. ユーザは、不動産のリストを表示しながら、住宅ローン金利が魅力的な銀行のリンクに気が付ききます。
3. ユーザはリンクをクリックします。
4. 不動産業者のサイトでは、アイデンティティ プロバイダとして機能するエンティティがユーザの情報を **SAML** アサーション内にパッケージングした後、**SAML 2.0 POST** バインディングを使用して、アサーションを銀行のサイトに転送します。
5. 銀行のサイトでは、サービス プロバイダとして機能するエンティティが、アイデンティティ プロバイダに関連付けられた **SAML 2.0** 認証方式を使用して、銀行のサイト上のリソースに対してユーザを検証します。  
この検証は、ユーザに透過的です。
6. ユーザは、正常に検証されると、銀行のサイト上でレート情報の表示が許可されます。

## SAML 2.0 擬似コード例

このセクション内の擬似コードは、以下の操作を示しています。

1. API を初期化します。
2. サービス プロバイダのアフィリエイト ドメインを取得します。
3. メタデータ定数を変数に割り当てます。
4. サービス プロバイダ メタデータに値を割り当てます。
5. サービス プロバイダを作成します。
6. アフィリエイト ドメインに関連付けられたディレクトリからユーザを取得します。

7. サービスプロバイダにユーザを追加します。
8. サービスプロバイダのデフォルト スキュー時間を 100 に更新します。
9. 更新を保存します。
10. 更新されたスキュー時間を印刷します。

```
# 1. Initialize the API
use Netegrity::PolicyMgtAPI;
$policyapi = Netegrity::PolicyMgtAPI->New();
$session = $policyapi->CreateSession("adminid", "adminpwd");

# 2. Retrieve the affiliate domain for the Service Provider
$affDom=$session->GetAffDomain("AffiliateDomain");

# 3. Assign metadata constants to variables
$SAML_NAME=SAML_NAME;
$SAML_SP_AUTHENTICATION_URL=SAML_SP_AUTHENTICATION_URL;
$SAML_KEY_SPID=SAML_KEY_SPID;
$SAML_SP_IDPID=SAML_SP_IDPID;
$SAML_AUDIENCE=SAML_AUDIENCE;
$SAML_SP_ASSERTION_CONSUMER_DEFAULT_URL=
    SAML_SP_ASSERTION_CONSUMER_DEFAULT_URL;
$SAML_SP_NAMEID_ATTRNAME=SAML_SP_NAMEID_ATTRNAME;
$SAML_SKEWTIME=SAML_SKEWTIME;

# 4. Assign values to the Service Provider metadata
%hsh=($SAML_NAME=>'My Service Provider',
    $SAML_SP_AUTHENTICATION_URL=>
        'http://www.mysite.com/redirect.jsp',
    $SAML_KEY_SPID=>'http://www.spprovider.com',
    $SAML_SP_IDPID=>'http://www.idpprovider.com',
    $SAML_AUDIENCE=>'SSOAudience',
    $SAML_SP_ASSERTION_CONSUMER_DEFAULT_URL=>
        'http://www.defaultconsumer.com',
    $SAML_SP_NAMEID_ATTRNAME=>'attribute'
);

# 5. Create the Service Provider
$sp=$affDom->CreateSAMLServiceProvider(%hsh);

# 6. Retrieve users from the directory associated with the # affiliate
domain—in this case, users in the group HR
$userDir=$session->GetUserDir("MyNtDirectory");
$usr=$userDir->LookupEntry("HR");
```

```
# 7. Add the users to the Service Provider
$sp->AddUser($usr);

# 8. Update the Service Provider's default skewtime to 100
$sp->Property($SAML_SKEWTIME, "100");

# 9. Save the update
$sp->Save();

# 10. Print the updated skewtime
print "%n";
print $sp->Property($SAML_SKEWTIME);
```

### SAML 2.0 アフィリエーション

SAML 2.0 アフィリエーションは、名前 ID ネームスペースを共有するサービスプロバイダおよびアイデンティティプロバイダから構成されます。また、アイデンティティプロバイダは、アフィリエーションの全域でユーザ特定プロパティも共有しています。

SAML 2.0 アフィリエーションには、複数のサービスプロバイダおよびアイデンティティプロバイダを含めることができます。ただし、1つのサービスプロバイダやアイデンティティプロバイダは、1つの SAML 2.0 アフィリエーションにしか所属できません。

例：

プリンシパルは、セキュリティアサーションを共有することによって、1つのサイト（アイデンティティプロバイダとして機能するサイト）にログインした後、別のサイト（サービスプロバイダ）のリソースにアクセスでき、2番目のサイトでは明示的に認証情報を提供する必要がなくなります。

1. ユーザは住宅購入者で、不動産業者の Web サイトで認証します。  
ユーザの認証には、任意の認証方式を使用できます。
2. ユーザは、不動産のリストを表示しながら、住宅ローン金利が魅力的な銀行のリンクに気が付ききます。
3. ユーザはリンクをクリックします。
4. 不動産業者のサイトでは、アイデンティティプロバイダとして機能するエンティティがユーザの情報を SAML アサーション内にパッケージングした後、SAML 2.0 POST バインディングを使用して、アサーションを銀行のサイトに転送します。

5. 銀行のサイトでは、サービスプロバイダとして機能するエンティティが、アイデンティティプロバイダに関連付けられた **SAML 2.0** 認証方式を使用して、銀行のサイト上のリソースに対してユーザを検証します。  
この検証は、ユーザに透過的に行われます。
6. ユーザは、正常に検証されると、銀行のサイト上でレート情報の表示が許可されます。

## SAML 2.0 属性認証機関

**SiteMinder** では、リモートサイトで事前決定されたユーザ属性値を認可決定の根拠として使用する認可をサポートします。 要求にはセッション情報が含まれません。これは、ユーザが必ずしもリモートサイトで認証されるとは限らないためです。

たとえば、カスタマがレンタカー取扱店のサイトにログオンして、料金について問い合わせるとします。 カスタマは取扱店によって認証されますが、取扱店では、競争料金を提供するために、カスタマが指定する航空会社からの情報を使用します。 レンタカー取扱店は、カスタマの貯めたマイレージに基づくカスタマ品質コードを取得するために、航空会社の **Web** サイトに要求を入力します。 航空会社から品質コードの値 (たとえば **1A**) が返され、レンタカー取扱店はカスタマイズされた料金表を表示します。

この例では、レンタカー取扱店が **SAML** リクエスタとして機能し、航空会社が **SAML** 属性認証機関として機能します。 カスタマは属性認証機関によって認証されていません。

ポリシーサーバは、ポリシー式内の変数を使用することにより、この種の認可決定を実施します。 ポリシー式では、フェデレーション属性変数によって属性がリモート属性認証機関と関連付けられます。 ポリシーサーバはフェデレーション属性変数の解決を試行するときに、属性の値を要求する属性認証機関を決定します。

ポリシー管理 API では、**Sm\_PolicyApi\_SAMLRequesterAttr\_t** 構造体によって、**SAML** リクエスタから要求可能な属性が定義されています。 この構造体は、属性認証機関で認識されている属性の実際名と、フェデレーション属性変数で使用されているローカル名を指定します。 ローカル名は、**SAML 2.0** 認証方式で定義された変数にマップされます。

詳細情報:

[Sm\\_PolicyApi\\_SAMLRequesterAttr\\_t \(P. 203\)](#)

[SAML 2.0 属性権限関数 \(P. 285\)](#)

## SAML 2.0 インデックス付きエンドポイント

アイデンティティ プロバイダでシングル サインオンを設定する際、アサーション コンシューマ サービス (サービス プロバイダが SAML アサーションを消費できるようにするサービス) に対して複数のエンドポイント設定できます。設定したエンドポイントのそれぞれには、アサーション コンシューマ サービス URL への明示的な単一の参照の代わりに、一意のインデックス値が割り当てられます。

割り当てられたインデックスは、サービス プロバイダがアイデンティティ プロバイダに送信するアサーションに対するサービス プロバイダの要求の一部として使用できます。これにより、サービス プロバイダにおいて、異なるアサーション コンシューマ サービスに別のプロトコル バインディング (Artifact または POST) の使用が可能になります。

ポリシー管理 API では、たとえば以下のように

`Sm_PolicyApi_AddAssertionConsumerServiceToSAMLSP()` 関数をコールすることにより、新規のアサーション コンシューマ サービスをプログラマ的にサービス プロバイダに追加できます。

```
iSmApiRetCode = Sm_PolicyApi_AddAssertionConsumerServiceToSAMLSP (
    pSmApiSessionHandle,
    &structSAMLSPACS2,
    pszOid);
```

パラメータ :

- `pSmApiSessionHandle` は、管理セッションおよびクライアントセッションに関する情報を保持する構造体へのポインタです。
- `pstructSAMLSPACS2` は、`Sm_PolicyApi_SAMLSPAssertionConsumerService_t` へのポインタです。この構造体は、インデックス、バインディング、およびアサーション コンシューマ サービス URL を指定します。
- `pszOid` は、サービス プロバイダの OID を含む文字列へのポインタです。

さらに、API には、アサーション コンシューマ サービスを削除する関数、およびサービス プロバイダ オブジェクトで定義されたすべてのアサーション コンシューマ サービスを取得する関数が含まれます。C ポリシー管理 API サンプルプログラム (`smpolicyapiexample.cpp`) では、これらの関数の実装方法を示しています。

詳細情報:

[SAML 2.0 インデックス エンドポイント関数 \(P. 288\)](#)

## アフィリエイトに関するサンプル アプリケーション

C サンプルプログラム `smpolicyapi` は、アフィリエイト機能が強化されています。

アフィリエイト部分を実行するには

1. SiteMinder オプション パックをポリシー サーバにインストールします
2. アフィリエイト ポリシーストア オブジェクトを `smpolicy.smdif` に定義し、オブジェクトをポリシーストアにインポートします。
3. サンプルをインストールします。これにより、アフィリエイト サンプルで使用されるサンプル ユーザ ディレクトリが作成されます。

3つの選択肢が `smpolicyapi` プログラムに追加されています。

- **Affiliate Sample Api** をインストールしますか? (Y/N) :  
「Y」を選択すると、`smpolicyapi` は以下のように動作します。
  - アフィリエイト ドメインを追加
  - サンプル ユーザ ディレクトリをアフィリエイト ドメインに追加します。
  - アフィリエイトをアフィリエイト ドメインに追加します。
  - サンプル ユーザ ディレクトリからユーザをアフィリエイトに追加します。
  - アフィリエイト属性をアフィリエイトに追加します。

- アフィリエイト ドメインをトラバースしますか? (Y/N) :  
「Y」を選択すると、`smpolicyapi` は以下のように動作します。
  - アフィリエイト ドメインを取得し、それを印刷します。
  - サンプルアフィリエイト ドメイン内のユーザ ディレクトリ **OID** をすべて取得し、それらを印刷します。
  - サンプルアフィリエイト ドメイン内のアフィリエイト **OID** をすべて取得し、それらを印刷します。
  - サンプルアフィリエイト ドメイン内のアフィリエイトをすべて取得し、それらのプロパティを印刷します。
  - アフィリエイト内のユーザをすべて取得し、それらを印刷します。
  - アフィリエイト内の属性をすべて取得し、それらを印刷します。
- **Affiliate Sample Api** をアンインストールしますか? (Y/N) :  
「Y」を選択すると、`smpolicyapi` はアフィリエイト ドメインを削除します。

## WS-フェデレーション

WS フェデレーション指定情報は、パッシブクライアント（Web ブラウザなど）でどのようにフェデレーションフレームワークを実装するかを示すプロトコルを提供します。ADFS は Microsoft による WS フェデレーションパッシブリクエストプロファイルの実装です。

WS フェデレーション環境での Web SSO およびサインアウトは、アカウントパートナーおよびリソースパートナーを使用して実装されます。アカウントパートナーは、ユーザを認証して、WS フェデレーションセキュリティトークンを提供し、これらをリソースパートナーに渡します。リソースパートナーはセキュリティトークンを使用し、WS フェデレーションセキュリティトークンの内容に基づいてセッションを確立します。

SiteMinder がアカウントパートナーとして動作するためには、セキュリティトークンを消費するリソースパートナーを管理者が定義する必要があります。これは、アフィリエイトドメインでリソースパートナーを定義することにより実行されます。SiteMinder がリソースパートナーとして動作するためには、セキュリティトークンを提供するアカウントパートナーを管理者が定義する必要があります。これは、WS フェデレーション認証方式を定義することにより実行されます。

C ポリシー管理 API サンプルプログラム (`smpolicyapiexample.cpp`) には、リソースパートナーの定義、一覧表示、削除方法の例や、WS フェデレーション認証方式の定義、一覧表示、削除方法の例が含まれます。

詳細情報:

[Sm\\_PolicyApi\\_WSFEDProviderProp\\_t](#) (P. 227)

[Sm\\_PolicyApi\\_WSFEDResourcePartner\\_t](#) (P. 236)

[WS-Federation 関数](#) (P. 293)

## ポリシー管理 API データ構造体

それぞれのデータ構造体は、SiteMinder ポリシー ストア内のエンティティを表します。構造体は、以下の共通の形式を持ちます。

- 各構造体の最初のフィールドは、構造体のタイプを示し、正しいポリシー管理 API メモリ関数によって解放できるようになっています。構造体が入力パラメータとして使用される場合、最初のフィールドは、ポリシー管理 API によって無視されます。構造体に戻り値に使用されるとき、最初のフィールドは、ポリシー管理 API によって適切に設定されます。
- 各構造体の最後のフィールドは、返されたリンク リスト内の次のアイテムへのポインタです。入力時、このフィールドは無視されます。これは、ポリシー管理 API が入力パラメータとしてリンク リストを受け付けないためです。（リンク リストを入力パラメータとして提供すると、この関数はリストの最初のアイテムのみを調べます。）

以下の構造体内のすべての文字配列は **BFSIZE** に従ってサイズが調整されます。**BFSIZE** は 1024 文字に設定されています。ポリシー時間制限には、24 文字の **TIMESIZE** バッファが使用されます。

## Sm\_PolicyApi\_Admin\_t

SiteMinder 管理者オブジェクトを定義します。

## 構文

```
typedef struct Sm_PolicyApi_Admin_s
{
    int iStructId;
    char pszOid[BFSIZE];
    char pszName[BFSIZE];           /* Required */
    char pszDesc[BFSIZE];
    char pszPassword[BFSIZE];
    char pszUserDirOid[BFSIZE];
    char pszSchemeOid[BFSIZE];
    Sm_PolicyApi_AdminRights_t nRights; /* Required */
    struct Sm_PolicyApi_Admin_s* next;
} Sm_PolicyApi_Admin_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義された管理者データ構造体 ID。
<i>pszOid</i>	管理者オブジェクトのオブジェクト識別子。
<i>pszName</i>	管理者の名前。
<i>pszDesc</i>	管理者の簡単な説明。
<i>pszPassword</i>	これは、 <i>pszUserDirOid</i> が指定されない場合に必須です。SiteMinder は、そのディレクトリにパスワードを保存します。
<i>pszUserDirOid</i>	管理者が外部ディレクトリに保存されている場合のユーザディレクトリのオブジェクト識別子。
<i>pszSchemeOid</i>	管理者の認証に使用する認証方式のオブジェクト識別子。これは、 <i>pszUserDirOid</i> が指定される場合に必須です。
<i>nRights</i>	Sm_PolicyApi_AdminRights_t で指定された管理者の権限。
<i>next</i>	次の管理者構造体へのポインタ。

## Sm\_PolicyApi\_Affiliate\_t

アフィリエイト オブジェクトを表します。

### 構文

```
typedef struct Sm_PolicyApi_Affiliate_s
{
    int iStructId;
    char pszOid[BFSIZE];
    char pszName[BFSIZE];
    char pszDesc[BFSIZE];
    char pszAffiliateDomainOid[BFSIZE];
    char pszPassword[BFSIZE];
    bool bIsEnabled;
    bool bAllowNotification;
    char pszAuthURL[BFSIZE];
    char pszAudience[BFSIZE];
    long nValidityDuration;
    bool bSharedSession;
    long nSyncInterval;
    long nSkewTime;
    long nStartTime;
    long nEndTime;
    unsigned char pszTimeGrid[TIMESIZE];
    Sm_PolicyApi_IPAddress_t *pIPAddress;
    struct Sm_PolicyApi_Affiliate_s* next;
    long nSAMLVersion;
    char pszAssertionPluginClass[BFSIZE];
    char pszAssertionPluginParameters[BFSIZE];
    Sm_PolicyApi_SAML_Profile_t SAMLProfile;
    char pszConsumerURL[BFSIZE];
} Sm_PolicyApi_Affiliate_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義されたドメインデータ構造体 ID。
<i>pszOid</i>	アフィリエイト オブジェクトのオブジェクト識別子。
<i>pszName</i>	アフィリエイトの名前
<i>pszDesc</i>	アフィリエイトの簡単な説明。
<i>pszAffiliateDomainOid</i>	アフィリエイト ドメイン オブジェクトのオブジェクト識別子。

フィールド	説明
<i>pszPassword</i>	Null 終端文字列としてのアフィリエイトのためのパスワード。
<i>bIsEnabled</i>	アフィリエイトが有効かどうかを示すブール。
<i>bAllowNotification</i>	通知がアフィリエイトに許可されているかどうかを示すブール。
<i>pszAuthURL</i>	アフィリエイトの認証 URL。
<i>pszAudience</i>	ポータルとアフィリエイトの間の許諾契約の条件について説明するドキュメントの URI。
<i>nValidityDuration</i>	アサーションが有効な時間（秒）。
<i>bSharedSession</i>	ポータルとアフィリエイトの間のセッション情報の共有を有効にするブール。
<i>nSyncInterval</i>	アフィリエイトがセッションステータスを検証するためにポータルにアクセスする頻度（秒）。
<i>nSkewTime</i>	ポータルのシステムクロック時間とアフィリエイトのシステムクロック時間の差（秒）。
<i>nStartTime</i>	時間制限が有効になる時刻。この値は、標準の <code>time_t</code> 形式で保存されます。 <i>nStartTime</i> を 0 に設定すると、時間制限が即座に開始されます。
<i>nEndTime</i>	時間制限が終了する時刻。この値は、標準の <code>time_t</code> 形式で保存されます。 <i>nEndTime</i> を 0 に設定すると、時間制限が即座に終了します。
<i>pszTimeGrid</i>	1 週間全体の時間制限が含まれる配列。
<i>pIPAddress</i>	ポリシーによって管理されているリソースへのアクセスを取得するために使用する必要がある IP アドレス。
<i>next</i>	次のアフィリエイト構造体へのポインタ。

フィールド	説明
nSAMLVersion	<p>以下のいずれかの SAML バージョン定数です。</p> <ul style="list-style-type: none"> <li>■ SM_POLICY_API_SAML_1_0</li> <li>■ SM_POLICY_API_SAML_1_1</li> </ul> <p>ポリシー管理 API のセッションバージョンが SM_POLICY_API_VERSION_6_0_1 以上である場合にのみ、SAML バージョンは効力を持ちます。</p>
pszAssertion PluginClass	<p>カスタムアサーションジェネレータプラグインの完全修飾クラス名。このプラグインを使用して、SiteMinder がアフィリエイト用に生成するデフォルトアサーションをカスタマイズできます。</p> <p>ポリシー管理 API のセッションバージョンが SM_POLICY_API_VERSION_6_0_2 以上である場合にのみ、プラグインクラスおよびパラメータ文字列がサポートされます。</p> <p>カスタムアサーションジェネレータプラグインは、Java SDK で実装されます。</p>
pszAssertion PluginParameters	<p>カスタムアサーションジェネレータプラグインに渡すパラメータ文字列。</p>
SAMLProfile	<p>SAML アサーションの送信および受信に使用されたプロファイルのタイプ。Sm_PolicyApi_SAML_Profile_t で定義されます。</p> <p>有効なプロファイルは以下のとおりです。</p> <ul style="list-style-type: none"> <li>■ Sm_PolicyApi_SAML_Profile_Artifact。SAML アサーションは、アサーションプロデューサと関連付けられた URL から取得されます。URL は SAML アーチファクト認証方式の設定時に指定されます。</li> <li>■ Sm_PolicyApi_SAML_Profile_POST。生成された SAML アサーションは、pszConsumerURL で指定された URL にポストされます。</li> </ul> <p>ポリシー管理 API のセッションバージョンが SM_POLICY_API_VERSION_6_0_2 以上である場合にのみ、このプロファイルがサポートされます。</p>
pszConsumerURL	<p>SAML POST プロファイルの場合、このフィールドは、要求元のユーザのブラウザが、生成されたアサーションをポストする必要がある場所の URL を指定します。この URL と関連付けられたサイトは、アサーションを検証し、その内容を使用してアクセスを決定します。</p>

## Sm\_PolicyApi\_AffiliateAttr\_t

アフィリエイト属性を表します。アフィリエイト属性を操作するために、アフィリエイトメソッドと共に使用されます。

### 構文

```
typedef struct Sm_PolicyApi_AffiliateAttr_s
{
    int iStructId;
    Sm_PolicyApi_AffiliateAttrType_t nAttrType;
    char pszValue[BFSIZE];
    struct Sm_PolicyApi_AffiliateAttr_s* next;
} Sm_PolicyApi_AffiliateAttr_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義されたドメインデータ構造体 ID。
<i>nAttrType</i>	Sm_PolicyApi_AffiliateAttrType_t からのアフィリエイト属性タイプ。
<i>pszValue</i>	<p>アフィリエイト属性指定。 アフィリエイト属性の名前および値。以下の形式のいずれかで指定されます。</p> <ul style="list-style-type: none"> <li>■ スタティック属性 : variableName=value</li> <li>■ ユーザ属性 : variableName=&lt;%userattr="AttrName"%&gt;</li> <li>■ DN 属性 : variableName=&lt;#dn="DNSpec" attr="AttrName"#&gt;</li> </ul> <p>SiteMinder がネストされたグループから DN 属性を取得できるようにするには、<i>DNSpec</i> の最初に感嘆符 (!) を付けます。たとえば、次のようにします。 dn="!ou=People,o=security.com"</p>
<i>next</i>	次のアフィリエイト属性構造体へのポインタ。

## Sm\_PolicyApi\_AffiliateDomain\_t

アフィリエイト ドメインを表します。

### 構文

```
typedef struct Sm_PolicyApi_Domain_s
{
    int iStructId;
    char pszOid[BFSIZE];
    char pszName[BFSIZE];
    char pszDesc[BFSIZE];
    struct Sm_PolicyApi_Domain_s* next;
} Sm_PolicyApi_Domain_t, Sm_PolicyApi_AffiliateDomain_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義されたアフィリエイト ドメイン データ構造体 ID。
<i>pszOid</i>	アフィリエイト ドメインのオブジェクト識別子。
<i>pszName</i>	アフィリエイト ドメインの名前。
<i>pszDesc</i>	アフィリエイト ドメインの簡単な説明。
<i>next</i>	次のアフィリエイト ドメイン構造体へのポインタ。

## Sm\_PolicyApi\_Agent\_t

SiteMinder エージェント オブジェクトを定義します。

## 構文

```
typedef struct Sm_PolicyApi_Agent_s
{
    int iStructId;
    char pszOid[BFSIZE];
    char pszName[BFSIZE];           /* Required */
    char pszDesc[BFSIZE];
    char pszIpAddr[BFSIZE];
    char pszSecret[BFSIZE];
    char pszAgentTypeOid[BFSIZE];  /* Required */
    int nRealmHintAttrId;         /* Required */
    struct Sm_PolicyApi_Agent_s* next;
} Sm_PolicyApi_Agent_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義されたエージェント データ構造体 ID。
<i>pszOid</i>	エージェント オブジェクトのオブジェクト識別子。
<i>pszName</i>	エージェントの名前です。 エージェント名がポリシーストアに書き込まれると、エージェント名は小文字に変換されます。
<i>pszDesc</i>	エージェントの簡単な説明。
<i>pszIpAddr</i>	エージェントが配置されたサーバの IP アドレス。
<i>pszSecret</i>	セキュアなデータ交換の確立を目的として 2 つのパーティが既知な共有秘密キー。 共有秘密キーが提供される場合、エージェントは SiteMinder バージョン 4.x エージェントであると見なされます。共有秘密キーが提供されない場合、エージェントは SiteMinder バージョン 5.x または 6.0 エージェントであると見なされます。
<i>pszAgentTypeOid</i>	エージェントのタイプ。
<i>nRealmHintAttrId</i>	ヒント属性は、RADIUS クライアントデバイスによって送信される RADIUS 属性です。
<i>next</i>	次のエージェント構造体へのポインタ。

## Sm\_PolicyApi\_AgentConfig\_t

エージェント設定オブジェクトを定義します。

### 構文

```
typedef struct Sm_PolicyApi_AgentConfig_s
{
    int iStructId;
    char pszOid[BFSIZE];
    char pszName[BFSIZE];          /* Required */
    char pszDesc[BFSIZE];
    struct Sm_PolicyApi_AgentConfig_s* next;
} Sm_PolicyApi_AgentConfig_t;
```

フィールド	説明
<i>iStructId</i>	データ構造体 ID、Sm_PolicyApi_Structs_t で定義されています。
<i>pszOid</i>	エージェント設定オブジェクトのオブジェクト識別子。
<i>pszName</i>	エージェント設定の名前。
<i>pszDesc</i>	エージェント設定の簡単な説明。
<i>next</i>	次のエージェント設定オブジェクト構造体へのポインタ。

## Sm\_PolicyApi\_AgentType\_t

SiteMinder エージェント タイプ オブジェクトを定義します。

### 構文

```
typedef struct Sm_PolicyApi_AgentType_s
{
    int iStructId;
    char pszOid[BFSIZE];
    char pszName[BFSIZE];
    char pszDesc[BFSIZE];
    int nRfcId;
    int nAgentType;
    int nResourceType;
    int nAgentTypeSpecific;
    struct Sm_PolicyApi_AgentType_s* next;
} Sm_PolicyApi_AgentType_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義されたエージェント タイプ データ構造体 ID。
<i>pszOid</i>	エージェント タイプ オブジェクトのオブジェクト識別子。
<i>pszName</i>	エージェント タイプ オブジェクトの名前。
<i>pszDesc</i>	エージェント タイプ オブジェクトの簡単な説明。
<i>nRfcId</i>	IANA (Internet Assigned Numbers Authority) によって割り当てられた IETF ベンダー ID。 (RADIUS デバイスのみに適用されません。)
<i>nAgentType</i>	エージェント タイプ: RADIUS (value=0) または Web エージェント (value=1)。

フィールド	説明
<i>nResourceType</i>	エージェントが保護するリソースタイプ。以下の値のいずれかです。 <ul style="list-style-type: none"> <li>■ 0 = なし</li> <li>■ 1 = URL</li> <li>■ 2 = IpAddr</li> <li>■ 3 = IpAddrRange</li> <li>■ 4 = AgentAuth</li> <li>■ 5 = Radius 認証</li> </ul>
<i>nAgentTypeSpecific</i>	RADIUS ベンダーによって提供されるベンダーに固有のオフセット。(RADIUS デバイスのみに適用されます。) 必須フィールドではありません。
<i>next</i>	次のエージェントタイプ構造体へのポインタ。

## Sm\_PolicyApi\_AgentTypeAttr\_t

SiteMinder エージェントタイプ属性オブジェクトを定義します。

### 構文

```
typedef struct Sm_PolicyApi_AgentTypeAttr_s
{
    int iStructId;
    char pszOid[BFSIZE];
    char pszName[BFSIZE];
    char pszDesc[BFSIZE];
    char pszAgentTypeOid[BFSIZE];
    struct Sm_PolicyApi_AgentTypeAttr_s* next;
} Sm_PolicyApi_AgentTypeAttr_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義されたエージェントタイプ属性データ構造体 ID。
<i>pszOid</i>	エージェントタイプ属性オブジェクトのオブジェクト識別子。

フィールド	説明
<i>pszName</i>	エージェントタイプ属性オブジェクトの名前。
<i>pszDesc</i>	エージェントタイプ属性オブジェクトの簡単な説明。
<i>pszAgentTypeOid</i>	エージェントタイプのオブジェクト識別子。
<i>next</i>	次のエージェントタイプ属性構造体へのポインタ。

## Sm\_PolicyApi\_Association\_t

エージェント設定オブジェクトの設定パラメータ名およびその関連付けられた値を定義します。

### 構文

```
typedef struct Sm_PolicyApi_Association_s
{
    int iStructId;
    char pszOid[BFSIZE];
    char pszName[BFSIZE];           /* Required */
    char* pszValue;
    int iFlags;
    struct Sm_PolicyApi_Association_s* next;
} Sm_PolicyApi_Association_t;
```

フィールド	説明
<i>iStructId</i>	データ構造体 ID、Sm_PolicyApi_Structs_t で定義されています。
<i>pszOid</i>	設定パラメータ関連付けオブジェクトのオブジェクト識別子。
<i>pszName</i>	設定パラメータの名前。
<i>pszValue</i>	設定パラメータの値。
<i>iFlags</i>	「1」の場合、名前/値ペアは暗号化された形式で保存されます。 「0」の場合、保存はプレーンテキストで行われます。
<i>next</i>	次の設定パラメータ関連付け構造体へのポインタ。

## Sm\_PolicyApi\_AuthAzMap\_t

SiteMinder 認証および認可マッピング オブジェクトを定義します。

### 構文

```
typedef struct Sm_PolicyApi_AuthAzMap_s
{
    int iStructId;
    char pszOid[BFSIZE];
    char pszAuthDirOid[BFSIZE];           /* Required */
    char pszAuthDirName[BFSIZE];
    char pszAzDirOid[BFSIZE];           /* Required */
    char pszAzDirName[BFSIZE];
    Sm_PolicyApi_AuthAzMapType_t nMapType; /* Required */
    struct Sm_PolicyApi_AuthAzMap_s *next;
} Sm_PolicyApi_AuthAzMap_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義された認証および認可マッピング オブジェクト データ構造体 ID。
<i>pszOid</i>	認証および認可マッピング オブジェクトのオブジェクト識別子。
<i>pszAuthDirOid</i>	認証ディレクトリのオブジェクト識別子。
<i>pszAuthDirName</i>	認証ディレクトリの名前。
<i>pszAzDirOid</i>	許可ディレクトリのオブジェクト識別子。
<i>pszAzDirName</i>	許可ディレクトリの名前。
<i>nMapType</i>	認証ディレクトリと認可ディレクトリ間のマッピングのタイプ。マッピングは、DN、ユニバーサル識別子、またはディレクトリ内の属性に基づくことができます。
<i>next</i>	次の認証および認可マッピング構造体へのポインタ。

## Sm\_PolicyApi\_CertMap\_t

SiteMinder 認定マッピング オブジェクトを定義します。

## 構文

```
typedef struct Sm_PolicyApi_CertMap_s
{
    int iStructId;
    char pszOid[BFSIZE];
    char pszDesc[BFSIZE];
    char pszUserDirOid[BFSIZE];           /* Required */
    char pszIssuerDN[BFSIZE];           /* Required */
    char pszCaDN[BFSIZE];
    char pszMapAttr[BFSIZE];
    Sm_PolicyApi_CertMapFlags_t nFlags; /* Required */
    Sm_PolicyApi_CertMapAttrType_t nAttrType;
    Sm_PolicyApi_DirType_t nDirType;    /* Required */
    struct Sm_PolicyApi_CertMap_s *next;
} Sm_PolicyApi_CertMap_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義された証明書マッピング データ構造体 ID。
<i>pszOid</i>	証明書マッピング オブジェクトのオブジェクト識別子。
<i>pszDesc</i>	証明書マッピング オブジェクトの簡単な説明。
<i>pszUserDirOid</i>	ユーザ ディレクトリのオブジェクト識別子。
<i>pszIssuerDN</i>	サーバ証明書の DN。
<i>pszCaDN</i>	発行認証機関の DN。

フィールド	説明
<i>pszMapAttr</i>	<p>単一属性マッピングまたはカスタム マッピングを実行できます。</p> <ul style="list-style-type: none"> <li>■ 単一属性マッピング 属性の形式は次のとおりです。   <math display="block">\%{\langle \text{attribute name} \rangle}</math> <math>\langle \text{attribute name} \rangle</math> は、属性の名前です。ユーザの証明書のサブジェクト DN の単一の属性を、ユーザディレクトリに保存された単一の属性と照合して、ユーザのアイデンティティを確認します。</li> <li>■ カスタム マッピング 複数の属性の複雑なマッピングにカスタム マッピング式を使用することで、証明書マッピングを確立するためにユーザ DN から抽出される必要がある複数のユーザ属性を指定できます。カスタム マッピング式の構文は、マッピングの柔軟性を十分に高めるために設計された仕様になっています。この式は、証明書から取得する情報を識別し、その情報の適用先となるユーザディレクトリ内の場所を示します。基本的な構文は次のとおりです。UserAttribute=%{CertificateAttribute}、UserAttribute2=%{CertificateAttribute}</li> </ul>
<i>nFlags</i>	<p>以下のフラグの 1 つまたは複数を設定します。</p> <ul style="list-style-type: none"> <li>■ Sm_PolicyApi_CertMapFlags_CertRequired</li> <li>■ Sm_PolicyApi_CertMapFlags_UseDistributionPoints</li> <li>■ Sm_PolicyApi_CertMapFlags_VerifySignature</li> <li>■ Sm_PolicyApi_CertMapFlags_CRLCheck</li> <li>■ Sm_PolicyApi_CertMapFlags_Cache</li> </ul>

フィールド	説明
<i>nAttrType</i>	<p>この列挙は、X.509 クライアント証明書が認証ディレクトリ内のユーザ情報にどのようにマップするか指定します。</p> <ul style="list-style-type: none"><li>■ <b>Sm_PolicyApi_CertMapAttrType_Single = 1</b> ポリシー サーバがユーザの証明書のサブジェクト DN の単一属性を、ユーザ ディレクトリ内に保存された単一属性と照合して、ユーザの身元を確認するための単一属性を指定します。</li><li>■ <b>Sm_PolicyApi_CertMapAttrType_Custom = 2</b> カスタム マッピング式を指定して、ユーザの身元を確認できます。マッピングがカスタム式に基づく場合に、この属性タイプを指定します。</li><li>■ <b>Sm_PolicyApi_CertMapAttrType_Exact = 3</b> ポリシー サーバが証明書内のユーザの DN のすべてを、認証ディレクトリ内の DN すべてと照合するための正確な属性タイプを指定します。</li></ul>
<i>nDirType</i>	<p>ユーザの認証に使用されるディレクトリのタイプ。次のいずれかの値を取ります。</p> <ul style="list-style-type: none"><li>■ <b>Sm_PolicyApi_DirType_LDAP = 1</b></li><li>■ <b>Sm_PolicyApi_DirType_WinNT = 2</b></li><li>■ <b>Sm_PolicyApi_DirType_ODBC = 3</b></li></ul>
<i>next</i>	次の登録方式構造体へのポインタ。

## Sm\_PolicyApi\_Domain\_t

SiteMinder ドメイン オブジェクトを定義します。

### 構文

```
typedef struct Sm_PolicyApi_Domain_s
{
    int iStructId;
    char pszOid[BFSIZE];
    char pszName[BFSIZE];           /* Required */
    char pszDesc[BFSIZE];
    struct Sm_PolicyApi_Domain_s* next;
    Sm_PolicyApi_DomainFlags_t iFlags;
} Sm_PolicyApi_Domain_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義されたドメイン データ構造体 ID。
<i>pszOid</i>	ドメイン オブジェクトのオブジェクト識別子。
<i>pszName</i>	ドメインの名前。
<i>pszDesc</i>	ドメインの簡単な説明。
<i>next</i>	次のドメイン構造体へのポインタ。
<i>iFlags</i>	ドメインに関する処理を行うグローバル ポリシーを有効化または無効化するためのフラグ。

## Sm\_PolicyApi\_Group\_t

SiteMinder グループ オブジェクトを定義します。 SiteMinder グループは、 Sm\_PolicyApi\_Groups\_t 列挙で定義されます。

### 構文

```
typedef struct Sm_PolicyApi_Group_s
{
    int iStructId;
    char pszOid[BFSIZE];
    char pszName[BFSIZE];           /* Required */
    char pszDesc[BFSIZE];
    char pszAgentTypeOid[BFSIZE];  /* Required */
    struct Sm_PolicyApi_Group_s* next;
} Sm_PolicyApi_Group_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義されたグループ データ構造体 ID。
<i>pszOid</i>	グループ オブジェクトのオブジェクト識別子。
<i>pszName</i>	グループの名前。
<i>pszDesc</i>	グループの簡単な説明。
<i>pszpszAgentTypeOid</i>	エージェントのオブジェクト識別子。
<i>next</i>	次のグループ構造体へのポインタ。

## Sm\_PolicyApi\_HostConfig\_t

ホスト設定オブジェクトを定義します。

## 構文

```
typedef struct Sm_PolicyApi_HostConfig_s
{
    int iStructId;
    char pszOid[BFSIZE];
    char pszName[BFSIZE];           /* Required */
    char pszDesc[BFSIZE];
    char pszIPAddress[BFSIZE];
    bool bEnableFailOver;
    int iMaxSocketsPerPort;
    int iMinSocketsPerPort;
    int iNewSocketStep;
    int iRequestTimeout;
    struct Sm_PolicyApi_HostConfig_s* next;
    Sm_PolicyApi_Server_t* pServer;
    Sm_PolicyApi_Server_t* pCluster;
    long nFailoverThreshold;
} Sm_PolicyApi_HostConfig_t;
```

フィールド	説明
<i>iStructId</i>	データ構造体 ID、Sm_PolicyApi_Structs_t で定義されています。
<i>pszOid</i>	ホスト設定オブジェクトのオブジェクト識別子。
<i>pszName</i>	ホスト設定の名前。
<i>pszDesc</i>	ホスト設定の簡単な説明。
<i>pszIPAddr</i>	ホスト設定オブジェクトの IP アドレス。
<i>bEnableFailOver</i>	エージェントとポリシー サーバがフェールオーバーまたはラウンドロビンによって通信する必要があるかどうかを指定します。このパラメータは、 <i>pServer</i> パラメータで指定された非クラスト サーバに適用可能です。
<i>iMaxSocketsPerPort</i>	エージェントと特定のポリシー サーバ プロセスの間で開くことができる TCP/IP ソケットの最大数。

フィールド	説明
<i>iMinSocketsPerPort</i>	エージェントと特定のポリシー サーバプロセスの間で開く必要がある TCP/IP ソケットの最小数。
<i>iNewSocketStep</i>	要求が増加した場合に、エージェントと特定のポリシー サーバプロセスの間で開く TCP/IP ソケットの増分数。
<i>iRequestTimeout</i>	エージェントがポリシー サーバからのレスポンスを待機する時間長 (秒)
<i>next</i>	次のホスト設定構造体へのポインタ。
<i>pServer</i>	特定のポリシー サーバインストールに関する TCP/IP 接続性情報を説明する <i>Sm_PolicyApi_Server_t</i> 構造体のリンク リスト。
<i>pCluster</i>	<i>Sm_PolicyApi_Server_t</i> 構造体の配列へのポインタ。配列はクラスタ サーバを指定します。
<i>nFailoverThreshold</i>	フェールオーバーのしきい値パーセントを指定します。このパラメータは、 <i>pCluster</i> で指定されたクラスタ サーバに適用可能です。

## Sm\_PolicyApi\_IPAddress\_t

オブジェクトに対する IP アドレス制限を定義します。たとえば、適用するポリシーに対して適合する必要がある IP アドレス制限を定義できます。

単一ホスト IP アドレス、IP アドレス範囲、ホスト名、またはサブネットマスクを指定できます。

### 構文

```
typedef struct Sm_PolicyApi_IPAddress_s
{
    int iStructId;
    Sm_PolicyApi_IPAddressType_t iIPAddressType;
    unsigned long nIPAddress;
    unsigned long nEndIPAddress;
    unsigned long nSubnetMask;
    char pszHostName[BFSIZE];
    struct Sm_PolicyApi_IPAddress_s *next;
} Sm_PolicyApi_IPAddress_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義された IP アドレス構造体 ID。
<i>iIPAddressType</i>	Sm_PolicyApi_IPAddressType_t で列挙される IP アドレスのタイプ。
<i>nIPAddress</i>	開始 IP アドレス
<i>nEndIPAddress</i>	終了 IP アドレス。
<i>nSubnetMask</i>	<p>サブネットマスク値は、ビット数で表します。この値を取得するには、アドレスのバイナリ値内のビットをカウントします。</p> <p>たとえば、サブネットマスクが <b>255.255.255.128</b> であるとしみます。バイナリ形式では以下ようになります。</p> <p style="text-align: center;">11111111 11111111 11111111 10000000</p> <p>左から右に数えて、<i>nSubnetMask</i> に渡す数は <b>25</b> になります。</p>
<i>pszHostName</i>	ポリシーを適用するためユーザが使用する必要があるマシンのホスト名。
<i>next</i>	次の IP アドレス構造体へのポインタ。

## Sm\_PolicyApi\_ManagementCommand\_t

管理コマンドを定義します。管理コマンドを使用することにより、エージェントがポリシーサーバから情報を取得できます。

### 構文

```
typedef struct Sm_PolicyApi_ManagementCommand_s
{
    Sm_PolicyApi_ManagementCommands_t iCommand;
    char pszData[BFSIZE];
} Sm_PolicyApi_ManagementCommand_t;
```

フィールド	説明
<i>iCommand</i>	Management Commands で指定された管理コマンド。
<i>pszData</i>	予約済み。

## Sm\_PolicyApi\_ODBCQueryScheme\_t

SiteMinder ODBC クエリ方式オブジェクトを定義します。

## 構文

```
typedef struct Sm_PolicyApi_ODBCQueryScheme_s
{
    int iStructId;
    char pszOid[BFSIZE];
    char pszName[BFSIZE];           /* Required */
    char pszDesc[BFSIZE];
    char pszQueryEnumerate[BFSIZE]; /* Required */
    char pszQueryGetObjInfo[BFSIZE]; /* Required */
    char pszQueryLookup[BFSIZE];    /* Required */
    char pszQueryInitUser[BFSIZE];  /* Required */
    char pszQueryAuthenticateUser[BFSIZE]; /* Required */
    char pszQueryGetUserProp[BFSIZE]; /* Required */
    char pszQuerySetUserProp[BFSIZE]; /* Required */
    char pszQueryGetUserProps[BFSIZE]; /* Required */
    char pszQueryLookupUser[BFSIZE]; /* Required */
    char pszQueryGetGroups[BFSIZE];  /* Required */
    char pszQueryIsGroupMember[BFSIZE]; /* Required */
    char pszQueryGetGroupProp[BFSIZE]; /* Required */
    char pszQuerySetGroupProp[BFSIZE]; /* Required */
    char pszQueryGetGroupProps[BFSIZE]; /* Required */
    char pszQueryLookupGroup [BFSIZE]; /* Required */
    char pszQuerySetPassword [BFSIZE]; /* Required */
    struct Sm_PolicyApi_ODBCQueryScheme_s* next;
} Sm_PolicyApi_ODBCQueryScheme_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義された ODBC クエリ方式データ構造体 ID。
<i>pszOid</i>	ODBC クエリ方式オブジェクトのオブジェクト識別子。
<i>pszName</i>	ODBC クエリ方式の名前。
<i>pszDesc</i>	ODBC クエリ方式の簡単な説明。
<i>pszQueryEnumerate</i>	ディレクトリ内のユーザ オブジェクトの名前を一覧表示するクエリ。

フィールド	説明
<i>pszQueryGetObjInfo</i>	オブジェクトのクラスをフェッチするクエリ。
<i>pszQueryLookup</i>	クエリはグループテーブルで指定された属性に基づいてオブジェクトを返すクエリ。
<i>pszQueryInitUser</i>	指定された名前のユーザがデータベースに存在するかどうかを判別するクエリ。
<i>pszQueryAuthenticateUser</i>	ユーザからパスワードを取得するクエリ。
<i>pszQueryGetUserProp</i>	ユーザプロパティの値を取得するクエリ。プロパティは、 <i>pszQueryGetUserProps</i> でリスト表示されたプロパティの1つであることが必要です。
<i>pszQuerySetUserProp</i>	ユーザプロパティの値を設定するクエリ。プロパティは、 <i>pszQueryGetUserProps</i> に一覧表示されたプロパティの1つであることが必要です。
<i>pszQueryGetUserProps</i>	ユーザ名と同じテーブルに存在するユーザ属性のカンマ区切りリストを返すクエリ。
<i>pszQueryLookupUser</i>	ユーザテーブルの属性を使用して、ユーザ名を取得するクエリ。
<i>pszQueryGetGroups</i>	ユーザがメンバとして属しているグループの名前を取得するクエリ。
<i>pszQueryIsGroupMember</i>	特定のユーザのグループメンバシップを識別するクエリ。
<i>pszQueryGetGroupProp</i>	<i>pszQueryGetGroupProps</i> で定義されたプロパティの値を返すクエリ。
<i>pszQuerySetGroupProp</i>	グループプロパティの値を設定するクエリ。プロパティは、 <i>pszQueryGetGroupProps</i> に一覧表示されたプロパティの1つであることが必要です。
<i>pszQueryGetGroupProps</i>	グループ属性のカンマ区切りリストを返すクエリ。
<i>pszQueryLookupGroup</i>	グループテーブルの属性を使用して、グループ名を取得するクエリ。
<i>pszQuerySetPassword</i>	ユーザパスワードを変更するクエリ。

フィールド	説明
<i>next</i>	次の ODBC クエリ方式構造体へのポインタ。

## Sm\_PolicyApi\_Oid\_t

SiteMinder オブジェクト コレクションを取得するために、以下の関数で使用されます。

- Sm\_PolicyApi\_GetDomainObjects()
- Sm\_PolicyApi\_GetChildren()
- Sm\_PolicyApi\_GetGlobalObjects()
- Sm\_PolicyApi\_GetGroupOids()。

### 構文

```
typedef struct Sm_PolicyApi_Oid_s
{
    int iStructId;
    int iObjectId;
    char pszOid[BFSIZE];
    struct Sm_PolicyApi_Oid_s *next;
} Sm_PolicyApi_Oid_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義された OID データ構造体 ID。
<i>iObjectId</i>	オブジェクトタイプ識別子 (Sm_PolicyApi_Objects_t に列挙されます)。
<i>pszOid</i>	一意のオブジェクト識別子
<i>next</i>	次の OID 構造体へのポインタ。

## Sm\_PolicyApi\_PasswordMsgField\_t

パスワードポリシーメッセージフィールドを説明します。パスワードポリシーメッセージフィールドには、新しいパスワードの検証試行中に発生したエラーに関する情報が含まれます。

パスワードポリシーメッセージフィールドは、パスワードポリシーメッセージ識別子に関連付けられています。この識別子が `Sm_PolicyApi_GetPasswordMsg()` の `nMsgId` パラメータで返されます。

この構造体は、`Sm_PolicyApi_GetPasswordMsg()` の `ppStructMsgField` パラメータで返されます。

### 構文

```
typedef struct Sm_PolicyApi_PasswordMsgField_s
{
    int iStructId;
    Sm_PolicyApi_PasswordMsgFieldId_t nId;
    Sm_PolicyApi_FieldType_t nType;
    char pszMsg[BFSIZE];
    int nValue;
    struct Sm_PolicyApi_PasswordMsgField_s* next;
} Sm_PolicyApi_PasswordMsgField_t;
```

フィールド	説明
<code>iStructId</code>	<code>Sm_PolicyApi_Structs_t</code> で定義されたパスワードポリシーデータ構造体 ID。
<code>nId</code>	パスワードメッセージフィールド識別子。メッセージフィールド識別子は <code>Sm_PolicyApi_PasswordMsgFieldId_t</code> で列挙されます。
<code>nType</code>	<code>Sm_PolicyApi_FieldType_t</code> で列挙されるメッセージフィールドのデータ型（整数、文字列、なし）。 <code>nType</code> の値により、 <code>pszMsg</code> または <code>nValue</code> が入力されるかどうか決定されます。
<code>pszMsg</code>	エラーに関する情報を提供するテキスト。
<code>nValue</code>	エラーに関する情報を提供する整数。
<code>next</code>	次のパスワードメッセージフィールド構造体へのポインタ。

## Sm\_PolicyApi\_PasswordPolicy\_t

SiteMinder パスワード ポリシー オブジェクトを定義します。

## 構文

```
typedef struct Sm_PolicyApi_PasswordPolicy_s
{
    int iStructId;
    char pszOid[BFSIZE];
    char pszName[BFSIZE]; /* Required */
    char pszDesc[BFSIZE];
    bool bEntireDir; /* Required */
    bool bIsEnabled; /* Required */
    char pszUserDirectoryOid[BFSIZE]; /* Required */
    char pszPath[BFSIZE]; /* Required */
    char pszClass[BFSIZE]; /* Required */
    int nResolution; /* Required */
    int nLoginMaxFailures; /* Required */
    int nLoginMaxInactivity; /* Required */
    int nLoginInactivityWarn;
    int nLoginDaysGrace; /* Required */
    char pszDictionaryName[BFSIZE];
    int nDictionaryPartial; /* Required */
    int nExpirationDelay; /* Required */
    int nReenabement; /* Required */
    int nPasswordBehavior; /* Required */
    char pszPasswordServicesRedirect[BFSIZE];
    int nPWMaxLength; /* Required */
    int nPWMinLength; /* Required */
    int nPWMaxRepeatingChar; /* Required */
    int nPWMinAlphaNum; /* Required */
    int nPWMinAlpha; /* Required */
    int nPWMinNonAlpha; /* Required */
    int nPWMinNonPrintable; /* Required */
    int nPWMinNumbers; /* Required */
    int nPWMinPunctuation; /* Required */
    int nPWReuseCount; /* Required */
    int nPWReuseDelay; /* Required */
    int nPWPercentDifferent; /* Required */
    int nPWPercentSequence; /* Required */
    int nPWSpecialsLength; /* Required */
    struct Sm_PolicyApi_PasswordPolicy_s* next;
    int nPriority; /* Required */
    int nPWMinLowerAlpha; /* Required */
    int nPWMinUpperAlpha; /* Required */
    int nReserved1;
    int nReserved2;
    int nReserved3;
}
```

```

int nReserved4[BFSIZE];
} Sm_PolicyApi_PasswordPolicy_t;

```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義されたパスワードポリシー データ構造体 ID。
<i>pszOid</i>	パスワードポリシー オブジェクトのオブジェクト識別子。
<i>pszName</i>	パスワードポリシーの名前。
<i>pszDesc</i>	パスワードポリシー オブジェクトの簡単な説明。
<i>bEntireDir</i>	フラグ：LDAP ディレクトリ全体にパスワードポリシーを適用する場合に true に設定します。
<i>blsEnabled</i>	フラグ：パスワードポリシーを有効にする場合に true に設定します。
<i>pszUserDirectoryOid</i>	パスワードポリシーを適用する対象のユーザディレクトリのオブジェクト識別子。
<i>pszPath</i>	<p><i>bEntireDir</i> が false に設定されている場合には、パスワードポリシーを適用する対象ユーザを指定するためにこのフィールドを使用します。たとえば、以下のような値を指定できます。</p> <ul style="list-style-type: none"> <li>■ 完全なユーザ DN</li> <li>■ パスワードポリシーの適用対象ユーザが含まれる組織またはグループの完全な DN</li> <li>■ パスワードポリシーの適用対象ユーザを表す検索式</li> </ul> <p>詳細については、「<i>nResolution</i>」を参照してください。</p>
<i>pszClass</i>	<p><i>bEntireDir</i> が false に設定されている場合、パスワードポリシーを適用するオブジェクトクラスを指定するためにこのフィールドを使用します。</p> <p>特定のユーザディレクトリによって認識されているオブジェクトクラスを指定します (LDAP における <i>organizationalUnit</i> や NT における <i>User</i> など)。</p>

フィールド	説明
<i>nResolution</i>	<p><i>pszPath</i> で指定されたユーザにパスワード ポリシー適用する方法を示す定数。例：</p> <ul style="list-style-type: none"> <li>■ <i>pszPath</i> がグループへのパスである場合、以下の定数は、パスワード ポリシーがグループのメンバに適用されることを示します。</li> </ul> <p><b>Sm_PolicyResolution_UserGroup</b></p> <ul style="list-style-type: none"> <li>■ <i>pszPath</i> が特定の属性が含まれるすべてのグループを検索する検索式である場合、以下の定数は、パスワード ポリシーがマッチンググループのメンバに適用されることを示します。</li> </ul> <p><b>Sm_PolicyResolution_GroupProp</b></p> <p>ポリシー解決定数は、<b>Sm_PolicyResolution_t</b> で定義されます。</p>
<i>nLoginMaxFailures</i>	ユーザ アカウントが無効になるまでにユーザが試行可能なログインの最大失敗回数。
<i>nLoginMaxInactivity</i>	ユーザ パスワードが期限切れになるまでに非アクティブ状態を継続できる日数。ユーザが期限切れ後にログインを試行するまで、アカウントは無効になりません。
<i>nLoginInactivityWarn</i>	予約済み。
<i>nLoginDaysGrace</i>	パスワードが期限切れになることを事前にユーザに通知する先行日数。
<i>pszDictionaryName</i>	パスワードで使用できない単語を一覧表示する辞書ファイルの場所。
<i>nDictionaryPartial</i>	辞書確認を適用する最小文字数。たとえば、4 文字以下の単語は確認しないようにするには、このフィールドを 5 に設定します。辞書内の単語に完全一致するパスワードのみを拒否するには、このフィールドを 0 に設定します。
<i>nExpirationDelay</i>	ユーザ アカウントが無効化されるまでに非アクティブ状態を継続できる日数。

フィールド	説明
<i>nReenablement</i>	<p>再度ログインを試行可能になるまで、またはユーザアカウントが再度有効化されるまでユーザが待つ必要がある時間を分で指定します。</p> <p><i>nReenablement</i> 期間後に適用される条件は、以下のフラグによって決定されます（このフラグは <i>nPasswordBehavior</i> で設定されます）。</p> <p><i>Sm_PasswordPolicyBehavior_FullReenable</i></p>
<i>nPasswordBehavior</i>	<p><i>Sm_PasswordPolicyBehavior_t</i> で定義されるパスワードポリシーの動作を式するビットマスクフラグ。</p> <p>以下に示す追加のパスワード動作フラグによって、パスワードポリシーの再帰的な動作を設定することもできます。 <i>pszPath</i> で指定されたグループ、およびそのグループ内でネストされたグループにパスワードポリシーが適用されることを示す場合に、このフラグを使用します。</p> <p><i>Sm_PolicyBehavior_Recursive_Yes</i></p> <p>すべてのパスワード動作フラグは、<i>SmApi.h</i> で定義されます。</p>
<i>pszPasswordServices</i> リダイレクト	<p>無効なパスワードが入力された場合に、ユーザのリダイレクト先となる URL。これはパスワードサービス CGI の URL である必要があります。</p>
<i>nPWMaxLength</i>	<p>ユーザパスワードの最大文字数。</p>
<i>nPWMinLength</i>	<p>ユーザパスワード用の最小文字数。</p>
<i>nPWMaxRepeatingChar</i>	<p>パスワード内で連続して使用可能な同一文字の最大数。</p>
<i>nPWMinAlphaNum</i>	<p>パスワードに含まれる必要がある英数文字（A～Z、a～z、または0～9）の最小数。 <i>PWMinAlpha</i> や <i>PWMinNumbers</i> と組み合わせて設定されることもあります。たとえば、 <i>PWMinAlphaNum</i> および <i>PWMinNumbers</i> が 4 に設定されている場合、パスワード 1234 は有効です。</p>

フィールド	説明
<i>nPWMinAlpha</i>	パスワードに含まれる必要があるアルファベット文字 (A ~ Z、a ~ z) の最小数。
<i>nPWMinNonAlpha</i>	パスワードに含まれる必要がある非英数文字の最小数。これらの文字としては、句読点やその他のキーボードの記号 (「@」、「\$」、「*」など) があります。
<i>nPWMinNonPrintable</i>	パスワードに含まれる必要がある非印刷可能文字の最小数。印刷不能文字は、コンピュータ画面には表示されません。
<i>nPWMinNumbers</i>	パスワードに含まれる必要がある数字 (0 ~ 9) の最小数。
<i>nPWMinPunctuation</i>	パスワードに含まれる必要がある句読点の最小数。これらの文字としては、ピリオド、カンマ、感嘆符、スラッシュ、ハイフン、ダッシュや、句読点に使用されるその他の記号などがあります。
<i>nPWReuseCount</i>	古いパスワードの再利用が可能になるまでに、使用される必要がある新しいパスワードの数。
<i>nPWReuseDelay</i>	パスワードの再使用が可能なるまでにユーザが待機する必要がある日数。
<i>nPWPercentDifferent</i>	以前のパスワードの文字と異なる文字が新しいパスワードに含まれる必要があるパーセンテージ。この値を 100 に設定した場合、 <i>nPWPercentSequence</i> を 0 に設定していない限り、以前のパスワードで使用されていた文字はすべて、新しいパスワードに含めることができないこととなります。

フィールド	説明
<i>nPWPercentSequence</i>	以前の文字のパーセンテージの計算時に、シーケンス (文字位置) を無視するかどうかを示すフラグ。文字位置を無視するには、 <i>nPWPercentSequence</i> を 1 に設定します。このフラグは、 <i>nPWPercentDifferent</i> と関係して作動します。このパラメータが <i>nPWPercentDifferent</i> とどのように関係するかの例については、150 ページの図 31 を参照してください。
<i>nPWSpecialsLength</i>	ユーザの個人情報と照合する最小の文字シーケンスを指定します。たとえば、この値が 4 に設定されている場合、 <i>SiteMinder</i> は、ユーザの個人情報に含まれる連続した 4 文字 (ユーザの電話番号の最後の 4 桁など) の使用を禁止します。 このフィールドにより、ユーザがパスワードに個人情報を組み込むことが防止されます。 <i>SiteMinder</i> は、ユーザのディレクトリ エントリ内の属性に照らしてパスワードを確認します。
<i>next</i>	次の登録方式構造体へのポインタ。
<i>nPriority</i>	複数のパスワード ポリシーが適用される場合のパスワード ポリシーの優先度。値には、負の値を含め、任意の整数を指定できます。数字が大きい方が、小さな数字よりも高い優先度になります。
<i>nPWMinLowerAlpha</i>	小文字のアルファベット文字の最小数。
<i>nPWMinUpperAlpha</i>	大文字のアルファベット文字の最小数。

## Sm\_PolicyApi\_Policy\_t

SiteMinder Policy オブジェクトを定義します。

### 構文

```
typedef struct Sm_PolicyApi_Policy_s
{
    int iStructId;
    char pszOid[BFSIZE];
    char pszName[BFSIZE];           /* Required */
    char pszDesc[BFSIZE];
    char pszActiveExpr[BFSIZE];
    bool bIsEnabled;               /* Required */
    char pszDomainOid[BFSIZE];     /* Required */
    long nStartTime;
    long nEndTime;
    unsigned char pszTimeGrid[TIMESIZE];
    Sm_PolicyApi_IPAddress_t *pIPAddress;
    struct Sm_PolicyApi_Policy_s* next;
    char pszVariableExpr[BFSIZE]
    Sm_PolicyApi_Oid_t* pVariableList;
} Sm_PolicyApi_Policy_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義されたポリシー データ構造体 ID。
<i>pszOid</i>	ポリシー オブジェクトのオブジェクト識別子。
<i>pszName</i>	ポリシーの名前
<i>pszDesc</i>	ポリシーの簡単な説明。

フィールド	説明
<i>pszActiveExpr</i>	<p>ポリシーと関連付けられたアクティブな式。この式は、以下の形式の変数定義文字列です。</p> <p>非英数文字は必須の文字です。例：</p> <pre>&lt;@lib="LibName" func="FuncName"   param="FuncParam"@&gt;</pre> <p>この形式の例において、</p> <ul style="list-style-type: none"> <li>■ <b>LibName</b> は、アクティブなポリシーをサポートする共有ライブラリの名前です。</li> <li>■ <b>FuncName</b> は、アクティブなポリシーを実装する共有ライブラリ内の実際の関数の名前です。</li> <li>■ <b>FuncParam</b> は、共有ライブラリ内の関数に渡すパラメータのリストです（オプション）。</li> </ul>
<i>blsEnabled</i>	ポリシーを有効化または無効化するフラグ。
<i>pszDomainOid</i>	ドメインのオブジェクト識別子。ドメインに固有のポリシーに必須。グローバルポリシーでは無視されます。
<i>nStartTime</i>	時間制限が有効になる時刻。この値は、標準の <code>time_t</code> 形式で保存されます。 <i>nStartTime</i> を 0 に設定すると、時間制限が即座に開始されます。
<i>nEndTime</i>	時間制限が終了する時刻。この値は、標準の <code>time_t</code> 形式で保存されます。 <i>nEndTime</i> を 0 に設定すると、時間制限が即座に終了します。
<i>pszTimeGrid</i>	1 週間全体の時間制限が含まれる配列。
<i>pIPAddress</i>	ポリシーによって管理されているリソースへのアクセスを取得するために使用する必要がある IP アドレス。
<i>next</i>	次のポリシー構造体へのポインタ。
<i>pszVariableExpr</i>	変数のタイプに対応した一意のオブジェクト識別子です。
<i>pVariableList</i>	この式によって使用される変数 <b>OID</b> のリンク リスト。



## Sm\_PolicyApi\_PolicyLink\_t

SiteMinder ポリシー リンク オブジェクトを定義します。

### 構文

```
typedef struct Sm_PolicyApi_PolicyLink_s
{
    int iStructId;
    char pszOid[BFSIZE];
    char pszRuleOid[BFSIZE];           /* Required */
    char pszResponseOid[BFSIZE];
    struct Sm_PolicyApi_PolicyLink_s* next;
} Sm_PolicyApi_PolicyLink_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義されたポリシー リンク データ構造体 ID。
<i>pszOid</i>	ポリシー リンク オブジェクトのオブジェクト識別子。
<i>pszRuleOid</i>	ルールのオブジェクト識別子。
<i>pszResponseOid</i>	レスポンスのオブジェクト識別子。
<i>next</i>	次のポリシー リンク構造体へのポインタ。

## Sm\_PolicyApi\_Realm\_t

Sm\_PolicyApi\_Realm\_t タイプは、SiteMinder レルム オブジェクトを定義します。

### 構文

```
typedef struct Sm_PolicyApi_Realm_s
{
    int iStructId;
    char pszOid[BFSIZE];
    char pszName[BFSIZE];           /* Required */
    char pszDesc[BFSIZE];
    char pszResourceFilter[BFSIZE];
    char pszAgentOid[BFSIZE];      /* Required */
    char pszSchemeOid[BFSIZE];     /* Required */
    char pszParentRealmOid[BFSIZE]; /* Required */
    char pszDomainOid[BFSIZE];     /* Required */
    char pszAzUserDirOid[BFSIZE];  /* Required */
    char pszRegSchemeOid[BFSIZE];  /* Required */
    bool bProcessAuthEvents;       /* Required */
    bool bProcessAzEvents;         /* Required */
    bool bProtectAll;              /* Required */
    int nMaxTimeout;               /* Required */
    int nIdleTimeout;              /* Required */
    bool bSyncAudit;               /* Required */
    struct Sm_PolicyApi_Realm_s* next;
} Sm_PolicyApi_Realm_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義されたレルム データ構造体 ID。
<i>pszOid</i>	レルム オブジェクトのオブジェクト識別子。
<i>pszName</i>	レルムの名前
<i>pszDesc</i>	レルムの簡単な説明。
<i>pszResourceFilter</i>	リソース フィルタのパス。
<i>pszAgentOid</i>	レルムに関連付けられたエージェントまたはエージェント グループのオブジェクト識別子。
<i>pszSchemeOid</i>	レルムに関連付けられた認証方式のオブジェクト識別子。

フィールド	説明
<i>pszParentRealmOid</i>	このレルムが追加される親レルムのオブジェクト識別子。追加されるレルムがレルム階層の最上位レルムの場合は、このフィールドをドメイン <b>OID</b> に設定します。それ以外の場合は、親レルム <b>OID</b> に設定します。
<i>pszDomainOid</i>	ドメインのオブジェクト識別子。
<i>bProcessAuthEvents</i>	認証の試行イベントでこのルールをトリガするブール。 認証イベント処理はパフォーマンスに影響します。レルム内のルールが認証イベントによってトリガされない場合は、このフィールドを <b>false</b> に設定します。
<i>bProcessAzEvents</i>	認可の試行イベントでこのルールをトリガするブール。 認可イベント処理はパフォーマンスに影響します。レルム内のルールが認可イベントによってトリガされない場合は、このフィールドを <b>false</b> に設定します。
<i>bProtectAll</i>	新しいレルムに含まれているリソースをすべて保護するブール。
<i>nMaxTimeout</i>	再認証が必要となる前に、ユーザがレルム内の保護されたリソースにアクセスできる最大時間。
<i>nIdleTimeout</i>	再認証が必要となる前に、リソースとやり取りすることなく、ユーザがレルムの保護されたリソースに対して認証された状態を維持できる時間長。
<i>bSyncAudit</i>	同期監査を有効にするブール。有効化されている場合、ユーザのアクティビティがポリシー サーバおよび <b>Web</b> エージェントの両方の監査ログに正常に記録されるまで、そのユーザはレルム内のリソースにアクセスできません。
<i>pszAzUserDirOid</i>	このレルム内のリソースにアクセスするユーザが認可されるディレクトリの <b>OID</b> 。
<i>pszRegSchemeOid</i>	このレルムのリソースにアクセスする新規ユーザの登録に使用される登録方式の <b>OID</b> 。
<i>next</i>	次のレルム構造体へのポインタ。

## Sm\_PolicyApi\_RegistrationScheme\_t

SiteMinder 登録方式オブジェクトを定義します。

## 構文

```
typedef struct Sm_PolicyApi_RegistrationScheme_s
{
    int iStructId;
    char pszOid[BFSIZE];
    char pszName[BFSIZE];           /* Required */
    char pszDesc[BFSIZE];
    char pszUserDirOid[BFSIZE];     /* Required */
    char pszWelcomePageURL[BFSIZE];
    char pszTemplatePath[BFSIZE];
    bool bEnableLogging;           /* Required */
    struct Sm_PolicyApi_RegistrationScheme_s* next;
} Sm_PolicyApi_RegistrationScheme_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義された登録方式データ構造体 ID。
<i>pszOid</i>	登録方式オブジェクトのオブジェクト識別子。
<i>pszName</i>	登録方式の名前です。
<i>pszDesc</i>	登録方式の簡単な説明。
<i>pszUserDirOid</i>	ユーザ情報が保存されている LDAP ユーザディレクトリのオブジェクト識別子。
<i>pszWelcomePageURL</i>	登録して正常に認証されたユーザの歓迎に使用されるフォームの場所。
<i>pszTemplatePath</i>	登録テンプレートの場所。
<i>bEnableLogging</i>	登録情報をログ記録するかどうかを示すフラグ。ログ記録を有効にするには、このパラメータを true に設定します。
<i>next</i>	次の登録方式構造体へのポインタ。

## Sm\_PolicyApi\_RegularExpression\_t

正規表現を定義します。

### 構文

```
typedef struct Sm_PolicyApi_RegularExpression_s
{
    int iStructId;
    char pszOid[BFSIZE];
    char pszName[BFSIZE];
    char ExpressionDef[BFSIZE];
    bool matchExpression;
    struct Sm_PolicyApi_RegularExpression_s *next;
} Sm_PolicyApi_RegularExpression_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義された正規表現構造体 ID。
<i>pszOid</i>	正規表現オブジェクトのオブジェクト識別子。
<i>pszName</i>	正規表現の名前。
<i>ExpressionDef</i>	正規表現の定義。
<i>matchExpression</i>	1 : パスワードがこの式に一致する必要がある場合 0 : パスワードがこの式に一致してはならない場合
<i>next</i>	次の正規表現構造体へのポインタ。

## Sm\_PolicyApi\_Response\_t

SiteMinder レスポンス オブジェクトを定義します。

### 構文

```
typedef struct Sm_PolicyApi_Response_s
{
    int iStructId;
    char pszOid[BFSIZE];
    char pszName[BFSIZE];           /* Required */
    char pszDesc[BFSIZE];
    char pszAgentTypeOid[BFSIZE];  /* Required */
    char pszDomainOid[BFSIZE];     /* Required */
    struct Sm_PolicyApi_Response_s* next;
} Sm_PolicyApi_Response_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義されたレスポンス データ構造体 ID。
<i>pszOid</i>	レスポンス オブジェクトのオブジェクト識別子。
<i>pszName</i>	レスポンスの名前。
<i>pszDesc</i>	レスポンスの簡単な説明。
<i>pszAgentTypeOid</i>	エージェントタイプのオブジェクト識別子。
<i>pszDomainOid</i>	ドメインのオブジェクト識別子。ドメインに固有のレスポンスに必須。グローバル レスポンスでは無視されます。
<i>next</i>	次のレスポンス構造体へのポインタ。

## Sm\_PolicyApi\_ResponseAttr\_t

SiteMinder レスポンス属性オブジェクトを定義します。

## 構文

```
typedef struct Sm_PolicyApi_ResponseAttr_s
{
    int iStructId;
    char pszOid[BFSIZE];
    char pszAgentTypeAttrOid [BFSIZE];      /* Required */
    char pszValue[BFSIZE];
    int iTTL;                               /* Required */
    int iFlags;
    struct Sm_PolicyApi_ResponseAttr_s* next;
    char pszActiveExpr[BFSIZE];
    Sm_PolicyApi_Oid_t* pVariableList;
} Sm_PolicyApi_ResponseAttr_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義されたレスポンス属性データ構造体 ID。
<i>pszOid</i>	レスポンス属性オブジェクトのオブジェクト識別子。
<i>pszAgentTypeAttrOid</i>	エージェントタイプ属性のオブジェクト識別子。
<i>pszValue</i>	レスポンス属性タイプ（「レスポンス属性のタイプ」参照）。
<i>iTTL</i>	レスポンス属性の値が再計算されるまでの経過時間（秒）。
<i>iFlags</i>	予約済み。
<i>next</i>	次のレスポンス属性構造体へのポインタ。
<i>pszActiveExpr</i>	レスポンス属性に関連付けられたアクティブな式。詳細については、「レスポンス属性のタイプ」の項目「アクティブレスポンス」を参照してください。
<i>pVariableList</i>	アクティブな式によって使用される変数 OID のリンクリスト。

## レスポンス属性のタイプ

*pszValue* フィールドには、以下のレスポンス属性タイプのいずれかが含まれます。

注: 以下の形式に含まれる非英数文字は必須の文字です。

- **スタティック**。SiteMinder レスポンスの一部の文字列です。文字列の形式は以下のとおりです。

`variable-name=variable-value`

この形式の例において、

- *variable-name* は、このレスポンス属性が Web エージェントに返す名前/値ペアの名前です。
  - *variable-value* は、名前/値ペアの後半として返されるスタティックテキストです。
- **ユーザ属性**。ユーザディレクトリ内のユーザのエントリからのプロフィール情報が含まれる文字列です。文字列の形式は以下のとおりです。

`User-Attr-variable-name=<%userattr="user-attr-name"%>`

この形式の例において、

- *User-Attr-variable-name* は、このレスポンス属性が Web エージェントに返す名前/値ペアの名前です。
  - *user-attr-name* は、LDAP、WinNT、または ODBC ユーザディレクトリから取得できるユーザ属性です。
- **DN 属性**。LDAP または ODBC のユーザディレクトリ内のディレクトリオブジェクトからのプロフィール情報が含まれる文字列です。文字列の形式は以下のとおりです。

`DN-Variable-Name=<#dn="DN-Spec" attr="DN-Attribute-Name"#>`

この形式の例において、

- *DN-Variable-Name* は、このレスポンス属性が Web エージェントに返す名前/値ペアの名前です。
- *DN-Spec* は、ユーザ属性の取得先のユーザグループの識別名です。
- *DN-Attribute-Name* は、グループや組織単位 (OU) のようにユーザが関係する LDAP または ODBC ディレクトリオブジェクトに関連付けられた属性です。

- **アクティブ レスポンス**。アクティブな式は、レスポンス属性に関連付けられています。式は、以下の形式の変数定義の文字列です。

```
Name=<@lib="LibName" func="FuncName" param="Param"@>
```

この形式の例において、

- **Name** は、アクティブな式の名前/値ペアと関連付けられた変数（WebAgent-HTTP-Header-Variable レスポンス属性と関連付けている）または Cookie（WebAgent-HTTP-Cookie-Variable レスポンス属性と関連付けている）の名前です。
- **LibName** は、アクティブなレスポンスをサポートする共有ライブラリの名前です。
- **FuncName** は、アクティブなレスポンスを実装する共有ライブラリ内の実際の関数の名前です。
- **Param** 共有ライブラリの関数に渡すパラメータのリストです（オプション）。

**注:** レスポンス、ルール、またはポリシーにおけるアクティブな式の設定の詳細については、「ポリシー デザイン」を参照してください。

## Sm\_PolicyApi\_Rule\_t

SiteMinder ルール オブジェクトを定義します。

### 構文

```
typedef struct Sm_PolicyApi_Rule_s
{
    int iStructId;
    char pszOid[BFSIZE];
    char pszName[BFSIZE];           /* Required */
    char pszDesc[BFSIZE];
    char pszRealmOid[BFSIZE];     /* Required */
    char pszAction[BFSIZE];       /* Required */
    char pszResource[2*BFSIZE];
    bool bAllowAccess;           /* Required */
    bool bRegularExpression;    /* Required */
    char pszActiveExpr[BFSIZE];
    bool bIsEnabled;            /* Required */
    long nStartTime;
    long nEndTime;
    unsigned char pszTimeGrid[TIMESIZE];
    struct Sm_PolicyApi_Rule_s* next;
    char pszAgentOid[BFSIZE];     /* Required */
} Sm_PolicyApi_Rule_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義されたルール データ構造体 ID。
<i>pszOid</i>	ルール オブジェクトのオブジェクト識別子。
<i>pszName</i>	ルールの名前。
<i>pszDesc</i>	ルールの簡単な説明。
<i>pszRealmOid</i>	ルールに関連付けられたレルムのオブジェクト識別子。ドメインに固有のルールに必須。グローバルルールでは無視されます。
<i>pszAction</i>	ルールが実行しているアクションのタイプ。サポートされる Web エージェントアクションは、次の HTTP 操作から構成されます： Get、Put、および Post
<i>pszResource</i>	ルールによって保護されたリソース。

フィールド	説明
<i>bAllowAccess</i>	ルールによって保護されたリソースへのアクセスを許可または拒否するフラグ。
<i>bRegularExpression</i>	正規表現パターン マッチングを実行するフラグ。正規表現は、文字列照合に使用するテキストパターンです。
<i>pszActiveExpr</i>	<p>ルールと関連付けられたアクティブな式。式は、以下の形式の変数定義の文字列です。</p> <p>The non-alphanumeric characters are required characters .For example:  <code>&lt;@lib="LibName" func="FuncName" param="FuncParam"@&gt;</code></p> <p>この形式の例において、</p> <ul style="list-style-type: none"> <li>■ <b>LibName</b> は、アクティブなルールをサポートする共有ライブラリの名前です。</li> <li>■ <b>FuncName</b> は、アクティブなルールを実装する共有ライブラリ内の実際の関数の名前です。</li> <li>■ <b>FuncParam</b> は、共有ライブラリ内の関数に渡すパラメータのリストです（オプション）。</li> </ul>
<i>dbIsEnabled</i>	ルールを有効化または無効化するフラグ。
<i>nStartTime</i>	時間制限が有効になる時刻。この値は、標準の <code>time_t</code> 形式で保存されます。 <i>nStartTime</i> を 0 に設定すると、時間制限が即座に開始されます。
<i>nEndTime</i>	時間制限が終了する時刻。この値は、標準の <code>time_t</code> 形式で保存されます。 <i>nEndTime</i> を 0 に設定すると、時間制限が即座に終了します。
<i>pszTimeGrid</i>	1 週間全体の時間制限が含まれる配列。
<i>next</i>	次のルール構造体へのポインタ。
<i>pszAgentOid</i>	グローバル ルールに関連付けられたエージェントまたはエージェント グループのオブジェクト識別子。

## Sm\_PolicyApi\_SAMLAffiliation\_t

SAML 2.0 アフィリエーションオブジェクトを定義します。SAML 2.0 アフィリエーションは、プリンシパル用の一意の名前 ID の単一のフェデレートされたネームスペースを共有する一連のエンティティです。

### 構文

```
typedef struct Sm_PolicyApi_SAMLAffiliation_s
{
    int iStructId;
    Sm_PolicyApi_SAMLProviderProp_t *pProps;
    struct Sm_PolicyApi_SAMLAffiliation_s *next;
} Sm_PolicyApi_SAMLAffiliation_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義された SAML アフィリエーション構造体 ID。
<i>pProps</i>	アフィリエーションに関連付けられた SAML 2.0 メタデータプロパティ。 デフォルト値に関連付けられたプロパティに値を割り当てない場合、デフォルトが割り当てられます。
<i>next</i>	次の SAML アフィリエーション構造体へのポインタ。

## Sm\_PolicyApi\_SAMLProviderProp\_t

SAML 2.0 メタデータ プロパティを名前/値ペアとして定義します。

Sm\_PolicyApi\_SAMLProviderProp\_t 構造体は、単一の名前/値ペアから構成されます。Sm\_PolicyApi\_SAMLProviderProp\_t 構造体のリンクリストを使用して、所定の SAML 2.0 オブジェクトの一連のプロパティを定義します。

以下の構造体および関数を使用して、SAML 2.0 サービスプロバイダ、アプリケーション、または認証方式および関連付けられたアイデンティティプロバイダに関する一連のプロパティを定義します。

- Sm\_PolicyApi\_SAMLSP\_t
- Sm\_PolicyApi\_SAMLAffiliation\_t
- Sm\_PolicyApi\_AddSAMLScheme()

### 構文

```
typedef struct Sm_PolicyApi_SAMLProviderProp_s
{
    int iStructId;
    char pszName[BFSIZE];
    char pszValue[BFSIZE];
    struct Sm_PolicyApi_SAMLProviderProp_s *next;
} Sm_PolicyApi_SAMLProviderProp_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義された SAML 2.0 プロパティ構造体 ID。
<i>pszName</i>	プロパティ名。有効なプロパティ名のリストについては、「プロパティリスト」を参照してください。
<i>pszValue</i>	<i>pszName</i> に割り当てる値。
<i>next</i>	次の SAML 2.0 プロパティ構造体へのポインタ。

## 例

`Sm_PolicyApi_SAMLProviderProp_t` のインスタンスを作成し、渡された名前/値ペアをこれに割り当てるヘルパー メソッドの例を以下に示します。

```
Sm_PolicyApi_SAMLProviderProp_t * CreateSAMLProp(
    const char *pszName,
    const char *pszValue)
{
    Sm_PolicyApi_SAMLProviderProp_t * pProp = new
        Sm_PolicyApi_SAMLProviderProp_t;
    memset(pProp, 0, sizeof(Sm_PolicyApi_SAMLProviderProp_t));
    pProp->iStructId = Sm_PolicyApi_SAMLProviderProp_ID;
    strncpy(pProp->pszName, pszName, BFSIZE);
    strncpy(pProp->pszValue, pszValue, BFSIZE);
    return pProp;
}
```

渡された名前を持つプロパティの値を取得するヘルパー メソッドの例を以下に示します。

```
char * FindSAMLParam(const Sm_PolicyApi_SAMLSP_t *pStructSP,
    const char *pszName)
{
    for(Sm_PolicyApi_SAMLProviderProp_t *pCurProp=pStructSP->pProps;
        pCurProp != NULL;
        pCurProp = pCurProp->next)
    {
        if (0 == strcmp(pCurProp->pszName, pszName))
        {
            return pCurProp->pszValue;
        }
    }

    return NULL;
}
```

## プロパティリスト

それぞれの `Sm_PolicyApi_SAMLProviderProp_t` 構造には、名前/値ペアとして定義された SAML 2.0 メタデータ プロパティが含まれています。特定の SAML 2.0 オブジェクトに関するプロパティの完全なセットは、`Sm_PolicyApi_SAMLProviderProp_t` 構造体のリンク リストとして定義されます。

以下のメタデータ プロパティは、SAML 2.0 オブジェクトの所定のタイプに適用されます。

- サービスプロバイダプロパティ
- SAMLアフィリエーションプロパティ
- SAML 2.0 認証方式プロパティ

注: いくつかのプロパティは、複数のオブジェクトタイプと共に使用されます。

#### サービスプロバイダプロパティ

サービスプロバイダオブジェクトには、サービスプロバイダ用のアサーションを生成するためにアイデンティティプロバイダが必要とする情報が含まれています。サービスプロバイダプロパティは、`Sm_PolicyApi_SAMLSP_t` 構造体内に保存されています。

SAML 2.0 サービスプロバイダオブジェクトのプロパティの一覧を以下に示します。

プロパティ名	コメント
<i>General Properties</i>	
SAML_NAME	文字列 (必須)
SAML_DESCRIPTION	文字列
SAML_SP_AUTHENTICATION_URL	文字列 (必須)
SAML_ENABLED	ブール値 デフォルト: SAML_TRUE
SAML_SP_DOMAIN	文字列 (必須)
<i>[名前ID] タブ</i>	
SAML_SP_NAMEID_FORMAT	文字列 デフォルト: 未指定
SAML_SP_NAMEID_TYPE	整数  デフォルト: 1
SAML_SP_NAMEID_STATIC	文字列 条件付きで必須

プロパティ名	コメント
SAML_SP_NAMEID_ATTRNAME	文字列 条件付きで必須
SAML_SP_NAMEID_DNSPEC	文字列 条件付きで必須
SAML_AFFILIATION	文字列
SAML_KEY_SPID	文字列 (必須)
	文字列 (必須)
SAML_MAJOR_VERSION	整数 デフォルト : 2
SAML_MINOR_VERSION	整数 デフォルト : 0
SAML_SKEWTIME	整数 デフォルト : 30
SAML_DISABLE_SIGNATURE_PROCESSING	ブール値 デフォルト : SAML_FALSE
SAML_DSIG_VERINFO_ISSUER_DN	文字列 条件付きで必須
SAML_DSIG_VERINFO_SERIAL_NUMBER	文字列 条件付きで必須
<i>SSO プロパティ</i>	
SAML_AUDIENCE	文字列 (必須)
SAML_SP_ASSERTION_CONSUMER_DEFAULT_URL	文字列 (必須)
SAML_ENABLE_SSO_ARTIFACT_BINDING	ブール値 デフォルト : SAML_FALSE
SAML_SP_ARTIFACT_ENCODING	文字列 デフォルト : FORM 値が提供されず、SAML_ENABLE_SSO_ARTIFACT_BINDING が SAML_TRUE の場合に適用されます。

プロパティ名	コメント
SAML_SP_IDP_SOURCEID	文字列 デフォルト: SAML_SP_IDPID 値の 16 進エンコードされた SHA-1 ハッシュ
SAML_SP_PASSWORD	文字列 条件付きで必須 (34 ページを参照)
SAML_ENABLE_SSO_POST_BINDING	ブール値 デフォルト: SAML_FALSE
SAML_SSOECPPROFILE	ブール値 デフォルト: SAML_FALSE
SAML_SP_REQUIRE_SIGNED_AUTHNREQUESTS	ブール値 デフォルト: SAML_FALSE
SAML_SP_AUTHENTICATION_LEVEL	整数 デフォルト: 5
SAML_SP_AUTHN_CONTEXT_CLASS_REF	文字列 デフォルト: urn:oasis:names:tc:SAML:2.0:ac:classes:Password
SAML_SP_VALIDITY_DURATION	整数 デフォルト: 60
SAML_SP_STARTTIME	long
SAML_SP_ENDTIME	long
<i>SLO</i> プロパティ	
SAML_SLO_REDIRECT_BINDING	ブール値 デフォルト: SAML_FALSE
SAML_SLO_SERVICE_VALIDITY_DURATION	整数 デフォルト: 60 値が提供されず、SAML_SLO_REDIRECT_BINDING が SAML_TRUE の場合に適用されます。
SAML_SLO_SERVICE_URL	文字列 条件付きで必須

プロパティ名	コメント
SAML_SLO_SERVICE_RESPONSE_URL	文字列
SAML_SLO_SERVICE_CONFIRM_URL	文字列
<i>IPD プロパティ</i>	
SAML_SP_ENABLE_IPD	ブール値 デフォルト : SAML_FALSE
SAML_SP_IPD_SERVICE_URL	文字列 条件付きで必須
SAML_SP_COMMON_DOMAIN	文字列 条件付きで必須
SAML_SP_PERSISTENT_COOKIE	ブール値 デフォルト : SAML_FALSE
<i>暗号化プロパティ</i>	
SAML_SP_ENCRYPT_ID	ブール値 デフォルト : SAML_FALSE
SAML_SP_ENCRYPT_ASSERTION	ブール値 デフォルト : SAML_FALSE
SAML_SP_ENCRYPT_BLOCK_ALGO	文字列  デフォルト : triledes
SAML_SP_ENCRYPT_KEY_ALGO	文字列  デフォルト : rsa-v15
SAML_SP_ENCRYPT_CERT_ISSUER_DN	文字列 条件付きで必須
SAML_SP_ENCRYPT_CERT_SERIAL_NUMBER	文字列 条件付きで必須
<i>属性サービス プロパティ</i>	
SAML_SP_ATTRSVS_ENABLE	ブール値 デフォルト : False (0)。

プロパティ名	コメント
SAML_SP_ATTRSVS_VALIDITY_DURATION	整数 デフォルト：60（秒）
SAML_SP_ATTRSVS_SIGN_ASSERTION	ブール値 デフォルト：False（0）。
SAML_SP_ATTRSVS_LDAP_SEARCH_SPEC	文字列
SAML_SP_ATTRSVS_ODBC_SEARCH_SPEC	文字列
SAML_SP_ATTRSVS_WINNT_SEARCH_SPEC	文字列
SAML_SP_ATTRSVS_CUSTOM_SEARCH_SPEC	文字列
SAML_SP_ATTRSVS_AD_SEARCH_SPEC	文字列
詳細プロパティ	
SAML_SP_PLUGIN_CLASS	文字列
SAML_SP_PLUGIN_PARAMS	文字列

### SAML アフィリエーションプロパティ

SAML 2.0 アフィリエーションオブジェクトのプロパティを以下に示します。プロパティは、[SAML アフィリエーションプロパティ] ダイアログボックスに表示される方法に基づいてグループ分けされています。

SAML アフィリエーションプロパティは、Sm\_PolicyApi\_SAMLAffiliation\_t 構造体内に保存されています。

プロパティ名	コメント
General Properties	
SAML_NAME	文字列（必須）
SAML_DESCRIPTION	文字列
SAML_KEY_AFFILIATION_ID	文字列（必須）
SAML_MAJOR_VERSION	整数 デフォルト：2

プロパティ名	コメント
SAML_MINOR_VERSION	整数 デフォルト： 0
SAML_OID	文字列 アフィリエーションオブジェクトが作成されると、SiteMinder がオブジェクト識別子を提供します。
[名前 ID] タブ	
SAML_SP_NAMEID_FORMAT	文字列  デフォルト： 未指定
SAML_SP_NAMEID_TYPE	整数  デフォルト： 1
SAML_SP_NAMEID_STATIC	文字列 条件付きで必須
SAML_SP_NAMEID_ATTRNAME	文字列 条件付きで必須
SAML_SP_NAMEID_DNSPEC	文字列 条件付きで必須
[ユーザ] タブ	
SAML_IDP_XPATH	文字列
SAML_IDP_LDAP_SEARCH_SPEC	文字列
SAML_IDP_ODBC_SEARCH_SPEC	文字列
SAML_IDP_WINNT_SEARCH_SPEC	文字列
SAML_IDP_CUSTOM_SEARCH_SPEC	文字列
SAML_IDP_AD_SEARCH_SPEC	文字列

### SAML 2.0 認証方式プロパティ

このセクションで示すプロパティは、以下の定義を行います。

- SAML 2.0 テンプレートに基づく認証方式。
- 関連するアイデンティティプロバイダのメタデータ プロパティ。プロパティは、認証方式と共に保存されています。

特定のサービスプロバイダに対応するアイデンティティプロバイダを表すには、SAML 2.0 認証方式を定義します。

SAML 2.0 認証方式およびその関連するアイデンティティプロバイダのプロパティを以下に示します。プロパティは、SAML 2.0 テンプレートの [SAML 認証方式プロパティ] ダイアログボックスに表示される方法に基づいてグループ分けされています。

Sm\_PolicyApi\_AddSAMLScheme() をコールすることにより、SAML 2.0 認証方式およびその関連するアイデンティティプロバイダのプロパティを定義します。

プロパティ名	コメント
General Properties	
SAML_NAME	文字列 (必須)
SAML_DESCRIPTION	文字列
[方式のセットアップ] タブ	
SAML_IDP_SPID	文字列 (必須)
SAML_KEY_IDPID	文字列 (必須)
SAML_MAJOR_VERSION	整数 デフォルト : 2
SAML_MINOR_VERSION	整数 デフォルト : 0
SAML_SKEWTIME	整数 デフォルト : 30
SAML_DISABLE_SIGNATURE_PROCESSING	ブール値 デフォルト : SAML_FALSE

プロパティ名	コメント
SAML_DSIG_VERINFO_ISSUER_DN	文字列 条件付きで必須 (9 ページを参照)
SAML_DSIG_VERINFO_SERIAL_NUMBER	文字列 条件付きで必須 (10 ページを参照)
追加設定 -- [ユーザ] タブ	
SAML_IDP_XPATH	文字列
SAML_IDP_LDAP_SEARCH_SPEC	文字列
SAML_IDP_ODBC_SEARCH_SPEC	文字列
SAML_IDP_WINNT_SEARCH_SPEC	文字列
SAML_IDP_CUSTOM_SEARCH_SPEC	文字列
SAML_IDP_AD_SEARCH_SPEC	文字列
SAML_AFFILIATION	文字列
追加設定 -- [SSO] タブ	
SAML_IDP_SSO_REDIRECT_MODE	整数 デフォルト : 0
SAML_IDP_SSO_DEFAULT_SERVICE	文字列 (必須)
SAML_AUDIENCE	文字列 (必須)
SAML_IDP_SSO_TARGET	文字列
SAML_ENABLE_SSO_ARTIFACT_BINDING	ブール値 デフォルト : SAML_FALSE
SAML_KEY_IDP_SOURCEID	文字列 デフォルト : SAML_KEY_IDPID 値の 16 進エンコードされた SHA-1 ハッシュ
SAML_IDP_ARTIFACT_RESOLUTION_DEFAULT_SERVICE	文字列 条件付きで必須 (12 ページを参照)
SAML_IDP_BACKCHANNEL_AUTH_TYPE	整数 デフォルト : 0

プロパティ名	コメント
SAML_IDP_SPNAME	文字列 条件付きで必須 (19 ページを参照)
SAML_IDP_PASSWORD	文字列 条件付きで必須 (14 ページを参照)
SAML_ENABLE_SSO_POST_BINDING	ブール値 デフォルト : SAML_FALSE
SAML_IDP_SSO_ENFORCE_SINGLE_USE_POLICY	ブール値 デフォルト : SAML_TRUE
SAML_SSOECPPROFILE	ブール値 デフォルト : SAML_FALSE
SAML_IDP_SIGN_AUTHNREQUESTS	ブール値 デフォルト : SAML_FALSE
追加設定 -- [SLO] タブ	
SAML_SLO_REDIRECT_BINDING	ブール値 デフォルト : SAML_FALSE
SAML_SLO_SERVICE_VALIDITY_DURATION	整数 デフォルト : 60 値が提供されず、SAML_SLO_REDIRECT_BINDING が SAML_TRUE の場合に適用されます。
SAML_SLO_SERVICE_URL	文字列 条件付きで必須
SAML_SLO_SERVICE_RESPONSE_URL	文字列
SAML_SLO_SERVICE_CONFIRM_URL	文字列
追加設定 -- [暗号化] タブ	
SAML_IDP_REQUIRE_ENCRYPTED_ASSERTION	ブール値 デフォルト : SAML_FALSE
SAML_IDP_REQUIRE_ENCRYPTED_NAMEID	ブール値 デフォルト : SAML_FALSE
追加設定 -- [属性] タブ	

プロパティ名	コメント
SAML_IDP_SAMLREQ_ENABLE	ブール値 デフォルト : False (0)。
SAML_IDP_SAMLREQ_REQUIRE_SIGNED_ASSERTION	ブール値 デフォルト : False (0)。
SAML_IDP_SAMLREQ_ATTRIBUTE_SERVICE	文字列
追加設定 -- [NameID] タブ	
SAML_IDP_SAMLREQ_NAMEID_FORMAT	文字列
SAML_IDP_SAMLREQ_NAMEID_TYPE	整数 デフォルト : 1 (ユーザ属性)
SAML_IDP_SAMLREQ_NAMEID_STATIC	文字列
SAML_IDP_SAMLREQ_NAMEID_ATTR_NAME	文字列
SAML_IDP_SAMLREQ_NAMEID_DN_SPEC	文字列
SAML_IDP_SAMLREQ_NAMEID_ALLOW_NESTED	ブール値 デフォルト : False (0)。
追加設定 -- [詳細] タブ	
SAML_SP_PLUGIN_CLASS	文字列
SAML_SP_PLUGIN_PARAMS	文字列
SAML_IDP_REDIRECT_URL_USER_NOT_FOUND	文字列
SAML_IDP_REDIRECT_MODE_USER_NOT_FOUND	整数 デフォルト : 0
SAML_IDP_REDIRECT_URL_FAILURE	文字列
SAML_IDP_REDIRECT_MODE_FAILURE	整数 デフォルト : 0
SAML_IDP_REDIRECT_URL_INVALID	文字列
SAML_IDP_REDIRECT_MODE_INVALID	整数 デフォルト : 0

詳細情報:

[SAML 2.0 プロパティ参照 \(P. 886\)](#)

## Sm\_PolicyApi\_SAMLRequesterAttr\_t

AttributeQuery メッセージ内の SAML リクエストによって要求可能な属性を定義します。

### 構文

```
typedef struct Sm_PolicyApi_SAMLRequesterAttr_s
{
    int iStructId;
    Sm_PolicyApi_SAMLSAttrNameFormat_t nAttrNameFormat;
    char pszLocalName[BUFSIZE];
    char pszName[BUFSIZE];
    struct Sm_PolicyApi_SAMLRequesterAttr_s* next;
} Sm_PolicyApi_SAMLRequesterAttr_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義された SAML 属性構造体 ID。
<i>nAttrNameFormat</i>	SAML 2.0 標準によって定義された属性名の形式。
<i>pszLocalName</i>	SAML 2.0 認証方式で定義された属性の名前。
<i>pszName</i>	属性認証機関から要求された属性の実際の名前。
<i>next</i>	次のリクエスト属性構造体へのポインタ。

## Sm\_PolicyApi\_SAMLSP\_t

アイデンティティプロバイダの SAML 2.0 サービスプロバイダオブジェクトを定義します。

サービスプロバイダは、フェデレーション内のプリンシパルにサービス（アプリケーションや他のリソースへのアクセスなど）を提供します。

### 構文

```
typedef struct Sm_PolicyApi_SAMLSP_s
{
    int iStructId;
    Sm_PolicyApi_SAMLProviderProp_t *pProps;
    Sm_PolicyApi_IPAddress_t *pIPAddress;
    unsigned char pszTimeGrid[TIMESIZE];
    struct Sm_PolicyApi_SAMLSP_s *next;
} Sm_PolicyApi_SAMLSP_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義された SAML サービスプロバイダ構造体 ID。
<i>pProps</i>	サービスプロバイダに関連付けられた SAML 2.0 メタデータプロパティ。 デフォルト値に関連付けられたプロパティに値を割り当てない場合、デフォルトが割り当てられます。
<i>pIPAddress</i>	サービスプロバイダの IP アドレス。
<i>pszTimeGrid</i>	1 週間全体の時間制限が含まれる配列。
<i>next</i>	次のサービスプロバイダ構造体へのポインタ。

## Sm\_PolicyApi\_SAMLSPAssertionConsumerService\_t

アサーション コンシューマ サービスをサポートする構造体を定義します。

## 構文

```
typedef struct Sm_PolicyApi_SAMLSPAssertionConsumerService_s
{
    int iStructId;
    int iIndex;
    Sm_PolicyApi_SAMLSPACSBinding_t nACSBinding;
    char pszAssertionConsumerURL[BFSIZE];
    bool bIsDefault;
    struct Sm_PolicyApi_SAMLSPAssertionConsumerService_s* next;
} Sm_PolicyApi_SAMLSPAsserttionConsumer_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義された SAML アサーション コンシューマ サービス ID。
<i>iIndex</i>	このアサーション コンシューマ サービスに割り当てられたインデックス値。値は、ゼロ (0) または正の整数であることが必要です。
<i>nACSBinding</i>	アサーション コンシューマ サービスに関連付けられた以下のバインディングのいずれか。 <ul style="list-style-type: none"> <li>■ Sm_PolicyApi_SAMLSP_HTTP_Post</li> <li>■ Sm_PolicyApi_SAMLSP_HTTP_Artifact</li> <li>■ Sm_PolicyApi_SAMLSP_PAOS</li> </ul>
<i>pszAssertionConsumerURL</i>	アサーション コンシューマ サービスの場所。 HTTP-Artifact バインディングの場合、この URL に SAML Artifact およびターゲットがクエリ パラメータとして含まれます。これらのクエリ パラメータは、認証情報コレクタが SAML アサーションを取得し、ユーザをターゲットにリダイレクトするために使用されます。 HTTP-Post バインディングの場合、この URL は、ユーザのブラウザが生成済みアサーションをポストする必要がある送信先サイト URL となります。
<i>bIsDefault</i>	このアサーション コンシューマ サービスがデフォルトかどうかを指定します。

フィールド	説明
<i>next</i>	次のアサーション コンシューマ サービスの定義へのポインタ。

## Sm\_PolicyApi\_SAMLSPAttr\_t

特定の SAML 2.0 サービス プロバイダのプリンシパルの属性を定義します。

### 構文

```
typedef struct Sm_PolicyApi_SAMLSPAttr_s
{
    int iStructId;
    Sm_PolicyApi_SAMLSPAttrNameFormat_t nAttrNameFormat;
    char pszValue[BFSIZE];
    bool bEncrypted;
    struct Sm_PolicyApi_SAMLSPAttr_s* next;
    Sm_PolicyApi_SAMLSPAttrMode_t nMode;
} Sm_PolicyApi_SAMLSPAttr_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義された SAML 属性構造体 ID。
<i>nAttrNameFormat</i>	SAML 2.0 標準によって定義された属性名の形式。
<i>pszValue</i>	属性の名前および値。以下の形式のいずれかです。 <ul style="list-style-type: none"> <li>■ スタティック属性 : variableName=value</li> <li>■ ユーザ属性 : variableName=&lt;%userattr="AttrName"%&gt;</li> <li>■ DN 属性 : variableName=&lt;#dn="DNSpec" attr="AttrName"#&gt;</li> </ul> SiteMinder がネストされたグループから DN 属性を取得できるようにするには、 <i>DNSpec</i> の最初に感嘆符 (!) を付けます。例 : dn="!ou=People,o=security.com"
<i>bEncrypted</i>	属性が暗号化されるかどうかを指定します。

フィールド	説明
<i>next</i>	次の属性構造体へのポインタ。
<i>nMode</i>	以下の 3 つの属性の用途のいずれか。 <ul style="list-style-type: none"> <li>■ Sm_PolicyApi_SAMLSP_SSO_Only = 0</li> <li>■ Sm_PolicyApi_SAMLSP_Attribute_Only = 1</li> <li>■ Sm_PolicyApi_SAMLSP_Both = 2</li> </ul>

## Sm\_PolicyApi\_Scheme\_t

SiteMinder 認証方式オブジェクトを定義します。

### 構文

```
typedef struct Sm_PolicyApi_Scheme_s
{
    int iStructId;
    char pszOid[BFSIZE];
    char pszName[BFSIZE];           /* Required */
    char pszDesc[BFSIZE];
    int nLevel;                     /* Required */
    char pszLib[BFSIZE];
    char pszParam[BFSIZE];
    char pszSecret[BFSIZE];
    bool bIsTemplate;
    bool bIsUsedbyAdmin;
    Sm_Api_SchemeType_t nType;     /* Required */
    bool bAllowSaveCreds;          /* Required */
    bool bIsRadius;                /* Required */
    bool bIgnorePwCheck;           /* Required */
    struct Sm_PolicyApi_Scheme_s* next;
} Sm_PolicyApi_Scheme_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義された認証方式データ構造体 ID。
<i>pszOid</i>	認証方式オブジェクトのオブジェクト識別子。
<i>pszName</i>	認証方式の名前です。
<i>pszDesc</i>	認証方式の簡単な説明。

フィールド	説明
<i>nLevel</i>	認証方式の保護レベル。レベルは <b>1 ~ 1000</b> の間で変更できます。数値が高いほど、方式の保護レベルが高くなります。匿名認証方式では、この値を <b>0</b> に設定します。
<i>pszLib</i>	カスタム認証方式を実装する共有ライブラリの名前。
<i>pszParam</i>	カスタム認証方式に渡される情報。
<i>pszSecret</i>	セキュアなデータ交換の確立を目的として <b>2</b> つのパーティが既知な共有秘密キー。この情報はカスタム認証方式に渡されます。
<i>blsTemplate</i>	認証方式がテンプレートかどうかを示すフラグ。 <b>注:</b> C ポリシー管理 API で認証方式をテンプレートとして設定する操作は、SDK v6.0 SP3 で廃止されました。
<i>blsUsedbyAdmin</i>	管理者の認証にカスタム認証方式を使用できるかどうかを示すフラグ。
<i>nType</i>	<code>Sm_Api_SchemeType_t</code> で定義された認証方式のタイプ。
<i>bAllowSaveCreds</i>	ユーザ認証情報の保存を許可するフラグ。
<i>blsRadius</i>	方式が <b>RADIUS</b> タイプかどうかを示すフラグ。
<i>blgnorePwCheck</i>	このフラグを <b>true</b> に設定すると、認証方式のパスワードポリシーは無効化されます。
<i>next</i>	次の方式構造体へのポインタ。

## Sm\_PolicyApi\_SharedSecretPolicy\_t

共有秘密キー ポリシーを定義します。

Administrative UI では、共有秘密キー ポリシーは、[キー管理] ダイアログ ボックスの [共有秘密キーのロールオーバー] タブで定義されます。

### 構文

```
typedef struct Sm_PolicyApi_SharedSecretPolicy_s
{
    int iStructID;
    bool bIsEnabled;
    int iRolloverPeriod;
    int iRolloverFrequency;
} Sm_PolicyApi_SharedSecretPolicy_t;
```

フィールド	説明
<i>iStructID</i>	Sm_PolicyApi_Structs_t で定義された認証方式データ構造体 ID。
<i>bIsEnabled</i>	共有秘密キーのロールオーバーが有効かどうか。
<i>iRolloverPeriod</i>	共有秘密キーが自動的に変更される頻度（たとえば、3 日ごと、2 か月ごとなど）を決定するために、 <i>iRolloverFrequency</i> と共に使用される時間の単位（毎時間、毎日、毎週、毎月）。有効な値は、Sm_PolicyApi_SecretRolloverPeriod_t で指定されます。
<i>iRolloverFrequency</i>	ロールオーバーの間隔における <i>iRolloverPeriod</i> 単位の数を指定します。

## Sm\_PolicyApi\_Server\_t

ポリシー サーバの TCP/IP 接続情報を定義します。

### 構文

```
typedef struct Sm_PolicyApi_Server_s
{
    int iStructId;
    char pszIpAddr[BFSIZE];          /* Required */
    long nPort[3];
    long nClusterSeq;
    struct Sm_PolicyApi_Server_s* next;
} Sm_PolicyApi_Server_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義されたポリシー サーバ構造体 ID。
<i>pszIpAddr</i>	ポリシー サーバの TCP/IP アドレス。
<i>nPort</i>	SiteMinder v6.0 より前のバージョンでは、アカウントिंग、認証、および認可のサービス用の TCP/IP ポートの配列です。SiteMinder v6.0 以降では、ポリシー サーバ ポートのみを指定する必要があります。
<i>nClusterSeq</i>	このサーバのクラスタ シーケンス番号。非クラスタ サーバの場合、このパラメータは省略します。クラスタ サーバの場合、クラスタ シーケンス番号を指定します（最初は 1）。既存のシーケンス番号と同じシーケンス番号を指定するとエラーになります。
<i>next</i>	次のサーバ構造体へのポインタ。

## Sm\_PolicyApi\_TrustedHost\_t

トラステッドホストオブジェクトを定義します。

## 構文

```
typedef struct Sm_PolicyApi_TrustedHost_s
{
    int iStructId;
    char pszOid[BFSIZE];
    char pszName[BFSIZE];
    char pszDesc[BFSIZE];
    char pszIPAddr[BFSIZE];
    char pszSecret[BFSIZE];
    bool bIs4xHost;
    struct Sm_PolicyApi_TrustedHost_s* next;
    bool bRolloverEnabled;
} Sm_PolicyApi_TrustedHost_t;
```

フィールド	説明
<i>iStructId</i>	データ構造体 ID、Sm_PolicyApi_Structs_t で定義されています。
<i>pszOid</i>	トラステッドホストオブジェクトのオブジェクト識別子。
<i>pszName</i>	トラステッドホスト名。
<i>pszDesc</i>	トラステッドホストの簡単な説明。
<i>pszIPAddr</i>	トラステッドホストの IP アドレス。
<i>pszSecret</i>	トラステッドホストの共有秘密キー。
<i>bIs4xHost</i>	内部的使用のみ。
<i>next</i>	次のトラステッドホスト構造体へのポインタ。
<i>bRolloverEnabled</i>	このトラステッドホストに対して共有秘密キーのロールオーバーを有効化するかどうかを示します。

## Sm\_PolicyApi\_User\_t

SiteMinder ユーザ オブジェクトを定義します。

### 構文

```
typedef struct Sm_PolicyApi_User_s
{
    int iStructId;
    char pszUserPolicyOid[BFSIZE];          /* Required */
    char pszUserDirOid[BFSIZE];            /* Required */
    char pszPath[BFSIZE];                  /* Required */
    char pszClass[BFSIZE];                 /* Required */
    Sm_PolicyResolution_t nPolicyResolution; /* Required */
    int nFlags;                             /* Required */
    struct Sm_PolicyApi_User_s* next;
} Sm_PolicyApi_User_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義されたユーザ データ構造体 ID。
<i>pszUserPolicyOid</i>	ユーザ ポリシーのオブジェクト識別子。
<i>pszUserDirOid</i>	ユーザ ディレクトリのオブジェクト識別子
<i>pszPath</i>	ユーザの識別名 (DN) 。
<i>pszClass</i>	ユーザ ディレクトリによって認識されているオブジェクトクラス (LDAP における <code>organizationalUnit</code> や NT における <code>User</code> など) 。
<i>nPolicyResolution</i>	2 つのポリシー オブジェクト間の関係。ポリシー解決は Sm_PolicyResolution_t で列挙されます。
<i>nFlags</i>	ポリシー フラグで定義されたビットを使用するビットマスク。
<i>next</i>	次のユーザ構造体へのポインタ。

## Sm\_PolicyApi\_UserContext\_t

ユーザ コンテキスト情報をポリシー管理 API のコール元に利用可能にします。

### 構文

```
typedef struct Sm_PolicyApi_UserContext_s
{
    int iStructId;
    Sm_Api_UserContext_t *pUserContext;
    struct Sm_PolicyApi_UserContext_s* next;
} Sm_PolicyApi_UserContext_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義されたユーザ データ構造体 ID。
<i>pUserContext</i>	ユーザ コンテキスト情報へのポインタ。
<i>next</i>	このフィールドは、常に <b>Null</b> に設定されます。

詳細情報:

[Sm\\_PolicyApi\\_GetTrustedHostByName\(\)](#) (P. 524)

## Sm\_PolicyApi\_UserDir\_t

SiteMinder ユーザディレクトリ オブジェクトを定義します。

### 構文

```
typedef struct Sm_PolicyApi_UserDir_s
{
    int iStructId;
    char pszOid[BFSIZE];
    char pszName[BFSIZE];           /* Required */
    char pszDesc[BFSIZE];
    char pszNamespace[BFSIZE];     /* Required */
    char pszServer[BFSIZE];        /* Required */
    char pszSearchRoot[BFSIZE];
    char pszUserLookupStart[BFSIZE];
    char pszUserLookupEnd[BFSIZE];
    char pszUsername[BFSIZE];
    char pszPassword[BFSIZE];
    int nSearchResults;            /* Required */
    int nSearchScope;             /* Required */
    int nSearchTimeout;           /* Required */
    bool bSecureConnection;       /* Required */
    bool bRequireCredentials;     /* Required */
    char pszDisabledAttr[BFSIZE];
    char pszUniversalIDAttr[BFSIZE];
    char pszODBCQuerySchemeOid[BFSIZE];
    char pszAnonymousId[BFSIZE];
    char pszPasswordData[BFSIZE];
    char pszPasswordAttribute[BFSIZE];
    char pszEmailAddressAttr[BFSIZE];
    char pszChallengeRespAttr[BFSIZE];
    struct Sm_PolicyApi_UserDir_s* next;
} Sm_PolicyApi_UserDir_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義されたユーザディレクトリ データ構造体 ID。
<i>pszOid</i>	ユーザディレクトリ オブジェクトのオブジェクト識別子。
<i>pszName</i>	ユーザディレクトリ の名前。
<i>pszDesc</i>	ユーザディレクトリ の簡単な説明。

フィールド	説明
<i>pszNamespace</i>	接続先の特定のディレクトリ サービスを指定する必須フィールド (LDAP:、ODBC:、WinNT:、AD:、Custom: など)。
<i>pszServer</i>	<p>必須フィールド。これは、内容がネームスペースに依存するオーバーロードされたフィールドです。</p> <ul style="list-style-type: none"> <li>■ ODBC -- データ ソース名。</li> <li>■ NT - ドメイン名。</li> <li>■ LDAP または AD - IP アドレス、または IP アドレスとポート番号 (形式: <i>IP_address:port_number</i>)。ポート番号が指定されない場合、ポート番号 389 と見なされます。</li> <li>■ Custom - ライブラリ名。</li> </ul>
<i>pszSearchRoot</i>	<p>以下の値のいずれかです。</p> <ul style="list-style-type: none"> <li>■ LDAP ディレクトリでは、ディレクトリ接続の開始ポイントである LDAP ツリー内の場所。通常は、組織 (o) または組織単位 (ou)。ポリシー サーバは、ルートからユーザの検索を開始します。</li> <li>■ カスタム ディレクトリの場合、カスタム ライブラリに渡すべき任意のパラメータを指定します。</li> </ul>
<i>pszUserLookupStart</i>	<p>ユーザ DN 検索の先頭文字列を使用すると、ユーザは、ユーザ名の一部だけを入力して認証できます。DN 文字列全体を入力する必要はありません。ユーザ DN 文字列の一意および非一意のセグメントの識別では、これが行われます。</p> <p>このフィールドは LDAP ディレクトリでのみ使用します。</p>
<i>pszUserLookupEnd</i>	<p>ユーザ DN 検索の終端文字列を使用すると、ユーザは、ユーザ名の一部だけを入力して認証できます。DN 文字列全体を入力する必要はありません。</p> <p>このフィールドは LDAP ディレクトリでのみ使用します。</p>
<i>pszUsername</i>	ユーザ ディレクトリにアクセスする必要があるユーザ名。

フィールド	説明
<i>pszPassword</i>	ユーザディレクトリのアクセスに必要なパスワード。
<i>nSearchResults</i>	LDAP ディレクトリまたはカスタム ディレクトリの検索から返すことができるレコードの最大数。
<i>nSearchScope</i>	LDAP ディレクトリの <i>pszSearchRoot</i> の下で SiteMinder がユーザおよびユーザ グループを検索する範囲 - ルート (サブツリー) の下の全レベル、またはルートの下での 1 レベルだけ。 ルートの 1 レベル下の場合は 1、サブツリーの場合は 2 を指定します。
<i>nSearchTimeout</i>	SiteMinder が LDAP またはカスタム ディレクトリをクエリする最大時間 (秒単位)。
<i>bSecureConnection</i>	SSL 経由で LDAP またはカスタム ディレクトリにアクセスする場合は、このフラグを有効にする必要があります。セキュア接続を有効化することは、SiteMinder がセキュアな認証および暗号化された転送を実行することを意味します。
<i>bRequireCredentials</i>	ユーザディレクトリに対して認証する必要があるクレデンシャルを指定するフラグ
<i>pszDisabledAttr</i>	ユーザの有効状態または無効状態の追跡に SiteMinder が使用するユーザディレクトリ属性の名前。 LDAP ディレクトリおよび ODBC ディレクトリに適用されます。カスタム ディレクトリに適用される場合もあります。
<i>pszUniversalIDAttr</i>	ユニバーサル ID として指定されたユーザディレクトリ属性の名前。通常、ユニバーサル ID は、ユーザのログイン ID とは異なり、ユーザ情報を検索するために使用されます。 LDAP ディレクトリ、ODBC ディレクトリ、WinNT ディレクトリ、および場合によってはカスタム ディレクトリに適用されます。
<i>pszODBCQuerySchemeOid</i>	SiteMinder が ODBC ディレクトリをクエリするために使用する一連の ODBC クエリのオブジェクト識別子。

フィールド	説明
<i>pszAnonymousId</i>	匿名ユーザ DN として指定されているユーザディレクトリ属性の名前。この DN は、匿名認証方式で定義されます。匿名ユーザは、この DN を偽装して、匿名認証方式に関連付けられているリソースにアクセスします。 LDAP ディレクトリに適用されます。カスタムディレクトリに適用される場合もあります。
<i>pszPasswordData</i>	パスワードポリシー情報の保存に SiteMinder が使用するユーザディレクトリ属性の名前。 LDAP ディレクトリおよび ODBC ディレクトリに適用されます。カスタムディレクトリに適用される場合もあります。
<i>pszPasswordAttribute</i>	パスワードサービスを使用して定義されるユーザパスワードが含まれるユーザディレクトリ属性の名前。 LDAP ディレクトリおよび ODBC ディレクトリに適用されます。カスタムディレクトリに適用される場合もあります。
<i>pszEmailAddressAttr</i>	将来使用のために予約済みです。
<i>pszChallengeRespAttr</i>	ユーザに返すレスポンスを含むユーザディレクトリ属性の名前。忘れたパスワードのヒントなど。 LDAP ディレクトリに適用されます。カスタムディレクトリに適用される場合もあります。
<i>next</i>	次のディレクトリ構造体へのポインタ。

### 注釈

個別のディレクトリタイプが指定されている場合を除き、フィールドはすべてのタイプのディレクトリ（LDAP、ODBC、WinNT、カスタム）に適用されます。

LDAP ディレクトリに適用されるフィールドは、Active Directories にも適用されます。

## Sm\_PolicyApi\_UserPasswordState\_t

すべての PasswordState 仮想属性に関する情報が SmPolicyApi\_UserPasswordState\_t 構造体を使用して返されます。この構造体は User オブジェクトと共存し、これはユーザディレクトリ OID とユーザ DN 文字列によって制限されます。この構造体は、C ポリシー管理 API を使用して取得、作成、更新できます。

### 構文

```
typedef struct Sm_PolicyApi_UserPasswordState_s
{
    int iLoginFailures;
    time_t tLastLogin;
    time_t tPrevLogin;
    time_t tDisabled;
    time_t tLastPWChange;
} Sm_PolicyApi_UserPasswordState_t;
```

フィールド	説明
<i>iLoginFailures</i>	最後の正常なログインの後、ユーザがログインに失敗した回数を指定します。
<i>tLastLogin</i>	ユーザがログインに成功した最後の時刻を指定します。
<i>tPrevLogin</i>	ユーザがログインに成功した 2 番目のログインから最終ログインまでを指定します。
<i>tDisabled</i>	ユーザが無効化された時刻を指定します。
<i>tLastPWChange</i>	<p>ユーザがパスワードを変更した最後の時刻を指定します。</p> <p>この値がパスワードが最後に変更された時のユーザディレクトリ設定を更新し、パスワードが SiteMinder の外部でリセットされた場合、パスワード再利用を防ぐパスワードポリシーが予期したように動作しない場合があります。</p> <p>以下の場合、このフィールドで値 0 が返される可能性があります。</p> <ul style="list-style-type: none"> <li>■ ユーザは、パスワードを変更する手順を開始したが、それを完了していない。</li> <li>■ Sm_PolicyApi_SetUserPasswordState() を呼び出してパスワード履歴がクリアされた。</li> </ul>

## Sm\_PolicyApi\_Variable\_t

ポリシーやレスポンスの変数式で使用可能な変数オブジェクトを定義します。変数オブジェクトは変数関数によって管理されます。

変数は、許可リクエスト中に値に解決される動的なオブジェクトです。変数は、ポリシーまたはレスポンスに対して定義されたアクティブな式の中で使用されます。

変数は以下のように使用されます。

- ポリシーの場合、変数は許可の制約として使用されます。ユーザがリソースへのアクセスを要求し、そのリソースには1つ以上の変数が含まれるアクティブな式が含まれる場合、変数はユーザに関連する値に解決されます。それらの値は評価され、ユーザを認可するかどうかについての判断に使用されます。

たとえば、銀行のクレジットカード申込を保護するポリシーに、**Credit Rating** 変数および **Salary** 変数のアクティブな式が含まれるものとします。ユーザがフォームにアクセスしようとする時、そのユーザの信用格付けと給料がこれらの変数の最小値に一致するかそれ以上の場合にのみ、ユーザが許可されます。

- レスポンスでは、変数は戻り値として使用されます。たとえば、リモート Web サービスから取得したトランザクションの追跡番号を返すようにレスポンス属性を設定するなどです。

### 構文

```
typedef struct Sm_PolicyApi_Variable_s
{
    int iStructId;
    char pszOid[BFSIZE];
    char pszName[BFSIZE];
    char pszDesc[BFSIZE];
    char pszVariableTypeOid[BFSIZE];
    char pszDefinition[BFSIZE];
    char pszMetaData[BFSIZE];
    int nReturnType;
    bool bPreFetchFlag;
    char pszDomainOid[BFSIZE];
    Sm_PolicyApi_Oid_t* pNestedVariableList;
    struct Sm_PolicyApi_Variable_s* next;
} Sm_PolicyApi_Variable_t;
```

フィールド	説明
<i>iStructId</i>	データ構造体 ID、Sm_PolicyApi_Structs_t で定義されています。
<i>pszOid</i>	可変オブジェクトの一意のオブジェクト ID。
<i>pszName</i>	可変オブジェクトのユーザ定義の名前。
<i>pszDesc</i>	可変オブジェクトについて説明するオプションのテキスト。
<i>pszVariableTypeOid</i>	可変タイプの一意のオブジェクト ID。
<i>pszDefinition</i>	ランタイムに変数の値を取得する必要がある情報。
<i>pszMetaData</i>	オプションの CA TransactionMinder 製品で使用するために予約済みです。
<i>nReturnType</i>	変数値のデータ型 : <ul style="list-style-type: none"> <li>■ Sm_PolicyApi_VarReturnTypes_Boolean</li> <li>■ Sm_PolicyApi_VarReturnTypes_Number</li> <li>■ Sm_PolicyApi_VarReturnTypes_String</li> <li>■ Sm_PolicyApi_VarReturnTypes_Date</li> </ul>
<i>bPreFetchFlag</i>	現在は使用されていません。
<i>pszDomainOid</i>	関連するドメインの一意のオブジェクト ID。
<i>pNestedVariableList</i>	この変数の定義の一部であるネストされた変数 OID のリンクリスト。
<i>next</i>	次の可変オブジェクト構造体へのポインタ。

## 変数定義

変数の値の検出が可能な場所を指定することにより変数を定義します。そのためには *pszDefinition* フィールドを使用します。

このフィールドの値は、変数タイプに応じて単純な文字列または一連の XML 要素が有効です。SiteMinder 変数タイプ、およびタイプごとの *pszDefinition* フィールドの説明を以下に示します。

### ■ Post

*pszDefinition* フィールドには、HTML フォーム上のフィールドの名前が含まれます。POST アクションでは、変数値はフィールドに割り当てられた値から引き出されます。

### ■ RequestContext

*pszDefinition* フィールドには以下の XML コードが含まれます。

```
<RequestContextVariableDef>
  <ItemName></ItemName>
</RequestContextVariableDef>
```

変数値は、ItemName エレメント内で以下の属性名のどれが使用されるかによって異なります。

- **Action**。このアイテム名では、可変値は、リクエスト（GET、POST など）で指定されたアクションのタイプです。
- **Resource**。このアイテム名です、可変値はターゲットリソースです（たとえば、/directory\_name/）。
- **Server**。このアイテム名では、可変値は要求で指定された完全サーバ名です（たとえば、server.company.com）。

### ■ Static

*pszDefinition* フィールドには、実行時にユーザ指定データと比較される実際の値が含まれます。たとえば、戻り型が

Sm\_PolicyApi\_VarReturnTypes\_Date の Static 変数は文字列値 2004-01-01 を割り当てられる場合があります。許可時に、この割り当てられた日付は、ユーザ提供の日付と比較されます。

■ **UserContext**

*pszDefinition* フィールドには、以下の XML コードのいくつかまたはすべてが含まれます。

```
<UserContextVariableDef>
  <ItemName></ItemName>
  <PropertyName></PropertyName>
  <DN></DN>
  <BufferSize></BufferSize>
</UserContextVariableDef>
```

変数値は、ユーザディレクトリ接続の属性（セッション ID など）またはユーザディレクトリの内容（ユーザ名など）に基づいています。変数値の基になる属性の名前は、XML エレメント **ItemName** 内で使用されます。

**ItemName** 要素は、以下のいずれかの値を含むことができます。

- DirectoryEntryProperty
- DirectoryNameSpace
- DirectoryPath
- DirectoryServer
- IsUserContext
- SessionId
- UserPath
- UserProperty
- ユーザ名

エレメント **PropertyName**、**DN**、および **BufferSize** は、以下の方法のみで使用されます。

- **ItemName** に **DirectoryEntryProperty** が含まれる場合、エレメント **PropertyName**、**DN**、および **BufferSize** が使用されます。
- **ItemName** に **UserProperty** が含まれる場合、エレメント **PropertyName** および **BufferSize** が使用されます。

■ **WebService**

*pszDefinition* フィールドには以下の基本的な XML 構造が含まれます。

```
<WebServiceVariableDefn xmlns:NeteWS=
    "http://www.netegrity.com/2003/SM6.0";>
  <NeteWS:RemoteURL></NeteWS:RemoteURL>
  <NeteWS:SSL/>
  <NeteWS:RemoteMethod></NeteWS:RemoteMethod>
  <NeteWS:ResultQuery></NeteWS:ResultQuery>
  <NeteWS:AuthCredentials>
    <NeteWS:Username></NeteWS:Username>
    <NeteWS>Password></NeteWS>Password>
    <NeteWS:Hash></NeteWS:Hash>
  </NeteWS:AuthCredentials>
  <NeteWS:Document>
    <SOAP:Envelope xmlns:SOAP=
      "http://schemas.xmlsoap.org/soap/envelope/">
      <SOAP:Header></SOAP:Header>
      <SOAP:Body></SOAP:Body>
    </SOAP:Envelope>
  </NeteWS:Document>
</WebServiceVariableDefn>
```

Web サービスから変数値を取得するため、ポリシー サーバは、Web サービスに、*pszDefinition* 引数で指定されているとおりの SOAP 要求ドキュメントを送信し、変数値を SOAP レスポンスから抽出します。

以下の表には、WebService 変数の設定に使用される XML エレメントの説明を示します。

エレメント	説明
RemoteURL	WebService 変数を解決する Web サービスの URL。
SSL	ポリシー サーバと Web サービスの間の接続に SSL を使用するよう指定します。
RemoteMethod	このエレメントを POST に設定します。
ResultQuery	XPath 形式でのリターンクエリ。ポリシー サーバは、SOAP レスポンス ドキュメントで変数値を検索するためにこの情報を使用します。

エレメント	説明
AuthCredentials	<p>必要に応じて、以下の要素によってユーザの Web サービス クレデンシャルを指定します。</p> <ul style="list-style-type: none"> <li>■ Username</li> <li>■ パスワード (SHA-1 パスワード要約またはクリア テキスト パスワードのいずれかを使用)</li> </ul> <p>オプションで、Hash エレメントを使用して、パスワードのハッシュが WS -セキュリティ パスワードに含まれることを指定します。</p>
Document	<p>オプションで、以下のエレメントによって SOAP ヘッダまたは SOAP 本文 (またはその両方) を定義するためにこのエレメントを使用します。</p> <ul style="list-style-type: none"> <li>■ Envelope。 SOAP ネームスペースは次のとおりです。 <a href="http://schemas.xmlsoap.org/soap/envelope">http://schemas.xmlsoap.org/soap/envelope</a></li> <li>■ Header。 ユーザ定義の SOAP ヘッダ。 ユーザの Web サービス 認証情報が指定される場合、 WS-Security ヘッダが自動的に追加されます。</li> <li>■ Body。 ユーザ定義の SOAP 本文。</li> </ul> <p>RequestContext、 UserContext、 Post、 および Static タイプのネストされた変数をヘッダと本文内で使用できます。それらの値は、要求ドキュメントがリモート Web サービスに送信される前に、解決および代入されます。</p> <p>ネストされた変数は、以下のように指定します。</p> <p><code>\$variable-name\$</code></p>

注: 上に表示された XML エレメント構造は、読みやすいようにフォーマットされます。 *pszDefinition* フィールドによって提供された XML 文字列は、スペース、タブ、リターン文字でフォーマットしません。たとえば、Resource 属性の RequestContext 変数は、次のように *pszDefinition* で渡されます。

```
<RequestContextVariableDef><ItemName>Resource</ItemName></RequestContextVariableDef>
```

## Sm\_PolicyApi\_VariableType\_t

サポートされる変数オブジェクトタイプを定義します。変数タイプは読み取り専用です。これらは、ポリシー管理変数関数で作成したり削除することはできません。

### 構文

```
typedef struct Sm_PolicyApi_VariableType_s
{
    int iStructId;
    char pszOid[BFSIZE];
    char pszName[BFSIZE];
    char pszDesc[BFSIZE];
    char pszFilter[BFSIZE];
    struct Sm_PolicyApi_VariableType_s* next;
} Sm_PolicyApi_VariableType_t;
```

フィールド	説明
<i>iStructId</i>	Sm_PolicyApi_Structs_t で定義されたデータ構造体 ID。
<i>pszOid</i>	可変タイプ オブジェクトの一意のオブジェクト ID。
<i>pszName</i>	以下のいずれかのオブジェクトタイプ名。 <ul style="list-style-type: none"> <li>■ Post</li> <li>■ RequestContext</li> <li>■ Static</li> <li>■ UserContext</li> <li>■ WebService</li> </ul> <p>オプションの CA SOA Security Manager 製品をインストールした場合は、以下の変数タイプも使用できます。</p> <ul style="list-style-type: none"> <li>■ SAMLAssertion</li> <li>■ Transport</li> <li>■ XMLAgent</li> <li>■ XMLBody</li> <li>■ XMLEnvelopeHeader</li> </ul> <p>SDK を使用して、これらの変数を作成することはできません。Administrative UI を使用する必要があります。</p>

フィールド	説明
<i>pszDesc</i>	オブジェクト変数タイプ（たとえば、フォーム ポスト変数など）の説明。
<i>pszFilter</i>	現在は使用されていません。
<i>next</i>	次の変数タイプ オブジェクト構造体へのポインタ。

## Sm\_PolicyApi\_WSFEDProviderProp\_t

WS フェデレーションプロバイダプロパティのリンク リストを定義します (名前/値ペア)。

Sm\_PolicyApi\_WSFEDProviderProp\_t 構造体は、単一の名前/値ペアから構成されます。Sm\_PolicyApi\_WSFEDProviderProp\_t 構造体のリンク リストを使用して、所定の WS フェデレーションオブジェクトの一連のプロパティを定義します。

### 構文

```
typedef struct Sm_PolicyApi_WSFEDProviderProp_s
{
    int iStructId;
    char pszName[BFSIZE];
    char pszValue[BFSIZE];
    Sm_PolicyApi_WSFEDProviderProp_t* next;
} Sm_PolicyApi_WSFEDProviderProp_t;
```

### パラメータ

iStructId

Sm\_PolicyAp\_Structs\_t 内の構造体の ID。

Sm\_PolicyApi\_WSFEDProviderProp\_ID に設定する必要があります。

pszName

WS フェデレーションプロバイダプロパティの名前。

pszValue

WS フェデレーションプロバイダプロパティの値。

next

リンク リスト内の次の WS フェデレーションプロバイダプロパティデータへのポインタ。

それぞれの Sm\_PolicyApi\_WSFEDProviderProp\_t 構造には、名前/値ペアとして定義された WS フェデレーション メタデータ プロパティが含まれます。特定のオブジェクトのプロパティの一式は、Sm\_PolicyApi\_WSFEDProviderProp\_t 構造体のリンク リストとして定義されます。

以下のメタデータ プロパティは、WS フェデレーションオブジェクトタイプに適用されます。

- 共通プロパティ
- リソースパートナーを定義するためのプロパティ
- アカウントパートナーを定義するためのプロパティ

オプションのプロパティは、角かっこで指定されます。

ブール値の場合、値 **1** は **true** を示し、他のすべての値は **false** を示します。

[ポリシー名] 列には、対応する C ポリシー管理 API マクロ名も含まれています。

### 共通プロパティ

以下の表には、リソースパートナーやアカウントパートナーの定義に共通のメタデータ プロパティを示します。

プロパティ名	タイプ	説明
<i>一般</i>		
名前 WSFED_NAME	文字列	プロバイダの名前
[Description] WSFED_DESCRIPTION	文字列	プロバイダの簡単な説明。
[SkewTime] WSFED_SKEW_TIME	文字列	コンシューマ側とプロデューサ側間のスキュー時間 (秒)。この値は、アサーションおよび SLO 要求の有効期間の計算に使用されます。デフォルト値は <b>30</b> です。
<i>バージョン</i>		
[WSFEDMajorVersion] WSFED_MAJOR_VERSION	整数	このプロバイダによってサポートされている WSFED プロトコルのバージョン。このプロパティの値は <b>1</b> に設定する必要があります。
[WSFEDMinorVersion] WSFED_MINOR_VERSION	整数	このプロバイダによってサポートされている WSFED プロトコルのバージョン。このプロパティの値は <b>0</b> に設定する必要があります。
[WSFEDSAMLMajorVersion] WSFED_SAML_MAJOR_VERSION	整数	このプロバイダによってサポートされている SAML プロトコルのバージョン。このプロパティの値は <b>1</b> に設定する必要があります。

プロパティ名	タイプ	説明
[WSFEDSAMLMinorVersion] WSFED_SAML_MINOR_VERSION	整数	このプロバイダによってサポートされている WSFED プロトコルのバージョン。このプロパティの値は 1 に設定する必要があります。

### リソース パートナー プロパティ

以下の表には、リソース パートナーを定義するために使用されるメタデータ プロパティを示します。

プロパティ名	タイプ	説明
ドメイン WSFED_RP_DOMAIN	OID	このリソース パートナーが定義されているドメイン OID。
[Enabled] WSFED_ENABLED	ブール	プロバイダが有効かどうかを示すブール。指定しない場合は、デフォルトにより true になります。このプロパティは、プロパティコレクションに物理的に保存されるものではありませんが、基本ポリシーを有効化するために使用されます。
NetegrityAffiliateMinderAuthURL WSFED_RP_AUTHENTICATION_URL	文字列	リソース パートナー ユーザの認証に使用する保護された URL。
<i>名前 ID</i>		
[NameIdFormat] WSFED_RP_NAMEID_FORMAT	文字列	WSFED 名識別子の URI。
[NameIdType] WSFED_RP_NAMEID_TYPE	整数	名前識別子のタイプを表します。 0 -- スタティック テキスト 1 -- ユーザ属性 2 -- DN 属性 デフォルトは 1

プロパティ名	タイプ	説明
[NameIdStatic] WSFED_RP_NAMEID_STATIC	文字列	NameIdType == 0 の場合にスタティックテキストとして使用される名前識別子。このプロパティに対して値が指定されず、NameIdType==0 である場合、ポリシー管理 API はエラーを返します。
[NameIdAttrName] WSFED_RP_NAMEID_ATTR_NAME	文字列	NameIdType == 1 または NameIdType == 2 の場合に名前識別子を保持する属性名（ユーザまたは DN）。「NameIdType」が「1」または「2」に設定されている場合、 「NameIdAttrName」プロパティに値が指定されていることが必要です。そうでない場合、ポリシー管理 API がエラーを返します。
[NameIdDNSpec] WSFED_RP_NAMEID_DN_SPEC	文字列	NameIdType == 2 の場合の DN 指定。「NameIdType」が「2」に設定されている場合、 「NameIdDNSpec」プロパティに値が指定されていることが必要です。そうでない場合、ポリシー管理 API がエラーを返します。
[NameIdAllowNested] WSFED_RP_NAMEID_ALLOWED_NESTED	ブール	名前識別子の DN 属性の選択時にネストされたグループが許可されるかどうかを示すフラグ。デフォルトはゼロです。
一般		

プロパティ名	タイプ	説明
KEY_RPID WSFED_KEY_RPID	文字列	WSFED アサーション コンシューマのリソース パートナー ID。 1024 文字未満の長さの URI である必要があります。また、これはプロバイダに関連付けたプロパティを検索することができるキーでもあります。
APID WSFED_APID	文字列	WSFED アサーションプロバイダのリソース パートナーID。
SSO		
[AuthenticationMethod] WSFED_RP_AUTHENTICATION_METHOD	文字列	アサーションで使用する認証方式。
[ValidityDuration] WSFED_RP_VALIDITY_DURATION	整数	生成されたアサーションの有効期間の秒数を表す整数。 リソース パートナーの作成中に指定されない場合、デフォルトは 60 秒です。
AssertionConsumerDefaultURL WSFED_RP_ASSERTION_CONSUMER_DEFAULT_URL	文字列	使用するデフォルト WSFED アサーション コンシューマ。
[AuthenticationLevel] WSFED_RP_AUTHENTICATION_LEVEL	整数	プリンシパルは、少なくともこのレベル以上の認証方式によってレルム内で認証される必要があります。 リソース パートナーの作成中に指定されない場合、デフォルトで 5 に設定されます。
サインアウト		
[SLOEnabled] WSFED_RP_SLO_ENABLED	ブール	リソース パートナーに対してサインアウトが有効かどうかを示すブール。

プロパティ名	タイプ	説明
[SignOutCleanupURL] WSFED_RP_SIGNOUT_CLEANUP_URL	文字列	リソース パートナーのサインアウト クリーンアップ URL。 SLOEnabled が true の場合、このプロパティは必須です。
[SignOutConfirmURL] WSFED_RP_SIGNOUT_CONFIRM_URL	文字列	アカウント パートナーでサインアウトが完了した以降にユーザがリダイレクトされる URL。 (利用可能な複数のリソース パートナーがある場合は、最後のリソース パートナーのサインアウト確認 URL を適用できます。)
<i>詳細</i>		
[AssertionPluginClass] WSFED_RP_PLUGIN_CLASS	文字列	使用されるアサーション ジェネレータ プラグイン クラスの完全修飾 Java クラス名。
[AssertionPluginParameters] WSFED_RP_PLUGIN_PARAMS	文字列	アサーション ジェネレータ プラグインに渡されるパラメータが含まれる文字列。

### アカウント パートナー プロパティ

以下の表には、アカウント パートナーを定義するために使用されるメタデータ プロパティを示します。

プロパティ名	タイプ	説明
<i>一般</i>		
KEY_APID WSFED_KEY_APID	文字列	アカウント パートナーの識別子。特に、アサーション 発行者の識別にはこの識別子が使用されます。また、これはアカウント パートナーに関連付けたプロパティを検索することができるキーでもあります。

プロパティ名	タイプ	説明
RPID WSFED_RPID	文字列	リソースパートナーの識別子。
<i>署名</i>		
[DisableSignatureProcessing] WSFED_DISABLE_SIGNATURE_PROCESSING	ブール	署名処理が無効かどうかを指定します。この設定は、アカウントパートナーの初期設定時に便利です。プロバイダが稼働中のときは、セキュリティの影響を避けるため、この設定を <b>false</b> に設定する必要があります。デフォルト値はゼロです。
[DsigVerInfoIssuerDN] WSFED_DSIG_VERINFO_ALIAS	文字列	これがインラインで提供されない場合、キーストア内でプロバイダの証明書を検索するために使用されます。
<i>ユーザ</i>		
[XPath] WSFED_AP_XPATH	文字列	プリンシパルの特定を行うための XPath クエリ。
[LDAPSearchSpec] WSFED_AP_LDAP_SEARCH_SPEC	文字列	LDAP ディレクトリの検索仕様。
[ODBCSearchSpec] WSFED_AP_ODBC_SEARCH_SPEC	文字列	ODBC ディレクトリの検索仕様。
[WinNTSearchSpec] WSFED_AP_WINNT_SEARCH_SPEC	文字列	WinNT ディレクトリの検索仕様。
[CustomSearchSpec] WSFED_AP_CUSTOM_SEARCH_SPEC	文字列	カスタム ディレクトリの検索指定情報です。
[ADSearchSpec] WSFED_AP_AD_SEARCH_SPEC	文字列	AD ディレクトリの検索仕様。
SSO		

プロパティ名	タイプ	説明
[RedirectMode] WSFED_AP_SSO_REDIRECT_MODE	整数	アサーション属性のリダイレクトモード。以下の値が有効です。 0 -- 302 データなし 1 -- 302 Cookie データ 2 -- サーバリダイレクト 3 -- 属性の保持 デフォルトはゼロです。
[SSODefaultService] WSFED_AP_SSO_DEFAULT_SERVICE	文字列	シングルサインオンサービスのデフォルトの場所。
[Target] WSFED_AP_SSO_TARGET	文字列	送信先サイトのターゲットリソース。
[EnforceSingleUsePolicy] ENFORCE_SINGLE_USE_POLICY	ブール	1 の場合、POST アサーションの使い捨てポリシーが適用されます。0 の場合は適用されません。デフォルトで 1 設定されています。
<i>サインアウト</i>		
[SLOEnabled] WSFED_AP_SLO_ENABLED	ブール	アカウント パートナーに対してサインアウトが有効かどうかを示すブール。アカウント パートナーの作成中に指定されない場合、デフォルトで無効に設定されます。
[SignOutURL] WSFED_AP_SIGNOUT_URL	文字列	アカウント パートナーのサインアウト URL。 SLOEnabled が true の場合、このプロパティは必須です。
<i>メッセージコンシューマ プラグイン</i>		
[APPluginClass] WSFED_AP_PLUGIN_CLASS	文字列	アサーション使用のカスタマイズを実装する Java クラスの名前。

プロパティ名	タイプ	説明
[APPluginParameters] WSFED_AP_PLUGIN_PARAMS	文字列	アサーション使用のカスタマイズを実装する Java クラスのパラメータ。パラメータはすべて 1 行に連結されます。
<i>後処理 URL サポート</i>		
[UserNotFoundRedirectURL] WSFED_AP_USER_NOT_FOUND_REDIRECT_URL	文字列	以下の場合に使用されるオプションのリダイレクト URL が含まれます。 - 設定済みのクエリ文字列を与えられたときに、認証方式がフェデレーションメッセージから LoginID を取得できない場合 - 設定済みのユーザストア検索文字列を与えられたときに、認証方式が特定のユーザディレクトリでユーザを検索しない場合
[UserNotFoundRedirectMode] WSFED_AP_USER_NOT_FOUND_REDIRECT_MODE	0/1	デフォルトは 0 です。 0 : Http 302 リダイレクト (フェデレーションメッセージの受け渡しなし) 1 : Http フォーム ポスト リダイレクト
[FailureRedirectURL] WSFED_AP_FAILURE_REDIRECT_URL	文字列	アサーション処理に失敗したときに使用されるオプションのリダイレクト URL が含まれています。
[FailureRedirectMode] WSFED_AP_FAILURE_REDIRECT_MODE	0/1	デフォルトは 0 です。 0 : Http 302 リダイレクト (フェデレーションメッセージの受け渡しなし) 1 : Http フォーム ポスト リダイレクト

プロパティ名	タイプ	説明
[InvalidRedirectURL] WSFED_AP_INVALID_REDIRECT_URL	文字列	アサーションが無効なときに使用されるオプションのリダイレクト URL が含まれています。
[InvalidRedirectMode] WSFED_AP_INVALID_REDIRECT_MODE	0/1	デフォルトは 0 です。 0 : Http 302 リダイレクト (フェデレーションメッセージの受け渡しなし) 1 : Http フォーム ポスト リダイレクト

## Sm\_PolicyApi\_WSFEDResourcePartner\_t

WS フェデレーション リソース パートナー データを定義します。

### 構文

```
typedef struct Sm_PolicyApi_WSFEDResourcePartner_s
{
    int iStructId;
    Sm_PolicyApi_WSFEDProviderProp_t* pProps;
    Sm_PolicyApi_WSFEDResourcePartner_t* next;
} Sm_PolicyApi_WSFEDResourcePartner_t;
```

### パラメータ

**iStructId**

Sm\_PolicyApi\_Structs\_t 内の構造体の ID。

Sm\_PolicyApi\_WSFEDResourcePartner\_ID に設定する必要があります。

**pProp**

リソース パートナー プロパティのリンク リストへのポインタ。

**next**

リンク リスト内の次のリソース パートナーデータへのポインタ。

## エクスポートされたタイプ

### 管理者権限

`Sm_PolicyApi_AdminRights_t` は、管理者の権限を列挙します。これらの値は、個別に使用されたり、組み合わせて複数の権限を設定することがあります。最終的な値が `Sm_PolicyApi_Admin_t` 構造体内の属性の 1 つとして `Sm_PolicyApi_AddAdmin()` に渡されます。

名前	値
<code>Sm_PolicyApi_AdminRights_ManageAllDomains</code>	0x01
<code>Sm_PolicyApi_AdminRights_ManageObjects</code>	0x02
<code>Sm_PolicyApi_AdminRights_ManageUsers</code>	0x04
<code>Sm_PolicyApi_AdminRights_ManageKeys</code>	0x08
<code>Sm_PolicyApi_AdminRights_ManagePasswordPolicy</code>	0x08
<code>Sm_PolicyApi_AdminRights_ManageReports</code>	0x10

以下の表では、これらの値を使用して管理者権限を設定する方法を示します。

スコープ	タスク	設定と権限
システム	システムとドメインオブジェクトの管理	<p>以下の権限を設定するには、管理者権限を以下の両方に設定します。</p> <p><b>Sm_PolicyApi_AdminRights_ManageAllDomains</b>  <b>Sm_PolicyApi_AdminRights_ManageObjects</b></p> <p>権限：</p> <p>エージェント、エージェントグループ、ディレクトリ、ポリシードメイン、認証方式、エージェントタイプ、ODBCセットアップ、ディレクトリマッピング、証明書マッピング、および登録方式の作成、編集、および削除</p> <p>すべてのドメインでの親レルムの作成と削除</p> <p>管理者の作成、編集、および削除</p> <p>キャッシュされたリソースを含むすべてのキャッシュのクリア</p> <p>グローバル設定の変更</p> <p>以下の一覧に示すドメインオブジェクトを管理するためのすべての権限</p>
ドメイン	ドメインオブジェクトの管理	<p>以下の権限を設定するには、管理者権限を以下に設定します。</p> <p><b>Sm_PolicyApi_AdminRights_ManageObjects</b></p> <p>権限：</p> <p>管理対象ドメインでの、ルール、ルールグループ、レスポンス、レスポンスグループ、ポリシーの作成、編集、および削除</p> <p>管理対象ドメイン内の最上位レルムの編集（リソースフィルタを除く）</p> <p>管理対象ドメイン内のネストされたレルムの作成、編集、および削除</p> <p>リソースキャッシュからの特定のレルムのクリア、およびキャッシュからのすべてのリソース（権限を付与されたドメイン内）のクリア</p>

スコープ	タスク	設定と権限
システム	レポートの表示	<p>以下の権限を設定するには、管理者権限を以下の両方に設定します。</p> <p><b>Sm_PolicyApi_AdminRights_ManageAllDomains</b>  <b>Sm_PolicyApi_AdminRights_ManageUsers</b></p> <p>権限：  システムおよびドメイン レポートをすべて表示します。</p>
ドメイン	レポートの表示	<p>以下の権限を設定するには、管理者権限を以下に設定します。</p> <p><b>Sm_PolicyApi_AdminRights_ManageUsers</b></p> <p>権限：  管理対象ドメインに関するレポートの表示</p>
システム	キーとパスワードポリシーの管理	<p>以下の権限を設定するには、管理者権限を以下の両方に設定します。</p> <p><b>Sm_PolicyApi_AdminRights_ManageAllDomains</b>  <b>Sm_PolicyApi_AdminRights_ManageKeys</b></p> <p>権限：  パスワードポリシーの作成、編集、および削除  キーの管理</p>
ドメイン	パスワードポリシーの管理	<p>以下の権限を設定するには、管理者権限を以下に設定します。</p> <p><b>Sm_PolicyApi_AdminRights_ManagePasswordPolicy</b></p> <p>権限：  管理対象ドメインに紐付けられたユーザに対するパスワードポリシーの作成、編集、および削除</p>
システム	ユーザの管理	<p>以下の権限を設定するには、管理者権限を以下の両方に設定します。</p> <p><b>Sm_PolicyApi_AdminRights_ManageAllDomains</b>  <b>Sm_PolicyApi_AdminRights_ManageReports</b></p> <p>権限：  すべてのユーザセッションキャッシュのクリア、または全ディレクトリのあらゆる個別ユーザのユーザセッションキャッシュクリア  全ディレクトリのユーザの有効化と無効化  全ディレクトリのすべてのユーザに対するパスワード変更の強制</p>

## エクスポートされたタイプ

スコープ	タスク	設定と権限
ドメイン	ユーザの管理	以下の権限を設定するには、管理者権限を以下に設定します。 <b>Sm_PolicyApi_AdminRights_ManageReports</b> 権限： 管理対象ドメインに紐付けられたディレクトリ内の個別ユーザのユーザセッションキャッシュのクリア 管理対象ドメインに紐付けられたディレクトリ内のユーザの有効化と無効化 管理対象ドメインに紐付けられたディレクトリ内のユーザに対するパスワード変更の強制

## アフィリエイト属性タイプ

`Sm_PolicyApi_AffiliateAttrType_t` は、アフィリエイト関数で使用するアフィリエイト属性を操作するための有効なアフィリエイト属性タイプを列挙します。

名前	値
<code>Sm_PolicyApi_Affiliate_HTTP_Header_Variable</code>	1
<code>Sm_PolicyApi_Affiliate_HTTP_Cookie_Variable</code>	2

## 属性モードタイプ

`Sm_PolicyApi_SAMLSPAttrMode_t` は、SAML 2.0 属性認証機関のサポートで使用される有効な属性取得タイプを列挙します。

名前	値
<code>Sm_PolicyApi_SAMLSP_SSO_Only</code>	0
<code>Sm_PolicyApi_SAMLSP_Attribute_Only</code>	1

これらの値の1つを `Sm_PolicyApi_SAMLSPAttr_t` 構造体の `nMode` エレメントに指定する必要があります。

## 認証と認可のマッピングタイプ

Sm\_PolicyApi\_AuthAzMapType\_t は、認証と認可のマッピングタイプを列挙します。

名前	値
Sm_PolicyApi_AuthAzMapType_DN	1
Sm_PolicyApi_AuthAzMapType_UniversalId	2
Sm_PolicyApi_AuthAzMapType_Attr	3

## 証明書マッピング属性タイプ

Sm\_PolicyApi\_CertMapAttrType\_t は、X.509 クライアント証明書を認証ディレクトリ内のユーザ情報にマッピングする方法を決定するマッピングのタイプを列挙します。

名前	値
Sm_PolicyApi_CertMapAttrType_Single	1
Sm_PolicyApi_CertMapAttrType_Custom	2
Sm_PolicyApi_CertMapAttrType_Exact	3

## 証明書マッピング ディレクトリタイプ

Sm\_PolicyApi\_DirType\_t は、ユーザの認証に使用可能なディレクトリのタイプを列挙します。

名前	値
Sm_PolicyApi_DirType_LDAP	1
Sm_PolicyApi_DirType_WinNT	2
Sm_PolicyApi_DirType_ODBC	3

## 証明書マッピング フラグ定義

`Sm_PolicyApi_CertMapFlags_t` は、証明書マッピング プロパティを表すフラグを列挙します。

フラグ	値
<p><code>Sm_PolicyApi_CertMapFlags_CertRequired</code></p> <p>このフラグをセットすると、<code>SiteMinder</code> は、ユーザから提示された証明書が、認証ディレクトリ内のユーザのエントリに保存されている証明書と一致することを確認します。認証ディレクトリは、LDAP ユーザ ディレクトリであることが必要です。</p>	0x01
<p><code>Sm_PolicyApi_CertMapFlags_UseDistributionPoints</code></p> <p>証明書失効リスト (CRL) が配布ポイントを使用する場合に、このフラグをセットします。大規模な CRL には配布ポイントが複数含まれています。これらの配布ポイントは失効したユーザの検索に使用できます。配布ポイントは CRL LDAP ディレクトリの開始点を示します。また、配布ポイントは CRL チェックの開始ポイントを示し、特定ユーザの検索に CRL 全体を対象とした場合にかかるであろう処理時間を短縮します。</p> <p>このフラグがセットされている場合、<code>SiteMinder</code> は、ユーザの証明書から配布ポイントを取得し、これを使用して、LDAP ディレクトリで CRL に適切なエントリポイントを検索します。</p>	0x02
<p><code>Sm_PolicyApi_CertMapFlags_VerifySignature</code></p> <p>署名の検証を有効にする場合に、このフラグをセットします。その場合、ポリシーサーバは、ポリシー データベースに保存された署名に照らして認証機関の公開証明書を確認します。</p>	0x04
<p><code>Sm_PolicyApi_CertMapFlags_CRLCheck</code></p> <p>このフラグをセットすると、<code>SiteMinder</code> が証明書失効リストの確認を実行します。証明書失効リスト (CRL) は、認証機関によって発行された取り消し済みの X.509 クライアント証明書のリストです。CRL と証明書を照合することは、証明書が有効なことを確認する 1 つの方法です。このような証明書を持つユーザが保護されたリソースにアクセスしようとする時、<code>SiteMinder</code> は、このユーザの証明書を CRL で検出し、認証を拒否します。</p>	0x08
<p><code>Sm_PolicyApi_CertMapFlags_Cache</code></p> <p>このフラグを設定すると、<code>SiteMinder</code> は CRL の <code>NextUpdate</code> フィールドで指定された日付までキャッシュされた CRL 情報を使用します。</p>	0x10

## ディレクトリ機能

Sm\_PolicyApi\_GetUserDirCapabilities() は、SmApi.h で定義された Sm\_DirectoryCapability\_t に列挙される値を使用します。

ディレクトリ機能	値
Sm_DirCapability_CreatePasswordPolicy パスワードポリシーを作成できます。以下の属性がユーザディレクトリ (Sm_PolicyApi_UserDir_t) で影響を受けます: <i>pszPasswordData</i> 、 <i>pszDisabledAttr</i> 、および <i>pszPasswordAttribute</i>	0x00000001
Sm_DirCapability_CreateRegistrationPolicy 登録ポリシーを作成できます。以下の属性がユーザディレクトリ (Sm_PolicyApi_UserDir_t) で影響を受けます: <i>pszAnonymousId</i> 、 <i>pszEmailAddressAttr</i> 、 <i>pszChallengeRespAtt</i> 、および <i>pszPasswordAttribute</i>	0x00000002
Sm_DirCapability_ResetUserPassword ユーザパスワードをリセットできます。これは <i>pszPasswordAttribute</i> に影響します。	0x00000004
Sm_DirCapability_ChangeUserPassword ユーザパスワードを変更できます。これは <i>pszPasswordAttribute</i> に影響します。	0x00000008
Sm_DirCapability_DisableUser ユーザアカウントを無効化できます。これは <i>pszDisabledAttr</i> に影響します。	0x00000010
Sm_DirCapability_DmsCapable 分散代行管理システム (DMS) によって書き込むことができます。	0x00000020
Sm_DirCapability_Recursive 再帰をサポートできます。	0x00000040
Sm_DirCapability_DisabledAttr 読み取り/書き込み無効属性。この属性は、ユーザディレクトリに対して設定されます。	0x00100000
Sm_DirCapability_UniversalIdAttr 読み取り専用ユニバーサル ID。この属性は、ユーザディレクトリに対して設定されます。	0x00200000

ディレクトリ機能	値
Sm_DirCapability_AnonymousIdAttr 読み取り/書き込み匿名 ID 属性。この属性は、ユーザディレクトリに対して設定されます。	0x00400000
Sm_DirCapability_PasswordDataAttr 読み取り/書き込みパスワードデータ属性。この属性は、ユーザディレクトリに対して設定されます。	0x00800000
Sm_DirCapability_UserPasswordAttr 読み取り/書き込みパスワード属性。この属性は、ユーザディレクトリに対して設定されます。	0x01000000
Sm_DirCapability_EmailAddressAttr 読み取り専用電子メール属性。この属性は、ユーザディレクトリに対して設定されます。	0x02000000
Sm_DirCapability_ChallengeRespAttr 読み取り/書き込みチャレンジおよびレスポンス属性。この属性は、ユーザディレクトリに対して設定されます。	0x04000000

注: 属性マスクは、ディレクトリ ユーザプロファイル属性です。これらは、ディレクトリ内で利用可能です。各属性は、読み取り専用属性または読み取り/書き込み属性です。読み取り/書き込み属性は他のアプリケーションでは使用されません。

## ドメイン フラグ

Sm\_PolicyApi\_DomainFlags\_t は、ドメイン全体の影響に関連するフラグを列挙します。

名前	値
Sm_PolicyApi_DomainFlags_GlobalPoliciesApply このフラグがセットされている場合、ドメインは、ドメイン内のすべてのレルムに対するグローバルポリシーを処理します。このフラグがセットされていない場合、ドメインはグローバルポリシーを処理しません。	0x02

## グループタイプ

`Sm_PolicyApi_Groups_t` は、グループ関数を実行可能なグループのタイプを列挙します。

名前	値
<code>Sm_PolicyApi_NULL_Group_Prop</code>	0
<code>Sm_PolicyApi_Rule_Group_Prop</code>	1
<code>Sm_PolicyApi_Response_Group_Prop</code>	2
<code>Sm_PolicyApi_Agent_Group_Prop</code>	3

## IP アドレスタイプ

`Sm_PolicyApi_IPAddressType_t` は、`Sm_PolicyApi_IPAddress_t` でオブジェクトに対して定義されている IP アドレス制限のタイプを列挙します。

IP アドレスタイプ	値
<p><code>Sm_PolicyApi_IPAddressType_SingleHost</code></p> <p>単一ホスト IP アドレスでは、以下のフィールドが設定されていることが必要です。</p> <ul style="list-style-type: none"> <li>■ <code>iStructId</code>。以下で定義される IP アドレス データ構造体 ID。 <code>Sm_PolicyApi_Structs_t</code></li> <li>■ <code>iIPAddressType</code>。IP アドレス タイプを <code>Sm_PolicyApi_IPAddressType_SingleHost</code> に設定します。</li> <li>■ <code>nIPAddress</code>。有効な IP アドレス。この IP アドレスは、ロング形式で指定されま す。</li> </ul>	1
<p><code>Sm_PolicyApi_IPAddressType_HostName</code></p> <p>ホスト名 IP アドレスでは、以下のフィールドが設定されていることが必要です。</p> <ul style="list-style-type: none"> <li>■ <code>iStructId</code>。以下で定義される IP アドレス データ構造体 ID。 <code>Sm_PolicyApi_Structs_t</code></li> <li>■ <code>iIPAddressType</code>。IP アドレス タイプを <code>Sm_PolicyApi_IPAddressType_HostName</code>。</li> <li>■ <code>pszHostName[BFSIZE]</code>。ユーザがアクション（たとえば、ポリシーの適用）を実 行するために使用していると判断されるマシンのホスト名。</li> </ul>	2

IP アドレス タイプ	値
<p>Sm_PolicyApi_IPAddressType_AddressAndSubNetMask</p> <p>サブネット マスクでは、以下のフィールドが設定されている必要があります。</p> <ul style="list-style-type: none"> <li>■ <i>iStructId</i>。以下で定義される IP アドレス データ構造体 ID。 Sm_PolicyApi_Structs_t</li> <li>■ <i>iIPAddressType</i>。IP アドレス タイプを Sm_PolicyApi_IPAddressType_AddressAndSubnetMask に設定します。</li> <li>■ <i>nIPAddress</i>。有効な IP アドレス。この IP アドレスは、ロング形式で指定されま す。</li> <li>■ <i>nSubnetMask</i>。サブネット マスクを指定します。</li> </ul>	3
<p>Sm_PolicyApi_IPAddressType_Range</p> <p>IP アドレスの範囲では、以下のフィールドが設定されている必要があります。</p> <ul style="list-style-type: none"> <li>■ <i>iStructId</i>。以下で定義される IP アドレス データ構造体 ID。 Sm_PolicyApi_Structs_t</li> <li>■ <i>iIPAddressType</i>。IP アドレス タイプを Sm_PolicyApi_IPAddressType_Range に設定します。</li> <li>■ <i>nIPAddress</i>。開始 IP アドレス この IP アドレスは、ロング形式で指定されま す。</li> <li>■ <i>nEndIPAddress</i>。終了 IP アドレス。この IP アドレスは、ロング形式で指定され ます。</li> </ul>	4

## 管理コマンド

`Sm_PolicyApi_ManagementCommands_t` は、キャッシュのクリア、エージェント暗号化キーの管理、および共有秘密キーのロールオーバーの目的で `Sm_PolicyApi_ManagementCommand()` に渡される可能性がある値を列挙します。

値を設定する前には、必ず構造体をゼロ (`memset`) に初期化してください。ハードコードされた整数コマンド値ではなく、シンボリック列挙型の値を使用してください。

値は、`Sm_PolicyApi_ManagementCommand_t` 構造体の `iCommand` フィールドに渡されます。

管理コマンド	値
<code>Sm_PolicyApi_ManagementCommand_FlushAll</code> すべての SiteMinder キャッシュをクリアします。ポリシーストア キャッシュ、リソース キャッシュ、およびユーザ情報キャッシュがこのコマンドによってクリアされます。 <code>Sm_PolicyApi_ManagementCommand_t</code> の <code>pszData</code> フィールドのデータは不要です。	1
<code>Sm_PolicyApi_ManagementCommand_FlushUsers</code> ユーザ情報キャッシュをクリアします。 <code>Sm_PolicyApi_ManagementCommand_t</code> の <code>pszData</code> フィールドのデータは不要です。	2
<code>Sm_PolicyApi_ManagementCommand_FlushRealms</code> リソース キャッシュをクリアします。 <code>Sm_PolicyApi_ManagementCommand_t</code> の <code>pszData</code> フィールドのデータは不要です。	3
<code>Sm_PolicyApi_ManagementCommand_ChangeDynamicKeys</code> ダイナミック エージェント キーを変更します。 <code>Sm_PolicyApi_ManagementCommand_t</code> の <code>pszData</code> フィールドのデータは不要です。 C API を使用してダイナミック エージェント キーを変更する前に、[ポリシー サーバ キー管理] ダイアログ ボックスの [エージェント キー] 設定が [ダイナミック エージェント キーを使用] に設定される必要があります。ポリシー サーバ UI でこのダイアログ ボックスにアクセスするには、[ツール] > [キーの管理] をクリックします。次に、[エージェント キー] タブで、[ダイナミック エージェント キーを使用] を選択します。	4

管理コマンド	値
Sm_PolicyApi_ManagementCommand_ChangePersistentKey 永続キーまたはスタティック キーを変更します。 Sm_PolicyApi_ManagementCommand_t 構造体のデータ フィールド <i>pszData</i> には、オプションのキー値が含まれる場合があります。 <i>pszData</i> が空の場合、永続キーはランダムに生成されます。	5
Sm_PolicyApi_ManagementCommand_ChangeSessionKey セッション キーを変更します。Sm_PolicyApi_ManagementCommand_t 構造体のデータ フィールド <i>pszData</i> には、オプションのキー値が含まれる場合があります。 <i>pszData</i> が空の場合、セッション キーがランダムに生成されます。	6
Sm_PolicyApi_ManagementCommand_RolloverSharedSecrets ロールオーバーが有効化されたトラステッド ホストに対して共有秘密キーをロールオーバーします。	7

## パスワード メッセージ

Sm\_PolicyApi\_PasswordMsgId\_t はパスワード メッセージ ID を列挙します。

パスワード メッセージには、指定されたディレクトリのパスワード ポリシー要件を新しいパスワードが満たさない場合に Sm\_PolicyApi\_SetPassword() に返されるエンコード済みエラー メッセージが記述されています。

パスワード メッセージ ID	値
Sm_PolicyApi_PasswordMsgId_None	0
Sm_PolicyApi_PasswordMsgId_ChangePassword	1
Sm_PolicyApi_PasswordMsgId_PassswordGeneralFailure	1000
Sm_PolicyApi_PasswordMsgId_PasswordShort	1001
Sm_PolicyApi_PasswordMsgId_PasswordLong	1002
Sm_PolicyApi_PasswordMsgId_PasswordOldPasswordBad	1003
Sm_PolicyApi_PasswordMsgId_PasswordReuse	1004
Sm_PolicyApi_PasswordMsgId_PasswordSimilar	1005
Sm_PolicyApi_PasswordMsgId_PasswordRepeatingChars?	1006

パスワード メッセージ ID	値
Sm_PolicyApi_PasswordMsgId_PasswordDictionaryMatch	1007
Sm_PolicyApi_PasswordMsgId_PasswordContentLetters	1008
Sm_PolicyApi_PasswordMsgId_PasswordContentDigits	1009
Sm_PolicyApi_PasswordMsgId_PasswordContentAlphaNum	1010
Sm_PolicyApi_PasswordMsgId_PasswordContentPunctuation	1011
Sm_PolicyApi_PasswordMsgId_PasswordCont は entNonPrintable です	1012
Sm_PolicyApi_PasswordMsgId_PasswordContentNonAlphaNum	1013
Sm_PolicyApi_PasswordMsgId_PasswordProfileMatch	1014
Sm_PolicyApi_PasswordMsgId_PasswordGraceDays	1015
Sm_PolicyApi_PasswordMsgId_PasswordSystemPIN	1016
Sm_PolicyApi_PasswordMsgId_PasswordUserMaxNumPIN	1017
Sm_PolicyApi_PasswordMsgId_PasswordUserMinMaxNumPIN	1018
Sm_PolicyApi_PasswordMsgId_PasswordUserMaxAlphaPIN	1019
Sm_PolicyApi_PasswordMsgId_PasswordUserMinMaxAlphaPIN	1020
Sm_PolicyApi_PasswordMsgId_PasswordAcceptPIN	1021
Sm_PolicyApi_PasswordMsgId_PasswordContentLowerAlpha	1022
Sm_PolicyApi_PasswordMsgId_PasswordContentUpperAlpha	1023
Sm_PolicyApi_PasswordMsgId_PasswordContentNoLowerAlpha	1024
Sm_PolicyApi_PasswordMsgId_PasswordContentNoUpperAlpha	1025
Sm_PolicyApi_PasswordMsgId_PasswordContentNoDigits	1026
Sm_PolicyApi_PasswordMsgId_PasswordContentNoPunctuation	1027
Sm_PolicyApi_PasswordMsgId_PasswordCont は entNoNonPrintable です	1028
Sm_PolicyApi_PasswordMsgId_PasswordContentNoNonAlphaNum	1029
Sm_PolicyApi_PasswordMsgId_PasswordContentNoAlphaNum	1030
Sm_PolicyApi_PasswordMsgId_PasswordContentMatchRegExp	1031
Sm_PolicyApi_PasswordMsgId_PasswordContentNoMatchRegExp	1032
Sm_PolicyApi_PasswordMsgId_PasswordUserMinNumPIN	1033
Sm_PolicyApi_PasswordMsgId_PasswordUserDigitsPIN	1034

パスワード メッセージ ID	値
Sm_PolicyApi_PasswordMsgId_PasswordUserAlphaNumPIN	1035

エラー メッセージに関する追加の情報は、パスワード メッセージに関連付けられたパスワード メッセージ フィールドで入手できます。

## パスワード メッセージ フィールド

Sm\_PolicyApi\_PasswordMsgFieldId\_t はパスワード メッセージ フィールド ID を列挙します。

パスワード メッセージ フィールドには、前セクションのパスワード メッセージに関する追加の情報が含まれます。この追加情報は、構造体 Sm\_PolicyApi\_PasswordMsgField\_t にあります。

パスワード メッセージ フィールド ID	値
Sm_PolicyApi_PasswordMsgFieldId_None	0
Sm_PolicyApi_PasswordMsgFieldId_Min	1
Sm_PolicyApi_PasswordMsgFieldId_Max	2
Sm_PolicyApi_PasswordMsgFieldId_OldPW	3
Sm_PolicyApi_PasswordMsgFieldId_NewPW	4
Sm_PolicyApi_PasswordMsgFieldId_Days	5
Sm_PolicyApi_PasswordMsgFieldId-Token	6

フィールドには、整数型または文字列型があります。また、型指定なしの場合もあります。

## パスワード メッセージ フィールド タイプ

Sm\_PolicyApi\_FieldType\_t は、パスワード メッセージ フィールドの推定されるデータ型を列挙します。

パスワード メッセージ フィールド タイプ	値
Sm_PolicyApi_FieldType_None	0

パスワード メッセージ フィールド タイプ	値
Sm_PolicyApi_FieldType_Int	1
Sm_PolicyApi_FieldType_String	2

## パスワード ポリシー 動作フラグ

Sm\_PasswordPolicyBehavior\_t は、パスワード ポリシーの動作特性を列挙します。

パスワード ポリシー 動作フラグ	値
Sm_PasswordPolicy_DontTrackLogins このフラグは、SiteMinder v6.0 SP3 で以下のフラグと置き換えられています。	0x00000004
<ul style="list-style-type: none"> <li>■ Sm_PasswordPolicy_DontTrackSuccessLogins</li> <li>■ Sm_PasswordPolicy_DontTrackFailedLogins</li> </ul> 新しいフラグにより、成功したログインと失敗したログインを別々に追跡可能になりました。	
Sm_PasswordPolicy_DontTrackLogins は、下位互換性を目的として現在も残されています。このフラグを設定すると、ログインの成功および失敗に関するログイン追跡は行われません。	
Sm_PasswordPolicy_AllowFailedWrites パスワードデータをユーザディレクトリに書き込むことができない場合にも、ユーザがログインできるようにします。	0x00000008
Sm_PasswordPolicy_InactivityForcePWChange ユーザパスワードが非アクティブ状態により無効になった後、次のログイン試行時にパスワード変更を要求します。	0x00000010
Sm_PasswordPolicy_PWExpiredForcePWChange ユーザパスワードが期限切れになった後、次のログイン試行時にパスワード変更を要求します。	0x00000020

パスワード ポリシー動作フラグ	値
<p><b>Sm_PasswordPolicyBehavior_FullReenable</b></p> <p>このフラグは、間違っただパスワードが連続して入力されたためにユーザアカウント無効化された場合に、所定の期間後にアカウントを再度有効にします。 <b>Sm_PolicyApi_PasswordPolicy_t</b> の <i>nReenablement</i> フィールドに時間を指定してください。</p> <p>このフラグをセットしない場合、ユーザは、所定の <i>nReenablement</i> の期間後に別のログイン試行が可能になります。</p>	0x00000040
<p><b>Sm_PasswordPolicy_StopPriorityChaining</b></p> <p>現行パスワードポリシーより優先度の低いパスワードポリシーの評価を回避します。</p>	0x00000080
<p><b>Sm_PasswordPolicy_ExpireDisablePassword</b></p> <p>パスワードが期限切れの場合、ユーザアカウントではなくパスワードのみを無効化します。</p>	0x00000100
<p><b>Sm_PasswordPolicy_FailuresDisablePassword</b></p> <p>認証失敗の最大回数を超過した場合、ユーザアカウントではなくパスワードのみを無効化します。</p>	0x00000200
<p><b>Sm_PasswordPolicy_ForceCase</b></p> <p><b>Sm_PasswordPolicy_CaseSelect</b> ビットで指定されているパスワードの大文字と小文字の使い分けが適用されます。</p>	0x00000400
<p><b>Sm_PasswordPolicy_CaseSelect</b></p> <p><b>Sm_PasswordPolicy_ForceCase</b> が設定されている場合、設定すると <b>Sm_PasswordPolicy_ForceCase</b> は大文字のパスワードを強制し、クリアすると小文字のパスワードを強制します。</p>	0x00000800
<p><b>Sm_PasswordPolicy_CaseBits</b></p> <p>以下のビットの両方をセットします(大文字のパスワードを要求します)。</p> <ul style="list-style-type: none"> <li>■ <b>Sm_PasswordPolicy_ForceCase</b></li> <li>■ <b>Sm_PasswordPolicy_CaseSelect</b></li> </ul>	0x00000c00
<p><b>Sm_PasswordPolicy_StripLeadingWhiteSpace</b></p> <p>パスワードから先頭のスペースをすべて削除します。</p>	0x00001000
<p><b>Sm_PasswordPolicy_StripTrailingWhiteSpace</b></p> <p>パスワードから末尾のスペースをすべて削除します。</p>	0x00002000

パスワード ポリシー動作フラグ	値
<b>Sm_PasswordPolicy_StripFlankingWhiteSpace</b> 以下のビットの両方をセットします (先頭および末尾のスペースを削除します)。 <ul style="list-style-type: none"> <li>■ Sm_PasswordPolicy_StripLeadingWhiteSpace</li> <li>■ Sm_PasswordPolicy_StripTrailingWhiteSpace</li> </ul>	0x00003000
<b>Sm_PasswordPolicy_StripEmbeddedWhiteSpace</b> パスワード内のスペースをすべて削除します。	0x00004000
<b>Sm_PasswordPolicy_WhiteSpaceBits</b> 以下のビットをすべてセットします (先頭、末尾、および文字間のスペースを削除します)。 <ul style="list-style-type: none"> <li>■ Sm_PasswordPolicy_StripLeadingWhiteSpace</li> <li>■ Sm_PasswordPolicy_StripTrailingWhiteSpace</li> <li>■ Sm_PasswordPolicy_StripEmbeddedWhiteSpace</li> </ul>	0x00007000
<b>Sm_PasswordPolicy_PreProcessBits</b> 以下のビットをすべてセットします (大文字のパスワードを要求し、先頭、末尾、および文字間のスペースを削除します)。 <ul style="list-style-type: none"> <li>■ Sm_PasswordPolicy_ForceCase</li> <li>■ Sm_PasswordPolicy_CaseSelect</li> <li>■ Sm_PasswordPolicy_StripLeadingWhiteSpace</li> <li>■ Sm_PasswordPolicy_StripTrailingWhiteSpace</li> <li>■ Sm_PasswordPolicy_StripEmbeddedWhiteSpace</li> </ul>	0x00007c00
<b>Sm_PasswordPolicy_DontTrackSuccessLogins</b> ログイン時にディレクトリ更新を実行します。このフラグがセットされていない場合、パスワード ポリシーは、最終のログイン時刻など、成功したユーザを追跡します。	0x00008000
<b>Sm_PasswordPolicy_DontTrackFailedLogins</b> ログイン時にディレクトリ更新を実行します。このフラグがセットされていない場合、パスワード ポリシーは、失敗したユーザ ログインの試行を追跡します。	0x00010000

注: 値 0x00000400 から 0x00007c00 までは、パスワードのプリプロセスに利用されます。プリプロセス中、パスワードの処理や保存が行われる前に、パスワードが確認されます。

## ポリシー フラグ

Sm\_PolicyApi\_AddUsersToPolicy() は、以下の値を使用します（これらは SmApi.h で定義されています）。

フラグ	値
Sm_PolicyBehavior_Exclude_Mask	0x01
ビット 0x01 により、ユーザ ポリシーが「ユーザ」を除外するか含めるかが決定されます。	
Sm_PolicyBehavior_Exclude_No	0x00
Sm_PolicyBehavior_Exclude_Yes	0x01
Sm_PolicyBehavior_Recursive_Mask	0x02
ビット 0x02 により、ユーザ ポリシーが再帰的かどうか決定されます。これは、ネスト可能なディレクトリ オブジェクトクラスに適用できます。	
Sm_PolicyBehavior_Recursive_No	0x00
Sm_PolicyBehavior_Recursive_Yes	0x02
Sm_PolicyBehavior_AND_Mask	0x04
ビット 0x04 により、ユーザ ポリシー間に AND 関係があるかどうか決定されます。これは、ポリシー内の特定のユーザ ディレクトリのメンバであるユーザ ポリシーに適用可能です。	
Sm_PolicyBehavior_AND_No	0x00
Sm_PolicyBehavior_AND_Yes	0x04

## ポリシー管理 API 初期化フラグ

Sm\_PolicyApi\_InitFlags\_t は、Sm\_PolicyApi\_Init() によって使用される初期化フラグを列挙します。これらのフラグは、API 動作に影響を与えます。

フラグ	値
Sm_PolicyApi_InitFlags_EnableCache	0x01
SiteMinder がユーザの要求に迅速に回答できるようにするために、ポリシーストア、リソース、ユーザ情報のキャッシュを有効化します。	

フラグ	値
Sm_PolicyApi_InitFlags_PreLoadCache	0x02
<p>ポリシー管理 API が SiteMinder キャッシュを事前ロードできるようにします。</p> <p>注: このフラグを省略することによって、カスタム ポリシー管理アプリケーションがポリシーストアを変更する時間を短縮できます。</p>	
Sm_PolicyApi_InitFlags_LoadAgentTypeDictionary	0x04
<p>ポリシー管理 API が SiteMinder エージェント タイプ辞書を事前ロードできるようにします。</p>	
Sm_PolicyApi_InitFlags_DisableValidation	0x08
<p>ポリシー オブジェクトの検証を無効にします。</p>	
Sm_PolicyApi_InitFlags_DisableAudit	0x10
<p>無効にします。</p> <ul style="list-style-type: none"> <li>■ 認証、認可、管理アクティビティなど、ユーザアクティビティの監査。（管理アクティビティには、ポリシーストアへの変更が含まれます）。</li> <li>■ ユーザセッションの監視。</li> </ul>	
Sm_PolicyApi_InitFlags_DisableCacheUpdates	0x20
<p>キャッシュの更新を無効にします。キャッシュ更新が無効化されておらず、Sm_PolicyApi_InitFlags_EnableCache がオフの場合、ポリシー管理 API は引き続きキャッシュ更新を発行します。</p>	
Sm_PolicyApi_InitFlags_DisableManagementWatchDog	0x40
<p>SiteMinder 管理 Watchdog を無効にします。Watchdog は、デフォルトで有効化されています。ウォッチドッグは内部的に使用され無効にはしません。</p>	

## ポリシー オブジェクト ID

Sm\_PolicyApi\_Objects\_t では、取得、設定、削除できるポリシーストアのプロパティが記述されています。

注: 値 0 の Sm\_PolicyApi\_NULL\_Domain\_Props は予約済みです。

以下の表には、Sm\_PolicyApi\_GetDomainObjects() に渡すことができるドメイン オブジェクト タイプの値を示します。

名前	値
Sm_PolicyApi_Rule_Prop	1
Sm_PolicyApi_RuleGroup_Prop	2
Sm_PolicyApi_Policy_Prop	3
Sm_PolicyApi_PolicyLink_Prop	4
Sm_PolicyApi_UserPolicy_Prop	5
Sm_PolicyApi_Realm_Prop	6
Sm_PolicyApi_ResponseGroup_Prop	7
Sm_PolicyApi_Response_Prop	8
Sm_PolicyApi_ResponseAttr_Prop	9
Sm_PolicyApi_UserDir_Prop	10
Sm_PolicyApi_Admins_Prop	17
Sm_PolicyApi_ActiveExpr_Prop	23
Sm_PolicyApi_Variable_Prop	25
Sm_PolicyApi_Affiliate_Prop	33
Sm_PolicyApi_SAMLSP_Prop	35

以下の表には、Sm\_PolicyApi\_GetGlobalObjects() に渡すことができるグローバル オブジェクト タイプの名前を示します。

名前	値
Sm_PolicyApi_Rule_Prop	1
Sm_PolicyApi_Policy_Prop	3

名前	値
Sm_PolicyApi_Response_Prop	8
Sm_PolicyApi_UserDir_Prop	10
Sm_PolicyApi_Scheme_Prop	11
認証方式のオブジェクト ID。	
Sm_PolicyApi_Agent_Prop	12
Sm_PolicyApi_AgentGroup_Prop	13
Sm_PolicyApi_AgentType_Prop	14
Sm_PolicyApi_AgentTypeAttr_Prop	15
Sm_PolicyApi_Domain_Prop	16
Sm_PolicyApi_Admins_Prop	17
Sm_PolicyApi_ODBCQueryScheme_Prop	18
Sm_PolicyApi_RegistrationScheme_Prop	19
Sm_PolicyApi_PasswordPolicy_Prop	20
Sm_PolicyApi_AuthAzMap_Prop	21
認証認可オブジェクトのオブジェクト ID。	
Sm_PolicyApi_CertMap_Prop	22
認定マッピング オブジェクトのオブジェクト ID。	
Sm_PolicyApi_VariableType_Prop	24
Sm_PolicyApi_TrustedHost_Prop	26
Sm_PolicyApi_HostConfig_Prop	27
Sm_PolicyApi_AgentConfig_Prop	28
Sm_PolicyApi_Association_Prop	29
エージェント設定オブジェクト内の設定名/値ペアのオブジェクト ID。	
Sm_PolicyApi_AffiliateDomain_Prop	32
Sm_PolicyApi_SharedSecretPolicy_Prop	34
Sm_PolicyApi_SAMLIdP_Prop	36
Sm_PolicyApi_SAMLAffiliation_Prop	37
Sm_PolicyApi_WSFEDResourcePartner_Prop	38

## ポリシー解決

`Sm_PolicyResolution_t` は、`SmApi.h` で定義され、2 つのポリシー オブジェクト間の関係を説明する値を列挙します。

詳細情報:

[Sm\\_PolicyResolution\\_t](#) (P. 870)

## リターンコード

API によって返すことができる値コードは、`Sm_PolicyApi_Status_t` で列挙されています。値の意味は以下のとおりです。

- ゼロのリターンコードは成功を示します。
- 負のリターンコードは失敗を示します。

ほとんどのコード名は自明です。ただし、必須の入力パラメータの 1 つまたは複数を提供されない場合には、`Sm_PolicyApi_BadArgument (-10)` が返される点に注意が必要です。たとえば、ドメイン OID などの引数が `Null` の場合や長さがゼロの文字列を表す場合、`Sm_PolicyApi_BadArgument` がコール元に返されます。

値が -100 未満のリターンコード (値が -105 の `Sm_PolicyApi_NotUnique` は除きます) は、この API で返されることはほとんどありません。ここでは、すべてを網羅するために、それらも含めています。

リターンコード	値
<code>Sm_PolicyApi_Success</code>	0
<code>Sm_PolicyApi_Failure</code>	-1
<code>Sm_PolicyApi_InvalidHandle</code>	-2
<code>Sm_PolicyApi_ErrorLogin</code>	-3
<code>Sm_PolicyApi_NoPrivilege</code>	-4
<code>Sm_PolicyApi_InvalidPasswordSyntax</code>	-5
<code>Sm_PolicyApi_InvalidPassword</code>	-6

Sm_PolicyApi_DuplicateEntry	-7
Sm_PolicyApi_DoesNotExist	-8
Sm_PolicyApi_NotFound	-9
Sm_PolicyApi_BadArgument	-10
Sm_PolicyApi_WrongNumberOfElements	-11
Sm_PolicyApi_UserDirNotPartOfDomain	-12
Sm_PolicyApi_UserDirNotValid	-13
Sm_PolicyApi_ErrorUserDir	-14
Sm_PolicyApi_AgentNotFound	-15
Sm_PolicyApi_AgentTypeNotFound	-16
Sm_PolicyApi_AgentTypeAttrNotFound	-17
Sm_PolicyApi_AgentTypeMismatch	-18
Sm_PolicyApi_ODBCQuerySchemeNotFound	-19
Sm_PolicyApi_UserDirNotFound	-20
Sm_PolicyApi_DomainNotFound	-21
Sm_PolicyApi_AdminNotFound	-22
Sm_PolicyApi_SchemeNotFound	-23
Sm_PolicyApi_RegistrationSchemeNotFound	-24
Sm_PolicyApi_PasswordPolicyNotFound	-25
Sm_PolicyApi_SchemelsRequired	-26
Sm_PolicyApi_PasswordPolicyConfig	-27
Sm_PolicyApi_RealmNotFound	-28
Sm_PolicyApi_NoChildren	-29
Sm_PolicyApi_RuleNotFound	-30
Sm_PolicyApi_ResponseNotFound	-31
Sm_PolicyApi_ResponseAttrNotFound	-32
Sm_PolicyApi_PolicyNotFound	-33
Sm_PolicyApi_PolicyLinkNotFound	-34
Sm_PolicyApi_UserPolicyNotFound	-35
Sm_PolicyApi_BadGroup	-36

Sm_PolicyApi_GroupNotFound	-37
Sm_PolicyApi_Invalid	-38
Sm_PolicyApi_InvalidHandleVersion	-39
Sm_PolicyApi_DomainNotAffiliate	-41
Sm_PolicyApi_InvalidOid	-100
Sm_PolicyApi_NotImplemented	-101
Sm_PolicyApi_NotSearchable	-102
Sm_PolicyApi_NotStorable	-103
Sm_PolicyApi_NotCollection	-104
Sm_PolicyApi_NotUnique	-105
Sm_PolicyApi_InvalidProp	-106
Sm_PolicyApi_NotInitted	-107
Sm_PolicyApi_NoSession	-108
Sm_PolicyApi_OidInUseByRealm	-109
Sm_PolicyApi_OidInUseByRule	-110
Sm_PolicyApi_OidInUseByAdmin	-111
Sm_PolicyApi_MissingProperty	-112
Sm_PolicyApi_GroupMemberName	-113
Sm_PolicyApi_RadiusIpAddrNotUnique	-114
Sm_PolicyApi_GroupAgentType	-115
Sm_PolicyApi_RadiusRealmNotUnique	-116
Sm_PolicyApi_RealmFilterNotUnique	-117
Sm_PolicyApi_InvalidCharacters	-118
Sm_PolicyApi_AgentTypeCantBeDeleted	-119
Sm_PolicyApi_ProvNotImplemented	-120
Sm_PolicyApi_ProvNotUnique	-121
Sm_PolicyApi_RealmCantBeUsedInRule	-122
Sm_PolicyApi_OidInUserByCertMap	-123
Sm_PolicyApi_OidInUseBySelfReg	-124
Sm_PolicyApi_OidInUseByUserDirectory	-125

Sm_PolicyApi_SchemeCantBeDeleted	-126
Sm_PolicyApi_BasicSchemeUpdate	-127
Sm_PolicyApi_NonHtmlForm	-128
Sm_PolicyApi_IllegalRealmOperation	-129
Sm_PolicyApi_NameNotUnique	-130
Sm_PolicyApi_FeatureNotSupported	-132
Sm_PolicyApi_AssertionConsumerDefaultMissing	-133
Sm_PolicyApi_SAMLSP_AuthenticationURLMissing	-134
Sm_PolicyApi_SAMLSP_DomainOidMissing	-135
Sm_PolicyApi_SAMLSP_IdPIDMissing	-136
Sm_PolicyApi_SAMLSP_NameMissing	-137
Sm_PolicyApi_SAMLSP_NameldFormatMissing	-138
Sm_PolicyApi_SAMLSP_NameldTypeMissing	-139
Sm_PolicyApi_SAMLSP_NameldStaticMissing	-140
Sm_PolicyApi_SAMLSP_NameldAttrNameMissing	-141
Sm_PolicyApi_SAMLSP_NameldDNSpecMissing	-142
Sm_PolicyApi_SAMLSP_ProviderIDMissing	-143
Sm_PolicyApi_SAMLSP_ProviderIDNotUnique	-144
Sm_PolicyApi_SAML_UnSupportedSAMLVersion	-145
Sm_PolicyApi_SAMLIDP_IncorrectParameters	-146
Sm_PolicyApi_SAMLIDP_ProviderIDNotUnique	-147
Sm_PolicyApi_SAMLAFF_NameMissing	-148
Sm_PolicyApi_SAMLAFF_NameldFormatMissing	-149
Sm_PolicyApi_SAMLAFF_NameldTypeMissing	-150
Sm_PolicyApi_SAMLAFF_NameldStaticMissing	-151
Sm_PolicyApi_SAMLAFF_NameldAttrNameMissing	-152
Sm_PolicyApi_SAMLAFF_NameldDNSpecMissing	-153
Sm_PolicyApi_SAMLAFF_AffiliationIDMissing	-154
Sm_PolicyApi_SAMLAFF_AffiliationIDNotUnique	-155
Sm_PolicyApi_SAMLAFF_AffiliationHasMembers	-156

Sm_PolicyApi_SAML_UnknownProperty	-157
Sm_PolicyApi_WSFEDRP_AssertionConsumerDefaultMissing	-158
Sm_PolicyApi_WSFEDRP_AuthenticationURLMissing	-159
Sm_PolicyApi_WSFEDRP_DomainOidMissing	-160
Sm_PolicyApi_WSFEDRP_APIDMissing	-161
Sm_PolicyApi_WSFEDRP_NameMissing	-162
Sm_PolicyApi_WSFEDRP_NameIdFormatMissing	-163
Sm_PolicyApi_WSFEDRP_NameIdTypeMissing	-164
Sm_PolicyApi_WSFEDRP_NameIdStaticMissing	-165
Sm_PolicyApi_WSFEDRP_NameIdAttrNameMissing	-166
Sm_PolicyApi_WSFEDRP_NameIdDNSpecMissing	-167
Sm_PolicyApi_WSFEDRP_ProviderIdMissing	-168
Sm_PolicyApi_WSFEDRP_ProviderIdNotUnique	-169
Sm_PolicyApi_WSFEDRP_UnsupportedSAMLVersion	-170
Sm_PolicyApi_WSFEDRP_UnkownProperty	-171
Sm_PolicyApi_WSFEDAP_IncorrectParameters	-172
Sm_PolicyApi_WSFEDAP_ProviderIDNotUnique	-173
Sm_PolicyAPI_InsufficientRPData	-174
Sm_PolicyAPI_WSFED_UnSupportedWSFEDVersion	-175
Sm_PolicyAPI_DuplicateAttribute	-176
Sm_PolicyAPI_SAMLSP_ACSDuplicateIndex	-177
Sm_PolicyAPI_SAMLSP_ACSIndexedEndpointInUse	-178
Sm_PolicyAPI_SAMLSP_ACSIndexedEndpointNotFound	-179
Sm_PolicyAPI_SAMLSP_CantDeleteDefaultACSIndex	-180
Sm_PolicyAPI_SAMLSP_ACSMaxExceeded	-181
Sm_PolicyAPI_InConsistentANDBitMask	-182

---

## SAML1x リダイレクト URL タイプ

Sm\_PolicyApi\_SAML1\_STATUS\_REDIRECT\_URL\_TYPE\_t では、Sm\_PolicyApi\_AddRedirectURLToSAML1xScheme() および Sm\_PolicyApi\_GetRedirectURLFromSAML1xScheme() で指定されるリダイレクトのタイプが定義されています。

Sm\_PolicyApi\_SAML1\_STATUS\_REDIRECT\_URL\_TYPE\_t は SmPolicyApi45.h でリスト表示されます。

名前	値
Sm_PolicyApi_SAML1_STATUS_REDIRECT_URL_USER_NOT_FOUND_TYPE	0
Sm_PolicyApi_SAML1_STATUS_REDIRECT_URL_INVALID_SSO	1
Sm_PolicyApi_SAML1_STATUS_REDIRECT_URL_UNACCEPTABLE_USER_CREDENTIALS	2

## SAML アサーション コンシューマ サービス バインディング

以下の値は、[アサーション コンシューマ サービス] の各行に対して指定できる SAML プロトコルのバインドです。

名前	値
Sm_PolicyApi_SAMLSP_HTTP_Post	0
Sm_PolicyApi_SAMLSP_HTTP_Artifact	1
Sm_PolicyApi_SAMLSP_PAOS	2

## SAML 属性名の形式識別子

`Sm_PolicyApi_SAMLSPAttrNameFormat_t` では、プリンシパルに適用される属性の指定に使用する形式が定義されています。形式の指定は `Sm_PolicyApi_SAMLSPAttr_t` 構造内に作成されます。

形式識別子は SAML 2.0 標準で定義されます。

`Sm_PolicyApi_SAMLSPAttrNameFormat_t` は、`SmPolicyApi45.h` にリストされています。

名前	値
<code>Sm_PolicyApi_SAMLSP_Unspecified</code>	0
<code>Sm_PolicyApi_SAMLSP_URI</code>	1
<code>Sm_PolicyApi_SAMLSP_Basic</code>	2

## SAML プロファイル

`Sm_PolicyApi_SAML_Profile_t` は、特定のアフィリエイトオブジェクトに対する SAML アサーションの送受信に使用される通信プロファイルを指定します。プロファイルは `Sm_PolicyApi_Affiliate_t` 構造の属性の 1 つとして指定されます。`Sm_PolicyApi_SAML_Profile_t` は `SmPolicyApi45.h` でリスト表示されます。

名前	値
<code>Sm_PolicyApi_SAML_Profile_Artifact</code>	1
<code>Sm_PolicyApi_SAML_Profile_POST</code>	2

## 方式タイプ

`Sm_Api_SchemeType_t` は、`SmPolicyApi_Scheme_t` 構造体の属性の 1 つとして `Sm_PolicyApi_AddScheme()` に渡せる値を説明します。

`Sm_Api_SchemeType_t` は `SmApi.h` にリストされています。

方式タイプ	値
<code>Sm_Api_SchemeType_Basic</code>	1

Sm_Api_SchemeType_CryptoCard	2
Sm_Api_SchemeType_Encotone	3
Sm_Api_SchemeType_HTMLForm	4
Sm_Api_SchemeType_BasicOverSSL	5
Sm_Api_SchemeType_RadiusServer	6
Sm_Api_SchemeType_SafeWordServer	7
Sm_Api_SchemeType_ACEServer	8
Sm_Api_SchemeType_X509ClientCert	9
Sm_Api_SchemeType_X509ClientCertAndBasic	10
Sm_Api_SchemeType_X509ClientCertOrBasic	11
Sm_Api_SchemeType_RadiusChapPap	12
Sm_Api_SchemeType_Anonymous	13
Sm_Api_SchemeType_NTLM	14
Sm_Api_SchemeType_Custom	15
Sm_Api_SchemeType_ACEServerHTMLForm	16
Sm_Api_SchemeType_SafeWordHTMLForm	17
Sm_Api_SchemeType_XMLDsig	18
Sm_Api_SchemeType_X509ClientCertOrForm	19
Sm_Api_SchemeType_X509ClientCertAndForm	20
Sm_Api_SchemeType_MSPassport	21
Sm_Api_SchemeType_XMLDocumentCredentialCollector	22
Sm_Api_SchemeType_SAMLSessionTicket	25
Sm_Api_SchemeType_SAMLArtifact	26
Sm_Api_SchemeType_Impersonation	27
Sm_Api_SchemeType_SAMLPOST	28
Sm_Api_SchemeType_SAML2	29
Sm_Api-SchemeType_WSFED	30

## 共有秘密キーのロールオーバー パラメータ

`Sm_PolicyApi_SecretRolloverPeriod_t` は、ロールオーバーの頻度設定と結合する場合、共有秘密キーのロールオーバーの発生頻度を決定する時間単位を列挙します。たとえば、ロールオーバー期間が `RolloverHOURS` で、頻度 `12` の場合、共有秘密キーは `12` 時間ごとに変更されることを意味します。

ロールオーバーの期間は `Sm_PolicyApi_SharedSecretPolicy_t` 構造体の `iRolloverPeriod` フィールドで定義されており、頻度は `iRolloverFrequency` フィールドで定義されています。

名前	値
<code>RolloverNEVER</code>	0
<code>RolloverHOURS</code>	1
<code>RolloverDAYS</code>	2
<code>RolloverWEEKS</code>	3
<code>RolloverMONTHS</code>	4

## 構造体 ID

`Sm_PolicyApi_Structs_t` は、ポリシー管理 API に渡す、または渡されるデータ構造を列挙します。列挙される項目は以下のとおりです。

名前	値
<code>Sm_PolicyApi_NULL_ID</code>	0
<code>Sm_PolicyApi_Rule_ID</code>	1
<code>Sm_PolicyApi_Policy_ID</code>	2
<code>Sm_PolicyApi_Realm_ID</code>	3
<code>Sm_PolicyApi_Response_ID</code>	4
<code>Sm_PolicyApi_UserDir_ID</code>	5
<code>Sm_PolicyApi_Agent_ID</code>	6
<code>Sm_PolicyApi_Domain_ID</code>	7
<code>Sm_PolicyApi_PolicyLink_ID</code>	8

名前	値
Sm_PolicyApi_ResponseAttr_ID	9
Sm_PolicyApi_User_ID	10
Sm_PolicyApi_Scheme_ID	11
Sm_PolicyApi_Admin_ID	12
Sm_PolicyApi_Group_ID	13
Sm_PolicyApi_ODBCQueryScheme_ID	14
Sm_PolicyApi_Object_ID	15
Sm_PolicyApi_AgentType_ID	16
Sm_PolicyApi_AgentTypeAttr_ID	17
Sm_PolicyApi_RegistrationScheme_ID	18
Sm_PolicyApi_PasswordPolicy_ID	19
Sm_PolicyApi_IPAddress_ID	20
Sm_PolicyApi_AuthAzMap_ID	21
Sm_PolicyApi_CertMap_ID	22
Sm_PolicyApi_PasswordMsgField_ID	23
Sm_PolicyApi_VariableType_ID	25
Sm_PolicyApi_Variable_ID	26
Sm_PolicyApi_TrustedHost_ID	27
Sm_PolicyApi_HostConfig_ID	28
Sm_PolicyApi_AgentConfig_ID	29
Sm_PolicyApi_Association_ID	30
Sm_PolicyApi_UserContext_ID	31
Sm_PolicyApi_Affiliate_ID	36
Sm_PolicyApi_AffiliateAttr_ID	37
Sm_PolicyApi_SharedSecretPolicy_ID	38
Sm_PolicyApi_UserContext_ID	40
Sm_PolicyApi_SAMLSP_ID	41
Sm_PolicyApi_SAMLProviderProp_ID	42

名前	値
Sm_PolicyApi_SAMLAffiliation_ID	43
Sm_PolicyApi_SAMLSPAttr_ID	44
Sm_PolicyApi_WSFEDResourcePartner_ID	45
Sm_PolicyApi_WSFEDProviderProp_ID	46
Sm_PolicyApi_WSFEDRPAttr_ID	47
Sm_PolicyApi_SAMLRequesterAttr_ID	48
Sm_PolicyApi_SAMLSPAssertionConsumerService_ID	49

## ポリシー アプリケーションの構造体

ポリシー アプリケーションは以下の操作を実行する必要があります。

- **初期化**： Sm\_PolicyApi\_Init() または Sm\_PolicyApi\_InitEx() が、API クライアントセッションによって呼び出される最初の関数になります。この関数が、SiteMinder ポリシー ストアへの接続を初期化し、*init handle* を確立します。（*init handle* は Sm\_PolicyApi\_Login() および Sm\_PolicyApi\_Release() を呼び出すときに渡されます）
- **ログイン**： 初期化の後、API の他の関数を呼び出す前に、Sm\_PolicyApi\_Login() を呼び出す必要があります。この関数によって管理者のログイン認証情報（ユーザ名とパスワード）または管理者の有効性を確認できます。  
管理者が認証されると、関数は内部データ構造体とリソースを初期化します。管理者がログインすると、ポリシー サーバがハンドルを初期化します。このハンドルは、後続のポリシー管理 API 関数への入力パラメータとして使用されます。クライアントセッション中に、Sm\_PolicyApi\_Login() を 2 回以上呼び出すことができます。
- **ログアウト**： Sm\_PolicyApi\_Logout() は管理者をログアウトします。
- **解放**： Sm\_PolicyApi\_Release() が、API クライアントセッションによって呼び出される最後の関数になります。それは、ポリシー ストアから切断し、API によって確保されていたメモリとリソースを解放します。この関数は、クライアントセッションにつき 1 回呼び出される必要があります。この関数を呼び出さないとメモリ リークが生じます。

## ポリシー管理 API のカテゴリ別の関数

ポリシー管理 API 内の大半の関数は、指定した関数によって影響を受ける SiteMinder のポリシーストア オブジェクト（たとえば、エージェント、ポリシー、ルールなど）に従って分類されます。そのほかに、カテゴリ必須の関数、ユーザ状態関数、キャッシュ、エージェント暗号化キー管理などのユーティリティ関数があります。これらは、特定の関数で実行されるサービスのタイプ別に分類されます。

これらのカテゴリは、カスタム ポリシー管理アプリケーションで使用する特定のポリシー管理 API 関数を探す際に役立ちます。

ポリシー管理 API 関数には次のカテゴリがあります。

- 必須関数
- 管理者関数
- エージェント関数
- エージェント設定関数
- 認証/許可マップ関数
- 認証方式関数
- 証明書マッピング関数
- ドメイン関数
- フェデレーション関数
- 一般オブジェクト関数
- グループ関数
- ODBC クエリ方式関数
- パスワード ポリシー関数
- ポリシー関数
- レルム関数
- 登録方式関数
- 正規表現関数
- レスポンス関数
- ルール関数
- SAML 2.0 属性権限関数

- SAML 2.0 設定関数
- SAML 2.0 インデックス エンドポイント関数
- ユーザ ディレクトリ関数
- ユーザおよびユーザ状態関数
- ユーザ パスワード状態関数
- ユーティリティ関数
- 変数関数
- WS-Federation 関数

### 必須関数

以下の関数は、すべてのポリシー管理アプリケーションで使用される必要があります。

関数	説明
<a href="#">Sm_PolicyApi_Init()</a> (P. 550)	ポリシー サーバへの接続を初期化します。
<a href="#">Sm_PolicyApi_InitEx()</a> (P. 551)	SiteMinder ポリシー ストアへの接続を初期化し、指定のバージョンに基づいて <code>init</code> ハンドルを確立します。バージョン <code>SM_POLICY_API_VERSION_6_0</code> 以上のクライアントに必須です。
<a href="#">Sm_PolicyApi_Login()</a> (P. 554)	管理者を認証します。
<a href="#">Sm_PolicyApi_Logout()</a> (P. 556)	管理者をログアウトします。
<a href="#">Sm_PolicyApi_Release()</a> (P. 561)	ポリシー サーバから切断し、API によって確保されていたメモリとリソースを解放します。

## 管理者関数

以下の関数は、SiteMinder 管理者オブジェクトを管理します。

関数	説明
<a href="#">Sm_PolicyApi_AddAdmin()</a> (P. 295)	管理者オブジェクトを作成または更新します。
<a href="#">Sm_PolicyApi_AddAdminToDomain()</a> (P. 298)	指定されたドメインを管理するための管理者権限を与えます。
<a href="#">Sm_PolicyApi_DeleteAdmin()</a> (P. 380)	管理者を削除します。
<a href="#">Sm_PolicyApi_GetAdmin()</a> (P. 415)	オブジェクト識別子で管理者の内容を取得します。
<a href="#">Sm_PolicyApi_GetAdminByName()</a> (P. 416)	名前で管理者の内容を取得します。
<a href="#">Sm_PolicyApi_GetGlobalPolicyByName()</a> (P. 466)	指定したグローバル ポリシーを名前で取得します。
<a href="#">Sm_PolicyApi_GetGlobalResponseByName()</a> (P. 468)	指定したグローバル レスポンスを名前で取得します。
<a href="#">Sm_PolicyApi_GetGlobalRuleByName()</a> (P. 469)	指定したグローバル ルールを名前で取得します。
<a href="#">Sm_PolicyApi_RemoveAdminFromDomain()</a> (P. 563)	指定したドメインから管理者の関連付けを解除します。

## エージェント関数

以下の関数は、SiteMinder エージェント オブジェクトを管理します。

注: エージェント タイプまたはエージェント タイプ属性を作成するためのファシリティがありません。

関数	説明
<a href="#">Sm_PolicyApi_AddAgent()</a> (P. 303)	エージェント オブジェクトを作成または更新します。

関数	説明
<a href="#">Sm_PolicyApi_ConvertFromLegacyAgent()</a> (P. 369)	v4.x エージェントを v5.x エージェントに変換します。
<a href="#">Sm_PolicyApi_ConvertToLegacyAgent()</a> (P. 370)	v5.x エージェントを v4.x エージェントに変換します。
<a href="#">Sm_PolicyApi_DeleteAgent()</a> (P. 384)	エージェントを削除します。
<a href="#">Sm_PolicyApi_GetAgent()</a> (P. 431)	OID でエージェントの内容を取得します。
<a href="#">Sm_PolicyApi_GetAgentByName()</a> (P. 433)	名前でエージェントの内容を取得します。
<a href="#">Sm_PolicyApi_GetAgentType()</a> (P. 438)	OID でエージェントタイプオブジェクトの内容を取得します。
<a href="#">Sm_PolicyApi_GetAgentTypeByName()</a> (P. 439)	名前でエージェントタイプオブジェクトの内容を取得します。
<a href="#">Sm_PolicyApi_GetAgentTypeAttr()</a> (P. 441)	1 つまたはすべてのエージェントタイプ属性を返します。
<a href="#">Sm_PolicyApi_GetAgentTypeAttrByName()</a> (P. 442)	名前で指定したエージェントタイプ属性オブジェクトを返します。

## エージェント設定関数

以下の関数は、エージェントを集中的に設定するための設定オブジェクト（エージェント設定オブジェクト、ホスト設定オブジェクト、トラステッドホストオブジェクト）を管理します。

トラステッドホストは、以下のいずれの方法でも作成されます。

- エージェントのインストール時
- `smreghost` ツールを使用するコマンドラインで。このツールの使用の詳細については、「[Web エージェントインストールガイド](#)」を参照してください。
- `Sm_PolicyApi_AddTrustedHost()` 関数で。

関数	説明
<a href="#">Sm_PolicyApi_AddAgentConfig()</a> (P. 305)	エージェント設定オブジェクトを追加または変更します。
<a href="#">Sm_PolicyApi_AddAgentConfigAssociation()</a> (P. 306)	設定パラメータ名および、指定したエージェント設定オブジェクトの対応する値を追加または変更します。
<a href="#">Sm_PolicyApi_AddHostConfig()</a> (P. 320)	ホスト設定オブジェクトを追加または変更します。
<a href="#">Sm_PolicyApi_AddTrustedHost()</a> (P. 350)	エージェントとポリシーサーバの間に接続がないとき、オブジェクトストアのトラステッドホストオブジェクトを作成または変更します。
<a href="#">Sm_PolicyApi_DeleteAgentConfig()</a> (P. 385)	エージェント設定オブジェクトを削除します。
<a href="#">Sm_PolicyApi_DeleteHostConfig()</a> (P. 389)	ホスト設定オブジェクトを削除します。
<a href="#">Sm_PolicyApi_DeleteTrustedHost()</a> (P. 401)	トラステッドホストオブジェクトを削除します。

関数	説明
<a href="#">Sm_PolicyApi_GetAgentConfig()</a> (P. 434)	エージェント設定オブジェクトを取得します。
<a href="#">Sm_PolicyApi_GetAgentConfigAssociations()</a> (P. 437)	エージェント設定オブジェクト用の設定パラメータのリストを取得します。
<a href="#">Sm_PolicyApi_GetAgentConfigByName()</a> (P. 435)	名前でエージェント設定オブジェクトを取得します。
<a href="#">Sm_PolicyApi_GetHostConfig()</a> (P. 477)	ホスト設定オブジェクトを取得します。
<a href="#">Sm_PolicyApi_GetHostConfigByName()</a> (P. 478)	名前でホスト設定オブジェクトを取得します。
<a href="#">Sm_PolicyApi_GetSharedSecretPolicy()</a> (P. 519)	現在の共有秘密キー ポリシーを取得します。
<a href="#">Sm_PolicyApi_GetTrusted</a> (P. 523) <a href="#">Host()</a> (P. 523)	OID でトラステッドホストオブジェクトを取得します。
<a href="#">Sm_PolicyApi_GetTrustedHostByName()</a> (P. 524)	名前でトラステッドホストオブジェクトを取得します。
<a href="#">Sm_PolicyApi_RemoveAgentConfigAssociation()</a> (P. 565)	指定したエージェント設定オブジェクトから設定パラメータ名/値のペアを削除します。
<a href="#">Sm_PolicyApi_SetSharedSecretPolicy()</a> (P. 599)	現在の <code>SharedSecretPolicy</code> を設定します。

## 認証/許可マップ関数

以下の関数は、SiteMinder 認証および許可ディレクトリ マッピング オブジェクトを管理します。

関数	説明
<a href="#">Sm_PolicyApi_CreateAuthAzMap()</a> (P. 371)	認証および許可ディレクトリ マッピング オブジェクトを作成または更新します。
<a href="#">Sm_PolicyApi_DeleteAuthAzMap()</a> (P. 386)	認証と許可ディレクトリ マップを削除します。
<a href="#">Sm_PolicyApi_GetAuthAzMap()</a> (P. 454)	認証および許可ディレクトリ マップの内容を取得します。

## 認証方式関数

以下の関数は、SiteMinder 認証方式を管理します。

関数	説明
<a href="#">Sm_PolicyApi_AddScheme()</a> (P. 345)	認証方式を作成または更新します。
<a href="#">Sm_PolicyApi_DeleteScheme()</a> (P. 400)	認証方式を削除します。
<a href="#">Sm_PolicyApi_GetScheme()</a> (P. 518)	OID で認証方式の内容を取得します。
<a href="#">Sm_PolicyApi_GetSchemeByName()</a> (P. 519)	名前で認証方式の内容を取得します。

## 証明書マッピング関数

以下の関数は、SiteMinder 証明書マッピング オブジェクトを管理します。

関数	説明
<a href="#">Sm_PolicyApi_CreateCertMap()</a> (P. 372)	証明書マッピング オブジェクトを作成または更新します。

関数	説明
<a href="#">Sm_PolicyApi_DeleteCertMap()</a> (P. 387)	証明書マップを削除します。
<a href="#">Sm_PolicyApi_GetCertMap()</a> (P. 455)	証明書マップの内容を取得します。

---

## ドメイン関数

以下の関数は、SiteMinder ドメイン オブジェクトを管理します。

関数	説明
<a href="#">Sm_PolicyApi_AddDomain()</a> (P. 312)	ドメインを作成または更新します。
<a href="#">Sm_PolicyApi_DeleteDomain()</a> (P. 388)	ドメインおよびドメインの子（ルール、レスポンス、レルム、ポリシー）を削除します。
<a href="#">Sm_PolicyApi_GetDomain()</a> (P. 460)	OID でドメインの内容を取得します。
<a href="#">Sm_PolicyApi_GetDomainByName()</a> (P. 462)	名前で指定したドメインの内容を取得します。
<a href="#">Sm_PolicyApi_GetDomainObjects()</a> (P. 463)	指定したドメイン内の指定のオブジェクトタイプに対するドメインオブジェクトのOIDを取得します。

---

## フェデレーション関数

以下の関数は、SAML アサーションを生成する際に必要なポリシーストアデータ（アフィリエイトドメインとアフィリエイトオブジェクト）の操作をサポートします。

関数	説明
<a href="#">Sm_PolicyApi_AddAdminToAffiliateDomain()</a> (P. 296)	アフィリエイトドメインに管理者を追加します。

---

関数	説明
<a href="#">Sm_PolicyApi_AddAffiliate()</a> (P. 299)	新規のアフィリエイトオブジェクトを作成するか、または既存のアフィリエイトオブジェクトを更新します。
<a href="#">Sm_PolicyApi_AddAffiliateDomain()</a> (P. 301)	新規のアフィリエイトドメインを作成するか、または既存のアフィリエイトドメインを更新します。
<a href="#">Sm_PolicyApi_AddAttributeToAffiliate()</a> (P. 309)	アフィリエイトに新規属性を追加します。
<a href="#">Sm_PolicyApi_AddUserDirToAffiliateDomain()</a> (P. 352)	アフィリエイトドメインにユーザディレクトリを追加します。
<a href="#">Sm_PolicyApi_AddUsersToAffiliate()</a> (P. 357)	ユーザディレクトリエントリをアフィリエイトに追加します。
<a href="#">Sm_PolicyApi_DeleteAffiliate()</a> (P. 381)	アフィリエイトを削除します。
<a href="#">Sm_PolicyApi_DeleteAffiliateDomain()</a> (P. 383)	アフィリエイトドメインを削除します。
<a href="#">Sm_PolicyApi_GetAffiliate()</a> (P. 418)	OID でアフィリエイトを取得します。
<a href="#">Sm_PolicyApi_GetAffiliateByName()</a> (P. 420)	名前でアフィリエイトを取得します。
<a href="#">Sm_PolicyApi_GetAffiliateDomain()</a> (P. 422)	OID でアフィリエイトドメインを取得します。
<a href="#">Sm_PolicyApi_GetAffiliateDomainByName()</a> (P. 423)	名前でアフィリエイトドメインを取得します。

関数	説明
<a href="#">Sm_PolicyApi_GetAffiliateDomainObjects()</a> (P. 425)	アフィリエイト ドメイン内の指定のオブジェクトタイプに対するドメインオブジェクトのOIDを取得します。
<a href="#">Sm_PolicyApi_GetAffiliateDomainUserDirSearchOrder()</a> (P. 426)	アフィリエイト ドメインのユーザディレクトリ検索順序を取得します。
<a href="#">Sm_PolicyApi_GetAffiliateUsers()</a> (P. 430)	アフィリエイトのユーザディレクトリ エントリを取得します。
<a href="#">Sm_PolicyApi_GetAllAffiliateAttributes()</a> (P. 443)	アフィリエイトの全属性を取得します。
<a href="#">Sm_PolicyApi_GetAllAffiliates()</a> (P. 445)	アフィリエイト ドメインの全アフィリエイトを取得します。
<a href="#">Sm_PolicyApi_GetGlobalObjects()</a> (P. 469)	複数のアフィリエイト ドメインを取得します。 (フェデレーション関数に限りません。)
<a href="#">Sm_PolicyApi_InitEx()</a> (P. 551)	必須関数を参照してください。
<a href="#">Sm_PolicyApi_RemoveAdminFromAffiliateDomain()</a> (P. 562)	アフィリエイト ドメインから管理者を削除します。
<a href="#">Sm_PolicyApi_RemoveAttributeFromAffiliate()</a> (P. 567)	アフィリエイトから属性を削除します。
<a href="#">Sm_PolicyApi_RemoveUserDirFromAffiliateDomain()</a> (P. 576)	アフィリエイト ドメインからユーザディレクトリを削除します。
<a href="#">Sm_PolicyApi_RemoveUsersFromAffiliate()</a> (P. 579)	アフィリエイトからユーザディレクトリのエントリを削除します。

関数	説明
<a href="#">Sm_PolicyApi_SetAffiliateDomainUserDirSearchOrder()</a> (P. 586)	アフィリエイト ドメインのユーザ ディレクトリ 検索順序を設定します。

## 一般オブジェクト関数

以下の関数は、SiteMinder オブジェクトの複数のタイプに作用します。

関数	説明
<a href="#">Sm_PolicyApi_GetGlobalObjects()</a> (P. 469)	指定したオブジェクトタイプに対するグローバル オブジェクトの OID を取得します。
<a href="#">Sm_PolicyApi_RenameObject()</a> (P. 585)	オブジェクトの名前を変更します。

## グループ関数

以下の関数は、SiteMinder グループ オブジェクトを管理します。  
Sm\_PolicyApi\_Groups\_t で列挙されるように、エージェント グループ、レスポンス グループ、ルール グループを作成できます。

**注:** グローバル オブジェクトの グループはサポートされていません。

グループには、それ自体のタイプの個別のアイテムまたはグループを含めることができます。たとえば、ルール グループはルールおよびルールのグループを含むことができます。

関数	説明
<a href="#">Sm_PolicyApi_AddGroup()</a> (P. 318)	エージェント、レスポンスまたはルール グループ オブジェクトを作成または更新します。
<a href="#">Sm_PolicyApi_AddToGroup()</a> (P. 348)	エージェント、レスポンスまたはルール アイテムをグループに追加します。
<a href="#">Sm_PolicyApi_DeleteGroup()</a> (P. 390)	グループを削除します。

関数	説明
<a href="#">Sm_PolicyApi_GetGroup()</a> (P. 471)	OID でグループの内容を取得します。
<a href="#">Sm_PolicyApi_GetGroupByName()</a> (P. 473)	名前でグループの内容を取得します。
<a href="#">Sm_PolicyApi_GetGroupOids()</a> (P. 475)	グループに含まれる OID を取得します。
<a href="#">Sm_PolicyApi_IsGroup()</a> (P. 552)	指定されたアイテムがグループ内に含まれるかどうかを判定します。
<a href="#">Sm_PolicyApi_RemoveFromGroup()</a> (P. 571)	グループから指定のアイテムを削除します。

## ODBC クエリ方式関数

以下の関数は、SiteMinder ODBC クエリ方式を管理します。

関数	説明
<a href="#">Sm_PolicyApi_CreateODBCQueryScheme()</a> (P. 374)	ODBC クエリ方式を作成または更新します。
<a href="#">Sm_PolicyApi_DeleteODBCQueryScheme()</a> (P. 391)	ODBC クエリ方式を削除します。
<a href="#">Sm_PolicyApi_GetODBCQueryScheme()</a> (P. 480)	OID で ODBC クエリ方式の内容を取得します。
<a href="#">Sm_PolicyApi_GetODBCQuerySchemeByName()</a> (P. 481)	名前で ODBC クエリ方式の内容を取得します。

## パスワードポリシー関数

以下の関数は、SiteMinder 証明書パスワードポリシーオブジェクトを管理します。

関数	説明
<a href="#">Sm_PolicyApi_AddPasswordPolicy()</a> (P. 323)	パスワードポリシーオブジェクトを作成または更新します。

関数	説明
<a href="#">Sm_PolicyApi_DeletePasswordPolicy()</a> (P. 392)	パスワード ポリシーを削除します。
<a href="#">Sm_PolicyApi_GetPasswordPolicy()</a> (P. 487)	OID でパスワード ポリシーの内容を取得します。
<a href="#">Sm_PolicyApi_GetPasswordPolicyByName()</a> (P. 488)	名前でパスワード ポリシーの内容を取得します。

## ポリシー関数

以下の関数は、SiteMinder ポリシーおよびポリシー リンク オブジェクトを管理します。ポリシー リンクは、ポリシー、ルール、およびオプションでレスポンスの関連付けです。

関数	説明
<a href="#">Sm_PolicyApi_AddGlobalPolicy()</a> (P. 314)	グローバル ポリシー オブジェクトを作成または更新します。
<a href="#">Sm_PolicyApi_AddPolicy()</a> (P. 325)	ポリシー オブジェクトを作成または更新します。
<a href="#">Sm_PolicyApi_AddPolicyLink()</a> (P. 327)	指定したポリシーのポリシー リンクを作成します。
<a href="#">Sm_PolicyApi_DeletePolicy()</a> (P. 393)	ポリシーを削除します。
<a href="#">Sm_PolicyApi_GetPasswordPolicyByName()</a> (P. 488)	名前でパスワード ポリシーの内容を取得します。
<a href="#">Sm_PolicyApi_GetPolicy()</a> (P. 489)	OID でポリシーの内容を取得します。
<a href="#">Sm_PolicyApi_GetPolicyLinks()</a> (P. 492)	指定したポリシーと関連付けられるすべてのポリシー リンクのリンク リストを取得します。
<a href="#">Sm_PolicyApi_RemovePolicyLinkFromPolicy()</a> (P. 573)	指定したポリシーからポリシー リンクを削除します。

## レルム関数

以下の関数は、SiteMinder レルム オブジェクトを管理します。

関数	説明
<a href="#">Sm_PolicyApi_AddRealm()</a> (P. 329)	レルム オブジェクトを作成または更新します。
<a href="#">Sm_PolicyApi_DeleteRealm()</a> (P. 394)	レルムを削除します。
<a href="#">Sm_PolicyApi_GetChildren()</a> (P. 456)	階層レルムおよびルール ツリーを構築します。
<a href="#">Sm_PolicyApi_GetRealm()</a> (P. 495)	OID でレルムの内容を取得します。
<a href="#">Sm_PolicyApi_GetRealmByName()</a> (P. 496)	名前でレルムの内容を取得します。

## 登録方式関数

以下の関数は、SiteMinder 登録方式を管理します。

関数	説明
<a href="#">Sm_PolicyApi_AddRegistrationScheme()</a> (P. 332)	登録方式を作成または更新します。
<a href="#">Sm_PolicyApi_DeleteRegistrationScheme()</a> (P. 395)	登録方式を削除します。
<a href="#">Sm_PolicyApi_GetRegistrationScheme()</a> (P. 499)	OID で登録方式の内容を取得します。
<a href="#">Sm_PolicyApi_GetRegistration</a> (P. 500) <a href="#">SchemeByName()</a> (P. 500)	名前で登録方式の内容を取得します。

## 正規表現関数

以下の関数は、SiteMinder の正規表現を管理します。

関数	説明
<a href="#">Sm_PolicyApi_AddRegularExpressionToPasswordPolicy()</a> (P. 333)	正規表現を作成または更新します。
<a href="#">Sm_PolicyApi_RemoveRegularExpressionFromPasswordPolicy()</a> (P. 574)	正規表現を削除します。
<a href="#">Sm_PolicyApi_GetRegularExpressions()</a> (P. 501)	指定したパスワードポリシーに属している正規表現を取得します。

## レスポンス関数

以下の関数は、SiteMinder レスポンス オブジェクトを管理します。

関数	説明
<a href="#">Sm_PolicyApi_AddGlobalResponse()</a> (P. 315)	グローバル レスポンス オブジェクトを作成または更新します。
<a href="#">Sm_PolicyApi_AddResponse()</a> (P. 335)	レスポンス オブジェクトを作成または更新します。
<a href="#">Sm_PolicyApi_AddResponseAttr()</a> (P. 336)	指定したレスポンスのレスポンス属性を作成します。
<a href="#">Sm_PolicyApi_DeleteResponse()</a> (P. 396)	レスポンスを削除します。
<a href="#">Sm_PolicyApi_GetResponse()</a> (P. 502)	OID でレスポンスの内容を取得します。
<a href="#">Sm_PolicyApi_GetResponseAttrs()</a> (P. 505)	指定されたレスポンスに対するレスポンス属性のリンク リストを取得します。

関数	説明
<a href="#">Sm_PolicyApi_GetResponseByName()</a> (P. 504)	名前でレスポンスの内容を取得します。
<a href="#">Sm_PolicyApi_RemoveResponseAttr()</a> (P. 575)	指定したレスポンスからレスポンス属性の関連付けを解除します。
<a href="#">Sm_PolicyApi_SetResponseInPolicyLink()</a> (P. 597)	レスポンスまたはレスポンスグループをルールまたはルールグループに設定するか、またはルールまたはルールグループからレスポンスまたはレスポンスグループを削除します。

## ルール関数

以下の関数は、SiteMinder ルール オブジェクトを管理します。

関数	説明
<a href="#">Sm_PolicyApi_AddGlobalRule()</a> (P. 317)	グローバルルール オブジェクトを作成または更新します。
<a href="#">Sm_PolicyApi_AddRule()</a> (P. 338)	ルール オブジェクトを作成または更新します。
<a href="#">Sm_PolicyApi_DeleteRule()</a> (P. 397)	ルールを削除します。
<a href="#">Sm_PolicyApi_GetRule()</a> (P. 507)	OID でルールの内容を取得します。
<a href="#">Sm_PolicyApi_GetRuleByName()</a> (P. 508)	名前でルールの内容を取得します。

## SAML1.x 設定関数

以下の関数は、SAML 1.x 設定をサポートします。

関数	説明
<code>Sm_PolicyApi_AddMessageConsumerPluginToSAML1Scheme()</code>	メッセージ コンシューマ プラグイン設定を SAML1.x 認証方式に追加または更新します。
<a href="#">Sm_PolicyApi_AddRedirectURLToSAML1xScheme()</a> (P. 331)	リダイレクト URL を SAML1.x 認証方式に追加または更新します。
<code>Sm_PolicyApi_GetMessageConsumerPluginFromSAML1Scheme()</code>	メッセージ コンシューマ プラグイン設定を SAML1.x 認証方式から取得します。
<code>Sm_PolicyApi_GetRedirectURLFromSAML1Scheme()</code>	SAML 1.x 認証方式からリダイレクト URL 設定を取得します。

## SAML 2.0 属性権限関数

以下の関数は、SAML リクエスト 用の属性を管理します。

関数	説明
<a href="#">Sm_PolicyApi_AddAttributeToSAMLScheme()</a> (P. 310)	SAML 2.0 認証方式に定義される SAML リクエストに属性を追加します。
<a href="#">Sm_PolicyApi_GetAllSAMLSchemeAttributes()</a> (P. 447)	SAML リクエストに対して定義された属性をすべて取得します。
<a href="#">Sm_PolicyApi_RemoveAttributeFromSAMLScheme()</a> (P. 568)	SAML 2.0 認証方式に定義される SAML リクエストから属性を削除します。

## SAML 2.0 設定関数

以下の関数は、SAML 2.0 アフィリエーション、サービスプロバイダ、およびアイデンティティプロバイダを管理します。

関数	説明
<a href="#">Sm_PolicyApi_AddAttributeToSAMLSP()</a> (P. 311)	サービスプロバイダに対して SAML 2.0 属性を定義します。
<a href="#">Sm_PolicyApi_AddSAMLAffiliation()</a> (P. 339)	新規 SAML アフィリエーションオブジェクトを追加するか、または既存の SAML アフィリエーションオブジェクトを変更します。
<a href="#">Sm_PolicyApi_AddSAMLScheme()</a> (P. 341)	新規 SAML 2.0 認証方式オブジェクトを追加するか、または既存のものを変更します。
<a href="#">Sm_PolicyApi_AddSAMLServiceProvider()</a> (P. 343)	新規 SAML 2.0 サービスプロバイダオブジェクトを追加するか、または既存のものを変更します。
<a href="#">Sm_PolicyApi_AddUsersToSAMLServiceProvider()</a> (P. 360)	ユーザディレクトリエントリを SAML 2.0 サービスプロバイダに関連付けます。
<a href="#">Sm_PolicyApi_DeleteSAMLAffiliation()</a> (P. 398)	指定された SAML アフィリエーションを削除します。
<a href="#">Sm_PolicyApi_DeleteSAMLServiceProvider()</a> (P. 399)	指定されたサービスプロバイダを削除します。
<a href="#">Sm_PolicyApi_GetAffiliatedSAMLAuthSchemes()</a> (P. 427)	指定された SAML アフィリエーションと関連付けられた SAML 認証方式をすべて取得します。

関数	説明
<a href="#">Sm_PolicyApi_GetAffiliatedSAMLServiceProviders()</a> (P. 428)	指定された SAML アフィリエーションと関連付けられているサービスプロバイダをすべて取得します。
<a href="#">Sm_PolicyApi_GetAllSAMLAffiliations()</a> (P. 446)	既存の SAML アフィリエーションオブジェクトをすべて取得します。
<a href="#">Sm_PolicyApi_GetAllSAMLServiceProviders()</a> (P. 448)	指定されたアフィリエイトドメインのサービスプロバイダをすべて取得します。
<a href="#">Sm_PolicyApi_GetAllSAMLSPAttributes()</a> (P. 451)	指定されたサービスプロバイダと関連付けられた属性をすべて取得します。
<a href="#">Sm_PolicyApi_GetSAMLAffiliation()</a> (P. 510)	ポリシーストア内のその OID によって指定される SAML アフィリエーションを取得します。
<a href="#">Sm_PolicyApi_GetSAMLAffiliationById()</a> (P. 511)	ポリシーストア内のその固有のアフィリエーション識別子 (URI) によって指定される SAML アフィリエーションを取得します。
<a href="#">Sm_PolicyApi_GetSAMLScheme()</a> (P. 513)	SAML 2.0 認証方式に関する情報を取得します。
<a href="#">Sm_PolicyApi_GetSAMLServiceProvider()</a> (P. 514)	ポリシーストア内のその OID によって指定されるサービスプロバイダを取得します。
<a href="#">Sm_PolicyApi_GetSAMLServiceProviderById()</a> (P. 515)	ポリシーストア内のその固有のプロバイダ ID によって指定されるサーバプロバイダを取得します。

関数	説明
<a href="#">Sm_PolicyApi_GetSAMLServiceProviderUsers()</a> (P. 517)	指定されたサービスプロバイダと関連付けられたユーザディレクトリエントリを取得します。
<a href="#">Sm_PolicyApi_RemoveAttributeFromSAMLSP()</a> (P. 570)	指定された SAML 属性をサービスプロバイダから削除します。
<a href="#">Sm_PolicyApi_RemoveUsersFromSAMLServiceProvider()</a> (P. 582)	指定されたユーザをサービスプロバイダから削除します。

## SAML 2.0 インデックス エンドポイント関数

以下の関数は、サービスプロバイダのインデックス エンドポイントを管理します。

関数	説明
<a href="#">Sm_PolicyApi_AddAssertionConsumerServiceToSAMLSP()</a> (P. 307)	サービスプロバイダに新規のインデックス エンドポイント参照を（インデックス、バインディング、アサーションコンシューマ サービス URL で）追加します。
<a href="#">Sm_PolicyApi_GetAllSAMLSPAssertionConsumerService()</a> (P. 450)	ポリシーストアに存在するすべてのアサーションコンシューマ サービスの一覧を取得します。
<a href="#">Sm_PolicyApi_RemoveAssertionConsumerService()</a> (P. 566)	アサーションコンシューマ サービスへのインデックス エンドポイント参照を削除します。

## ユーザ ディレクトリ関数

以下の関数は、SiteMinder のユーザ ディレクトリ オブジェクトを管理します。

関数	説明
<a href="#">Sm_PolicyApi_AddUserDirToDomain()</a> (P. 354)	ディレクトリ オブジェクトを、指定されたドメインと関連付けます。
<a href="#">Sm_PolicyApi_CreateUserDir()</a> (P. 376)	ユーザ ディレクトリ オブジェクトを作成または更新します。
<a href="#">Sm_PolicyApi_DeleteUserDir()</a> (P. 402)	ユーザ ディレクトリを削除します。
<a href="#">Sm_PolicyApi_GetDirectoryContents()</a> (P. 457)	指定されたユーザ ディレクトリのユーザ構造のリンク リストを取得します。
<a href="#">Sm_PolicyApi_GetUserContext()</a> (P. 527)	ポリシー管理 API の呼び出し元によるユーザ コンテキスト情報へのアクセスを可能にします。
<a href="#">Sm_PolicyApi_GetUserDir()</a> (P. 529)	OID でユーザ ディレクトリの内容を取得します。
<a href="#">Sm_PolicyApi_GetUserDirByName()</a> (P. 530)	名前ユーザ ディレクトリの内容を取得します。
<a href="#">Sm_PolicyApi_GetUserDirCapabilities()</a> (P. 532)	ユーザ ディレクトリの機能を取得します。

関数	説明
<a href="#">Sm_PolicyApi_GetUserDirSearchOrder()</a> (P. 533)	指定されたドメインと関連付けられたユーザディレクトリのOIDを取得します。
<a href="#">Sm_PolicyApi_LookupDirectoryEntry()</a> (P. 557)	特定のユーザディレクトリで、指定された検索パターンに基づいてユーザ仕様を検索します。
<a href="#">Sm_PolicyApi_RemoveUserDirFromDomain()</a> (P. 578)	指定されたドメインからユーザディレクトリの関連付けを解除します。
<a href="#">Sm_PolicyApi_SetUserDirSearchOrder()</a> (P. 600)	指定されたドメインと関連付けられたユーザディレクトリの検索順序を再調整します。
<a href="#">Sm_PolicyApi_ValidateDirectoryEntry()</a> (P. 603)	指定されたパスおよびユーザディレクトリ内のユーザ仕様を検証します。

## ユーザおよびユーザ状態関数

以下の関数は、ユーザ状態、および SiteMinder ユーザディレクトリ内のユーザエントリに対する操作を実行します。

フィールド	説明
<a href="#">Sm_PolicyApi_AddUsersToPolicy()</a> (P. 358)	指定されたポリシーにユーザを追加します。
<a href="#">Sm_PolicyApi_DisableUser()</a> (P. 406)	ユーザを無効にします。
<a href="#">Sm_PolicyApi_EnableUser()</a> (P. 407)	ユーザを有効にします。

フィールド	説明
<a href="#">Sm_PolicyApi_GetDisabledUserState()</a> (P. 459)	ユーザの無効状態を取得します。
<a href="#">Sm_PolicyApi_GetPasswordMsg()</a> (P. 484)	新規パスワードの検証時に発生したエラーに関する情報を取得します。
<a href="#">Sm_PolicyApi_GetPolicyUsers()</a> (P. 493)	指定されたポリシーおよびオプションでユーザディレクトリと関連付けられているユーザの構造のリンクリストを取得します。
<a href="#">Sm_PolicyApi_GetUserGroups()</a> (P. 534)	ユーザがメンバであるグループのリストを取得します。
<a href="#">Sm_PolicyApi_RemoveUsersFromPolicy()</a> (P. 580)	指定されたポリシーからユーザの関連付けを解除します。
<a href="#">Sm_PolicyApi_SetDisabledUserState()</a> (P. 588)	ユーザの無効な状態を設定します。
<a href="#">Sm_PolicyApi_SetPassword()</a> (P. 592)	ユーザパスワードを変更または検証します。
<a href="#">Sm_PolicyApi_SetPath()</a> (P. 595)	指定されたポリシーでユーザのパスを設定します。

## ユーザパスワード状態関数

以下の関数は、SiteMinder のパスワード状態オブジェクトを管理します。

関数	説明
<a href="#">Sm_PolicyApi_GetUserPasswordState()</a> (P. 536)	UserPasswordState オブジェクトを返します。
<a href="#">Sm_PolicyApi_SetUserPasswordState()</a> (P. 601)	UserPasswordState オブジェクトを追加/更新します。

## ユーティリティ関数

以下の関数は、メモリ、キャッシュ、およびエージェント暗号化キー管理など、さまざまなサービスを提供します。

関数	説明
<a href="#">Sm_PolicyApi_FlushRealm()</a> (P. 409)	指定されたレルムをリソース キャッシュからクリアします。
<a href="#">Sm_PolicyApi_FlushUser()</a> (P. 410)	指定されたユーザをユーザ キャッシュからクリアします。
<a href="#">Sm_PolicyApi_FreeMemory()</a> (P. 411)	ポリシー管理 API によって割り当てられたメモリを解放します。
<a href="#">Sm_PolicyApi_FreeMemoryEx()</a> (P. 412)	ポリシー管理 API によって割り当てられたメモリを解放します。バージョン <code>SM_POLICY_API_VERSION_6_0</code> 以上のクライアントに必須です。
<a href="#">Sm_PolicyApi_FreeString()</a> (P. 413)	ポリシー管理 API によって割り当てられた単一の文字列を解放します。
<a href="#">Sm_PolicyApi_FreeStringArray()</a> (P. 414)	ポリシー管理 API によって割り当てられた文字列の配列を解放します。
<a href="#">Sm_PolicyApi_ManagementCommand()</a> (P. 560)	ユーザ、キー、およびリソース管理のサービスを実行します。

## 変数関数

以下のメソッドは、変数を管理します。

関数	説明
<a href="#">Sm_PolicyApi_AddVariable()</a> (P. 363)	変数オブジェクトを追加します。
<a href="#">Sm_PolicyApi_DeleteVariable()</a> (P. 403)	変数オブジェクトを削除します。
<a href="#">Sm_PolicyApi_GetVariable()</a> (P. 539)	OID で指定された変数を取得します。
<a href="#">Sm_PolicyApi_GetVariableByName()</a> (P. 541)	名前で指定された変数を取得します。

関数	説明
<a href="#">Sm_PolicyApi_GetVariableType()</a> (P. 543)	OID で指定された変数タイプを取得します。
<a href="#">Sm_PolicyApi_GetVariableTypeByName()</a> (P. 545)	名前で指定された変数タイプを取得します。

## WS-Federation 関数

以下の表では、リソース パートナーとアカウント パートナーに対してサポートされている関数のリストを示します。

関数	説明
<a href="#">Sm_PolicyApi_AddWSFEDResourcePartner()</a> (P. 364)	WS-Federation リソース パートナー オブジェクトを作成または更新します。
<a href="#">Sm_PolicyApi_GetWSFEDResourcePartner()</a> (P. 547)	WS-Federation リソース パートナー オブジェクトを取得します。
<a href="#">Sm_PolicyApi_GetAllWSFEDResourcePartners()</a> (P. 453)	リンク リストとしてドメイン用の WS フェデレーション リソース パートナー オブジェクトをすべて取得します。

関数	説明
<a href="#">Sm_PolicyApi_DeleteWSFEDResourcePartner()</a> (P. 405)	WS-Federation リソース パートナーを削除します。それにより、次にリソース パートナー オブジェクトと関連付けられているエージェント、レルム、ルール、ポリシー、およびポリシー リンク オブジェクトが削除されます。
<a href="#">Sm_PolicyApi_AddUsersToWSFEDResourcePartner()</a> (P. 362)	ユーザ ディレクトリ エントリを WS-Federation リソース パートナーに関連付けます。
<a href="#">Sm_PolicyApi_RemoveUsersFromWSFEDResourcePartner()</a> (P. 584)	WS-Federation リソース パートナーから ユーザ ディレクトリ エントリの関連付け解除します。
<a href="#">Sm_PolicyApi_GetUsersFromWSFEDResourcePartner()</a> (P. 538)	WS-Federation リソース パートナーと関連付けられている ユーザ ディレクトリ エントリを取得します。
<a href="#">Sm_PolicyApi_AddWSFEDScheme()</a> (P. 367)	WS-Federation 認証方式を作成または更新します。
<a href="#">Sm_PolicyApi_GetWSFEDScheme()</a> (P. 549)	WS-Federation 認証方式を取得します。

## ポリシー管理 API の関数宣言

関数宣言には構文が含まれ、参照用にポリシー管理 API の各関数に対して値を返します。

### Sm\_PolicyApi\_AddAdmin()

グローバル範囲で新規 SiteMinder 管理者オブジェクトを作成します。管理者の属性は *pstructAdmin* 構造に含まれています。

管理者オブジェクトが存在し、*bUpdate* が「true」の場合、アイテムが更新されます。

#### タイプ

管理者関数、グローバル範囲。

#### 構文

```
int SM_EXTERN Sm_PolicyApi_AddAdmin (  
    void*                pSessionHandle,  
    Sm_PolicyApi_Admin_t* pstructAdmin,  
    const bool           bUpdate  
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pstructAdmin</i>	I	値の入力が完了している管理者構造体へのポインタです。
<i>bUpdate</i>	I	既存のオブジェクトが見つかる場合にそれが更新されることを示すフラグです。

### 戻り値

- Sm\_PolicyApi\_Success。 管理者オブジェクトは正常に作成されました。
- Sm\_PolicyApi\_Failure。 管理者オブジェクトは正常に作成されませんでした。
- Sm\_PolicyApi\_InvalidHandle。 このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。 有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。 管理者には、管理者オブジェクトを作成する権限がありません。
- Sm\_PolicyApi\_InvalidOid :
  - 管理者オブジェクト識別子が更新中に見つかりませんでした。
  - ユーザディレクトリ オブジェクト識別子は見つかりませんでした。
  - 方式のオブジェクト識別子が見つかりませんでした。

## Sm\_PolicyApi\_AddAdminToAffiliateDomain()

アフィリエイト ドメインに管理者を追加します。

### タイプ

フェデレーション関数

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddAdminToAffiliateDomain (  
    void*          pSessionHandle,  
    const char*    pszAdminOid,  
    const char*    pszAffiliateDomainOid  
);
```

---

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。 この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

---

---

パラメータ	I/O	説明
<i>pszAdminOid</i>	I	既存の管理者のオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszAffiliate DomainOid</i>	I	既存のアフィリエイト ドメインのオブジェクト識別子が含まれ、NULL で終わる文字列です。

---

### 戻り値

- **Sm\_PolicyApi\_Success**。管理者がアフィリエイト ドメインに追加されました。
- **Sm\_PolicyApi\_Failure**。管理者がアフィリエイト ドメインに追加されませんでした。
- **Sm\_PolicyApi\_InvalidHandle**。このコールに先立つ有効な初期化はありませんでした。
- **Sm\_PolicyApi\_NoSession**。有効な管理者セッションがありません。
- **Sm\_PolicyApi\_NoPrivilege**。管理者には、アフィリエイト ドメインを削除する権限がありません。
- **Sm\_PolicyApi\_InvalidOid**。アフィリエイト ドメイン **OID** が更新中に見つかりませんでした。
- **Sm\_PolicyApi\_DuplicateEntry**。指定された管理者オブジェクト識別子は、すでにアフィリエイト ドメインに存在します。

## Sm\_PolicyApi\_AddAdminToDomain()

指定されたドメインを管理するために指定の管理者権限を与え、*szDomainOid* によって識別されるドメインと *szAdminOid* によって識別される管理者オブジェクトとを関連付けます。

### タイプ

管理者関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddAdminToDomain (
    void*          pSessionHandle,
    const char*    pszAdminOid,
    const char*    pszDomainOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAdminOid</i>	I	既存の管理者のオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszDomainOid</i>	I	既存のドメインのオブジェクト識別子が含まれ、NULL で終わる文字列です。

### 戻り値

- `Sm_PolicyApi_Success`。管理者がドメインに追加されました。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、管理者をドメインに追加する権限がありません。
- `Sm_PolicyApi_InvalidOid`。管理者オブジェクト識別子またはドメインオブジェクト識別子が見つかりませんでした。
- `Sm_PolicyApi_DuplicateEntry`。指定された管理者オブジェクト識別子は、すでにドメインに存在します。

## Sm\_PolicyApi\_AddAffiliate()

新規のアフィリエイト オブジェクトを作成するか、または既存のアフィリエイト オブジェクトを更新します。この関数は、また `AffiliateDomain` `OID` に基づいて `PropertyCollection` オブジェクトを取得します。

### タイプ

フェデレーション関数

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddAffiliate (
    void*                pSessionHandle,
    Sm_PolicyApi_Affiliate_t* pstructAffiliate,
    const bool           bUpdate
);
```

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<code>pstructAffiliate</code>	I	値の入力が完了しているアフィリエイト構造体へのポインタです。

パラメータ	I/O	説明
<i>bUpdate</i>	I	既存のオブジェクトが見つかる場合にそれが更新されることを示すフラグです。

### 注釈

この関数は、Artifact プロファイルまたは POST プロファイルに基づいたアフィリエイトオブジェクトを作成します (Sm\_PolicyApi\_Affiliate\_t フィールドの SAMLProfile を参照してください)。POST プロファイルに基づいたアフィリエイトオブジェクトを作成するには、少なくとも SM\_POLICY\_API\_VERSION\_6\_0\_2 の API バージョンが必要です。以前のバージョンが含まれている場合、POST プロファイルリクエストは無視され (Sm\_PolicyApi\_Affiliate\_t の任意の POST に固有のフィールドと共に)、Artifact プロファイルに基づいたアフィリエイトオブジェクトを作成しようとします。

### 戻り値

- Sm\_PolicyApi\_Success。アフィリエイトが正常に作成されました。
- Sm\_PolicyApi\_Failure。アフィリエイトが正常に作成されませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、アフィリエイトを作成する権限がありません。
- Sm\_PolicyApi\_DomainNotFound。アフィリエイトドメインOIDが見つかりませんでした。
- Sm\_PolicyApi\_DomainNotAffiliate。ドメインはアフィリエイトドメインではありません。

## Sm\_PolicyApi\_AddAffiliateDomain()

新規のアフィリエイト ドメインを作成するか、または既存のアフィリエイト ドメインを更新します。 *bIsAffiliate* を TRUE に設定します。

### タイプ

フェデレーション関数

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddAffiliateDomain (
    void*                pSessionHandle,
    Sm_PolicyApi_AffiliateDomain_t *pstructAffiliateDomain,
    const bool           bUpdate
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pstructAffiliate ドメイン</i>	I	値の入力が完了しているアフィリエイト ドメインの構造体へのポインタです。
<i>bUpdate</i>	I	既存のオブジェクトが見つかる場合にそれが更新されることを示すブールフラグです。

### 戻り値

- **Sm\_PolicyApi\_Success**。アフィリエイト ドメインが正常に作成されました。
- **Sm\_PolicyApi\_Failure**。アフィリエイト ドメインが正常に作成されませんでした。
- **Sm\_PolicyApi\_InvalidHandle**。このコールに先立つ有効な初期化はありませんでした。
- **Sm\_PolicyApi\_NoSession**。有効な管理者セッションがありません。
- **Sm\_PolicyApi\_NoPrivilege**。管理者には、アフィリエイト ドメインを作成する権限がありません。
- **Sm\_PolicyApi\_DomainNotFound**。アフィリエイト ドメイン OID が更新中に見つかりませんでした。
- **Sm\_PolicyApi\_DomainNotAffiliate**。ドメインはアフィリエイト ドメインではありません。

## Sm\_PolicyApi\_AddAgent()

新規 SiteMinder エージェントを作成します。エージェントの属性は *pstructAgent* 構造に含まれています。

管理者オブジェクトが存在し、*bUpdate* フラグが「真」の場合、アイテムが更新されます。

この呼び出しでエージェントタイプを指定する必要があります。エージェントに対してエージェントタイプ OID を取得するには、*Sm\_PolicyApi\_GetGlobalObjects()* を使用します。

### タイプ

エージェント関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddAgent (
    void*                pSessionHandle,
    Sm_PolicyApi_Agent_t* pstructAgent,
    const bool           bUpdate
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pstructAgent</i>	I	値の入力が完了しているエージェント構造へのポインタです。
<i>bUpdate</i>	I	既存のオブジェクトが見つかる場合にそれが更新されることを示すフラグです。

### 戻り値

- `Sm_PolicyApi_Success`。 エージェントが正常に作成されました。
- `Sm_PolicyApi_Failure`。 エージェントが正常に作成されませんでした。
- `Sm_PolicyApi_InvalidHandle`。 このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。 有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。 管理者には、エージェントを作成する権限がありません。
- `Sm_PolicyApi_InvalidOid` :
  - エージェントタイプ `OID` が見つからなかったか、または特定のエージェントタイプではありません。
  - エージェント `OID` が見つかりませんでした。 これは、この関数が呼び出される時 `bUpdate` の設定が「真」であり、`Sm_PolicyApi_Agent_t` の `pszOid` フィールドが更新中のエージェントの `OID` を保持する場合に生じます。
- `Sm_PolicyApi_NotUnique`。 同じ名前のエージェントが存在します。
- `Sm_PolicyApi_RadiusIpAddrNotUnique`。 同一の `RADIUS IP` アドレスが別に存在します。
- `Sm_PolicyApi_MissingProperty`。 必須フィールドの 1 つが設定されていません。

## Sm\_PolicyApi\_AddAgentConfig()

ポリシーストアのエージェント設定オブジェクトを作成または変更します。

### タイプ

エージェント設定関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddAgentConfig (
    void*                pSessionHandle,
    Sm_PolicyApi_AgentConfig_t* pstructAgentConfig,
    const bool           bUpdate
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pstructAgentConfig</i>	I	エージェント設定オブジェクトを定義する構造体のアドレスです。
<i>bUpdate</i>	I	真の場合、オブジェクトが更新されています。

### 戻り値

- Sm\_PolicyApi\_Success。操作は成功しました。
- Sm\_PolicyApi\_Failure。一般的な障害です。
- Sm\_PolicyApi\_InvalidOid。エージェント設定オブジェクトに一意の ID が対応していません。
- Sm\_PolicyApi\_NoPrivilege。呼び出し元に、エージェント設定オブジェクトを追加または変更する権限がありません。
- Sm\_PolicyApi\_InvalidHandle。セッションポインタが無効です。
- Sm\_PolicyApi\_NoSession。API ユーザが正しくログインしていません。

## Sm\_PolicyApi\_AddAgentConfigAssociation()

設定パラメータ名および、指定したエージェント設定オブジェクトの対応する値を追加または変更します。

### タイプ

エージェント設定関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddAgentConfigAssociation (
    void*                pSessionHandle,
    const char*          pszAgentConfigOid,
    Sm_PolicyApi_Association_t* pstructAssociation,
    bool                 bUpdate
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAgentConfigOid</i>	I	エージェント設定オブジェクトの一意の識別子です。
<i>pstructAssociation</i>	I	エージェント設定オブジェクトで追加または変更する、名前/値のペアです。
<i>bUpdate</i>	I	真の場合、オブジェクトが更新されています。

### 戻り値

- Sm\_PolicyApi\_Success。 操作は成功しました。
- Sm\_PolicyApi\_Failure。 一般的な障害です。
- Sm\_PolicyApi\_InvalidOid。 エージェント設定オブジェクトに一意の ID が対応していません。
- Sm\_PolicyApi\_NoPrivilege。 呼び出し元に、エージェント設定オブジェクト パラメータを追加する権限がありません。
- Sm\_PolicyApi\_InvalidHandle。 セッション ポインタが無効です。
- Sm\_PolicyApi\_NoSession。 API ユーザが正しくログインしていません。

## Sm\_PolicyApi\_AddAssertionConsumerServiceToSAMLSP

インデックス、バインディング、アサーション コンシューマ サービス URL などの新規インデックス エンドポイントを、サービス プロバイダに追加します。

注: 既存のインデックス エンドポイント参照は変更できません。

### タイプ

フェデレーション関数

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddAssertionConsumerServiceToSAMLSP (
    void*                                pSessionHandle,
    const Sm_PolicyApi_SAMLSPAssertionConsumerService_t*
        pstructSAMLSPAssertionConsumerService,
    const char*                            pszSAMLSPoid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pstructSAMLSPAssertionConsumerService</i>	I	アサーション コンシューマ サービス構造体へのポインタです。

パラメータ	I/O	説明
<i>pszSAMLSPoid</i>	I	サービス プロバイダの OID を含む文字列へのポインタです。

---

### 戻り値

- **Sm\_PolicyApi\_Success**。アサーション コンシューマ サービス は正常に追加されました。
- **Sm\_PolicyApi\_Failure** - アサーション コンシューマ サービスは正常に追加されませんでした。
- **Sm\_PolicyApi\_ACSDuplicateIndex** - 同じインデックス値を持つサービス プロバイダに割り当てられたアサーション コンシューマ サービスがすでにあります。
- **Sm\_PolicyApi\_InvalidHandle** - このコールに先立つ有効な初期化はありませんでした。
- **Sm\_PolicyApi\_NoSession** - 有効な管理者セッションがありません。
- **Sm\_PolicyApi\_NoPrivilege** - 管理者には、アサーション コンシューマ サービスを追加する権限がありません。

### 注釈

**Sm\_PolicyApi\_SAMLSPAssertionConsumerService\_t** 構造の以下のフィールドが評価されます。

- インデックスの値は 65535 未満の一意の正の整数です。
- アサーション コンシューマ URL は **http://** または **https://** で始まります。
- バインディングは **HTTP-Post** または **HTTP\_Artifact** のいずれかです。

## Sm\_PolicyApi\_AddAttributeToAffiliate()

アフィリエイトに新規属性を追加します。

### タイプ

フェデレーション関数

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddAttributeToAffiliate (
    void*                pSessionHandle,
    const Sm_PolicyApi_AffiliateAttr_t* pstructAffiliateAttr,
    const char*          pszAffiliateOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pstructAffiliateAttr</i>	I	値の入力が完了しているアフィリエイト属性の構造体へのポインタです。
<i>pszAffiliateOid</i>	I	既存のアフィリエイトのオブジェクト識別子が含まれ、NULL で終わる文字列です。

### 戻り値

- Sm\_PolicyApi\_Success。アフィリエイト属性が正常に作成されました。
- Sm\_PolicyApi\_Failure - アフィリエイト属性が正常に作成されませんでした。
- Sm\_PolicyApi\_InvalidHandle - このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession - 有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege - 管理者には、アフィリエイト属性を作成する権限がありません。
- Sm\_PolicyApi\_InvalidOid - アフィリエイト OID が見つかりませんでした。

## Sm\_PolicyApi\_AddAttributeToSAMLScheme()

設定された属性サービスからリクエストできる属性を、SAML 2.0 認証方式に追加します。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddAttributeToSAMLScheme(
    void*                pHandle,
    const Sm_PolicyApi_Scheme_t*    pstructScheme,
    const Sm_PolicyApi_SAMLRequesterAttr_t*    pAttr
);
```

パラメータ	I/O	説明
<i>pHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pstructScheme</i>	I	SAML 2.0 方式に対して値が完全に入力されている構造体へのポインタです。
<i>pAttr</i>	I	追加される属性を含む Sm_PolicyApi_SAMLRequesterAttr_t 構造体へのポインタです。

### 戻り値

- Sm\_PolicyApi\_Success。属性が正常に追加されました。
- Sm\_PolicyApi\_Failure。属性が正常に追加されませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、属性を SAML リクエストに追加する権限がありません。
- Sm\_PolicyApi\_SAMLIDP\_IncorrectParameters。入力された SAML 提供のプロパティが未完了か、または正しくありません。
- Sm\_PolicyApi\_DuplicateAttribute。同じ「名前」と NameFormat の属性がすでに存在します。

## Sm\_PolicyApi\_AddAttributeToSAMLSP()

サービス プロバイダに対して SAML 2.0 属性を定義します。

SAML 2.0 属性には、プリンシパルのユーザ DN など、サービス プロバイダのリソースにアクセスしようとするプリンシパルの情報が含まれます。

定義された属性は、サービス プロバイダに対して作成されるすべての SAML 2.0 アサーション用の属性ステートメントに含まれています。

### タイプ

SAML 2.0 設定関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddAttributeToSAMLSP (
    void*                pHandle,
    const Sm_PolicyApi_SAMLSPAttr_t* pstructSAMLSPAttr,
    const char*          pszSAMLSPoid
);
```

パラメータ	I/O	説明
<i>pHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pstructSAMLSPAttr</i>	I	値の入力が完了している属性の構造体へのポインタです。
<i>pszSAMLSPoid</i>		既存のサービス プロバイダのオブジェクト識別子が含まれ、NULL で終わる文字列です。

### 戻り値

- Sm\_PolicyApi\_Success。 属性が正常に追加されました。
- Sm\_PolicyApi\_Failure。 属性が正常に追加されませんでした。
- Sm\_PolicyApi\_InvalidHandle。 このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。 有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。 管理者には、属性を SAML サービス プロバイダに追加する権限がありません。
- Sm\_PolicyApi\_DuplicateAttribute。 同じ「名前」と NameFormat の属性がすでに存在します。

## Sm\_PolicyApi\_AddDomain()

新規 SiteMinder ドメインを作成します。 ドメインの属性は *pstructDomain* 構造に含まれています。

ドメインが存在し、*bUpdate* フラグが「true」の場合、アイテムが更新されます。

### タイプ

ドメイン関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddDomain (
    void*                pSessionHandle,
    Sm_PolicyApi_Domain_t* pstructDomain,
    const bool           bUpdate
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pstructDomain</i>	I	値の入力が完了しているドメイン構造体へのポインタです。

---

パラメータ	I/O	説明
<i>bUpdate</i>	I	既存のオブジェクトが見つかる場合にそれが更新されることを示すフラグです。

---

#### 戻り値

- Sm\_PolicyApi\_Success。ドメインが正常に作成されました。
- Sm\_PolicyApi\_Failure。ドメインが正常に作成されませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、ドメインを作成する権限がありません。
- Sm\_PolicyApi\_InvalidOid。ドメイン OID が更新中に見つかりませんでした。

## Sm\_PolicyApi\_AddGlobalPolicy()

オブジェクトストアで新規のグローバルポリシーを作成します。ポリシーの属性は *pStructPolicy* 構造に含まれています。

ポリシーが存在し、*bUpdate* フラグが「true」の場合、アイテムが更新されます。

### タイプ

ポリシー関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddGlobalPolicy (
    void*                pSessionHandle,
    Sm_PolicyApi_Policy_t* pStructPolicy,
    const bool           bUpdate
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pStructPolicy</i>	I	値の入力が完了しているポリシー構造体へのポインタです。構造体のドメイン <b>OID</b> が無視されます。
<i>bUpdate</i>	I	既存のオブジェクトが見つかる場合にそれが更新されることを示すフラグです。

### 戻り値

- `Sm_PolicyApi_Success`。グローバルポリシーが正常に作成されました。
- `Sm_PolicyApi_Failure`。グローバルポリシーが正常に作成されませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、グローバルポリシーを作成する権限がありません。
- `Sm_PolicyApi_NotImplemented` : この関数を呼び出したクライアントが、`SM_POLICY_API_VERSION_6_0` より以前のバージョンで API を初期化しました。

## Sm\_PolicyApi\_AddGlobalResponse()

オブジェクトストアで新規のグローバルレスポンスを作成します。

### タイプ

ルール関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddGlobalResponse (
    void*                pSessionHandle,
    Sm_PolicyApi_Response_t* pStructResponse,
    const bool           bUpdate
);
```

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<code>pStructResponse</code>	I	値の入力が完了しているレスポンス構造体へのポインタです。構造体のドメイン <code>OID</code> が無視されます。

パラメータ	I/O	説明
<i>bUpdate</i>	I	既存のオブジェクトが見つかる場合にそれが更新されることを示すフラグです。

### 戻り値

- **Sm\_PolicyApi\_Success**。グローバル レスポンスが正常に作成されました。
- **Sm\_PolicyApi\_Failure**。グローバル レスポンスが正常に作成されませんでした。
- **Sm\_PolicyApi\_InvalidHandle**。このコールに先立つ有効な初期化はありませんでした。
- **Sm\_PolicyApi\_NoSession**。有効な管理者セッションがありません。
- **Sm\_PolicyApi\_NoPrivilege**。管理者には、グローバル レスポンスを作成する権限がありません。
- **Sm\_PolicyApi\_InvalidOid**：一致する **OID** のグローバル レスポンスが更新中に見つかりませんでした。
- **Sm\_PolicyApi\_NotImplemented**：この関数を呼び出したクライアントが、**SM\_POLICY\_API\_VERSION\_6\_0** より以前のバージョンで **API** を初期化しました。

## Sm\_PolicyApi\_AddGlobalRule()

オブジェクトストアで新規のグローバルルールを作成します。

### タイプ

ルール関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddGlobalRule (  
    void*                pSessionHandle,  
    Sm_PolicyApi_Rule_t* pStructRule,  
    const bool           bUpdate  
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pStructRule</i>	I	値の入力が完了しているルール構造体へのポインタです。構造内のレلم OID が無視されます。
<i>bUpdate</i>	I	既存のオブジェクトが見つかる場合にそれが更新されることを示すフラグです。

### 戻り値

- `Sm_PolicyApi_Success`。 グローバル ルールが正常に作成されました。
- `Sm_PolicyApi_Failure`。 グローバルルールが正常に作成されませんでした。
- `Sm_PolicyApi_InvalidHandle`。 このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。 有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。 管理者には、グローバルルールを作成する権限がありません。
- `Sm_PolicyApi_InvalidOid` :
  - 一致する `OID` のグローバルルールが更新中に見つかりませんでした。
  - 一致する `OID` のエージェントまたはエージェント グループは見つかりませんでした。
- `Sm_PolicyApi_NotImplemented` : この関数を呼び出したクライアントが、`SM_POLICY_API_VERSION_6_0` より以前のバージョンで API を初期化しました。

### `Sm_PolicyApi_AddGroup()`

Siteminder ポリシー ストアに新規のグループ オブジェクトを追加します。グループの属性は `pStructGroup` 構造に含まれています。

注: グローバル オブジェクトの グループはサポートされていません。

`pszDomainOid` パラメータはルール グループまたはレスポンス グループに必須です。 エージェント グループはドメインベースのオブジェクトではないので、ドメイン `OID` を必要としません。

グループ オブジェクトが存在し、`bUpdate` フラグが「true」の場合、アイテムが更新されます。

### タイプ

グループ関数、グローバル範囲（エージェント）またはドメイン範囲（レスポンス、ルール）。

## 構文

```
int SM_EXTERN Sm_PolicyApi_AddGroup (
    void*                pSessionHandle,
    Sm_PolicyApi_Groups_t dwGroup,
    const char*          pszDomainOid,
    Sm_PolicyApi_Group_t* pStructGroup,
    const bool           bUpdate
);
```

パラメータ	I/O	説明
<i>pSession Handle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>dwGroup</i>	I	追加されるグループのタイプを示します。
<i>pszDomainOid</i>	I	既存のドメインの名前が含まれ、NULL で終わる文字列です。ルールまたはレスポンス グループの必須パラメータです。（グローバルルールまたはレスポンス グループはサポートされていません。）
<i>pStructGroup</i>	I	値の入力が完了しているグループ構造体へのポインタです。
<i>bUpdate</i>	I	既存のオブジェクトが見つかる場合にそれが更新されることを示すフラグです。

## 戻り値

- `Sm_PolicyApi_Success`。追加が成功しました。
- `Sm_PolicyApi_Failure`。追加は成功しませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、グループを追加する権限がありません。
- `Sm_PolicyApi_InvalidOid`。ドメイン OID が見つかりませんでした（ドメインベースのグループについて）。
- `Sm_PolicyApi_BadGroup`。 *dwGroup* パラメータはルール、レスポンス、またはエージェント グループタイプではありません。

## Sm\_PolicyApi\_AddHostConfig()

ポリシーストアのホスト設定オブジェクトを作成または更新します。

### タイプ

エージェント設定関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddHostConfig (
    void*                pSessionHandle,
    Sm_PolicyApi_HostConfig_t* pstructHostConfig,
    const bool           bUpdate
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pstructHostConfig</i>	I	ホスト設定オブジェクトを定義する構造体のアドレスです。
<i>bUpdate</i>	I	真の場合、オブジェクトが更新されています。

### 戻り値

- Sm\_PolicyApi\_Success。操作は成功しました。
- Sm\_PolicyApi\_Failure。一般的な障害です。
- Sm\_PolicyApi\_InvalidOid。一意の ID がホスト設定オブジェクトに対応しません。
- Sm\_PolicyApi\_NoPrivilege。呼び出し元に、ホスト設定オブジェクトを作成または変更する権限がありません。
- Sm\_PolicyApi\_InvalidHandle。セッションポインタが無効です。
- Sm\_PolicyApi\_NoSession。API ユーザが正しくログインしていません。

## Sm\_PolicyApi\_AddMessageConsumerPluginToSAML1xScheme()

メッセージ コンシューマ プラグイン設定を SAML 1.x 認証方式に追加または更新します。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddMessageConsumerPluginToSAML1xScheme(  
    void*    pHandle,  
    char*    pszSchemeOID,  
    char*    pluginClass,  
    char*    pluginParam  
);
```

### パラメータ

#### pHandle

[in] 内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

#### pszSchemeOid

[in] 更新中の認証方式の OID へのポインタです。

#### pluginClass

[in] 設定するメッセージ コンシューマ プラグイン クラスの名前へのポインタです。

#### pluginParam

[in] 設定するメッセージ コンシューマ プラグイン クラスのパラメータへのポインタです。

### 戻り値

- Sm\_PolicyApi\_Success。操作は正常に完了しました。
- Sm\_PolicyApi\_Failure。動作が失敗しました。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_SchemeNotFound。OID に対応する認証方式が見つからなかったか、SAML 1.x 認証方式ではありませんでした。
- Sm\_PolicyApi\_InvalidOid。認証方式の OID は NULL です。

## Sm\_PolicyApi\_AddOneTimeUsePropToAffiliate()

SAML 1.x アフィリエイトでアサーションの **OneTimeUse** プロパティを追加または更新します。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddOneTimeUsePropToAffiliate(  
    void*    pHandle,  
    char*    pszAffiliateOID,  
    bool     bOneTimeUse  
);
```

### パラメータ

#### pHandle

[in] 内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

#### pszAffiliateOid

[in] 既存の SAML 1.x アフィリエイトの OID へのポインタです。

#### bOneTimeUse

[in] アサーションがこのアフィリエイトで 1 回のみ使用されるかどうかを指定するブール値です。

### 戻り値

- Sm\_PolicyApi\_Success。操作は正常に完了しました。
- Sm\_PolicyApi\_Failure。動作が失敗しました。
- Sm\_PolicyApi\_NoSession。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoPrivilege。管理者に十分なアクセス権限がありませんでした。

## Sm\_PolicyApi\_AddPasswordPolicy()

パスワード ポリシー オブジェクトを追加します。

Sm\_PolicyApi\_PasswordPolicy\_t に関して、以下の点に注意してください。

- パスワード ポリシーがディレクトリ全体に適用される場合は (*bEntireDir* は「true」に設定)、*pszPath* と *pszClass* を空の文字列に設定する必要があります。これらの文字列が空になっていない場合、ポリシー管理 API は Sm\_PolicyAPI\_BadArgument を返します。
- *pszPasswordServicesRedirect* フィールドが空の文字列に設定されているとき、ポリシー管理 API は以下のデフォルト URL にそれを設定します。

```
/siteminderagent/forms/smpwservices.fcc
```

### タイプ

パスワード ポリシー関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddPasswordPolicy (
    void*                pSessionHandle,
    Sm_PolicyApi_PasswordPolicy_t* pstructPasswordPolicy,
    const bool           bUpdate
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pstructPasswordPolicy</i> ポリシー	I	パスワード ポリシーに関する情報を含んだ、Sm_PolicyApi_PasswordPolicy_t 構造へのポインタのアドレスです。
<i>bUpdate</i>	I	既存のオブジェクトが見つかる場合にそれが更新されることを示すフラグです。

### 戻り値

- Sm\_PolicyApi\_Success。 作成操作が成功しました。
- Sm\_PolicyApi\_Failure。 作成操作が成功しませんでした。
- Sm\_PolicyApi\_InvalidHandle。 このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。 有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。 管理者には、パスワードポリシーを作成する権限がありません。
- Sm\_PolicyApi\_InvalidOid :
  - パスワードポリシー OID が更新中に見つかりませんでした。
  - ユーザディレクトリ OID が見つかりませんでした。
- Sm\_PolicyApi\_BadArgument。 パスとクラスは空です。

## Sm\_PolicyApi\_AddPolicy()

新規 SiteMinder ポリシーを作成します。ポリシーの属性は *pStructPolicy* 構造に含まれています。

ポリシーが存在し、*bUpdate* フラグが「true」の場合、アイテムが更新されます。

### タイプ

ポリシー関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddPolicy (  
    void*                pSessionHandle,  
    Sm_PolicyApi_Policy_t* pStructPolicy,  
    const bool           bUpdate  
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pStructPolicy</i>	I	値の入力が完了しているポリシー構造体へのポインタです。
<i>bUpdate</i>	I	既存のオブジェクトが見つかる場合にそれが更新されることを示すフラグです。

### 戻り値

- `Sm_PolicyApi_Success`。ポリシーが正常に作成されました。
- `Sm_PolicyApi_Failure`。ポリシーが正常に作成されませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、ポリシーを作成する権限がありません。
- `Sm_PolicyApi_InvalidOid` :
  - ポリシー `OID` が更新中に見つかりませんでした。
  - ドメイン `OID` が見つかりませんでした。

## Sm\_PolicyApi\_AddPolicyLink()

*pszPolicyOid* によって識別されたポリシーに対して新規 SiteMinder ポリシー リンクを作成します。

ポリシー リンク オブジェクトは、ポリシーをルールおよびオプションでレスポンスにバインドします。新規ポリシー リンクの属性は、*pstructPolicyLink* 構造に含まれています。

ポリシー リンクが存在し、*bUpdate* フラグが「true」の場合、アイテムが更新されます。

### タイプ

ポリシー関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddPolicyLink (
    void*                pSessionHandle,
    const                pszPolicyOid,
    Sm_PolicyApi_PolicyLink_t* pstructPolicyLink
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszPolicyOid</i>	I	既存のポリシーのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pstructPolicyLink</i>	I	値の入力が完了しているポリシー リンク構造体へのポインタです。

### 戻り値

- `Sm_PolicyApi_Success`。ポリシー リンクが正常に作成されました。
- `Sm_PolicyApi_Failure` :
  - ルール `OID` のドメインはポリシー `OID` のドメインと同じではありません。
  - レスポンス `OID` のドメインはポリシー `OID` のドメインと同じではありません。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、ポリシー リンクを作成する権限がありません。
- `Sm_PolicyApi_InvalidOid` :
  - ポリシー `OID` が見つかりませんでした。
  - ルールまたはルールグループ `OID` が見つかりませんでした。
  - レスポンスまたはレスポンスグループ `OID` が見つかりませんでした。

## Sm\_PolicyApi\_AddRealm()

*pStructRealm* で指定されたドメイン内の新規 SiteMinder レルムを作成します。*pStructRealm* 構造には、レルムを保護するエージェントまたはエージェントグループなどの、レルムの他の属性も含まれます。

保護するエージェントまたはエージェントグループが指定されていない場合、*Sm\_PolicyApi\_AddRealm()* は失敗します。

新規レルムがトップレベルのレルムである場合は、*pszParentRealmOid* (*Sm\_PolicyApi\_Realm\_t* 内の) をドメイン OID に設定します。トップレベルのレルムでない場合は、*pszParentRealmOid* を新規レルムの親レルムの OID に設定します。

レルムが存在し、*bUpdate* フラグが「true」の場合、既存アイテムが更新されます。

許可ディレクトリをレルムに正常に追加するためには、クライアントアプリケーションで以下の条件を満たす必要があります。

- 許可ユーザディレクトリと認証ユーザディレクトリとの間のディレクトリマッピングが存在する。

**注:** ポリシー管理 API は、許可ユーザディレクトリが存在するかどうかの確認のみ検証します。指定された許可ユーザディレクトリに対してディレクトリマッピングが存在するかどうかは検証しません。

- 関連する認証ユーザディレクトリが、ドメインのユーザディレクトリコレクションに存在する。

### タイプ

レルム関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddRealm (
    void*                pSessionHandle,
    Sm_PolicyApi_Realm_t* pStructRealm,
    const bool           bUpdate
);
```

パラメータ	I/O	説明
-------	-----	----

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pStructRealm</i>	I	値の入力が完了している構造体へのポインタです。
<i>bUpdate</i>	I	既存のオブジェクトが見つかる場合にそれが更新されることを示すフラグです。

### 戻り値

- Sm\_PolicyApi\_Success。 レルムが正常に作成されました。
- Sm\_PolicyApi\_Failure。 レルムが正常に作成されませんでした。
- Sm\_PolicyApi\_NoSession。 有効な管理者セッションがありません。
- Sm\_PolicyApi\_InvalidHandle。 このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoPrivilege。 管理者には、レルムを作成する権限がありません。
- Sm\_PolicyApi\_InvalidOid :
  - レルム OID が更新中に見つかりませんでした。
  - ドメイン OID が見つかりませんでした。
  - エージェント OID またはエージェントグループ OID が見つかりませんでした。
  - 方式 OID が見つかりませんでした。
  - 親 OID を見つけることができませんでした。(親 OID は、レルム OID またはドメイン OID になることができます。)

## Sm\_PolicyApi\_AddRedirectURLToSAML1xScheme()

SAML1.x 認証方式でリダイレクト URL 設定を追加または更新します。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddRedirectURLToSAML1xScheme(  
    void*          pSessionHandle,  
    const char*    pszSchemeOid,  
    int            iTypeURL,  
    char*          URL,  
    int            redirectMode  
);
```

### パラメータ

#### pSessionHandle

[in] 内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

#### pszSchemeOid

[in] 更新中の認証方式のオブジェクト識別子が含まれ、NULL で終わる文字列です。

#### iTypeUrl

[in] リダイレクト URL のタイプを指定する整数であり、Sm\_PolicyAPI\_SAML1\_STATUS\_REDIRECT\_URL\_TYPE\_t で以下のように定義されます。

- Sm\_PolicyAPI\_SAML1\_STATUS\_REDIRECT\_URL\_USER\_NOT\_FOUND\_TYPE = 0
- Sm\_PolicyAPI\_SAML1\_STATUS\_REDIRECT\_URL\_INVALID\_SSO = 1
- Sm\_PolicyAPI\_SAML1\_STATUS\_REDIRECT\_URL\_UNACCEPTABLE\_USER\_CREDENTIALS = 2

#### URL

[in] 入力リダイレクト URL へのポインタです。

#### redirectMode

[in] 入力リダイレクトモードを指定する整数です。「302 データなし」(302 No Data) の場合に 0、または「Http-Post」の場合に 1 になります。

### 戻り値

- Sm\_PolicyApi\_Success。操作は正常に完了しました。
- Sm\_PolicyApi\_Failure。動作が失敗しました。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_SchemeNotFound。OID に対応する認証方式が見つからなかったか、SAML 1.x 認証方式ではありませんでした。
- Sm\_PolicyApi\_InvalidOid。認証方式の OID は NULL です。

## Sm\_PolicyApi\_AddRegistrationScheme()

登録方式を追加します。

### タイプ

登録方式関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddRegistrationScheme (
    void*                pSessionHandle,
    Sm_PolicyApi_RegistrationScheme_t*
                        pstructRegistrationScheme,
    const bool           bUpdate
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pstruct</i> 登録方式	I	登録方式に関する情報を含んだ、Sm_PolicyApi_RegistrationScheme_t 構造へのポインタのアドレスです。
<i>bUpdate</i>	I	既存のオブジェクトが見つかる場合にそれが更新されることを示すフラグです。

### 戻り値

- Sm\_PolicyApi\_Success。登録方式が正常に作成されました。
- Sm\_PolicyApi\_Failure。登録方式が正常に作成されませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、登録方式を作成する権限がありません。
- Sm\_PolicyApi\_InvalidOid :
  - 登録方式オブジェクト識別子が更新中に見つかりませんでした。
  - ユーザディレクトリオブジェクト識別子は見つかりませんでした。

## Sm\_PolicyApi\_AddRegularExpressionToPasswordPolicy()

参照されるパスワードポリシーに正規表現を追加します。セッションのバージョンが SM\_POLICY\_API\_VERSION\_6\_0 に設定される場合にのみ実装されます。

### タイプ

正規表現関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddRegularExpressionToPasswordPolicy (
    void*                pSessionHandle,
    const char*          pszPasswordPolicyOid,
    Sm_PolicyApi_RegularExpression_t* pstructRegExpr
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

パラメータ	I/O	説明
<i>pszPassword PolicyOid</i>	I	正規表現を追加するパスワードポリシーの <b>OID</b> です。
<i>pstructRegExpr</i>	I	追加する正規表現構造へのポインタです。

---

#### 戻り値

- **Sm\_PolicyApi\_Success**。正規表現が正常に追加されました。
- **Sm\_PolicyApi\_Failure**。正規表現が正常に追加されませんでした。
- **Sm\_PolicyApi\_NoSession**。有効な管理者セッションがありません。
- **Sm\_PolicyApi\_InvalidHandle**。このコールに先立つ有効な初期化はありませんでした。
- **Sm\_PolicyApi\_NoPrivilege**。管理者には、正規表現を追加する権限がありません。
- **Sm\_PolicyApi\_InvalidOid**：パスワードポリシー **OID** が見つかりませんでした。

## Sm\_PolicyApi\_AddResponse()

新規 SiteMinder レスポンスを作成します。レスポンス自体の属性は *pStructResponse* 構造に含まれています。

レスポンスが存在し、*bUpdate* フラグが「true」の場合、アイテムが更新されます。

### タイプ

レスポンス関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddResponse (  
    void*                pSessionHandle,  
    Sm_PolicyApi_Response_t* pStructResponse,  
    const bool           bUpdate  
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pStructResponse</i>	I	値の入力が完了しているレスポンス構造体へのポインタです。
<i>bUpdate</i>	I	既存のオブジェクトが見つかる場合にそれが更新されることを示すフラグです。

### 戻り値

- `Sm_PolicyApi_Success`。 レスポンスが正常に作成されました。
- `Sm_PolicyApi_Failure`。 レスポンスが正常に作成されませんでした。
- `Sm_PolicyApi_NoSession`。 有効な管理者セッションがありません。
- `Sm_PolicyApi_InvalidHandle`。 このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoPrivilege`。 管理者には、レスポンスを作成する権限がありません。
- `Sm_PolicyApi_InvalidOid` :
  - レスポンス `OID` が更新中に見つかりませんでした。
  - ドメイン `OID` が見つかりませんでした。
  - エージェントタイプ `OID` が見つかりませんでした。

## Sm\_PolicyApi\_AddResponseAttr()

`pszResponseOid` によって識別されたレスポンス内で新規 SiteMinder レスポンス属性を作成します。レスポンス属性は `pstructResponseAttr` パラメータに含まれています。

### タイプ

レスポンス関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddResponseAttr (  
    void*                pSessionHandle,  
    const char*          pszResponseOid,  
    Sm_PolicyApi_ResponseAttr_t* pstructResponseAttr  
);
```

---

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

---

---

パラメータ	I/O	説明
<i>pszResponseOid</i>	I	既存のレスポンスのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pstructResponseAttr</i>	I	値の入力が完了しているレスポンス構造体へのポインタです。

---

### 戻り値

- `Sm_PolicyApi_Success`。レスポンス属性が正常に追加されました。
- `Sm_PolicyApi_Failure`。レスポンス属性が正常に追加されませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoPrivilege`。管理者には、レスポンス属性を追加する権限がありません。
- `Sm_PolicyApi_InvalidOid` :

レスポンス OID が見つかりませんでした。

- エージェントタイプ属性 OID が見つかりませんでした。

## Sm\_PolicyApi\_AddRule()

新規 SiteMinder ルールを作成します。ルール自体の属性は、*pStructRule* によって参照される構造体に含まれています。

ルールが存在し、*bUpdate* フラグが「true」の場合、アイテムが更新されます。

注: ルールは、レルムと常に関連付けられます。ルール名は、ドメイン内ではなくレルム内で一意です。

### タイプ

ルール関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddRule (
    void*                pSessionHandle,
    Sm_PolicyApi_Rule_t* pStructRule,
    const bool           bUpdate
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pStructRule</i>	I	完了したルール構造体へのポインタです。
<i>bUpdate</i>	I	既存のオブジェクトが見つかる場合にそれが更新されることを示すフラグです。

### 戻り値

- Sm\_PolicyApi\_Success。 ルールが正常に作成されました。
- Sm\_PolicyApi\_Failure。 ルールが正常に作成されませんでした。
- Sm\_PolicyApi\_InvalidHandle。 このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。 有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。 管理者には、ルールを作成する権限がありません。
- Sm\_PolicyApi\_InvalidOid :
  - ルール OID が更新中に見つかりませんでした。
  - レルム OID が見つかりませんでした。

## Sm\_PolicyApi\_AddSAMLAffiliation()

新規 SAML アフィリエーションオブジェクトを追加するか、または既存の SAML アフィリエーションオブジェクトを変更します。

### タイプ

SAML 2.0 設定関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddSAMLAffiliation (
    void*                pSessionHandle,
    Sm_PolicyApi_SAMLAffiliation_t* pstructAffiliation,
    const bool           bUpdate,
    char**               pszErrMsg
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pstructAffiliation</i>	I	値が完全に入力されている SAML アフィリエーション構造体へのポインタです。

パラメータ	I/O	説明
bUpdate	I	更新操作を既存のオブジェクトで実行するかどうかを指定します。
pszErrMsg	O	<p>アフィリエーションが正常に追加または更新されない場合に、エラーメッセージを含む文字列です。</p> <p>ポリシー管理 API はこのパラメータに対してメモリを動的に割り当てます。Sm_PolicyApi_FreeString() の呼び出しによるメモリの解放は、カスタム アプリケーションで行います。</p> <p>Sm_PolicyApi_MissingProperty または Sm_PolicyApi_InvalidProp が返される場合、このフィールドには欠落または無効のプロパティ名が含まれます。</p>

### 戻り値

- Sm\_PolicyApi\_Success。 操作は成功しました。
- Sm\_PolicyApi\_Failure。 操作は成功しませんでした。
- Sm\_PolicyApi\_InvalidProp。 指定されたプロパティは無効です。プロパティ名が *pszErrMsg* で返されます。
- Sm\_PolicyApi\_MissingProperty。 必要なプロパティが指定されませんでした。プロパティ名が *pszErrMsg* で返されます。
- Sm\_PolicyApi\_InvalidHandle。 このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。 有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。 Sm\_PolicyApi\_InvalidOid : 管理者には、SAML アフィリエーションを作成する権限がありません。
- Sm\_PolicyApi\_InsufficientData。 SAML アフィリエーションの設定に必要なプロパティがありませんでした。

## Sm\_PolicyApi\_AddSAMLScheme()

新規 SAML 2.0 認証方式オブジェクトを追加するか、または既存のものを  
変更します。この関数はまた、関連するアイデンティティプロバイダの  
メタデータ プロパティを定義します。メタデータ プロパティは認証方式  
で保存されます。

### タイプ

SAML 2.0 設定関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddSAMLScheme
(
    void*                                pHandle,
    Sm_PolicyApi_Scheme_t*              pstructScheme,
    Sm_PolicyApi_SAMLProviderProp_t*    pProps,
    const bool                           bUpdate,
    char**                               pszErrMsg
);
```

パラメータ	I/O	説明
<i>pHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pstructScheme</i>	I	値の入力が完了している認証方式の構造体へのポインタです。
<i>pProps</i>	I	認証方式と関連付けられた SAML 2.0 メタデータ プロパティです。 デフォルト値に関連付けられたプロパティに値を割り当てない場合、デフォルトが割り当てられます。
<i>bUpdate</i>	I	更新操作を既存のオブジェクトで実行するかどうかを指定します。

パラメータ	I/O	説明
<i>pszErrMsg</i>	O	<p>認証方式が正常に追加または更新されない場合に、エラーメッセージを含む文字列です。</p> <p>ポリシー管理 API はこのパラメータに対してメモリを動的に割り当てます。Sm_PolicyApi_FreeString() の呼び出しによるメモリの解放は、カスタム アプリケーションで行います。</p> <p>Sm_PolicyApi_MissingProperty または Sm_PolicyApi_InvalidProp が返される場合、このフィールドには欠落または無効のプロパティ名が含まれます。</p>

### 戻り値

- Sm\_PolicyApi\_Success。 SAML 2.0 認証方式の操作が成功しました。
- Sm\_PolicyApi\_Failure。 SAML 2.0 認証の操作が成功しませんでした。
- Sm\_PolicyApi\_InvalidProp。 指定されたプロパティは無効です。プロパティ名が *pszErrMsg* で返されます。
- Sm\_PolicyApi\_MissingProperty。 必要なプロパティが指定されませんでした。プロパティ名が *pszErrMsg* で返されます。
- Sm\_PolicyApi\_InvalidHandle。 このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。 有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。 管理者には、SAML 2.0 認証方式を作成する権限がありません。
- Sm\_PolicyApi\_SAMLIDP\_IncorrectParameters。 指定された SAML プロバイダプロパティが未完了か、または正しくありません。
- Sm\_PolicyApi\_SAMLIDP\_ProviderIdNotUnique。 指定された SAML プロバイダ ID は一意ではありません。

## Sm\_PolicyApi\_AddSAMLServiceProvider()

新規 SAML 2.0 サービス プロバイダ オブジェクトを追加するか、または既存のものを変更します。

サービス プロバイダを作成できない場合は、試行中にポリシー ストアで作成された関連するオブジェクトがロールバックされます。

### タイプ

SAML 2.0 設定関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddSAMLServiceProvider
(
    void*                pSessionHandle,
    Sm_PolicyApi_SAMLSP_t* pstructSAMLSP,
    const bool           bUpdate,
    char**               pszErrMsg
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pstructSAMLSP</i>	I	値の入力が完了しているサービス プロバイダ構造体へのポインタです。
<i>bUpdate</i>	I	更新操作を既存のオブジェクトで実行するかどうかを指定します。
<i>pszErrMsg</i>	O	サービス プロバイダが正常に追加または更新されない場合に、エラー メッセージを含む文字列です。 ポリシー管理 API はこのパラメータに対してメモリを動的に割り当てます。Sm_PolicyApi_FreeString() の呼び出しによるメモリの解放は、カスタム アプリケーションで行います。 Sm_PolicyApi_MissingProperty または Sm_PolicyApi_InvalidProp が返される場合、このフィールドには欠落または無効のプロパティ名が含まれます。

### 戻り値

- `Sm_PolicyApi_Success`。 サービス プロバイダ操作が成功しました。
- `Sm_PolicyApi_Failure`。 サービス プロバイダ操作が成功しませんでした。
- `Sm_PolicyApi_InvalidProp`。 指定されたプロパティは無効です。 プロパティ名が `pszErrMsg` で返されます。
- `Sm_PolicyApi_MissingProperty`。 必要なプロパティが指定されませんでした。 プロパティ名が `pszErrMsg` で返されます。
- `Sm_PolicyApi_InvalidHandle`。 このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。 有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。 管理者には、サービス プロバイダを作成する権限がありません。
- `Sm_PolicyApi_DomainNotFound`。 アフィリエイト ドメイン OID が見つかりませんでした。
- `Sm_PolicyApi_DomainNotAffiliate`。 ドメインはアフィリエイト ドメインではありません。
- `Sm_PolicyApi_InsufficientSPData`。 サービス プロパティの設定に必要なプロパティがありませんでした。

## Sm\_PolicyApi\_AddScheme()

新規の SiteMinder 認証方式を作成します。認証方式の属性は *pstructScheme* 構造に含まれています。

認証方式が存在し、*bUpdate* フラグが「true」の場合、アイテムが更新されます。

### タイプ

認証方式関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddScheme (  
    void*                pSessionHandle,  
    Sm_PolicyApi_Scheme_t* pstructScheme,  
    const bool           bUpdate  
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pstructScheme</i>	I	値の入力が完了している方式構造体へのポインタです。
<i>bUpdate</i>	I	既存のオブジェクトが見つかる場合にそれが更新されることを示すフラグです。

### 戻り値

- `Sm_PolicyApi_Success`。方式が正常に作成されました。
- `Sm_PolicyApi_Failure`。方式が正常に作成されませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、方式を作成する権限がありません。
- `Sm_PolicyApi_InvalidOid`。方式オブジェクト識別子が更新中に見つかりませんでした。

## Sm\_PolicyAPI\_AddTargetConfigToSAML1xScheme

SAML 1.x 認証方式内にデフォルトのターゲット設定を追加します。

### 構文

Sm\_PolicyApi\_AddTargetConfigToSAML1xScheme 関数の構文は、以下のとおりです。

```
int SM_EXTERN Sm_PolicyApi_AddTargetConfigToSAML1xScheme(  
    void*          pHandle,  
    const char*   pszSchemeOid,  
    const char*   pszURL,  
    int           iQPOverridesTarget  
);
```

### パラメータ

Sm\_PolicyApi\_AddTargetConfigToSAML1xScheme 関数は、以下のパラメータを受け取ります。

#### pHandle

[in] 内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

#### pszSchemeOid

[in] 更新中の認証方式のオブジェクト識別子が含まれ、NULL で終わる文字列です。

#### pszURL

[in] 有効なデフォルトのターゲット設定 URL へのポインタです。

#### iQPOverrides Target

[in] [クエリ パラメータでデフォルト ターゲットをオーバーライドする] チェック ボックスで指定される値を決める整数です。

iQPOverridesTarget の有効な値は次のとおりです。

- 0 を指定すると、クエリ パラメータによってターゲットが上書きされません。
- 1 を指定すると、クエリ パラメータによってターゲットが上書きされます。

### 戻り値

`Sm_PolicyApi_AddTargetConfigToSAML1xScheme` 関数は、以下の値のいずれかを返します。

- `Sm_PolicyApi_Success`。操作は正常に完了しました。
- `Sm_PolicyApi_Failure`。動作が失敗しました。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_SchemeNotFound`。OID に対応する認証方式が見つからなかったか、SAML 1.x 認証方式ではありませんでした。
- `Sm_PolicyApi_InvalidOid`。認証方式の OID は NULL です。

### `Sm_PolicyApi_AddToGroup()`

`pszGroupOid` によって識別されたグループに、アイテムを追加します。

アイテム（グループの場合もある）およびグループが存在し、かつ同じタイプである必要があります。

`pszDomainOid` パラメータはルールグループまたはレスポンスグループに必須です。エージェントグループはドメインベースのオブジェクトではないので、ドメイン OID を必要としません。

#### タイプ

グループ関数、グローバル範囲（エージェント）またはドメイン範囲（レスポンス、ルール）。

## 構文

```
int SM_EXTERN Sm_PolicyApi_AddToGroup (
    void*                pSessionHandle,
    Sm_PolicyApi_Groups_t dwGroup,
    const char*          pszItemOid,
    const char*          pszGroupOid,
    const char*          pszDomainOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>dwGroup</i>	I	グループのタイプ。
<i>pszItemOid</i>	I	グループと同じタイプの既存のアイテムのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszGroupOid</i>	I	<i>dwGroup</i> によって示されるタイプのグループのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszDomainOid</i>	I	既存のドメインのオブジェクト識別子が含まれ、NULL で終わる文字列です。ルールおよびレスポンス オブジェクトに必須です。

## 戻り値

- `Sm_PolicyApi_Success`。追加が成功しました。
- `Sm_PolicyApi_Failure`。追加は成功しませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、ルール、レスポンスまたはエージェント OID をその各グループに追加する権限がありません。

- Sm\_PolicyApi\_InvalidOid :
  - ドメイン OID が見つかりませんでした（ドメインベースのグループについて）。
  - グループ OID が見つかりませんでした。
  - ルール、レスポンス、またはエージェント OID またはグループ OID が見つかりませんでした。
- Sm\_PolicyApi\_BadGroup。パラメータ *dwGroup* はルール、レスポンスまたはエージェント タイプではありません。

## Sm\_PolicyApi\_AddTrustedHost()

オブジェクトストアでトラステッドホスト オブジェクトを作成または変更します。

この関数を使用して、トラステッドホストの「オフライン」を登録します。つまり、エージェントとポリシーサーバ間の接続を行いません。この関数を使用するとき、SiteMinder smreghost ツールを実行してホスト設定ファイル（デフォルト名は *SmHost.conf*）で共有秘密キーを定義する必要があります。smreghost ツールの *-sh* オプションで共有秘密キーを定義します。

### タイプ

エージェント設定関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddTrustedHost (
    void*                pSessionHandle,
    Sm_PolicyApi_TrustedHost_t* pstructTrustedHost,
    bool                 bUpdate,
    bool                 bGenSharedSecret
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	セッションの識別子を指定します。
<i>pstructTrustedHost</i>	I/O	Sm_PolicyApi_TrustedHost_t タイプの構造体のアドレスです。構造体は、関数を呼び出す前に呼び出し元によって入力されます。

パラメータ	I/O	説明
<i>bUpdate</i>	I	オブジェクトストアの既存のオブジェクトを更新するために関数が呼び出されたかどうかを指定します。
<i>bGenSharedSecret</i>	I	共有秘密キーを生成するかどうかを示します。

### 注釈

*bGenSharedSecret* が「真」の場合、関数では 128 バイト値を生成し、*pstructTrustedHost* の *pszSecret* フィールドを更新します。  
*bGenSharedSecret* が「偽」の場合、新規または更新するトラステッドホストオブジェクト用の共有秘密キーの値は、*pstructTrustedHost* 構造の *pszSecret* フィールドから取得されます。

*bGenSharedSecret* と *bUpdate* の両方が「真」の場合、関数では *pstructTrustedHost* の *pszSecret* フィールドで指定された値を無視し、新たに値を生成し、オブジェクトストアおよび *pstructTrustedHost* の *pszSecret* フィールドにおけるオブジェクトを更新します。

関数で共有秘密キーを生成する場合、*smreghost* の *-sh* オプションで定義できるように、生成された共有秘密キーをクリアテキストで取得する必要があります。共有秘密キーを取得するには、*Sm\_PolicyApi\_GetTrustedHost()* を呼び出します。

過去のリリースでは、ポリシーサーバのエージェント登録で常に 128 バイトのランダムな ASCII 共有秘密キーを使用していました。新しいモデルでは、共有秘密キーとしてユーザ定義の文字列値を使用できるようになっています。これは潜在的なセキュリティの脆弱性です。管理者はオフラインのエージェントホスト設定を選択した場合、強力な共有秘密キーを作成し、それを安全に格納する必要があります。*bGenSharedSecret* パラメータを「真」に設定して、*Sm\_PolicyApi\_AddTrustedHost()* 関数を呼び出すことを強くお勧めします。これにより、推測困難な共有秘密キーが自動生成されます。

この関数を使用する際に、トラステッドホストの作成や、ホスト設定の定義は要求されません。*SiteMinder* がインストール中および、*-sh* オプションを使用せずに *smreghost* を実行するときに、自動的にトラステッドホストを作成し設定します。

### 戻り値

- `Sm_PolicyApi_Success`。 トラストド ホスト オブジェクトが作成または変更されました。
- `Sm_PolicyApi_Failure`。 一般的な障害です。
- `Sm_PolicyApi_InvalidOid`。 トラストド ホスト オブジェクトに一意の ID が対応しません。
- `Sm_PolicyApi_NoPrivilege`。 呼び出し元に、トラストド ホスト オブジェクトを追加または変更するための適切な権限がありません。
- `Sm_PolicyApi_InvalidHandle`。 セッション ポインタが無効です。
- `Sm_PolicyApi_NotUnique`。 共有秘密キーはすでに存在します。
- `Sm_PolicyApi_NoSession`。 API ユーザが正しくログインしていません。
- `Sm_PolicyApi_InvalidProp`。 指定された共有秘密キーの値は無効です。
- `Sm_PolicyApi_GenSharedSecretFailure`。 128 -バイトの共有秘密キーの値が生成できませんでした。

## Sm\_PolicyApi\_AddUserDirToAffiliateDomain()

既存のアフィリエイト ドメインにユーザ ディレクトリを追加します。

### タイプ

フェデレーション関数

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddUserDirToAffiliateDomain (  
    void*      pSessionHandle,  
    const char* pszUserDirOid,  
    const char* pszAffiliateDomainOid  
);
```

---

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

---

パラメータ	I/O	説明
<i>pszUserDirOid</i>	I	既存のユーザディレクトリのオブジェクト識別子が含まれ、NULLで終わる文字列です。
<i>pszAffiliateDomainOid</i>	I	既存のアフィリエイトドメインのオブジェクト識別子が含まれ、NULLで終わる文字列です。

### 戻り値

- **Sm\_PolicyApi\_Success**。ユーザディレクトリが、アフィリエイトドメインに正常に追加されました。
- **Sm\_PolicyApi\_Failure**。ユーザディレクトリがアフィリエイトドメインに追加されませんでした。
- **Sm\_PolicyApi\_NoSession**。有効な管理者セッションがありません。
- **Sm\_PolicyApi\_InvalidHandle**。このコールに先立つ有効な初期化はありませんでした。
- **Sm\_PolicyApi\_NoPrivilege**。管理者には、ユーザディレクトリをアフィリエイトドメインに追加する権限がありません。
- **Sm\_PolicyApi\_InvalidOid**。アフィリエイトドメインOIDまたはユーザディレクトリOIDが更新中に見つかりませんでした。
- **Sm\_PolicyApi\_DuplicateEntry**。ユーザディレクトリがドメインのユーザディレクトリコレクションに存在します。

## Sm\_PolicyApi\_AddUserDirToDomain()

*pszUserDirOid* によって識別されるディレクトリ オブジェクトを、*pszDomainOid* によって識別されるドメインに関連付けます。ディレクトリ オブジェクトが検索順序の最後に追加されます。

### タイプ

ユーザ ディレクトリ 関数、グローバル 範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddUserDirToDomain (
    void*          pSessionHandle,
    const char*    pszUserDirOid,
    const char*    pszDomainOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszUserDirOid</i>	I	既存のユーザ ディレクトリのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszDomainOid</i>	I	既存のドメインのオブジェクト識別子が含まれ、NULL で終わる文字列です。

### 戻り値

- `Sm_PolicyApi_Success`。追加の操作が成功しました。
- `Sm_PolicyApi_Failure`。追加の操作は成功しませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoPrivilege`。管理者には、ユーザディレクトリをドメインに追加する権限がありません。
- `Sm_PolicyApi_InvalidOid`。ユーザディレクトリまたはドメイン OID が見つかりませんでした。
- `Sm_PolicyApi_DuplicateEntry`。ユーザディレクトリがドメインのユーザディレクトリコレクションに存在します。

## Sm\_PolicyApi\_AddUseSecureAuthPropToAffiliate()

UseSecureAuthURL プロパティを SAML 1.x アフィリエイトに追加または更新します。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddUseSecureAuthPropToAffiliate(  
    void*    pHandle,  
    char*    pszAffiliateOID,  
    bool     bUseSecureAuthURL  
);
```

### パラメータ

#### pHandle

[in] 内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

#### pszAffiliateOid

[in] 既存の SAML 1.x アフィリエイトの OID へのポインタです。

#### bUseSecureAuthURL

[in] このアフィリエイトに対し安全な認証 URL を使用するかどうかを指定するブール値です。

### 戻り値

- Sm\_PolicyApi\_Success。操作は正常に完了しました。
- Sm\_PolicyApi\_Failure。動作が失敗しました。
- Sm\_PolicyApi\_NoSession。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoPrivilege。管理者に十分なアクセス権限がありませんでした。

## Sm\_PolicyApi\_AddUsersToAffiliate()

ユーザディレクトリ エントリをアフィリエイトに追加します。

## タイプ

フェデレーション関数

## 構文

```
int SM_EXTERN Sm_PolicyApi_AddUsersToAffiliate (
    void*                pSessionHandle,
    const char*          pszAffiliateOid,
    Sm_PolicyApi_User_t *pStructUsers,
    int                  iPolicyFlags
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAffiliateOid</i>	I	既存のアフィリエイトのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pStructUsers</i>	I	Sm_PolicyApi_User_t 構造体へのポインタ。ユーザディレクトリに関する情報が含まれます。
<i>iPolicyFlags</i>	I	1 ビットのフィールド値。ポリシーがユーザを含むか除外するかを示し、さらにポリシーが繰り返し適用されるかどうかを示します。ビットの定義については、112 ページの図 21 に一覧にしてあります。

### 戻り値

- Sm\_PolicyApi\_Success。 ユーザが正常に追加されました。
- Sm\_PolicyApi\_Failure。 ユーザが正常に追加されませんでした。
- Sm\_PolicyApi\_InvalidHandle。 このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。 有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。 管理者には、ユーザを追加する権限がありません。
- Sm\_PolicyApi\_InvalidOid： アフィリエイト OID が見つかりませんでした。
- Sm\_PolicyApi\_DuplicateEntry。 ユーザはすでにアフィリエイトに含まれています。

## Sm\_PolicyApi\_AddUsersToPolicy()

*pszPolicyOid* によって識別されるポリシーに、ユーザディレクトリエントリを追加します。1 ユーザの指定（まとめた指定の場合も）に限り、一度に追加できます。

### タイプ

ユーザとユーザ状態関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddUsersToPolicy (
    void*                pSessionHandle,
    const char*          pszPolicyOid,
    Sm_PolicyApi_User_t* pStructUsers,
    int                  iPolicyFlags
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

パラメータ	I/O	説明
<i>pszPolicyOid</i>	I	NULL で終わる文字列であり、ユーザが追加される既存のポリシーのオブジェクト識別子が含まれます。
<i>pStructUsers</i>	I	<b>Sm_PolicyApi_User_t</b> 構造体へのポインタ。ユーザディレクトリに関する情報が含まれます。
<i>iPolicyFlags</i>	I	1 ビットのフィールド値。ポリシーがユーザを含むか除外するかを示し、さらにポリシーが繰り返し適用されるかどうかを示します。ビットの定義はポリシーフラグに一覧されています。

### 戻り値

- **Sm\_PolicyApi\_Success**。追加が成功しました。
- **Sm\_PolicyApi\_Failure**。ユーザディレクトリはドメインユーザディレクトリコレクションに含まれていません。
- **Sm\_PolicyApi\_InvalidHandle**。このコールに先立つ有効な初期化はありませんでした。
- **Sm\_PolicyApi\_NoSession**。有効な管理者セッションがありません。
- **Sm\_PolicyApi\_NoPrivilege**。管理者には、ユーザをポリシーに追加する権限がありません。
- **Sm\_PolicyApi\_InvalidOid** :
  - ポリシー OID が見つかりませんでした。
  - ユーザディレクトリ OID が見つかりませんでした。
- **Sm\_PolicyApi\_DuplicateEntry**。ユーザはすでにポリシーに含まれています。
- **Sm\_PolicyApi\_InconsistentANDBitMask**。ポリシーの下にあるこの特定のユーザディレクトリの既存ユーザポリシーには、追加されるものに対して、**Sm\_PolicyBehavior\_AND\_Mask** セットの別の値があります。

## Sm\_PolicyApi\_AddUsersToSAMLServiceProvider()

ユーザディレクトリ エントリを SAML 2.0 サービス プロバイダに関連付けます。

### タイプ

SAML 2.0 設定関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddUsersToSAMLServiceProvider
(
    void*                pSessionHandle,
    const char*          pszProviderOid,
    Sm_PolicyApi_User_t* pStructUsers,
    int                  iPolicyFlags
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszProviderOid</i>	I	既存の SAML サービス プロバイダのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pStructUsers</i>	I	Sm_PolicyApi_User_t 構造体へのポインタ。ユーザディレクトリに関する情報が含まれます。
<i>iPolicyFlags</i>	I	1 ビットのフィールドであり、以下を示します。 <ul style="list-style-type: none"> <li>■ SAML サービス プロバイダに対して作成されたポリシーにユーザが含まれるかどうか</li> <li>■ ポリシーが再帰的に適用されるべきかどうか</li> </ul>

### 戻り値

- `Sm_PolicyApi_Success`。 ユーザが正常に追加されました。
- `Sm_PolicyApi_Failure`。 ユーザが正常に追加されませんでした。
- `Sm_PolicyApi_InvalidHandle`。 このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。 有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。 管理者には、ユーザを追加する権限がありません。
- `Sm_PolicyApi_InvalidOid`。 サービスプロバイダ `OID` が見つかりませんでした。
- `Sm_PolicyApi_DuplicateEntry`。 ユーザはすでにサービスプロバイダに含まれています。

## Sm\_PolicyApi\_AddUsersToWSFEDResourcePartner()

ユーザディレクトリ エントリを **WS**-フェデレーション リソース パートナーと関連付けます。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddUsersToWSFEDResourcePartner (  
    void* pSessionHandle,  
    const char * pszProviderOid,  
    Sm_PolicyApi_User_t *pStructUsers,  
    int iPolicyFlags  
);
```

### パラメータ

pSessionHandle

[in] 内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

pszProviderOid

[in] 既存の **WS** フェデレーション リソース パートナーのオブジェクト識別子が含まれ、**NULL** で終わる文字列です。

pStructUsers

[in] Sm\_PolicyApi\_User\_t 構造へのポインタ。ユーザディレクトリに関する情報が含まれます。

iPolicyFlags

[in] 1 ビットのフィールド。**WS** フェデレーション リソース パートナーに対して作成されたポリシーがユーザを含むか除外するかを示し、さらにポリシーが繰り返し適用されるかどうかを示します。

### 戻り値

- Sm\_PolicyApi\_Success。ユーザが正常に追加されました。
- Sm\_PolicyApi\_Failure。ユーザが正常に追加されませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、ユーザを追加する権限がありません。

- `Sm_PolicyApi_InvalidOid`。リソース パートナー OID が見つかりませんでした。
- `Sm_PolicyApi_DuplicateEntry`。ユーザはすでにリソース パートナーに含まれています。

## Sm\_PolicyApi\_AddVariable()

変数オブジェクトを追加します。

### タイプ

変数関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddVariable (  
    void*                pSessionHandle,  
    Sm_PolicyApi_Variable_t* pstructVariable,  
    const bool           bUpdate  
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pstructVariable</i>	I	<code>Sm_PolicyApi_Variable_t</code> 構造へのポインタ。変数に関する情報が含まれます。
<i>bUpdate</i>	I	既存のオブジェクトが見つかる場合にそれが更新されることを示すフラグです。

### 戻り値

- `Sm_PolicyApi_Success`。追加の操作が成功しました。
- `Sm_PolicyApi_Failure`。追加の操作は成功しませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、変数を追加する権限がありません。
- `Sm_PolicyApi_FeatureNotSupported`。この関数を呼び出したクライアントが、`SM_POLICY_API_VERSION_6_0` より以前のバージョンで API を初期化しました。

## Sm\_PolicyApi\_AddWSFEDResourcePartner()

WS フェデレーション リソース パートナーの新規オブジェクトを作成するか、または既存のオブジェクトを更新します。プロパティ（値および他のプロパティへの依存性）の検証が実行されます。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddWSFEDResourcePartner (  
    void* pSessionHandle,  
    Sm_PolicyApi_WSFEDResourcePartner_t* structServiceProvider,  
    const bool bUpdate,  
    char **pszErrMsg  
);
```

### パラメータ

#### pSessionHandle

[in] 内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

#### pstructServiceProvider

[in] 値の入力が完了している WS フェデレーション リソース パートナー構造体へのポインタです。

**bUpdate**

[in] 既存のオブジェクトが見つかる場合はそれを更新することを示すフラグです。

**pszErrMsg**

[out] この関数を呼び出して `Sm_PolicyApi_SAML_UnknownProperty`、`Sm_PolicyApi_MissingProperty` または `Sm_PolicyApi_InvalidProp` の値が返される場合、この変数にはエラーを発生させたプロパティ名が含まれます。`Sm_PolicyApi_FreeString()` を呼び出して、この変数に割り当てられたメモリを解放します。

**戻り値**

- `Sm_PolicyApi_Success`。 リソース パートナーが正常に作成されました。
- `Sm_PolicyApi_Failure`。 リソース パートナーが正常に作成されませんでした。
- `Sm_PolicyApi_InvalidHandle`。 このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。 有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。 管理者には、アフィリエイトを作成する権限がありません。
- `Sm_PolicyApi_DomainNotFound`。 アフィリエイト ドメイン OID が見つかりませんでした。
- `Sm_PolicyApi_DomainNotAffiliate`。 ドメインはアフィリエイト ドメインではありません。
- `Sm_PolicyApi_InsufficientRPData`。 リソース パートナーの設定に必要なプロパティがありませんでした。
- `Sm_PolicyApi_WSFED_UnknownProperty`。 不明なプロパティ名が指定されました。
- `Sm_PolicyApi_WSFEDRP_AssertionConsumerDefaultMissing`。 潜在的に依存性のために必要とされるプロパティが提供されませんでした。
- `Sm_PolicyApi_WSFEDRP_AuthenticationURLMissing`。 潜在的に依存性のために必要とされるプロパティが提供されませんでした。
- `Sm_PolicyApi_WSFEDRP_DomainOidMissing`。 潜在的に依存性のために必要とされるプロパティが提供されませんでした。

- `Sm_PolicyApi_WSFEDRP_APIDMissing`。潜在的に依存性のために必要とされるプロパティが提供されませんでした。
- `Sm_PolicyApi_WSFEDRP_NameMissing`。潜在的に依存性のために必要とされるプロパティが提供されませんでした。
- `Sm_PolicyApi_WSFEDRP_NameldFormatMissing`。潜在的に依存性のために必要とされるプロパティが提供されませんでした。
- `Sm_PolicyApi_WSFEDRP_NameldTypeMissing`。潜在的に依存性のために必要とされるプロパティが提供されませんでした。
- `Sm_PolicyApi_WSFEDRP_NameldStaticMissing`。潜在的に依存性のために必要とされるプロパティが提供されませんでした。
- `Sm_PolicyApi_WSFEDRP_NameldAttrNameMissing`。潜在的に依存性のために必要とされるプロパティが提供されませんでした。
- `Sm_PolicyApi_WSFEDRP_NameldDNSpecMissing`。潜在的に依存性のために必要とされるプロパティが提供されませんでした。
- `Sm_PolicyApi_WSFEDRP_ProviderIdMissing`。潜在的に依存性のために必要とされるプロパティが提供されませんでした。
- `Sm_PolicyApi_WSFEDRP_ProviderIdNotUnique`。潜在的に依存性のために必要とされるプロパティが提供されませんでした。
- `Sm_PolicyApi_WSFEDRP_UnSupportedSAMLVersion`。潜在的に依存性のために必要とされるプロパティが提供されませんでした。
- `Sm_PolicyApi_WSFEDRP_UnknownProperty`。潜在的に依存性のために必要とされるプロパティが提供されませんでした。
- `Sm_PolicyApi_MissingProperty`。必要なプロパティが提供されませんでした。依存性が原因となった可能性があります。
- `Sm_PolicyApi_InvalidProp`。入力されたプロパティの値は無効です。

### 注釈

リソースパートナーを作成できない場合、ポリシーストアで作成されたオブジェクトがロールバックされる必要があります。また、*affiliate:* と *samlsp:* とを区別するために、プレフィクス文字列として *wsfed:* が「名前」プロパティに使用されます。

## Sm\_PolicyApi\_AddWSFEDScheme()

SiteMinder WSFED 認証方式の新規オブジェクトを作成するか、または既存のオブジェクトを更新します。プロパティ（値および他のプロパティへの依存性）の検証が実行されます。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddWSFEDScheme(  
    void* pSessionHandle,  
    Sm_PolicyApi_Scheme_t* pstructScheme,  
    Sm_PolicyApi_WSFEDProviderProp_t* pProps,  
    const bool bUpdate,  
    char **pszErrMsg  
);
```

### パラメータ

#### pSessionHandle

[in] 内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

#### pstructScheme

[in] 値の入力が完了している方式の構造体へのポインタです。

#### pProps

[in] WSFED プロバイダ プロパティのリンク リストへのポインタです。

#### bUpdate

[in] 既存のオブジェクトが見つかる場合にそれが更新されることを示すフラグです。

#### pszErrMsg

[out] この関数を呼び出して Sm\_PolicyApi\_WSFED\_UnknownProperty、Sm\_PolicyApi\_MissingProperty または Sm\_PolicyApi\_InvalidProp の値が返される場合、この変数にはエラーを発生させたプロパティ名が含まれます。Sm\_PolicyApi\_FreeString() を呼び出して、この変数に割り当てられたメモリを解放します。

### 戻り値

- Sm\_PolicyApi\_Success。WSFED 認証方式が正常に作成されました。
- Sm\_PolicyApi\_Failure。WSFED 認証方式が正常に作成されませんでした。

- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、WSFED 認証方式を作成する権限がありません。
- `Sm_PolicyApi_WSFEDAP_IncorrectParameters`。提供された「WSFED アカウント パートナー」プロパティの入力が完了していないか、または正しくありません。
- `Sm_PolicyApi_WSFEDAP_ProviderIdNotUnique`。入力された WSFED アカウント パートナー ID は一意ではありません。
- `Sm_PolicyApi_WSFED_UnknownProperty`。不明なプロパティ名が指定されました。
- `Sm_PolicyApi_MissingProperty`。必要なプロパティが提供されませんでした。依存性が原因となった可能性があります。
- `Sm_PolicyApi_InvalidProp`。入力されたプロパティの値は無効です。

## Sm\_PolicyApi\_ConvertFromLegacyAgent()

v4.x エージェントを v5.x エージェントに変換します。

この関数を呼び出すことは、[エージェントプロパティ] ダイアログ ボックスの [4.x エージェントをサポートする] チェック ボックスをオフにすることと同じです。

### タイプ

エージェント関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_ConvertFromLegacyAgent (
    void*                pSessionHandle,
    Sm_PolicyApi_Agent_t* pStructAgent
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pStructAgent</i>	I	値の入力が完了しているエージェント構造へのポインタです。

### 戻り値

- `Sm_PolicyApi_Success`。 エージェントは正常に変換されました。
- `Sm_PolicyApi_Failure`。 エージェントの変換に失敗しました。
- `Sm_PolicyApi_InvalidProp`。 *pStructAgent* で定義された 1 つ以上のプロパティは無効です。
- `Sm_PolicyApi_InvalidHandle`。 このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。 有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。 管理者には、エージェントを変更する権限がありません。

## Sm\_PolicyApi\_ConvertToLegacyAgent()

v5.x エージェントを v4.x エージェントに変換します。

この関数を呼び出すことは、[エージェントプロパティ]ダイアログボックスの [4.x エージェントをサポートする] チェックボックスをオンにすることと同じです。

### タイプ

エージェント関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_ConvertToLegacyAgent (
    void*                pSessionHandle,
    Sm_PolicyApi_Agent_t* pStructAgent
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pStructAgent</i>	I	値の入力が完了しているエージェント構造へのポインタです。

### 戻り値

- Sm\_PolicyApi\_Success。 エージェントは正常に変換されました。
- Sm\_PolicyApi\_Failure。 エージェントの変換に失敗しました。
- Sm\_PolicyApi\_InvalidProp。 *pStructAgent* で定義された 1 つ以上のプロパティは無効です。
- Sm\_PolicyApi\_InvalidHandle。 このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。 有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。 管理者には、エージェントを変更する権限がありません。

## Sm\_PolicyApi\_CreateAuthAzMap()

認証および許可ディレクトリ マッピング オブジェクトを作成します。

### タイプ

認証/許可マップ関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_CreateAuthAzMap (  
    void*                pSessionHandle,  
    Sm_PolicyApi_AuthAzMap_t* pAuthAzMap,  
    const bool           bUpdate  
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pAuthAzMap</i>	I	認証および許可マッピングに関する情報へのポインタのアドレスです。
<i>bUpdate</i>	I	既存のオブジェクトが見つかる場合にそれが更新されることを示すフラグです。

### 戻り値

- **Sm\_PolicyApi\_Success**。ディレクトリ マッピング オブジェクトが正常に作成されました。
- **Sm\_PolicyApi\_InvalidHandle**。このコールに先立つ有効な初期化はありませんでした。
- **Sm\_PolicyApi\_NoSession**。有効な管理者セッションがありません。
- **Sm\_PolicyApi\_NoPrivilege**。管理者には、ディレクトリ マッピング オブジェクトを作成する権限がありません。
- **Sm\_PolicyApi\_InvalidOid** :
  - 認証ユーザ ディレクトリ **OID** は見つかりませんでした。
  - 許可ユーザ ディレクトリ **OID** が見つかりませんでした。
  - ディレクトリ マッピング オブジェクト **OID** が見つかりませんでした。

## Sm\_PolicyApi\_CreateCertMap()

証明書マッピング オブジェクトを作成します。

### タイプ

証明書マップ関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_CreateCertMap (  
    void*                pSessionHandle,  
    Sm_PolicyApi_CertMap_t* pCertMap,  
    const bool           bUpdate  
);
```

---

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

---

---

パラメータ	I/O	説明
<i>pCertMap</i>	I	証明書マッピングに関する情報を含んだ、 <b>Sm_PolicyApi_CertMap_t</b> 構造へのポインタのアドレスです。
<i>Update</i>	I	既存のオブジェクトが見つかる場合にそれが更新されることを示すフラグです。

---

### 戻り値

- **Sm\_PolicyApi\_Success**。証明書マッピング オブジェクトが正常に作成されました。
- **Sm\_PolicyApi\_InvalidHandle**。このコールに先立つ有効な初期化はありませんでした。
- **Sm\_PolicyApi\_NoSession**。有効な管理者セッションがありません。
- **Sm\_PolicyApi\_NoPrivilege**。管理者には、証明書マッピング オブジェクトを作成する権限がありません。
- **Sm\_PolicyApi\_InvalidOid**。ユーザ ディレクトリ **OID** が見つからなかったか、または証明書マッピング オブジェクト **OID** が更新中に見つかりませんでした。
- **Sm\_PolicyApi\_Invalid**。ディレクトリ タイプが有効ではありません。

## Sm\_PolicyApi\_CreateODBCQueryScheme()

新規の ODBC クエリ方式を作成します。ODBC クエリ方式は ODBC ディレクトリの作成に使用されます。*pstructODBCQueryScheme* 構造にユーザディレクトリの属性が含まれています。

ユーザ ODBC クエリ方式オブジェクトが存在し、*bUpdate* フラグが「真」の場合、アイテムが更新されます。

### タイプ

ODBC クエリ方式関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_CreateODBCQueryScheme (
    void*                pSessionHandle,
    Sm_PolicyApi_ODBCQueryScheme_t* pstructODBCQueryScheme,
    const bool           bUpdate
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pstructODBCQueryScheme</i>	I	値の入力が完了している ODBC クエリ方式の構造体へのポインタです。
<i>bUpdate</i>	I	既存のオブジェクトが見つかる場合にそれが更新されることを示すフラグです。

### 戻り値

- Sm\_PolicyApi\_Success。 ODBC クエリ方式が正常に作成されました。
- Sm\_PolicyApi\_Failure。 ODBC クエリ方式が正常に作成されませんでした。
- Sm\_PolicyApi\_InvalidHandle。 このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。 有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。 管理者には、ODBC クエリ方式を作成する権限がありません。
- Sm\_PolicyApi\_InvalidOid。 ODBC クエリ方式 OID が見つかりませんでした。これは、この関数が呼び出される時 *bUpdate* の設定が「真」であり、Sm\_PolicyApi\_ODBCQueryScheme\_t の *pszOid* フィールドが更新中の ODBC クエリ方式の OID を保持する場合に生じます。
- Sm\_PolicyApi\_NotUnique。 同じ名前の ODBC クエリ方式が存在します。
- Sm\_PolicyApi\_MissingProperty。 必須フィールドの 1 つが設定されていません。

## Sm\_PolicyApi\_CreateUserDir()

グローバル範囲で新規 SiteMinder ユーザディレクトリ オブジェクトを作成します。 *pstructUserDir* 構造にユーザディレクトリの属性が含まれています。

ユーザディレクトリ オブジェクトが存在し、*bUpdate* フラグが「真」の場合、アイテムが更新されます。

### タイプ

ユーザディレクトリ関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_CreateUserDir (
    void*                pSessionHandle,
    Sm_PolicyApi_UserDir_t* pstructUserDir,
    const bool           bUpdate
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pstructUserDir</i>	I	値の入力が完了しているユーザディレクトリ構造へのポインタです。
<i>bUpdate</i>	I	既存のオブジェクトが見つかる場合にそれが更新されることを示すフラグです。

### SM\_PolicyAPI\_UserDir\_t フィールドの使用

ユーザディレクトリの異なるタイプに対して *Sm\_PolicyApi\_CreateUserDir()* が使用する *SM\_PolicyAPI\_UserDir\_t* フィールドを以下の表に示します。

フィールド	ユーザディレクトリのタイプ			
	ODBC	LDAP	Windows NT	カスタム

フィールド	ユーザ ディレクトリのタイプ			
	ODBC	LDAP	Windows NT	カスタム
<i>pszOid</i> 更新されているユーザ ディレクトリのオブジェクト識別子	X	X	X	X
<i>pszName</i> 必須フィールド	X	X	X	X
<i>pszDesc</i>	X	X	X	X
<i>pszNamespace</i> 必須フィールド	ODBC	LDAP	Windows NT	カスタム
<i>pszServer</i> 必須フィールド	ODBC データソース	IP アドレス	NT ドメイン名	共有ライブラリの名前
<i>pszODBCQuerySchemeOid</i> 必須フィールド	X			
<i>pszSearchRoot</i>		X		パラメータ文字列
<i>pszUserLookupStart</i>		X		
<i>pszUserLookupEnd</i>		X		
<i>bRequireCredentials</i>	X	X	X	X
<i>pszUsername</i> <i>bRequireCredentials</i> が「真」の場合、必須フィールド。	X	X	X	X
<i>pszPassword</i> <i>bRequireCredentials</i> が「真」の場合、必須フィールド。	X	X	X	X
<i>nSearchResults</i>		X		検索の上限
<i>nSearchScope</i>		X		

フィールド	ユーザ ディレクトリのタイプ			
	ODBC	LDAP	Windows NT	カスタム
<i>nSearchTimeout</i>		X		タイムアウトの上限
<i>bSecureConnection</i>		X		X
<i>pszDisabledAttr</i>	X	X		(異なる)
<i>pszUniversalIDAttr</i>	X	X	X	(異なる)
<i>pszAnonymousId</i>		X		(異なる)
<i>pszPasswordData</i>	X	X		(異なる)
<i>pszPasswordAttribute</i>	X	X		(異なる)
<i>pszEmailAddressAttr</i>				
<i>pszChallengeRespAttr</i>		X		(異なる)

注: カスタム ディレクトリで、「異なる」と表示されているフィールドは、作成されているユーザ ディレクトリ オブジェクトに適用できる場合と適用できない場合があります。

### 戻り値

- **Sm\_PolicyApi\_Success**。 ユーザ ディレクトリが正常に作成されました。
- **Sm\_PolicyApi\_Failure**。 ユーザ ディレクトリが正常に作成されませんでした。
- **Sm\_PolicyApi\_NoSession**。 有効な管理者セッションがありません。
- **Sm\_PolicyApi\_InvalidHandle**。 このコールに先立つ有効な初期化はありませんでした。
- **Sm\_PolicyApi\_NoPrivilege**。 管理者には、ユーザ ディレクトリを作成する権限がありません。
- **Sm\_PolicyApi\_InvalidOid** :
  - 指定された **OID** のユーザ ディレクトリ タイプは見つかりませんでした。
  - ユーザ ディレクトリ **OID** を見つけることができません。 これは、この関数が呼び出される時 **bUpdate** の設定が「真」であり、**pszOid** が更新中のユーザ ディレクトリの **OID** を保持する場合に生じます。
  - **ODBC** ユーザ ディレクトリを作成中です。 **OCBC** クエリ方式の **OID** が見つかりませんでした。
- **Sm\_PolicyApi\_NotUnique**。 同じ名前のユーザ ディレクトリが存在しません。
- **Sm\_PolicyApi\_MissingProperty**。 必須フィールドの 1 つが設定されていません。

## Sm\_PolicyApi\_DeleteAdmin()

*pszAdminOid* によって識別された管理者オブジェクトを削除します。

### タイプ

管理者関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_DeleteAdmin (
    void*          pSessionHandle,
    const char*    pszAdminOid
)
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAdminOid</i>	I	既存の管理者のオブジェクト識別子が含まれ、NULL で終わる文字列です。

### 戻り値

- Sm\_PolicyApi\_Success。削除の操作が成功しました。
- Sm\_PolicyApi\_Failure。削除の操作が成功しませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、管理者オブジェクトを削除する権限がありません。
- Sm\_PolicyApi\_InvalidOid。管理者オブジェクト識別子が見つかりませんでした。

## Sm\_PolicyApi\_DeleteAffiliate()

既存のアフィリエイトを削除します。

### タイプ

フェデレーション関数

### 構文

```
int SM_EXTERN Sm_PolicyApi_DeleteAffiliate (  
    void*          pSessionHandle,  
    const char*    pszAffiliateOid  
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAffiliateOid</i>	I	既存のアフィリエイトのオブジェクト識別子が含まれ、NULL で終わる文字列です。

### 注釈

この関数は、Artifact プロファイルまたは POST プロファイルに基づいたアフィリエイト オブジェクトを削除します (`Sm_PolicyApi_Affiliate_t` フィールドの `SAMLProfile` を参照してください)。POST プロファイルに基づいてアフィリエイト オブジェクトを削除するには、`SM_POLICY_API_VERSION_6_0_2` 以上の API バージョンが必要です。それ以前のバージョンが含まれているとき関数で POST プロファイルに基づいてアフィリエイト オブジェクトを指定した場合、リクエストは失敗します。

### 戻り値

- `Sm_PolicyApi_Success`。アフィリエイトが正常に削除されました。
- `Sm_PolicyApi_Failure`。アフィリエイトが正常に削除されませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoPrivilege`。管理者には、アフィリエイトを削除する権限がありません。
- `Sm_PolicyApi_InvalidOid`。アフィリエイト OID が見つかりませんでした。

## Sm\_PolicyApi\_DeleteAffiliateDomain()

既存のアフィリエイト ドメインを削除します。

### タイプ

フェデレーション関数

### 構文

```
int SM_EXTERN Sm_PolicyApi_DeleteAffiliateDomain (
    void*          pSessionHandle,
    const char*    pszAffiliateDomainOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAffiliateDomainOid</i>	I	既存のアフィリエイト ドメインのオブジェクト識別子が含まれ、NULL で終わる文字列です。

### 戻り値

- **Sm\_PolicyApi\_Success**。アフィリエイト ドメインが正常に削除されました。
- **Sm\_PolicyApi\_Failure**。アフィリエイト ドメインが正常に削除されませんでした。
- **Sm\_PolicyApi\_NoSession**。有効な管理者セッションがありません。
- **Sm\_PolicyApi\_InvalidHandle**。このコールに先立つ有効な初期化はありませんでした。
- **Sm\_PolicyApi\_NoPrivilege**。管理者には、アフィリエイト ドメインを削除する権限がありません。
- **Sm\_PolicyApi\_InvalidOid**。指定された OID でのアフィリエイトは更新中に見つかりませんでした。

## Sm\_PolicyApi\_DeleteAgent()

*pszAgentOid* によって識別されたエージェントを削除します。

### タイプ

エージェント関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_DeleteAgent (
    void*          pSessionHandle,
    const char*    pszAgentOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAgentOid</i>	I	既存のエージェントのオブジェクト識別子が含まれ、NULL で終わる文字列です。

### 戻り値

- Sm\_PolicyApi\_Success。 エージェントが正常に削除されました。
- Sm\_PolicyApi\_Failure。 エージェントが正常に削除されませんでした。
- Sm\_PolicyApi\_NoSession。 有効な管理者セッションがありません。
- Sm\_PolicyApi\_InvalidHandle。 このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoPrivilege。 管理者には、エージェントを削除する権限がありません。
- Sm\_PolicyApi\_InvalidOid。 指定された OID でのエージェントが見つかりませんでした。

## Sm\_PolicyApi\_DeleteAgentConfig()

エージェント設定オブジェクトを削除します。

### タイプ

エージェント設定関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_DeleteAgentConfig (
    void*          pSessionHandle,
    const char*    pszAgentConfigOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAgentConfigOid</i>	I	削除するエージェント設定オブジェクトの一意の識別子です。

### 戻り値

- Sm\_PolicyApi\_Success。削除の操作が成功しました。
- Sm\_PolicyApi\_Failure。一般的な障害です。
- Sm\_PolicyApi\_InvalidOid。エージェント設定オブジェクトに一意の ID が対応していません。
- Sm\_PolicyApi\_NoPrivilege。呼び出し元に、エージェント設定オブジェクトを削除する権限がありません。
- Sm\_PolicyApi\_InvalidHandle。セッションポインタが無効です。
- Sm\_PolicyApi\_NoSession。API ユーザが正しくログインしていません。

## Sm\_PolicyApi\_DeleteAuthAzMap()

認証および許可ディレクトリ マッピング オブジェクトを削除します。

### タイプ

認証/許可マップ関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_DeleteAuthAzMap (
    void*          pSessionHandle,
    const char*    pszAuthAzMapOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAuthAzMapOid</i>	I	ディレクトリ マッピング オブジェクトのオブジェクト識別子が含まれ、NULL で終わる文字列です。

### 戻り値

- Sm\_PolicyApi\_Success。削除の操作が成功しました。
- Sm\_PolicyApi\_Failure。削除の操作が成功しませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、ディレクトリ マッピング オブジェクトを削除する権限がありません。
- Sm\_PolicyApi\_InvalidOid。ディレクトリ マッピング オブジェクト OID が見つかりませんでした。

## Sm\_PolicyApi\_DeleteCertMap()

証明書マッピング オブジェクトを削除します。

### タイプ

証明書マップ関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_DeleteCertMap (
    void*          pSessionHandle,
    const char*    pszCertMapOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszCertMapOid</i>	I	証明書マッピング オブジェクトのオブジェクト識別子が含まれ、NULL で終わる文字列です。

### 戻り値

- Sm\_PolicyApi\_Success。削除の操作が成功しました。
- Sm\_PolicyApi\_Failure。削除の操作が成功しませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、証明書マッピング オブジェクトを削除する権限がありません。
- Sm\_PolicyApi\_InvalidOid。指定された OID での証明書マッピング オブジェクトは見つかりませんでした。

## Sm\_PolicyApi\_DeleteDomain()

ドメインの子（ルール、レスポンス、レーム、ポリシー）と共に、*pszDomainOid* によって識別されたドメインを削除します。

### タイプ

ドメイン関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_DeleteDomain (
    void*          pSessionHandle,
    const char*    pszDomainOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszDomainOid</i>	I	既存のドメインのオブジェクト識別子が含まれ、NULL で終わる文字列です。

### 戻り値

- **Sm\_PolicyApi\_Success**。ドメインが正常に削除されました。
- **Sm\_PolicyApi\_Failure**。ドメインが正常に削除されませんでした。
- **Sm\_PolicyApi\_InvalidHandle**。このコールに先立つ有効な初期化はありませんでした。
- **Sm\_PolicyApi\_NoSession**。有効な管理者セッションがありません。
- **Sm\_PolicyApi\_NoPrivilege**。管理者には、ドメインを削除する権限がありません。
- **Sm\_PolicyApi\_InvalidOid**。ドメイン OID が見つかりませんでした。

## Sm\_PolicyApi\_DeleteHostConfig()

ポリシーストアからホスト設定オブジェクトを削除します。

### タイプ

エージェント設定関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_DeleteHostConfig (
    void*                pSessionHandle,
    const char*          pszHostConfigOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszHostConfigOid</i>	I	削除するホスト設定オブジェクトの一意的識別子です。

### 戻り値

- Sm\_PolicyApi\_Success。削除の操作が成功しました。
- Sm\_PolicyApi\_Failure。一般的な障害です。
- Sm\_PolicyApi\_InvalidOid。一意の ID がホスト設定オブジェクトに対応しません。
- Sm\_PolicyApi\_NoPrivilege。呼び出し元に、ホスト設定オブジェクトを削除する権限がありません。
- Sm\_PolicyApi\_InvalidHandle。セッションポインタが無効です。
- Sm\_PolicyApi\_NoSession。API ユーザが正しくログインしていません。

## Sm\_PolicyApi\_DeleteGroup()

*pszGroupOid* によって識別されたグループ オブジェクトを削除します。

*pszDomainOid* パラメータはルール グループまたはレスポンス グループに必須です。エージェント グループはドメインベースのオブジェクトではないので、ドメイン OID を必要としません。

### タイプ

グループ関数、グローバル範囲（エージェント）またはドメイン範囲（レスポンス、ルール）。

### 構文

```
int SM_EXTERN Sm_PolicyApi_DeleteGroup (
    void*                pSessionHandle,
    Sm_PolicyApi_Groups_t dwGroup,
    const char*          pszGroupOid,
    const char*          pszDomainOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>dwGroup</i>	I	削除するグループのタイプを示します。
<i>pszGroupOid</i>	I	<i>dwGroup</i> によって示されるグループとタイプのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszDomainOid</i>	I	既存のドメインのオブジェクト識別子が含まれ、NULL で終わる文字列です。ルールまたはレスポンス グループの必須パラメータです。

### 戻り値

- Sm\_PolicyApi\_Success。削除が成功しました。
- Sm\_PolicyApi\_Failure。削除が成功しませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、ルールグループ、レスポンスグループまたはエージェントグループを削除する権限がありません。
- Sm\_PolicyApi\_InvalidOid :
  - グループ OID が見つかりませんでした。
  - ドメイン OID が見つかりませんでした（ドメインベースのグループについて）。
- Sm\_PolicyApi\_BadGroup。dwGroup パラメータはルール、レスポンスまたはエージェントタイプではありません。

## Sm\_PolicyApi\_DeleteODBCQueryScheme()

pszODBCQuerySchemeOid によって識別された ODBC クエリ方式を削除します。

### タイプ

ODBC クエリ方式関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_DeleteODBCQueryScheme (
    void*      pSessionHandle,
    const char* pszODBCQuerySchemeOid
);
```

パラメータ	I/O	説明
pSessionHandle	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

パラメータ	I/O	説明
<i>pszODBCQuerySchemeOid</i>	I	既存の ODBC クエリ方式のオブジェクト識別子が含まれ、NULL で終わる文字列です。

### 戻り値

- `Sm_PolicyApi_Success`。削除の操作が成功しました。
- `Sm_PolicyApi_Failure`。削除の操作が成功しませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、ODBC クエリ方式を削除する権限がありません。
- `Sm_PolicyApi_InvalidOid`。ODBC クエリ方式 OID が見つかりませんでした。

## Sm\_PolicyApi\_DeletePasswordPolicy()

パスワードポリシーを削除します。

### タイプ

パスワードポリシー関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_DeletePasswordPolicy (
    void*      pSessionHandle,
    const char* pszPasswordPolicyOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

パラメータ	I/O	説明
<i>pszPasswordPolicyOid</i>	I	パスワード ポリシーのオブジェクト識別子が含まれ、NULL で終わる文字列です。

### 戻り値

- `Sm_PolicyApi_Success`。削除の操作が成功しました。
- `Sm_PolicyApi_Failure`。削除の操作が成功しませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、パスワード ポリシーを削除する権限がありません。
- `Sm_PolicyApi_InvalidOid`。パスワード ポリシー OID が見つかりませんでした。

## Sm\_PolicyApi\_DeletePolicy()

*pszPolicyOid* によって識別されたポリシーを削除します。

### タイプ

ポリシー関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_DeletePolicy (
    void*          pSessionHandle,
    const char*    pszPolicyOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszPolicyOid</i>	I	既存のポリシーのオブジェクト識別子が含まれ、NULL で終わる文字列です。

### 戻り値

- `Sm_PolicyApi_Success`。ポリシーが正常に削除されました。
- `Sm_PolicyApi_Failure`。ポリシーが正常に削除されませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。- 有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、ポリシーを削除する権限がありません。
- `Sm_PolicyApi_InvalidOid`。ポリシー OID が見つかりませんでした。

## Sm\_PolicyApi\_DeleteRealm()

`pszRealmOid` によって識別されたレルムを削除します。

注: ルールとリンクしている場合にはレルムを削除できません。

### タイプ

レルム関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_DeleteRealm (
    void*          pSessionHandle,
    const char*    pszRealmOid
);
```

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<code>pszRealmOid</code>	I	既存のレルムのオブジェクト識別子が含まれ、NULL で終わる文字列です。

**戻り値**

- Sm\_PolicyApi\_Success。 レルムが正常に削除されました。
- Sm\_PolicyApi\_Failure。 レルムが正常に削除されませんでした。
- Sm\_PolicyApi\_NoSession。 有効な管理者セッションがありません。
- Sm\_PolicyApi\_InvalidHandle。 このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoPrivilege。 管理者には、レルムを削除する権限がありません。
- Sm\_PolicyApi\_InvalidOid。 レルム OID が見つかりませんでした。

**Sm\_PolicyApi\_DeleteRegistrationScheme()**

登録方式を削除します。

**タイプ**

登録方式関数、グローバル範囲。

**構文**

```
int SM_EXTERN Sm_PolicyApi_DeleteRegistrationScheme (
    void*          pSessionHandle,
    const char*    pszRegistrationSchemeOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszRegistrationOid</i>	I	登録方式のオブジェクト識別子が含まれ、NULL で終わる文字列です。

### 戻り値

- Sm\_PolicyApi\_Success。登録方式が正常に削除されました。
- Sm\_PolicyApi\_Failure。登録方式が正常に削除されませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、登録方式を削除する権限がありません。
- Sm\_PolicyApi\_InvalidOid。登録方式 OID が見つかりませんでした。

## Sm\_PolicyApi\_DeleteResponse()

*pszResponseOid* および関連するレスポンス属性によって識別されたレスポンスを削除します。

### タイプ

レスポンス関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_DeleteResponse (
    void*          pSessionHandle,
    const char*    pszResponseOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszResponseOid</i>	I	既存のレスポンスのオブジェクト識別子が含まれ、NULLで終わる文字列です。

### 戻り値

- `Sm_PolicyApi_Success`。 レスポンスが正常に削除されました。
- `Sm_PolicyApi_Failure`。 レスポンスが正常に削除されませんでした。
- `Sm_PolicyApi_NoSession`。 有効な管理者セッションがありません。
- `Sm_PolicyApi_InvalidHandle`。 このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoPrivilege`。 管理者には、レスポンスを削除する権限がありません。
- `Sm_PolicyApi_InvalidOid`。 レスポンス `OID` が見つかりませんでした。

## Sm\_PolicyApi\_DeleteRule()

`pszRuleOid` によって識別されたルールを削除します。

### タイプ

ルール関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_DeleteRule (
    void*          pSessionHandle,
    const char*    pszRuleOid
);
```

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<code>pszRuleOid</code>	I	既存のルールのオブジェクト識別子が含まれ、 <code>NULL</code> で終わる文字列です。

### 戻り値

- Sm\_PolicyApi\_Success。 ルールが正常に削除されました。
- Sm\_PolicyApi\_Failure。 ルールが正常に削除されませんでした。
- Sm\_PolicyApi\_InvalidHandle。 このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。 有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。 管理者には、ルールを削除する権限がありません。
- Sm\_PolicyApi\_InvalidOid。 ルール OID が見つかりませんでした。

## Sm\_PolicyApi\_DeleteSAMLAffiliation()

*pszAffiliationOid* によって識別された SAML アフィリエーションを削除します。

### タイプ

SAML 2.0 設定関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_DeleteSAMLAffiliation
(
    void*          pSessionHandle,
    const char*    pszAffiliationOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAffiliationOid</i>	I	既存の SAML アフィリエーションのオブジェクト識別子が含まれ、NULL で終わる文字列です。

**戻り値**

- `Sm_PolicyApi_Success`。SAML アフィリエーションは正常に削除されました。
- `Sm_PolicyApi_Failure`。SAML アフィリエーションは正常に削除されませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、SAML アフィリエーションを削除する権限がありません。
- `Sm_PolicyApi_InvalidOID`。SAML アフィリエーションのオブジェクト識別子が見つかりませんでした。

**Sm\_PolicyApi\_DeleteSAMLServiceProvider()**

`pszProviderOid` によって識別されたサービス プロバイダを削除します。

**タイプ**

SAML 2.0 設定関数、ドメイン範囲。

**構文**

```
int SM_EXTERN Sm_PolicyApi_DeleteSAMLServiceProvider(
void*          pSessionHandle,
const char*    pszProviderOid
);
```

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<code>pszProviderOid</code>	I	既存のサービス プロバイダのオブジェクト識別子が含まれ、NULL で終わる文字列です。

### 戻り値

- `Sm_PolicyApi_Success`。 サービスプロバイダが正常に削除されました。
- `Sm_PolicyApi_Failure`。 サービスプロバイダが正常に削除されませんでした。
- `Sm_PolicyApi_InvalidHandle`。 このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。 有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。 管理者には、サービスプロバイダを削除する権限がありません。
- `Sm_PolicyApi_InvalidOID`。 サービスプロバイダのオブジェクト識別子が見つかりませんでした。

## Sm\_PolicyApi\_DeleteScheme()

`pszSchemeOid` によって識別された認証方式を削除します。

### タイプ

認証方式関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_DeleteScheme (  
    void*          pSessionHandle,  
    const char*   pszSchemeOid  
);
```

---

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<code>pszSchemeOid</code>	I	既存の認証方式のオブジェクト識別子が含まれ、NULL で終わる文字列です。

---

### 戻り値

- `Sm_PolicyApi_Success`。方式が正常に削除されました。
- `Sm_PolicyApi_Failure`。方式が正常に削除されませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、方式を削除する権限がありません。
- `Sm_PolicyApi_InvalidOid`。方式のオブジェクト識別子が見つかりませんでした。

## Sm\_PolicyApi\_DeleteTrustedHost()

ポリシーストアから既存のトラステッドホストオブジェクトを削除します。

### タイプ

エージェント設定関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_DeleteTrustedHost (
    void*          pSessionHandle,
    const char*    pszTrustedHostOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszTrustedHostOid</i>	I	削除するトラステッドホストオブジェクトの一意の識別子です。

### 戻り値

- Sm\_PolicyApi\_Success。削除の操作が成功しました。
- Sm\_PolicyApi\_Failure。一般的な障害です。
- Sm\_PolicyApi\_InvalidOid。トラステッドホストオブジェクトに一意の ID が対応しません。
- Sm\_PolicyApi\_NoPrivilege。呼び出し元に、トラステッドホストオブジェクトを削除する権限がありません。
- Sm\_PolicyApi\_InvalidHandle。セッションポインタが無効です。
- Sm\_PolicyApi\_NoSession。API ユーザが正しくログインしていません。

## Sm\_PolicyApi\_DeleteUserDir()

*pszUserDirOid* によって識別されたユーザディレクトリを削除します。

### タイプ

ユーザディレクトリ関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_DeleteUserDir (  
    void*          pSessionHandle,  
    const char*   pszUserDirOid  
);
```

---

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszUserDirOid</i>	I	既存のユーザディレクトリのオブジェクト識別子が含まれ、NULL で終わる文字列です。

---

### 戻り値

- `Sm_PolicyApi_Success`。 ユーザディレクトリが正常に削除されました。
- `Sm_PolicyApi_Failure`。 ユーザディレクトリが正常に削除されませんでした。
- `Sm_PolicyApi_NoSession`。 有効な管理者セッションがありません。
- `Sm_PolicyApi_InvalidHandle`。 このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoPrivilege`。 管理者には、ユーザディレクトリを削除する権限がありません。
- `Sm_PolicyApi_InvalidOid`。 ユーザディレクトリ `OID` が見つかりませんでした。

## Sm\_PolicyApi\_DeleteVariable()

変数オブジェクトを削除します。

### タイプ

変数関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_DeleteVariable (
    void*          pSessionHandle,
    const char*    pszVariableOid
);
```

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<code>pszVariableOid</code>	I	既存の変数のオブジェクト識別子が含まれ、NULL で終わる文字列です。

### 戻り値

- `Sm_PolicyApi_Success`。削除の操作が成功しました。
- `Sm_PolicyApi_Failure`。削除の操作が成功しませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、変数を削除する権限がありません。
- `Sm_PolicyApi_InvalidOid`。変数のオブジェクト識別子が見つかりませんでした。
- `Sm_PolicyApi_FeatureNotSupported`。この関数を呼び出したクライアントが、`SM_POLICY_API_VERSION_6_0` より以前のバージョンで API を初期化しました。

## Sm\_PolicyApi\_DeleteWSFEDResourcePartner()

既存の WS-Federation リソース パートナー オブジェクトを削除します。

### 構文

```
int SM_EXTERN Sm_PolicyApi_DeleteWSFEDResourcePartner (  
    void* pSessionHandle,  
    const char * pszProviderOid,  
);
```

### パラメータ

#### pSessionHandle

[in] 内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

#### pszProviderOid

[in] 既存の WS フェデレーション リソース パートナーのオブジェクト識別子が含まれ、NULL で終わる文字列です。

### 戻り値

- Sm\_PolicyApi\_Success。 リソース パートナーが正常に取得されました。
- Sm\_PolicyApi\_Failure。 リソース パートナーが正常に取得されませんでした。
- Sm\_PolicyApi\_InvalidHandle。 このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。 有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。 管理者には、アフィリエイトを取得する権限がありません。
- Sm\_PolicyApi\_InvalidOID。 リソース パートナー OID が見つかりませんでした。

## Sm\_PolicyApi\_DisableUser()

理由の `Sm_Api_Disabled_AdminDisabled` でユーザを無効にします。それによって他の同時無効化の理由は変更されません。

この関数を機能させるには、無効なユーザを追跡するための属性をユーザディレクトリで設定する必要があります (`Sm_PolicyApi_UserDir_t` の `pszDisabledAttr` フィールド)。また、ポリシー サーバ UI を使用して、属性を設定することもできます。

### タイプ

ユーザとユーザ状態関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_DisableUser (
    void*      pSessionHandle,
    const char* pszUserDirOid,
    const char* pszUserDN,
    char**     pszErrMsg
);
```

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<code>pszUserDirOid</code>	I	ユーザが含まれているユーザディレクトリのオブジェクト識別子を含み、NULL で終わる文字列です。
<code>pszUserDN</code>	I	無効にするユーザの識別名です。
<code>pszErrMsg</code>	O	ユーザが正常に無効にならない場合にエラーメッセージが含まれる文字列です。 <code>Sm_PolicyApi_FreeString()</code> を呼び出して、この変数に割り当てられたメモリを解放します。

### 戻り値

- `Sm_PolicyApi_Success`。ユーザが正常に無効にされました。
- `Sm_PolicyApi_Failure`。ユーザは正常に無効にならなかったか、または、エラーメッセージ文字列にメモリが割り当てられませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、ユーザを無効にする権限がありません。
- `Sm_PolicyApi_InvalidOid`。ユーザディレクトリ OID が見つかりませんでした。

## Sm\_PolicyApi\_EnableUser()

無効なビットをすべてクリアすることによりユーザを有効にします。それによって修飾子ビットの `Sm_Api_Disabled_PWMustChange` はクリアされません。

### タイプ

ユーザとユーザ状態関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_EnableUser (
    void*          pSessionHandle,
    const char*    pszUserDirOid,
    const char*    pszUserDN,
    char**         pszErrMsg
);
```

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<code>pszUserDirOid</code>	I	ユーザが含まれているユーザディレクトリのオブジェクト識別子を含み、NULL で終わる文字列です。

パラメータ	I/O	説明
<i>pszUserDN</i>	I	有効にするユーザの識別名です。
<i>pszErrMsg</i>	O	ユーザが正常に有効にならない場合にエラーメッセージが含まれる文字列です。Sm_PolicyApi_FreeString() を呼び出して、この変数に割り当てられたメモリを解放します。

### 戻り値

- Sm\_PolicyApi\_Success。ユーザが正常に有効にされました。
- Sm\_PolicyApi\_Failure。ユーザは正常に有効にならなかったか、または、エラーメッセージ文字列にメモリが割り当てられませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、ユーザを有効にする権限がありません。
- Sm\_PolicyApi\_InvalidOid。ユーザディレクトリ OID が見つかりませんでした。

## Sm\_PolicyApi\_FlushRealm()

リソース キャッシュから *pszRealmOid* によって識別された、指定したレルムをクリアします。

### タイプ

ユーティリティ関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_FlushRealm (
    void*          pSessionHandle,
    const char*    pszRealmOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszRealmOid</i>	I	レルムのオブジェクト識別子が含まれ、NULL で終わる文字列です。

### 戻り値

- *Sm\_PolicyApi\_Success*。クリアの操作が成功しました。
- *Sm\_PolicyApi\_Failure*。クリアの操作が成功しませんでした。
- *Sm\_PolicyApi\_NoSession*。有効な管理者セッションがありません。
- *Sm\_PolicyApi\_InvalidHandle*。このコールに先立つ有効な初期化はありませんでした。
- *Sm\_PolicyApi\_NoPrivilege*。管理者には、レルムをクリアする権限がありません。
- *Sm\_PolicyApi\_InvalidOid*。レルム OID が見つかりませんでした。

## Sm\_PolicyApi\_FlushUser()

ユーザ キャッシュからユーザをクリアします。

### タイプ

ユーティリティ関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_FlushUser (
    void*      pSessionHandle,
    const char* pszUserDirOid,
    const char* pszUserDN,
    char**     pszErrMsg
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszUserDirOid</i>	I	既存のユーザ ディレクトリのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszUserDN</i>	I	ユーザ キャッシュからクリアされる既存のユーザ ディレクトリのユーザの名前を含み、NULL で終わる文字列です。
<i>pszErrMsg</i>	O	ポリシー管理 API によって返されたエラー メッセージです。Sm_PolicyApi_FreeString() を呼び出して、この変数に割り当てられたメモリを解放します。

### 戻り値

- `Sm_PolicyApi_Success`。クリアの操作が成功しました。
- `Sm_PolicyApi_Failure`。クリアの操作が成功しなかったか、または、エラーメッセージ文字列にメモリが割り当てられませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、ユーザをクリアする権限がありません。
- `Sm_PolicyApi_InvalidOid`。ユーザディレクトリ OID が見つかりませんでした。

## Sm\_PolicyApi\_FreeMemory()

注: この関数は `SM_POLICY_API_VERSION_6_0` 以上のクライアントに対しては廃止予定です。代わりに、`Sm_PolicyApi_FreeMemoryEx()` を使用します。

`Sm_PolicyApi_FreeMemory()` を呼び出して、ポリシー管理 API によって割り当てられたメモリを解放します。これには、`Sm_PolicyApi_HostConfig_t` 構造の一部として割り当てられた `Sm_PolicyApi_Server_t` 構造が含まれます。

### タイプ

ユーティリティ関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_FreeMemory (void* pMem);
```

パラメータ	I/O	説明
<code>pMem</code>	I	ポリシー管理 API によって割り当てられたメモリへの <code>void</code> ポインタです。

### 戻り値

`Sm_PolicyApi_Success` または `Sm_PolicyApi_Failure`。

## Sm\_PolicyApi\_FreeMemoryEx()

Sm\_PolicyApi\_FreeMemoryEx() を呼び出して、ポリシー管理 API によって割り当てられたメモリを解放します。これには、Sm\_PolicyApi\_HostConfig\_t 構造の一部として割り当てられた Sm\_PolicyApi\_Server\_t 構造が含まれます。

注: バージョン SM\_POLICY\_API\_VERSION\_6\_0 以上のクライアントは、Sm\_PolicyApi\_FreeMemory() の代わりにこの関数を使用する必要があります。

### タイプ

ユーティリティ関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_FreeMemoryEx (
    void* pInitHandle
    void* pMem
);
```

パラメータ	I/O	説明
<i>pInitHandle</i>	I	ポリシー管理 API によって割り当てられたメモリを指す void ポインタです。
<i>pMem</i>	I	ポリシー管理 API によって割り当てられたメモリを指す void ポインタです。

### 戻り値

- Sm\_PolicyApi\_Success。操作は成功しました。
- Sm\_PolicyApi\_Failure。操作が失敗しました。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。

## Sm\_PolicyApi\_FreeString()

ポリシー管理 API によって割り当てられた単一の文字列を解放します。

### タイプ

ユーティリティ関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_FreeString(  
    char* pszString  
);
```

パラメータ	I/O	説明
<i>pszString</i>	I	NULL で終わる文字列へのポインタです。

### 戻り値

この関数は常に `Sm_PolicyApi_Success` を返します。

## Sm\_PolicyApi\_FreeStringArray()

ポリシー管理 API によって割り当てられた文字列の配列を解放します。

### タイプ

ユーティリティ関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_FreeStringArray (  
    char* pszStringArray[]  
);
```

---

パラメータ	I/O	説明
<i>pszStringArray</i>	I	NULL で終わる文字列を指すポインタの配列へのポインタです。

---

### 戻り値

この関数は常に `Sm_PolicyApi_Success` を返します。

## Sm\_PolicyApi\_GetAdmin()

*szAdminOid* によって識別される管理者オブジェクトの内容を取得します。

この関数の結果は、*ppstructAdmin* によって参照される構造で返されます。  
*Sm\_PolicyApi\_FreeMemoryEx()* を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

管理者関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetAdmin (
    void*                pSessionHandle,
    const char*          pszAdminOid,
    Sm_PolicyApi_Admin_t** ppstructAdmin
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAdminOid</i>	I	既存の管理者のオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>ppstructAdmin</i>	O	管理者構造へのポインタのアドレスです。

### 戻り値

- `Sm_PolicyApi_Success`。取得の操作が成功しました。
- `Sm_PolicyApi_Failure`。 `Sm_PolicyApi_Admin_t` にメモリを割り当てることができませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、管理者オブジェクトを取得する権限がありません。
- `Sm_PolicyApi_InvalidOid`。管理者オブジェクト ID が見つかりませんでした。

## Sm\_PolicyApi\_GetAdminByName()

`szAdminName` によって識別される管理者オブジェクトの内容を取得します。

この関数の結果は、`ppstructAdmin` によって参照される構造で返されます。`Sm_PolicyApi_FreeMemoryEx()` を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

管理者関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetAdminByName (
    void*                pSessionHandle,
    const char*          pszAdminName,
    Sm_PolicyApi_Admin_t** ppstructAdmin
);
```

---

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

---

---

パラメータ	I/O	説明
<i>pszAdminName</i>	I	既存の管理者の名前が含まれ、NULL で終わる文字列です。
<i>ppstructAdmin</i>	O	管理者構造へのポインタのアドレスです。

---

### 戻り値

- `Sm_PolicyApi_Success`。取得の操作が成功しました。
- `Sm_PolicyApi_Failure`。 `Sm_PolicyApi_Admin_t` にメモリを割り当てるのができませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、管理者オブジェクトを取得する権限がありません。
- `Sm_PolicyApi_NotFound`。管理者名が見つかりませんでした。

## Sm\_PolicyApi\_GetAffiliate()

アフィリエイトを取得します。

### タイプ

フェデレーション関数

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetAffiliate (
    void*                pSessionHandle,
    const char*          pszAffiliateOid,
    Sm_PolicyApi_Affiliate_t** ppstructAffiliate
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAffiliateOid</i>	I	既存のアフィリエイトのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>ppstruct</i> アフィリエイト	O	アフィリエイト構造体へのポインタのアドレスです。

## 注釈

この関数は、Artifact プロファイルまたは POST プロファイルに基づいたアフィリエイト オブジェクトを取得します (Sm\_PolicyApi\_Affiliate\_t フィールドの SAMLProfile を参照してください)。POST プロファイルに基づいてアフィリエイト オブジェクトを取得するには、

SM\_POLICY\_API\_VERSION\_6\_0\_2 以上の API バージョンが必要です。それ以前のバージョンが含まれているとき関数で POST プロファイルに基づいてアフィリエイト オブジェクトを指定した場合、リクエストは失敗します。

## 戻り値

- Sm\_PolicyApi\_Success。アフィリエイトが正常に取得されました。
- Sm\_PolicyApi\_Failure。アフィリエイトが正常に取得されませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、アフィリエイトを取得する権限がありません。
- Sm\_PolicyApi\_InvalidOID。アフィリエイト OID が見つかりませんでした。

## Sm\_PolicyApi\_GetAffiliateByName()

アフィリエイトを取得します。

### タイプ

フェデレーション関数

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetAffiliateByName (
    void*                pSessionHandle,
    const char*          pszAffiliateOid,
    const char*          pszAffiliateName,
    Sm_PolicyApi_Affiliate_t** ppstructAffiliate
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAffiliateOid</i>	I	既存のアフィリエイトのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszAffiliateName</i>	I	既存のアフィリエイトの名前が含まれ、NULL で終わる文字列です。
<i>ppstruct</i> アフィリエイト	O	アフィリエイト構造体へのポインタのアドレスです。

## 注釈

この関数は、Artifact プロファイルまたは POST プロファイルに基づいたアフィリエイトオブジェクトを取得します (Sm\_PolicyApi\_Affiliate\_t フィールドの SAMLProfile を参照してください)。POST プロファイルに基づいてアフィリエイトオブジェクトを取得するには、

SM\_POLICY\_API\_VERSION\_6\_0\_2 以上の API バージョンが必要です。それ以前のバージョンが含まれているとき関数で POST プロファイルに基づいてアフィリエイトオブジェクトを指定した場合、リクエストは失敗します。

## 戻り値

- Sm\_PolicyApi\_Success。アフィリエイトが正常に取得されました。
- Sm\_PolicyApi\_Failure。アフィリエイトが正常に取得されませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、アフィリエイトを取得する権限がありません。
- Sm\_PolicyApi\_InvalidOID。アフィリエイト OID が見つかりませんでした。
- Sm\_PolicyApi\_NotFound。アフィリエイトの名前が見つかりませんでした。

## Sm\_PolicyApi\_GetAffiliateDomain()

1 個のアフィリエイト ドメインを取得します。

### タイプ

フェデレーション関数

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetAffiliateDomain (
    void*                pSessionHandle,
    const char*          pszAffiliateDomainOid,
    Sm_PolicyApi_AffiliateDomain_t** ppstructAffiliateDomain
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAffiliateDomainOid</i>	I	既存のアフィリエイト ドメインのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>ppstructAffiliateDomain</i>	O	アフィリエイト ドメイン構造体へのポインタのアドレスです。

### 戻り値

- `Sm_PolicyApi_Success`。アフィリエイト ドメインが正常に取得されました。
- `Sm_PolicyApi_Failure`。アフィリエイト ドメインが正常に取得されませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、アフィリエイト ドメインを取得する権限がありません。
- `Sm_PolicyApi_DomainNotFound`。アフィリエイト ドメイン OID が見つかりませんでした。
- `Sm_PolicyApi_DomainNotAffiliate`。ドメインはアフィリエイト ドメインではありません。

## Sm\_PolicyApi\_GetAffiliateDomainByName()

1 個のアフィリエイト ドメインを取得します。

### タイプ

フェデレーション関数

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetAffiliateDomainByName (
    void*                pSessionHandle,
    const char*          pszAffiliateDomainName,
    Sm_PolicyApi_AffiliateDomain_t** ppstructAffiliateDomain
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAffiliateDomainName</i>	I	既存のアフィリエイト ドメインの名前が含まれ、NULL で終わる文字列です。

パラメータ	I/O	説明
<code>ppstruct AffiliateDomain</code>	O	アフィリエイト ドメイン構造体へのポインタのアドレスです。

#### 戻り値

- `Sm_PolicyApi_Success`。アフィリエイト ドメインが正常に取得されました。
- `Sm_PolicyApi_Failure`。アフィリエイト ドメインが正常に取得されませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、アフィリエイト ドメインを取得する権限がありません。
- `Sm_PolicyApi_NotFound`。アフィリエイト ドメイン名が見つかりませんでした。
- `Sm_PolicyApi_DomainNotAffiliate`。ドメインはアフィリエイト ドメインではありません。

## Sm\_PolicyApi\_GetAffiliateDomainObjects()

アフィリエイト ドメイン オブジェクトを取得します。

### タイプ

フェデレーション関数

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetAffiliateDomainObjects (
    void*                pSessionHandle,
    const char*          pszAffiliateDomainOid,
    const Sm_PolicyApi_Objects_t nObjectId,
    Sm_PolicyApi_Oid_t** ppstructObject
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAffiliateDomainOid</i>	I	既存のアフィリエイト ドメインのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>nObjectId</i>	I	取得するドメイン オブジェクトのタイプです。アフィリエイト、管理者、およびユーザ ディレクトリ オブジェクトに対してのみ有効なタイプです。
<i>ppstructObject</i>	O	Sm_PolicyApi_Oid_t 構造体へのポインタのアドレス

### 戻り値

- Sm\_PolicyApi\_Success。 取得の操作が成功しました。
- Sm\_PolicyApi\_Failure。 取得の操作が成功しませんでした。
- Sm\_PolicyApi\_InvalidHandle。 このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。 有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。 管理者には、アフィリエイト ドメイン オブジェクトを取得する権限がありません。
- Sm\_PolicyApi\_InvalidOID。 アフィリエイト ドメイン OID が見つかりませんでした。

## Sm\_PolicyApi\_GetAffiliateDomainUserDirSearchOrder()

アフィリエイト ドメインのユーザディレクトリ検索順序を取得します。

### タイプ

フェデレーション関数

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetAffiliateDomainUserDirSearchOrder (
    void*          pSessionHandle,
    const char*    pszAffiliateDomainOid,
    char**         pszArray[]
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAffiliateDomainOid</i>	I	既存のアフィリエイト ドメインのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszArray</i>	O	要求されたオブジェクトのユーザディレクトリ OID の返された配列です。

### 戻り値

- `Sm_PolicyApi_Success`。関数は正常にユーザ ディレクトリ検索順序を返しました。
- `Sm_PolicyApi_Failure`。関数は正常にユーザ ディレクトリ検索順序を返しませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、アフィリエイト ドメインのユーザ ディレクトリ検索順序を取得する権限がありません。
- `Sm_PolicyApi_InvalidOID`。アフィリエイト ドメイン `OID` が見つかりませんでした。

## Sm\_PolicyApi\_GetAffiliatedSAMLAuthSchemes()

指定された SAML アフィリエイトと関連付けられた SAML 認証方式をすべて取得します。

### タイプ

SAML 2.0 設定関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetAffiliatedSAMLAuthSchemes
(
    void*                pSessionHandle,
    const char*          pszAffiliationOid,
    Sm_PolicyApi_Scheme_t** ppstructSchemes
);
```

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<code>pszAffiliationOid</code>	I	既存の SAML アフィリエイトのオブジェクト識別子が含まれ、NULL で終わる文字列です。

パラメータ	I/O	説明
<i>ppstructSchemes</i>	O	SAML 認証方式構造へのポインタのアドレスです。

### 戻り値

- `Sm_PolicyApi_Success`。 SAML 認証方式が正常に取得されました。
- `Sm_PolicyApi_Failure`。 SAML 認証方式が正常に取得されませんでした。
- `Sm_PolicyApi_InvalidHandle`。 このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。 有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。 管理者には、SAML 認証方式を取得する権限がありません。
- `Sm_PolicyApi_InvalidOID`。 SAML アフィリエーションオブジェクト識別子が見つかりませんでした。

## Sm\_PolicyApi\_GetAffiliatedSAMLServiceProviders()

指定された SAML アフィリエーションと関連付けられているサービスプロバイダをすべて取得します。

### タイプ

SAML 2.0 設定関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetAffiliatedSAMLServiceProviders
(
    void*                pSessionHandle,
    const char*          pszAffiliationOid,
    Sm_PolicyApi_SAMLSP_t** ppstructSAMLSPs
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

---

パラメータ	I/O	説明
<i>pszAffiliationOid</i>	I	既存の SAML アフィリエーションのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>ppstructSAMLSPs</i>	O	サービス プロバイダ構造へのポインタのアドレスです。

---

### 戻り値

- **Sm\_PolicyApi\_Success**。 サービス プロバイダが正常に取得されました。
- **Sm\_PolicyApi\_Failure**。 サービス プロバイダが正常に取得されませんでした。
- **Sm\_PolicyApi\_InvalidHandle**。 このコールに先立つ有効な初期化はありませんでした。
- **Sm\_PolicyApi\_NoSession**。 有効な管理者セッションがありません。
- **Sm\_PolicyApi\_NoPrivilege**。 管理者には、サービス プロバイダを取得する権限がありません。
- **Sm\_PolicyApi\_InvalidOID**。 SAML アフィリエーション オブジェクト識別子が見つかりませんでした。

## Sm\_PolicyApi\_GetAffiliateUsers()

アフィリエイトのユーザディレクトリ エントリを取得します。

### タイプ

フェデレーション関数

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetAffiliateUsers (
    void*                pSessionHandle,
    const char*          pszAffiliateOid,
    const char*          pszUserDirOid,
    Sm_PolicyApi_User_t **ppStructUsers
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAffiliateOid</i>	I	既存のアフィリエイトのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszUserDirOid</i>	I	既存のユーザディレクトリのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>ppStructUsers</i>	O	ユーザ構造のリンク リストへのポインタのアドレスです。

### 戻り値

- Sm\_PolicyApi\_Success。 ユーザが正常に取得されました。
- Sm\_PolicyApi\_Failure。 ユーザが正常に取得されませんでした。
- Sm\_PolicyApi\_InvalidHandle。 このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。 有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。 管理者には、ユーザを取得する権限がありません。
- Sm\_PolicyApi\_InvalidOID。 アフィリエイト OID またはユーザ ディレクトリ OID が見つかりませんでした。

## Sm\_PolicyApi\_GetAgent()

*pszAgentOid* によって識別されたエージェントの内容を取得します。 この関数の結果は、*ppstructAgent* によって参照される構造で返されます。

Sm\_PolicyApi\_FreeMemoryEx() を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

エージェント関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetAgent (
    void*                pSessionHandle,
    const char*          pszAgentOid,
    Sm_PolicyApi_Agent_t** ppstructAgent
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。 この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAgentOid</i>	I	既存のエージェントのオブジェクト識別子が含まれ、NULL で終わる文字列です。

パラメータ	I/O	説明
<i>ppstructAgent</i>	O	エージェント構造へのポインタのアドレスです。

---

#### 戻り値

- `Sm_PolicyApi_Success`。取得の操作が成功しました。
- `Sm_PolicyApi_Failure`。 `Sm_PolicyApi_Agent_t` にメモリを割り当てることができませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、エージェントを取得する権限がありません。
- `Sm_PolicyApi_InvalidOid`。指定された `OID` でのエージェントが見つかりません。

## Sm\_PolicyApi\_GetAgentByName()

*szAgentName* によって識別されたエージェントの内容を取得します。この関数の結果は、*ppstructAgent* によって参照される構造で返されます。

*Sm\_PolicyApi\_FreeMemoryEx()* を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

エージェント関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetAgentByName (
    void*                pSessionHandle,
    const char*          pszAgentName,
    Sm_PolicyApi_Agent_t** ppstructAgent
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAgentName</i>	I	既存のエージェントの名前が含まれ、NULL で終わる文字列です。
<i>ppstructAgent</i>	O	エージェント構造へのポインタのアドレスです。

### 戻り値

- `Sm_PolicyApi_Success`。取得の操作が成功しました。
- `Sm_PolicyApi_Failure`。 `Sm_PolicyApi_Agent_t` にメモリを割り当てることできませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、エージェントを取得する権限がありません。
- `Sm_PolicyApi_NotFound`。指定された名前でのエージェントが見つかりません。

## Sm\_PolicyApi\_GetAgentConfig()

既存のエージェント設定オブジェクトを取得します。

### タイプ

エージェント設定関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetAgentConfig (
    void*                pSessionHandle,
    const char*          pszAgentConfigOid,
    Sm_PolicyApi_AgentConfig_t** ppstructAgentConfig
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAgentConfigOid</i>	I	取得するエージェント設定オブジェクトの一意的識別子です。

パラメータ	I/O	説明
<i>ppstructAgentConfig</i>	O	エージェント設定オブジェクトを定義する構造へのポインタのアドレスです。関数によって構造が割り当てられます。

### 戻り値

- `Sm_PolicyApi_Success`。取得の操作が成功しました。
- `Sm_PolicyApi_Failure`。一般的な障害です。
- `Sm_PolicyApi_InvalidOid`。エージェント設定オブジェクトに一意の ID が対応していません。
- `Sm_PolicyApi_NoPrivilege`。呼び出し元に、エージェント設定オブジェクトを取得する権限がありません。
- `Sm_PolicyApi_InvalidHandle`。セッションポインタが無効です。

## Sm\_PolicyApi\_GetAgentConfigByName()

*pszAgentConfigName* で既存のエージェント設定オブジェクトを取得します。

### タイプ

エージェント設定関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetAgentConfigByName (
    void*                pSessionHandle,
    const char*          pszAgentConfigName,
    Sm_PolicyApi_AgentConfig_t** ppstructAgentConfig
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

パラメータ	I/O	説明
<i>pszAgentConfigName</i>	I	取得するエージェント設定オブジェクトの一意の名前です。
<i>ppstructAgentConfig</i>	O	エージェント設定オブジェクトを定義する構造へのポインタのアドレスです。関数によって構造が割り当てられます。

### 戻り値

- `Sm_PolicyApi_Success`。取得の操作が成功しました。
- `Sm_PolicyApi_Failure`。一般的な障害です。
- `Sm_PolicyApi_NotFound`。一意の名前がエージェント設定オブジェクトに対応しません。
- `Sm_PolicyApi_NoPrivilege`。呼び出し元に、エージェント設定オブジェクトを取得する権限がありません。
- `Sm_PolicyApi_InvalidHandle`。セッションポインタが無効です。
- `Sm_PolicyApi_NoSession`。API ユーザが正しくログインしていません。

## Sm\_PolicyApi\_GetAgentConfigAssociations()

エージェント設定オブジェクト用の設定パラメータのリストを取得します。

### タイプ

エージェント設定関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetAgentConfigAssociations (
    void*                pSessionHandle,
    const char*          pszAgentConfigOid,
    Sm_PolicyApi_Association_t** ppstructAssociations
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAgentConfigOid</i>	I	エージェント設定オブジェクトの一意的識別子です。
<i>ppstructAssociations</i>	O	エージェント設定オブジェクト用の設定パラメータを表す名前/値ペアのリストです。

### 戻り値

- Sm\_PolicyApi\_Success。 取得の操作が成功しました。
- Sm\_PolicyApi\_Failure。 一般的な障害です。
- Sm\_PolicyApi\_InvalidOid。 エージェント設定オブジェクトに一意的 ID が対応していません。
- Sm\_PolicyApi\_NoPrivilege。 呼び出し元には、エージェント設定オブジェクト用の設定パラメータを取得する権限がありません。
- Sm\_PolicyApi\_InvalidHandle。 セッション ポインタが無効です。
- Sm\_PolicyApi\_NoSession。 API ユーザが正しくログインしていません。

## Sm\_PolicyApi\_GetAgentType()

*pszAgentTypeOid* によって識別されたエージェント タイプ オブジェクトの内容を取得します。この関数の結果は、*ppstructAgentType* で参照される構造で返されます。

*Sm\_PolicyApi\_FreeMemoryEx()* を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

エージェント関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetAgentType (
    void*                pSessionHandle,
    const char*          pszAgentTypeOid,
    Sm_PolicyApi_AgentType_t** ppstructAgentType
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAgentTypeOid</i>	I	既存のエージェント タイプのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>ppstructAgentType</i>	O	エージェント タイプ構造へのポインタのアドレスです。

### 戻り値

- `Sm_PolicyApi_Success`。 取得の操作が成功しました。
- `Sm_PolicyApi_Failure`。 メモリは `Sm_PolicyApi_AgentType_t` に対して割り当てられることができませんでした。
- `Sm_PolicyApi_InvalidHandle`。 このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。 有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。 管理者には、エージェントタイプを取得する権限がありません。
- `Sm_PolicyApi_InvalidOid`。 指定された `OID` でのエージェントタイプが見つかりませんでした。

## Sm\_PolicyApi\_GetAgentTypeByName()

`pszAgentTypeName` によって識別されたエージェントタイプオブジェクトの内容を取得します。この関数の結果は、`ppstructAgentType` で参照される構造で返されます。

`Sm_PolicyApi_FreeMemoryEx()` を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

エージェント関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetAgentTypeByName (
    void*                pSessionHandle,
    const char*          pszAgentTypeName,
    Sm_PolicyApi_AgentType_t** ppstructAgentType
);
```

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

パラメータ	I/O	説明
<i>pszAgentTypeName</i>	I	既存のエージェントタイプの名前が含まれ、NULL で終わる文字列です。
<i>ppstructAgentType</i>	O	エージェントタイプ構造へのポインタのアドレスです。

### 戻り値

- **Sm\_PolicyApi\_Success**。 取得の操作が成功しました。
- **Sm\_PolicyApi\_Failure**。 メモリは **Sm\_PolicyApi\_AgentType\_t** に対して割り当てられることができませんでした。
- **Sm\_PolicyApi\_InvalidHandle**。 このコールに先立つ有効な初期化はありませんでした。
- **Sm\_PolicyApi\_NoSession**。 有効な管理者セッションがありません。
- **Sm\_PolicyApi\_NoPrivilege**。 管理者には、エージェントタイプを取得する権限がありません。
- **Sm\_PolicyApi\_NotFound**。 指定された名前でのエージェントタイプが見つかりませんでした。

## Sm\_PolicyApi\_GetAgentTypeAttr()

以下のように、1つまたはすべてのエージェント属性を返します。

- *pszOid* がタイプ `Sm_PolicyApi_AgentType_Prop` である場合、関数はすべてのエージェント属性を返します。
- *pszOid* がタイプ `Sm_PolicyApi_AgentTypeAttr_Prop` である場合、関数は1つのエージェントタイプ属性オブジェクトを返します。

### タイプ

エージェント関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetAgentTypeAttr (
    void*                pSessionHandle,
    const char*          pszOid,
    Sm_PolicyApi_AgentTypeAttr_t** ppstructAgentTypeAttr
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszOid</i>	I	既存のエージェントタイプまたはエージェントタイプ属性のオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>ppstructAgentTypeAttr</i>	O	エージェントタイプ属性構造へのポインタのアドレスです。

### 戻り値

- Sm\_PolicyApi\_Success。取得の操作が成功しました。
- Sm\_PolicyApi\_Failure。メモリは Sm\_PolicyApi\_AgentTypeAttr\_t に対して割り当てられることができませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、エージェントタイプまたはエージェントタイプ属性を取得する権限がありません。
- Sm\_PolicyApi\_InvalidOid。指定された OID でのエージェントタイプまたはエージェントタイプ属性が見つかりませんでした。

## Sm\_PolicyApi\_GetAgentTypeAttrByName()

名前でエージェントタイプ属性オブジェクトを返します。

### タイプ

エージェント関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetAgentTypeAttrByName (
    void*                pSessionHandle,
    const char*          szAgentTypeName,
    Sm_PolicyApi_AgentTypeAttr_t** ppstructAgentTypeAttr
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>szAgentTypeName</i>	I	既存のエージェントタイプ属性の名前が含まれ、NULL で終わる文字列です。

パラメータ	I/O	説明
<i>ppstructAgentTypeAttr</i>	O	エージェントタイプ属性構造へのポインタのアドレスです。

### 戻り値

- `Sm_PolicyApi_Success`。取得の操作が成功しました。
- `Sm_PolicyApi_Failure`。メモリは `Sm_PolicyApi_AgentTypeAttr_t` に対して割り当てられることができませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、エージェントタイプまたはエージェントタイプ属性を取得する権限がありません。
- `Sm_PolicyApi_NotFound`。指定された名前でのエージェントタイプ属性が見つかりませんでした。

## Sm\_PolicyApi\_GetAllAffiliateAttributes()

アフィリエイトの属性をすべて取得します。

### タイプ

フェデレーション関数

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetAllAffiliateAttributes (
    void*                pSessionHandle,
    const char*          pszAffiliateOid,
    Sm_PolicyApi_AffiliateAttr_t** ppstructAffiliateAttr
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

パラメータ	I/O	説明
<i>pszAffiliateOid</i>	I	既存のアフィリエイトのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>ppstruct AffiliateAttr</i>	O	アフィリエイト属性構造体へのポインタのアドレスです。

### 戻り値

- `Sm_PolicyApi_Success`。アフィリエイト属性が正常に取得されました。
- `Sm_PolicyApi_Failure`。アフィリエイト属性が正常に取得されませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、アフィリエイト属性を取得する権限がありません。
- `Sm_PolicyApi_InvalidOID`。アフィリエイト OID が見つかりませんでした。

## Sm\_PolicyApi\_GetAllAffiliates()

指定されたアフィリエイト ドメインのすべてのアフィリエイトを取得します。

### タイプ

フェデレーション関数

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetAllAffiliates (
    void*                pSessionHandle,
    const char*          pszAffiliateDomainOid,
    Sm_PolicyApi_Affiliate_t** ppstructAffiliates
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAffiliateDomainOid</i>	I	既存のアフィリエイト ドメインのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>ppstructアフィリエイト</i>	O	アフィリエイト構造体へのポインタのアドレスです。

### 戻り値

- Sm\_PolicyApi\_Success。アフィリエイトが正常に取得されました。
- Sm\_PolicyApi\_Failure。アフィリエイトが正常に取得されませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、アフィリエイトを取得する権限がありません。
- Sm\_PolicyApi\_InvalidOID。アフィリエイト ドメイン OID が見つかりませんでした。

## Sm\_PolicyApi\_GetAllSAMLAffiliations()

既存の SAML アフィリエーション オブジェクトをすべて取得します。

### タイプ

SAML 2.0 設定関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetAllSAMLAffiliations
(
    void*                pSessionHandle,
    Sm_PolicyApi_SAMLAffiliation_t** ppstructAffiliations
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>ppstructAffiliations</i>	O	SAML アフィリエーション構造へのポインタのアドレスです。

### 戻り値

- **Sm\_PolicyApi\_Success**。 SAML アフィリエーションは正常に取得されました。
- **Sm\_PolicyApi\_Failure**。 SAML アフィリエーションは正常に取得されませんでした。
- **Sm\_PolicyApi\_InvalidHandle**。 このコールに先立つ有効な初期化はありませんでした。
- **Sm\_PolicyApi\_NoSession**。 有効な管理者セッションがありません。
- **Sm\_PolicyApi\_NoPrivilege**。 管理者には、SAML アフィリエーションを取得する権限がありません。

## Sm\_PolicyApi\_GetAllSAMLSchemeAttributes()

SAML リクエストに対して定義されたすべての属性のリンク リストを返します。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddAttributeToSAMLScheme(  
    void*                pHandle,  
    const Sm_PolicyApi_Scheme_t*    pstructScheme,  
    const Sm_PolicyApi_SAMLRequesterAttr_t*    pAttr  
);
```

パラメータ	I/O	説明
<i>pHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pstructScheme</i>	I	値の入力が完了している方式構造体へのポインタです。
<i>pAttr</i>	O	返された属性リストの最初の <code>Sm_PolicyApi_SAMLRequesterAttr_t</code> 構造に割り当てられるポインタです。

### 戻り値

- `Sm_PolicyApi_Success`。SAML リクエストの属性が正常に返されました。
- `Sm_PolicyApi_Failure`。SAML リクエストの属性が正常に返されませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、SAML リクエスト属性を取得する権限がありません。
- `Sm_PolicyApi_SAMLIDP_IncorrectParameters`。入力された SAML 提供のプロパティが未完了か、または正しくありません。
- `Sm_PolicyApi_DuplicateAttribute`。同じ「名前」と `NameFormat` の属性がすでに存在します。

## Sm\_PolicyApi\_GetAllSAMLServiceProviders()

指定されたアフィリエイト ドメインのサービス プロバイダをすべて取得します。

### タイプ

SAML 2.0 設定関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetAllSAMLServiceProviders
(
    void*                                pSessionHandle,
    const char*                           pszAffiliateDomainOid,
    Sm_PolicyApi_SAMLSP_t**               ppstructSAMLSPs
);
```

---

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

---

---

パラメータ	I/O	説明
<i>pszAffiliateDomainOid</i>	I	既存のアフィリエイト ドメインのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>ppstructSAMLSPs</i>	O	サービス プロバイダ構造へのポインタのアドレスです。

---

### 戻り値

- **Sm\_PolicyApi\_Success**。 サービス プロバイダが正常に取得されました。
- **Sm\_PolicyApi\_Failure**。 サービス プロバイダが正常に取得されませんでした。
- **Sm\_PolicyApi\_InvalidHandle**。 このコールに先立つ有効な初期化はありませんでした。
- **Sm\_PolicyApi\_NoSession**。 有効な管理者セッションがありません。
- **Sm\_PolicyApi\_NoPrivilege**。 管理者には、サービス プロバイダを取得する権限がありません。
- **Sm\_PolicyApi\_InvalidOID**。 アフィリエイト ドメイン オブジェクト識別子が見つかりませんでした。

## Sm\_PolicyApi\_GetAllSAMLSPAssertionConsumerService()

現在ポリシーストア内のすべてのアサーション コンシューマ サービスの一覧を取得します。

**注:** この関数によって返されるアサーション コンシューマ サービス構造のリストは、Sm\_PolicyApi\_FreeMemory 関数を使って解放する必要があります。

### タイプ

フェデレーション関数

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetAllSAMLSPAssertionConsumerService(
    void* pSessionHandle,
    const Sm_PolicyApi_SAMLSPAssertionConsumerService_t**
        ppstructSAMLSPAssertionConsumerService,
    const char* pszSAMLSPoid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>ppstructSAMLSPAssertionConsumerServiceAttr</i>	I	アサーション コンシューマ サービス構造体の配列へのポインタです。
<i>pszSAMLSPoid</i>	I	サービス プロバイダの OID を含む文字列へのポインタです。

### 戻り値

- `Sm_PolicyApi_Success`。アサーション コンシューマ サービスの配列が正常に取得されました。
- `Sm_PolicyApi_Failure` - アサーション コンシューマ サービスの配列が正常に取得されませんでした。
- `Sm_PolicyApi_InvalidHandle` - このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession` - 有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege` - 管理者には、SAML サービス プロバイダから属性を削除する権限がありません。

## Sm\_PolicyApi\_GetAllSAMLSPAttributes()

指定されたサービス プロバイダと関連付けられた属性をすべて取得します。

### タイプ

SAML 2.0 設定関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetAllSAMLSPAttributes
(
    void*                                pHandle,
    const char*                          pszSAMLSPoid,
    Sm_PolicyApi_SAMLSPAttr_t**         ppstructSAMLSPAttr
);
```

パラメータ	I/O	説明
<i>pHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszSAMLSPoid</i>	I	既存のサービス プロバイダのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>ppstructSAMLSPAttr</i>	O	属性構造へのポインタのアドレスです。

### 戻り値

- `Sm_PolicyApi_Success`。属性が正常に取得されました。
- `Sm_PolicyApi_Failure`。属性が正常に取得されませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、SAML サービス プロバイダから属性を取得する権限がありません。

## Sm\_PolicyApi\_GetAllWSFEDResourcePartners()

既存のリソース パートナー オブジェクトをすべて取得します。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetAllWSFEDResourcePartners (  
    void* pSessionHandle,  
    const char * pszAffiliateDomainOid,  
    Sm_PolicyApi_WSFEDResourcePartner_t** ppResourcePartners  
);
```

### パラメータ

#### pSessionHandle

[in] 内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

#### pszAffiliateDomainOid

[in] 既存のアフィリエイト ドメインのオブジェクト識別子が含まれ、NULL で終わる文字列です。

#### ppResourcePartners

[out] WS フェデレーション リソース パートナー構造へのポインタのアドレスです。

### 戻り値

- Sm\_PolicyApi\_Success。リソース パートナーが正常に取得されました。
- Sm\_PolicyApi\_Failure。リソース パートナーが正常に取得されませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、アフィリエイトを取得する権限がありません。
- Sm\_PolicyApi\_InvalidOID。アフィリエイト ドメイン OID が見つかりませんでした。

## Sm\_PolicyApi\_GetAuthAzMap()

認証および許可ディレクトリ マッピング オブジェクトの内容を取得します。

### タイプ

認証/許可マップ関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetAuthAzMap (
    void*                pSessionHandle,
    const char*          pszAuthAzMapOid,
    Sm_PolicyApi_AuthAzMap_t** ppAuthAzMap
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAuthAzMapOid</i>	I	ディレクトリ マッピング オブジェクトのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszAuthAzMap</i>	O	Sm_PolicyApi_AuthAzMap_t 構造へのポインタのアドレスです。

### 戻り値

- Sm\_PolicyApi\_Success。取得の操作が成功しました。
- Sm\_PolicyApi\_Failure。メモリは Sm\_PolicyApi\_AuthAzMap\_t に対して割り当てられることができませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、ディレクトリ マッピング オブジェクトを取得する権限がありません。
- Sm\_PolicyApi\_InvalidOid。ディレクトリ マッピング OID が見つかりませんでした。

## Sm\_PolicyApi\_GetCertMap()

証明書マッピング オブジェクトを取得します。

### タイプ

証明書マップ関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetCertMap (
    void*                pSessionHandle,
    const char*          pszCertMapOid,
    Sm_PolicyApi_CertMap_t** ppCertMap
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszCertMapOid</i>	I	証明書マッピング オブジェクトのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>ppCertMap</i>	O	Sm_PolicyApi_CertMap_t 構造へのポインタのアドレスです。

### 戻り値

- Sm\_PolicyApi\_Success。取得の操作が成功しました。
- Sm\_PolicyApi\_Failure。メモリは Sm\_PolicyApi\_CertMap\_t に対して割り当てられないことができませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、証明書マッピング オブジェクトを取得する権限がありません。
- Sm\_PolicyApi\_InvalidOid。指定された OID に一致する証明書マッピング オブジェクトが見つかりませんでした。

## Sm\_PolicyApi\_GetChildren()

階層レلمおよびルール ツリーを構築します。

この関数では、OID のリストを取得します。OID はタイプレلم、またはレلمおよびルールです。関数がドメイン OID で呼び出される場合、それはトップレベルのレلم OID のリストを取得します。関数がレلم OID で呼び出される場合、それはそのレلم下のレلمとルールの OID のリストを取得します。Sm\_PolicyApi\_Oid\_t 内の *iObjectId* フィールドは、OID のタイプを指定します。

ドメインまたはレلم OID の子がない場合、関数は空のリストを返します。

### タイプ

レلم関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetChildren (
    void*                pSessionHandle,
    const char*          pszOid,
    Sm_PolicyApi_Oid_t** ppStructObject
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszOid</i>	I	ドメインまたはレلمのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>ppStructObject</i>	O	Sm_PolicyApi_Oid_t 構造へのポインタのアドレスです。

### 戻り値

- `Sm_PolicyApi_Success`。取得は正常に行われました。
- `Sm_PolicyApi_Failure`。 `Sm_PolicyApi_Oid_t` にメモリを割り当てることができませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoPrivilege`。管理者には、レلمとルール getList を取得する権限がありません。
- `Sm_PolicyApi_InvalidOid`。ドメインまたはレلم OID が見つかりませんでした。
- `Sm_PolicyApi_NoChildren`。ドメインにレلمがないか、またはレلمにレلمまたはルールがありません。

## Sm\_PolicyApi\_GetDirectoryContents()

特定のユーザディレクトリ用のユーザ構造 (`ppStructUsers` によって参照された) のリンクリストを取得します。

この関数に対するレスポンスの精度は以下のレジストリ エントリによって管理されます。

```
HKLM\SOFTWARE\Wow6423Node\Netegrity\SiteMinder\CurrentVersion\Ds\ClassFilters
```

`Sm_PolicyApi_FreeMemoryEx()` を呼び出して返された構造に割り当てられたメモリを解放します。

### タイプ

ユーザディレクトリ関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetDirectoryContents (
    void*                pSessionHandle,
    const char*          pszUserDirOid,
    Sm_PolicyApi_User_t** ppStructUserSpec
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszUserDirOid</i>	I	既存のユーザディレクトリのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>ppStructUserSpec</i>	O	ユーザ構造へのポインタのアドレスです。

### 戻り値

- `Sm_PolicyApi_Success`。取得の操作が成功しました。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_InvalidOid`。ユーザディレクトリ OID が見つかりませんでした。

## Sm\_PolicyApi\_GetDisabledUserState()

ユーザの無効状態を取得します。

この関数を正常に動作させるには、ディレクトリ属性を指定して無効化されたユーザを追跡する必要があります。この属性はユーザディレクトリの無効フラグで指定されます。無効化の理由は `Sm_Api_DisabledReason_t` に列挙され、それは `SmApi.h` で定義されています。

### タイプ

ユーザとユーザ状態関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetDisabledUserState (
    void*                pSessionHandle,
    const char*          pszUserDirOid,
    const char*          pszUserDN,
    Sm_Api_DisabledReason_t* nDisabledReason,
    char**               pszErrMsg
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszUserDirOid</i>	I	ユーザを含む既存ユーザディレクトリのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszUserDN</i>	I	無効状態が変更されるユーザの識別名を含み、NULL で終わる文字列です。
<i>nDisabledReason</i>	O	ユーザを無効または有効にする理由です。理由は <code>Sm_Api_DisabledReason_t</code> に列挙され、それは <code>SmApi.h</code> で定義されています。
<i>pszErrMsg</i>	O	取得が成功しなかった場合に、エラーメッセージがこの文字列に保持されます。 <code>Sm_PolicyApi_FreeString()</code> を呼び出して、この変数に割り当てられたメモリを解放します。

### 戻り値

- `Sm_PolicyApi_Success`。取得は正常に行われました。
- `Sm_PolicyApi_Failure` :
  - 無効化状態が取得されませんでした。
  - `pszErrMsg` にメモリを割り当てることができませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、ユーザ無効状態を取得する権限がありません。
- `Sm_PolicyApi_InvalidOid`。ユーザディレクトリ `OID` が見つかりませんでした。

## Sm\_PolicyApi\_GetDomain()

`pszDomainOid` によって識別されたドメインの内容を取得します。この関数の結果は、`ppstructDomain` によって参照される構造で返されます。

`Sm_PolicyApi_FreeMemoryEx()` を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

ドメイン関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetDomain (  
    void*                pSessionHandle,  
    const char*          pszDomainOid,  
    Sm_PolicyApi_Domain_t** ppstructDomain  
);
```

---

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

---

---

パラメータ	I/O	説明
<i>pszDomainOid</i>	I	既存のドメインのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>ppstructDomain</i>	O	ドメイン構造へのポインタのアドレスです。

---

### 戻り値

- `Sm_PolicyApi_Success`。 取得の操作が成功しました。
- `Sm_PolicyApi_Failure`。 `Sm_PolicyApi_Domain_t` にメモリを割り当てることができませんでした。
- `Sm_PolicyApi_InvalidHandle`。 このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。 有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。 管理者には、ドメインを取得する権限がありません。
- `Sm_PolicyApi_InvalidOid`。 ドメイン OID が見つかりませんでした。

## Sm\_PolicyApi\_GetDomainByName()

*pszDomainName* によって識別されたドメインの内容を取得します。この関数の結果は、*ppstructDomain* によって参照される構造で返されます。

*Sm\_PolicyApi\_FreeMemoryEx()* を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

ドメイン関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetDomainByName (
    void*                pSessionHandle,
    const char*          pszDomainName,
    Sm_PolicyApi_Domain_t** ppstructDomain
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszDomainName</i>	I	既存のドメインの名前が含まれ、NULL で終わる文字列です。
<i>ppstructDomain</i>	O	ドメイン構造へのポインタのアドレスです。

### 戻り値

- `Sm_PolicyApi_Success`。取得の操作が成功しました。
- `Sm_PolicyApi_Failure`。 `Sm_PolicyApi_Domain_t` にメモリを割り当てることができませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、ドメインを取得する権限がありません。
- `Sm_PolicyApi_NotFound`。ドメイン名が見つかりませんでした。

## Sm\_PolicyApi\_GetDomainObjects()

`pszDomainOid` によって識別されたドメイン内の指定されたオブジェクトタイプに対してドメインオブジェクトの **OID** を取得します。返される値は、`Sm_PolicyApi_Oid_t` 構造に含まれます。

`Sm_PolicyApi_FreeMemoryEx()` を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

ドメイン関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetDomainObjects (
    void*                pSessionHandle,
    const char*          pszDomainOid,
    const Sm_PolicyApi_Objects_t nObjectId,
    Sm_PolicyApi_Oid_t** ppstructObject
);
```

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

パラメータ	I/O	説明
<i>pszDomainOid</i>	I	既存のドメインのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>nObjectId</i>	I	Sm_PolicyApi_Objects_t で列挙されるように、取得するドメイン オブジェクトのタイプです。
<i>ppstructObject</i>	O	Sm_PolicyApi_Oid_t 構造へのポインタのアドレスです。

### 戻り値

- Sm\_PolicyApi\_Success。 取得の操作が成功しました。
- Sm\_PolicyApi\_Failure。 Sm\_PolicyApi\_Oid\_t にメモリを割り当てることができませんでした。
- Sm\_PolicyApi\_InvalidHandle。 このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。 有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。 管理者には、ドメイン ベースのオブジェクトを取得する権限がありません。
- Sm\_PolicyApi\_InvalidOid。 ドメイン OID が見つかりませんでした。

## Sm\_PolicyApi\_GetGlobalObjects()

指定されたタイプのグローバル オブジェクトのオブジェクト識別子を取得します。SiteMinder v6.0 以上で、この関数はグローバル オブジェクトとしてルール、ポリシーおよびレスポンス プロパティを受け取り、グローバル ルール、ポリシーおよびレスポンスを返します。

返される値は、Sm\_PolicyApi\_Oid\_t 構造に含まれます。

Sm\_PolicyApi\_FreeMemoryEx() を呼び出すことによりこの構造に割り当てられたメモリを解放します。

**注:** SiteMinder v4.5 より前のリリースでは、Sm\_PolicyApi\_GetGlobalObjects() によって提供される機能は、Sm\_PolicyApi\_GetGlobalObjectNames() によって提供されています。

### タイプ

一般オブジェクト関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetGlobalObjects (
    void*                pSessionHandle,
    const Sm_PolicyApi_Objects_t nObjectId,
    Sm_PolicyApi_Oid_t** ppstructObject
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>nObjectId</i>	I	取得するグローバル オブジェクトのタイプです。オブジェクトタイプは Sm_PolicyApi_Objects_t に列挙されます。
<i>ppstructObject</i>	O	Sm_PolicyApi_Oid_t 構造へのポインタのアドレスです。

### 戻り値

- `Sm_PolicyApi_Success`。 取得の操作が成功しました。
- `Sm_PolicyApi_Failure`。 `Sm_PolicyApi_Oid_t` にメモリを割り当てることができませんでした。
- `Sm_PolicyApi_NoSession`。 有効な管理者セッションがありません。
- `Sm_PolicyApi_InvalidHandle`。 このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoPrivilege`。 管理者には、グローバル オブジェクトを取得する権限がありません。
- `Sm_PolicyApi_Invalid`。 非グローバル オブジェクト識別子が指定されました。

## Sm\_PolicyApi\_GetGlobalPolicyByName()

指定されたグローバル ポリシーを取得します。

この関数の結果は、`ppstructPolicy` によって参照される構造で返されます。`Sm_PolicyApi_FreeMemoryEx()` を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

管理者関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetGlobalPolicyByName (
    void*                pHandle,
    const char*          szPolicyName,
    Sm_PolicyApi_Policy_t** ppstructPolicy
);
```

パラメータ	I/O	説明
<code>pHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<code>szPolicyName</code>	I	グローバル ポリシーに相当する一意の名前です。

---

パラメータ	I/O	説明
<i>ppstructPolicy</i>	O	ポリシー構造へのポインタのアドレスです。

---

### 戻り値

- `Sm_PolicyApi_Success`。取得の操作が成功しました。
- `Sm_PolicyApi_Failure`。 `Sm_PolicyApi_Policy_t` にメモリを割り当てることができませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、ポリシーを取得する権限がありません。
- `Sm_PolicyApi_NotFound`。ポリシー名が見つかりませんでした。

## Sm\_PolicyApi\_GetGlobalResponseByName()

指定されたグローバル レスポンスを取得します。

この関数の結果は、*ppstructResponse* によって参照される構造で返されま  
す。

*Sm\_PolicyApi\_FreeMemoryEx()* を呼び出すことによりこの構造に割り当て  
られたメモリを解放します。

### タイプ

管理者関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetGlobalResponseByName (
    void*                pHandle,
    const char*          szResponseName,
    Sm_PolicyApi_Response_t** ppstructResponse
);
```

パラメータ	I/O	説明
<i>pHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>szResponseName</i>	I	グローバル レスポンスに相当する一意の名前です。
<i>ppstructResponse</i>	O	レスポンス構造へのポインタのアドレスです。

### 戻り値

- `Sm_PolicyApi_Success`。取得の操作が成功しました。
- `Sm_PolicyApi_Failure`。 `Sm_PolicyApi_Response_t` にメモリを割り当てるできませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoPrivilege`。管理者には、方式を取得する権限がありません。
- `Sm_PolicyApi_NotFound`。グローバル レスポンス名が見つかりませんでした。

## Sm\_PolicyApi\_GetGlobalRuleByName()

指定されたグローバルルールを取得します。

この関数の結果は、`ppstructRule` によって参照される構造で返されます。

`Sm_PolicyApi_FreeMemoryEx()` を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

管理者関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetGlobalRuleByName (
    void*                pHandle,
    const char*          szRuleName,
    Sm_PolicyApi_Rule_t** ppstructRule
);
```

パラメータ	I/O	説明
<code>pHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

パラメータ	I/O	説明
<i>szRuleName</i>	I	グローバルルールに相当する一意の名前です。
<i>ppstructRule</i>	O	レスポンス構造へのポインタのアドレスです。

### 戻り値

- `Sm_PolicyApi_Success`。 取得の操作が成功しました。
- `Sm_PolicyApi_Failure`。 `Sm_PolicyApi_Rule_t` にメモリを割り当てることができませんでした。
- `Sm_PolicyApi_InvalidHandle`。 このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。 有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。 管理者には、方式を取得する権限がありません。
- `Sm_PolicyApi_NotFound`。 グローバルルール名が見つかりませんでした。

## Sm\_PolicyApi\_GetGroup()

*pszGroupOid* によって識別されたグループ オブジェクトの内容を取得します。この関数の結果は、*ppStructGroup* によって参照される構造で返されます。

*pszDomainOid* パラメータはルールまたはレスポンス グループに必須です。エージェントグループはドメイン ベースのオブジェクトではないので、ドメイン OID を必要としません。

*Sm\_PolicyApi\_FreeMemoryEx()* を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

グループ関数、グローバル範囲（エージェント）またはドメイン範囲（レスポンス、ルール）。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetGroup (
    void*                pSessionHandle,
    Sm_PolicyApi_Groups_t dwGroup,
    const char*          pszGroupOid,
    const char*          pszDomainOid,
    Sm_PolicyApi_Group_t** ppStructGroup
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>dwGroup</i>	I	取得されるグループのタイプです。
<i>pszGroupOid</i>	I	取得中のグループ オブジェクトのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszDomainOid</i>	I	既存のドメインのオブジェクト識別子が含まれ、NULL で終わる文字列です。ルールまたはレスポンス グループの必須パラメータです。

パラメータ	I/O	説明
<i>ppStructGroup</i>	O	グループ構造へのポインタのアドレスです。

---

#### 戻り値

- **Sm\_PolicyApi\_Success**。 取得は正常に行われました。
- **Sm\_PolicyApi\_Failure**。 取得が成功しませんでした。
- **Sm\_PolicyApi\_InvalidHandle**。 このコールに先立つ有効な初期化はありませんでした。
- **Sm\_PolicyApi\_NoSession**。 有効な管理者セッションがありません。
- **Sm\_PolicyApi\_NoPrivilege**。 管理者には、グループの内容を取得する権限がありません。
- **Sm\_PolicyApi\_InvalidOid** :
  - グループ OID が見つかりませんでした。
  - ドメイン OID が見つかりませんでした（ドメインベースのグループについて）。
- **Sm\_PolicyApi\_BadGroup**。 *dwGroup* パラメータ はルール、レスポンスまたはエージェント タイプではありません。

## Sm\_PolicyApi\_GetGroupName()

*pszGroupName* によって識別されたグループ オブジェクトの内容を取得します。この関数の結果は、*ppStructGroup* によって参照される構造で返されます。

*pszDomainOid* パラメータはルールまたはレスポンス グループに必須です。エージェントグループはドメイン ベースのオブジェクトではないので、ドメイン OID を必要としません。

*Sm\_PolicyApi\_FreeMemoryEx()* を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

グループ関数、グローバル範囲（エージェント）またはドメイン範囲（レスポンス、ルール）。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetGroupName (
    void*                pSessionHandle,
    Sm_PolicyApi_Groups_t dwGroup,
    const char*          pszGroupName,
    const char*          pszDomainOid,
    Sm_PolicyApi_Group_t** ppStructGroup
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>dwGroup</i>	I	取得されるグループのタイプです。
<i>pszGroupOid</i>	I	取得中のグループ オブジェクトのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszDomainName</i>	I	既存のドメインの名前が含まれ、NULL で終わる文字列です。ルールまたはレスポンス グループの必須パラメータです。

パラメータ	I/O	説明
<i>ppStructGroup</i>	O	グループ構造へのポインタのアドレスです。

---

#### 戻り値

- `Sm_PolicyApi_Success`。 取得は正常に行われました。
- `Sm_PolicyApi_Failure`。 取得が成功しませんでした。
- `Sm_PolicyApi_InvalidHandle`。 このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。 有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。 管理者には、グループの内容を取得する権限がありません。
- `Sm_PolicyApi_InvalidOid`。 ドメイン OID が見つかりませんでした（ドメインベースのグループについて）。
- `SmPolicyApi_Notfound`。 グループ名が見つかりませんでした。
- `Sm_PolicyApi_BadGroup`。 *dwGroup* パラメータ はルール、レスポンスまたはエージェントタイプではありません。

## Sm\_PolicyApi\_GetGroupOids()

グループ オブジェクト内に含まれているオブジェクト識別子を取得します。この関数の結果は、*ppStructObjects* によって参照される構造で返されます。

*Sm\_PolicyApi\_FreeMemoryEx()* を呼び出すことによりこの構造に割り当てられたメモリを解放します。

*pszDomainOid* はルール グループまたはレスポンス グループに必須です。エージェント グループはドメイン ベースのオブジェクトではないので、ドメイン OID を必要としません。

注: SiteMinder v4.5 より前のリリースでは、*Sm\_PolicyApi\_GetGroupOids()* によって提供される機能は、*Sm\_PolicyApi\_GetGroupNames()* によって提供されています。

### タイプ

グループ関数、グローバル範囲（エージェント）またはドメイン範囲（レスポンス、ルール）。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetGroupOids (
    void*                pSessionHandle,
    Sm_PolicyApi_Groups_t dwGroup,
    const char*          pszGroupOid,
    const char*          pszDomainOid,
    Sm_PolicyApi_Oid_t** ppStructObjects
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>dwGroup</i>	I	オブジェクト識別子が取得されるグループのタイプを示します。
<i>pszGroupOid</i>	I	<i>dwGroup</i> によって示されるタイプのグループのオブジェクト識別子が含まれ、NULL で終わる文字列です。

パラメータ	I/O	説明
<i>pszDomainOid</i>	I	既存のドメインのオブジェクト識別子が含まれ、NULLで終わる文字列です。ルールまたはレスポンスグループの必須パラメータです。
<i>ppStructObjects</i>	O	オブジェクト構造のアドレスへのポインタです。

### 戻り値

- **Sm\_PolicyApi\_Success**。 取得は正常に行われました。
- **Sm\_PolicyApi\_Failure**。 取得の操作が成功しなかったか、または **Sm\_PolicyApi\_Oid\_t** にメモリを割り当てることができませんでした。
- **Sm\_PolicyApi\_InvalidHandle**。 このコールに先立つ有効な初期化はありませんでした。
- **Sm\_PolicyApi\_NoSession**。 有効な管理者セッションがありません。
- **Sm\_PolicyApi\_NoPrivilege**。 管理者には、グループ **OID** を取得する権限がありません。
- **Sm\_PolicyApi\_InvalidOid** :
  - グループ **OID** が見つかりませんでした。
  - ドメイン **OID** が見つかりませんでした（ドメインベースのグループについて）。
- **Sm\_PolicyApi\_BadGroup**。 パラメータ *dwGroup* はルール、レスポンスまたはエージェントタイプではありません。

## Sm\_PolicyApi\_GetHostConfig()

既存のホスト設定オブジェクトを取得します。

### タイプ

エージェント設定関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetHostConfig (
    void*                pSessionHandle,
    const char*          pszHostConfigOid,
    Sm_PolicyApi_HostConfig_t** ppstructHostConfig
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszHostConfigOid</i>	I	取得するホスト設定オブジェクトの一意の識別子です。
<i>ppstructHostConfig</i>	O	ホスト設定オブジェクトを定義する構造へのポインタのアドレスです。関数によって構造が割り当てられます。

### 戻り値

- Sm\_PolicyApi\_Success。取得の操作が成功しました。
- Sm\_PolicyApi\_Failure。一般的な障害です。
- Sm\_PolicyApi\_InvalidOid。一意の ID がホスト設定オブジェクトに対応しません。
- Sm\_PolicyApi\_NoPrivilege。呼び出し元に、ホスト設定オブジェクトを取得する権限がありません。
- Sm\_PolicyApi\_InvalidHandle。セッションポインタが無効です。
- Sm\_PolicyApi\_NoSession。API ユーザーが正しくログインしていません。

## Sm\_PolicyApi\_GetHostConfigByName()

既存のホスト設定オブジェクトを取得します。

### タイプ

エージェント設定関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetHostConfigByName (
    void*                pSessionHandle,
    const char*          pszHostConfigName,
    Sm_PolicyApi_HostConfig_t** ppstructHostConfig
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszHostConfigName</i>	I	取得するホスト設定オブジェクトの一意の名前です。
<i>ppstructHostConfig</i>	O	ホスト設定オブジェクトを定義する構造へのポインタのアドレスです。関数によって構造が割り当てられます。

### 戻り値

- Sm\_PolicyApi\_Success。取得の操作が成功しました。
- Sm\_PolicyApi\_Failure。一般的な障害です。
- Sm\_PolicyApi\_NotFound。一意の名前がホスト設定オブジェクトに対応しません。
- Sm\_PolicyApi\_NoPrivilege。呼び出し元に、ホスト設定オブジェクトを取得する権限がありません。
- Sm\_PolicyApi\_InvalidHandle。セッションポインタが無効です。
- Sm\_PolicyApi\_NoSession。API ユーザーが正しくログインしていません。

## Sm\_PolicyAPI\_GetMessageConsumerPluginFromSAML1xScheme()

メッセージ コンシューマ プラグイン設定を SAML1.x 認証方式から取得します。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetMessageConsumerPluginFromSAML1xScheme(  
    void*    pHandle,  
    char*    pszSchemeOID,  
    char**   pluginClass,  
    char**   pluginParam  
);
```

### パラメータ

#### pHandle

[in] 内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

#### pszSchemeOid

[in] 更新中の認証方式の OID へのポインタです。

#### pluginClass

[out] 認証方式から読み取られる、プラグイン クラスの名前へのポインタです。

#### pluginParam

[out] 認証方式から読み取られる、プラグイン クラスのパラメータへのポインタです。

### 戻り値

- Sm\_PolicyApi\_Success。操作は正常に完了しました。
- Sm\_PolicyApi\_Failure。動作が失敗しました。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_SchemeNotFound。OID に対応する認証方式が見つからなかったか、SAML 1.x 認証方式ではありませんでした。
- Sm\_PolicyApi\_InvalidOid。認証方式の OID は NULL です。

## Sm\_PolicyApi\_GetODBCQueryScheme()

*pszODBCQuerySchemeOid* によって識別される ODBC クエリ方式オブジェクトの内容を取得します。

この関数の結果は、*pstructODBCQueryScheme* によって参照される構造で返されます。この構造に割り当てられたメモリは、*Sm\_PolicyApi\_FreeMemoryEx()* を呼び出して解放します。

### タイプ

ODBC クエリ方式関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetODBCQueryScheme (
    void*                pSessionHandle,
    const char*          pszODBCQuerySchemeOid,
    Sm_PolicyApi_ODBCQueryScheme_t** pstructODBCQueryScheme
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszODBCQuerySchemeOid</i>	I	既存の ODBC クエリ方式のオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pstructODBCQueryScheme</i>	O	ODBC クエリ方式へのポインタのアドレスです。

### 戻り値

- Sm\_PolicyApi\_Success。 取得の操作が成功しました。
- Sm\_PolicyApi\_Failure。 メモリは Sm\_PolicyApi\_ODBCQueryScheme\_t に対して割り当てられることができませんでした。
- Sm\_PolicyApi\_InvalidHandle。 このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。 有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。 管理者には、ODBC クエリ方式を取得する権限がありません。
- Sm\_PolicyApi\_InvalidOid。 ODBC クエリ方式 OID が見つかりませんでした。

## Sm\_PolicyApi\_GetODBCQuerySchemeByName()

*pszODBCQuerySchemeName* によって識別される ODBC クエリ方式オブジェクトの内容を取得します。

この関数の結果は、*pstructODBCQueryScheme* によって参照される構造で返されます。 この構造に割り当てられたメモリは、Sm\_PolicyApi\_FreeMemoryEx() を呼び出して解放します。

### タイプ

ODBC クエリ方式関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetODBCQuerySchemeByName (
    void*                pSessionHandle,
    const char*          pszODBCQuerySchemeName,
    Sm_PolicyApi_ODBCQueryScheme_t** pstructODBCQueryScheme
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。 この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

パラメータ	I/O	説明
<i>pszODBCQuerySchemeName</i>	I	既存の ODBC クエリ方式の名前が含まれ、NULL で終わる文字列です。
<i>pstructODBCQueryScheme</i>	O	ODBC クエリ方式へのポインタのアドレスです。

#### 戻り値

- `Sm_PolicyApi_Success`。取得の操作が成功しました。
- `Sm_PolicyApi_Failure`。メモリは `Sm_PolicyApi_ODBCQueryScheme_t` に対して割り当てられることができませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、ODBC クエリ方式を取得する権限がありません。
- `Sm_PolicyApi_NotFound`。ODBC クエリ方式名が見つかりませんでした。

## Sm\_PolicyApi\_GetOneTimeUsePropFromAffiliate()

SAML 1.x アフィリエイトでアサーションの **OneTimeUse** プロパティの値を取得します。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetOneTimeUsePropFromAffiliate(  
    void*    pHandle,  
    char*    pszAffiliateOID,  
    bool     &bOneTimeUse  
);
```

### パラメータ

#### pHandle

[in] 内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

#### pszAffiliateOid

[in] 既存の SAML 1.x アフィリエイトの OID へのポインタです。

#### bOneTimeUse

[in] アサーションがこのアフィリエイトで 1 回のみ使用されるかどうかを指定するブール値です。

### 戻り値

- Sm\_PolicyApi\_Success。操作は正常に完了しました。
- Sm\_PolicyApi\_Failure。動作が失敗しました。
- Sm\_PolicyApi\_NoSession。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoPrivilege。管理者に十分なアクセス権限がありませんでした。

## Sm\_PolicyApi\_GetPasswordMsg()

新規パスワードの検証時に発生したエラーに関する情報を取得します。

Sm\_PolicyApi\_SetPassword() がエラー コード Sm\_PolicyApi\_InvalidPasswordSyntax を返すとき、この関数を呼び出します。

### タイプ

ユーザとユーザ状態関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetPasswordMsg (
    void*                pSessionHandle,
    const char*          pszPasswordMsg,
    unsigned int*        nVersion,
    unsigned int*        nMsgId,
    unsigned int*        nArgs,
    Sm_PolicyApi_PasswordMsgField_t** ppStructMsgField
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszPasswordMsg</i>	I	関数 Sm_PolicyApi_SetPassword() の <i>pszErrMsg</i> パラメータで返されたエンコードされたエラーメッセージ。
<i>nVersion</i>	O	SiteMinder パスワード サービスのバージョン。
<i>nMsgId</i>	O	エンコードされたエラーメッセージに対して取得されたパスワード サービス メッセージ識別子。メッセージ識別子は Sm_PolicyApi_PasswordMsgId_t で列挙されます。

---

パラメータ	I/O	説明
<i>nArgs</i>	O	<i>ppStructMsgField</i> によって参照される <code>Sm_PolicyApi_PasswordMsgField_t</code> 構造内のフィールド数。
<i>ppStructMsgField</i>	O	パスワードエラーメッセージ情報が含まれる <code>Sm_PolicyApi_PasswordMsgField_t</code> 構造へのポインタのアドレス。

---

### 注釈

SiteMinder パスワード サービス エラーには一意のメッセージ識別子が含まれます。( `Sm_PolicyApi_PasswordMsgId_t` で列挙される) メッセージ識別子が関数 `Sm_PolicyApi_GetPasswordMsg()` の *nMsgId* パラメータで返されます。

さらに、メッセージ識別子はそれぞれエラーに関する追加の情報と関連付けられます。この追加の情報は、( `Sm_PolicyApi_PasswordMsgFieldId_t` で列挙される) フィールド識別子によって参照されます。

たとえば、管理者がパスワードの最小の長さを 7 文字に設定すると仮定します。ユーザがちょうど 6 文字が含まれる新しいパスワードを検証しようとする場合、以下のイベントが発生する場合があります。

1. パスワード検証の試行失敗の結果、`Sm_PolicyApi_SetPassword()` は `Sm_PolicyApi_InvalidPasswordSyntax` を返します。
2. ポリシー管理アプリケーションは、`Sm_PolicyApi_GetPasswordMsg()` を呼び出し、`pszPasswordMsg` パラメータ内の `Sm_PolicyApi_SetPassword()` から返された、エンコードされたエラーメッセージ渡すことで応答します。
3. `Sm_PolicyApi_GetPasswordMsg()` が戻ると、
  - `nMsgId` には `Sm_PolicyApi_PasswordMsgId_PasswordShort` が含まれます。
  - `ppStructMsgField` は、パスワードエラーに関する以下の追加情報が含まれる `Sm_PolicyApi_PasswordMsgField_t` 構造を参照します。

フィールド	値と意味
<code>iStructId</code>	<code>Sm_PolicyApi_PasswordMsgField_ID</code> 。 このエラー情報はパスワードポリシーに関連しています。
<code>nId</code>	<code>Sm_PolicyApi_PasswordMsgField_Id_Min</code> このエラーは、パスワード（パスワードの最初文字数）に関する最小の文字要件に違反します。
<code>nType</code>	<code>Sm_PolicyApi_FieldType_Int</code> 。 このエラー説明は整数です。したがって、 <code>nValue</code> が入力され <code>pszMsg</code> は入力されません。
<code>pszMsg</code>	""。 このエラーには適用されません。
<code>nValue</code>	7。 パスワードの最小文字数。要求されたパスワードの長さはこの値未満でした。

## Sm\_PolicyApi\_GetPasswordPolicy()

パスワード ポリシー オブジェクトの内容を取得します。

### タイプ

パスワード ポリシー関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetPasswordPolicy (
    void*                pSessionHandle,
    const char*          pszPasswordPolicyOid,
    Sm_PolicyApi_PasswordPolicy_t** ppstructPasswordPolicy
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszPasswordPolicyOid</i>	I	パスワード ポリシーのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>ppstructPasswordPolicy</i>	O	Sm_PolicyApi_PasswordPolicy_t へのポインタのアドレス。

### 戻り値

- Sm\_PolicyApi\_Success。取得は正常に行われました。
- Sm\_PolicyApi\_Failure。メモリは Sm\_PolicyApi\_PasswordPolicy\_t に割り当てられることができませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、パスワード ポリシーを取得する権限がありません。
- Sm\_PolicyApi\_InvalidOid。パスワード ポリシー OID が見つかりませんでした。

## Sm\_PolicyApi\_GetPasswordPolicyByName()

パスワード ポリシー オブジェクトの内容を取得します。

### タイプ

パスワード ポリシー関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetPasswordPolicyByName (
    void*                pSessionHandle,
    const char*          pszPasswordPolicyName,
    Sm_PolicyApi_PasswordPolicy_t** ppstructPasswordPolicy
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszPasswordPolicyName</i>	I	パスワード ポリシーの名前が含まれ、NULL で終わる文字列です。
<i>ppstructPasswordPolicy</i>	O	Sm_PolicyApi_PasswordPolicy_t へのポインタのアドレス。

### 戻り値

- Sm\_PolicyApi\_Success。取得は正常に行われました。
- Sm\_PolicyApi\_Failure。メモリは Sm\_PolicyApi\_PasswordPolicy\_t に割り当てられることができませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、パスワード ポリシーを取得する権限がありません。
- Sm\_PolicyApi\_NotFound。パスワード ポリシー名が見つかりませんでした。

## Sm\_PolicyApi\_GetPolicy()

*pszPolicyOid* で識別されるポリシーの内容を取得します。

この関数の結果は、*ppstructPolicy* によって参照される構造で返されます。*Sm\_PolicyApi\_FreeMemoryEx()* を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

ポリシー関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetPolicy (
    void*                pSessionHandle,
    const char*          pszPolicyOid,
    Sm_PolicyApi_Policy_t** ppstructPolicy
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszPolicyOid</i>	I	既存のポリシーのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>ppstructPolicy</i>	O	ポリシー構造へのポインタのアドレスです。

### 戻り値

- Sm\_PolicyApi\_Success。 取得の操作が成功しました。
- Sm\_PolicyApi\_Failure。 Sm\_PolicyApi\_Policy\_t にメモリを割り当てること  
ができませんでした。
- Sm\_PolicyApi\_InvalidHandle。 このコールに先立つ有効な初期化はあり  
ませんでした。
- Sm\_PolicyApi\_NoSession。 有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。 管理者には、ポリシーを取得する権限があ  
りません。
- Sm\_PolicyApi\_InvalidOid。 ポリシー OID が見つかりませんでした。

## Sm\_PolicyApi\_GetPolicyByName()

*pszPolicyName*、およびポリシーが存在するドメインの対応する  
*pszDomainOid* によって識別されるポリシーの内容を取得します。

この関数の結果は、*ppstructPolicy* によって参照される構造で返されます。  
Sm\_PolicyApi\_FreeMemoryEx() を呼び出すことによりこの構造に割り当て  
られたメモリを解放します。

### タイプ

ポリシー関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetPolicyByName (
    void*                pSessionHandle,
    const char*          szDomainOid,
    const char*          pszPolicyName,
    Sm_PolicyApi_Policy_t** ppstructPolicy
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この 構造体は、管理者セッションおよびクライアントセッ ションに関する情報を保持します。

---

パラメータ	I/O	説明
<i>szDomainOid</i>	I	既存のドメインのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszPolicyName</i>	I	既存のポリシーの名前が含まれ、NULL で終わる文字列です。
<i>ppstructPolicy</i>	O	ポリシー構造へのポインタのアドレスです。

---

### 戻り値

- `Sm_PolicyApi_Success`。 取得の操作が成功しました。
- `Sm_PolicyApi_Failure`。 `Sm_PolicyApi_Policy_t` にメモリを割り当てることができませんでした。
- `Sm_PolicyApi_InvalidHandle`。 このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。 有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。 管理者には、ポリシーを取得する権限がありません。
- `Sm_PolicyApi_InvalidOid`。 ドメイン OID が見つかりませんでした。
- `Sm_PolicyApi_NotFound`。 ポリシー名が見つかりませんでした。

## Sm\_PolicyApi\_GetPolicyLinks()

*pszPolicyOid* によって識別されるポリシーと関連付けられるポリシー リンクすべてのリンク リストを返します。

返されたリンク リストは、*ppstructPolicyLink* 構造によって参照されます。*Sm\_PolicyApi\_FreeMemoryEx()* を呼び出すことによりこれらの構造に割り当てられたメモリを解放します。

### タイプ

ポリシー関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetPolicyLinks (
    void*                pSessionHandle,
    const char*          pszPolicyOid,
    Sm_PolicyApi_PolicyLink_t** ppstructPolicyLink
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszPolicyOid</i>	I	既存のポリシーのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>ppstructPolicyLink</i>	O	ポリシー リンク構造のアドレス。

### 戻り値

- `Sm_PolicyApi_Success`。取得の操作が成功しました。
- `Sm_PolicyApi_Failure`。メモリは `Sm_PolicyApi_PolicyLink_t` に割り当てられません。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、ポリシー リnkのリストを取得する権限がありません。
- `Sm_PolicyApi_InvalidOid`。ポリシー OID が見つかりませんでした。

## Sm\_PolicyApi\_GetPolicyUsers()

`pszPolicyOid` によって識別されるポリシーと関連付けられ、オプションで `pszUserDirOid` によって識別されるユーザディレクトリに属するユーザに対応する `Sm_PolicyApi_User_t` 構造のリンク リストを返します。  
`Sm_PolicyApi_FreeMemoryEx()` を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

ユーザとユーザ状態関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetPolicyUsers (
    void*                pSessionHandle,
    const char*          pszPolicyOid,
    const char*          pszUserDirOid,
    Sm_PolicyApi_User_t** ppStructUsers
);
```

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

パラメータ	I/O	説明
<i>pszPolicyOid</i>	I	既存のポリシーのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszUserDirOid</i>	I	既存のユーザディレクトリのオブジェクト識別子が含まれ、NULL で終わる文字列です。このパラメータは任意です。これが空白の場合、関数は指定されたポリシー下のポリシー ユーザすべてを返します。これが指定されている場合、関数は指定されたポリシー下のこのディレクトリに属するポリシー ユーザを返します。
<i>ppStructUsers</i>	O	ユーザ構造のリンク リストへのポインタのアドレスです。

### 戻り値

- **Sm\_PolicyApi\_Success**。 取得の操作が成功しました。
- **Sm\_PolicyApi\_Failure**。 メモリを **Sm\_PolicyApi\_User\_t** に割り当てるできませんでした。
- **Sm\_PolicyApi\_InvalidHandle**。 このコールに先立つ有効な初期化はありませんでした。
- **Sm\_PolicyApi\_NoSession**。 有効な管理者セッションがありません。
- **Sm\_PolicyApi\_NoPrivilege**。 管理者には、ポリシーからユーザを取得する権限がありません。
- **Sm\_PolicyApi\_InvalidOid** :
  - ポリシー **OID** が見つかりませんでした。
  - ユーザディレクトリ **OID** が見つかりませんでした。

## Sm\_PolicyApi\_GetRealm()

*pszRealmOid* によって識別されるレルムの内容を取得します。

この関数の結果が *ppstructRealm* によって参照される構造で返されます。  
Sm\_PolicyApi\_FreeMemoryEx() を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

レルム関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetRealm (
    void*                pSessionHandle,
    const char*          pszRealmOid,
    Sm_PolicyApi_Realm_t** ppstructRealm
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszRealmOid</i>	I	既存のレルムのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>ppstructRealm</i>	O	レルム構造へのポインタのアドレス。

### 戻り値

- Sm\_PolicyApi\_Success。 取得の操作が成功しました。
- Sm\_PolicyApi\_Failure。 メモリは Sm\_PolicyApi\_Realm\_t に対して割り当てられることができませんでした。
- Sm\_PolicyApi\_NoSession。 有効な管理者セッションがありません。
- Sm\_PolicyApi\_InvalidHandle。 このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoPrivilege。 管理者には、レルムを取得する権限がありません。
- Sm\_PolicyApi\_InvalidOid。 レルム OID が見つかりませんでした。

## Sm\_PolicyApi\_GetRealmByName()

*pszRealmName*、およびレルムが存在するドメインの対応する *pszDomainOrParentRealmOid*、または子レルムの場合は親レルムの *Oid* によって識別されるレルムの内容を取得します。

この関数の結果が *ppstructRealm* によって参照される構造で返されます。 *Sm\_PolicyApi\_FreeMemoryEx()* を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

レルム関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetRealmByName (
    void*                pSessionHandle,
    const char*          pszRealmOid,
    Sm_PolicyApi_Realm_t** ppstructRealm
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。 この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

パラメータ	I/O	説明
<code>szDomainOrParentRealmOid</code>	I	既存のドメインまたはレルムのオブジェクト識別子が含まれる NULL で終わる文字列
<code>pszRealmName</code>	I	既存のレルムの名前が含まれ、NULL で終わる文字列です。
<code>ppstructRealm</code>	O	レルム構造へのポインタのアドレス。

### 戻り値

- `Sm_PolicyApi_Success`。取得の操作が成功しました。
- `Sm_PolicyApi_Failure`。メモリは `Sm_PolicyApi_Realm_t` に対して割り当てられることができませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoPrivilege`。管理者には、レルムを取得する権限がありません。
- `Sm_PolicyApi_InvalidOid`。ドメインまたは親レルム `OID` が見つかりませんでした。
- `Sm_PolicyApi_NotFound`。レルム名が見つかりませんでした。

## Sm\_PolicyAPI\_GetRedirectURLFromSAML1xScheme()

SAML 1.x 認証方式からリダイレクト URL 設定を取得します。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetRedirectURLFromSAML1xScheme(  
    void*          pSessionHandle,  
    const char*    pszSchemeOid,  
    int            iTypeURL,  
    char**         URL,  
    int            &redirectMode  
);
```

### パラメータ

#### pSessionHandle

[in] 内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

#### pszSchemeOid

[in] 更新中の認証方式のオブジェクト識別子が含まれ、NULL で終わる文字列です。

#### iTypeUrl

[in] リダイレクト URL のタイプを指定する整数であり、Sm\_PolicyAPI\_SAML1\_STATUS\_REDIRECT\_URL\_TYPE\_t で以下のように定義されます。

- Sm\_PolicyAPI\_SAML1\_STATUS\_REDIRECT\_URL\_USER\_NOT\_FOUND\_TYPE = 0
- Sm\_PolicyAPI\_SAML1\_STATUS\_REDIRECT\_URL\_INVALID\_SSO = 1
- Sm\_PolicyAPI\_SAML1\_STATUS\_REDIRECT\_URL\_UNACCEPTABLE\_USER\_CREDENTIALS = 2

#### URL

[out] 認証方式からのリダイレクト URL へのポインタ

#### redirectMode

[out] 入力されたリダイレクトモードを指定する整数（302 データなしの場合は 0、Http-Post の場合は 1 のどちらか）。

### 戻り値

- Sm\_PolicyApi\_Success。操作は正常に完了しました。
- Sm\_PolicyApi\_Failure。動作が失敗しました。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_SchemeNotFound。OID に対応する認証方式が見つからなかったか、SAML 1.x 認証方式ではありませんでした。
- Sm\_PolicyApi\_InvalidOid。認証方式の OID は NULL です。

## Sm\_PolicyApi\_GetRegistrationScheme()

登録方式を取得します。

### タイプ

登録方式関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetRegistrationScheme (
    void*                pSessionHandle,
    const char*          pszRegistrationSchemeOid,
    Sm_PolicyApi_RegistrationScheme_t**
                        ppstructRegistrationScheme
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszRegistrationSchemeOid</i>	I	登録方式のオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>ppstruct</i> 登録方式	O	Sm_PolicyApi_RegistrationScheme_t へのポインタのアドレス。

### 戻り値

- Sm\_PolicyApi\_Success。 取得は正常に行われました。
- Sm\_PolicyApi\_Failure。 メモリは Sm\_PolicyApi\_RegistrationScheme\_t に割り当てられることができませんでした。
- Sm\_PolicyApi\_InvalidHandle。 このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。 有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。 管理者には、登録方式を取得する権限がありません。
- Sm\_PolicyApi\_InvalidOid。 登録方式 OID が見つかりませんでした。

## Sm\_PolicyApi\_GetRegistrationSchemeByName()

登録方式を取得します。

### タイプ

登録方式関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetRegistrationSchemeByName (
    void*                pSessionHandle,
    const char*          pszRegistrationSchemeName,
    Sm_PolicyApi_RegistrationScheme_t**
                        ppstructRegistrationScheme
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszRegistrationSchemeName</i>	I	登録方式の名前が含まれ、NULL で終わる文字列です。
<i>ppstruct 登録 方式</i>	O	Sm_PolicyApi_RegistrationScheme_t へのポインタのアドレス。

### 戻り値

- `Sm_PolicyApi_Success`。取得は正常に行われました。
- `Sm_PolicyApi_Failure`。メモリは `Sm_PolicyApi_RegistrationScheme_t` に割り当てられることができませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、登録方式を取得する権限がありません。
- `Sm_PolicyApi_NotFound`。登録方式名が見つかりませんでした。

## Sm\_PolicyApi\_GetRegularExpressions()

参照されるパスワード ポリシーに属する正規表現のリストを取得します。セッションのバージョンが `SM_POLICY_API_VERSION_6_0` に設定される場合にのみ実装されます。

### タイプ

正規表現関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetRegularExpressions (
    void* pSessionHandle,
    const char* pszPasswordPolicyOid,
    Sm_PolicyApi_RegularExpression_t** ppstructRegExpr
);
```

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<code>pszPasswordPolicyOid</code>	I	パスワードポリシーの OID。
<code>ppstructRegExpr</code>	O	参照されるパスワードポリシーに属する正規表現のリンク リストへのポインタ。

### 戻り値

- Sm\_PolicyApi\_Success。 正規表現リストは正常に取得されました。
- Sm\_PolicyApi\_Failure。 正規表現リストは正常に取得されませんでした。
- Sm\_PolicyApi\_NoSession。 有効な管理者セッションがありません。
- Sm\_PolicyApi\_InvalidHandle。 このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoPrivilege。 管理者には、正規表現を取得する権限がありません。
- Sm\_PolicyApi\_InvalidOid: パスワードポリシー OID が見つかりませんでした。

## Sm\_PolicyApi\_GetResponse()

*pszResponseOid* によって識別されるレスポンスの内容を取得します。

この関数の結果は、*ppstructResponse* によって参照される構造で返されます。 *Sm\_PolicyApi\_FreeMemoryEx()* を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

レスポンス関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetResponse (
    void*                pSessionHandle,
    const char*          pszResponseOid,
    Sm_PolicyApi_Response_t** ppstructResponse
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszResponseOid</i>	I	既存のレスポンスのオブジェクト識別子が含まれ、NULL で終わる文字列です。

---

パラメータ	I/O	説明
<i>ppstructResponse</i>	O	レスポンス構造へのポインタのアドレスです。

---

### 戻り値

- `Sm_PolicyApi_Success`。取得操作は正常に完了しました。
- `Sm_PolicyApi_Failure`。 `Sm_PolicyApi_Response_t` にメモリを割り当てるできませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoPrivilege`。管理者には、レスポンスを取得する権限がありません。
- `Sm_PolicyApi_InvalidOid`。レスポンス `OID` が見つかりませんでした。

## Sm\_PolicyApi\_GetResponseByName()

*pszResponseName*、およびレスポンスが存在するドメインの対応する *pszDomainOid* によって識別されるレスポンスの内容を取得します。

この関数の結果は、*ppstructResponse* によって参照される構造で返されます。 *Sm\_PolicyApi\_FreeMemoryEx()* を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

レスポンス関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetResponseByName (
    void*                pSessionHandle,
    const char*          szDomainOid,
    const char*          pszResponseName,
    Sm_PolicyApi_Response_t** ppstructResponse
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>szDomainOid</i>		既存のドメインのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszResponseName</i>	I	既存のレスポンスの名前が含まれ、NULL で終わる文字列です。
<i>ppstructResponse</i>	O	レスポンス構造へのポインタのアドレスです。

### 戻り値

- `Sm_PolicyApi_Success`。取得操作は正常に完了しました。
- `Sm_PolicyApi_Failure`。 `Sm_PolicyApi_Response_t` にメモリを割り当てることができませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoPrivilege`。管理者には、レスポンスを取得する権限がありません。
- `Sm_PolicyApi_InvalidOid`。ドメイン OID が見つかりませんでした。
- `Sm_PolicyApi_NotFound`。レスポンス名が見つかりませんでした。

## Sm\_PolicyApi\_GetResponseAttrs()

`pszResponseOid` によって識別されるレスポンスと関連付けられるレスポンス属性のリンク リストを取得します。

返されるリンク リストは `ppstructResponseAttr` 構造によって参照されます。`Sm_PolicyApi_FreeMemoryEx()` を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

レスポンス関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetResponseAttrs (
    void*                pSessionHandle,
    const char*          pszResponseOid,
    Sm_PolicyApi_ResponseAttr_t** ppstructResponseAttr
);
```

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

パラメータ	I/O	説明
<i>pszResponseOid</i>	I	既存のレスポンスのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>ppstructResponseAttr</i>	O	レスポンス属性構造へのポインタ。

### 戻り値

- **Sm\_PolicyApi\_Success**。 取得の操作が成功しました。
- **Sm\_PolicyApi\_Failure**。 メモリは **Sm\_PolicyApi\_ResponseAttr\_t** に割り当てられることができませんでした。
- **Sm\_PolicyApi\_NoSession**。 有効な管理者セッションがありません。
- **Sm\_PolicyApi\_InvalidHandle**。 このコールに先立つ有効な初期化はありませんでした。
- **Sm\_PolicyApi\_NoPrivilege**。 管理者には、レスポンス属性のリストを取得する権限がありません。
- **Sm\_PolicyApi\_InvalidOid**。 レスポンス OID が見つかりませんでした。

## Sm\_PolicyApi\_GetRule()

*pszRuleOid* によって識別されるルールの内容を取得します。

この結果が *ppstructRule* によって参照される構造で返されます。  
**Sm\_PolicyApi\_FreeMemoryEx()** を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

ルール関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetRule (
    void*                pSessionHandle,
    const char*          pszRuleOid,
    Sm_PolicyApi_Rule_t** ppstructRule
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszRuleOid</i>	I	既存のルールのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>ppstructRule</i>	O	Sm_PolicyApi_Rule_t 構造へのポインタのアドレス。

### 戻り値

- `Sm_PolicyApi_Success`。 取得の操作が成功しました。
- `Sm_PolicyApi_Failure`。 `Sm_PolicyApi_Rule_t` にメモリを割り当てること  
ができませんでした。
- `Sm_PolicyApi_InvalidHandle`。 このコールに先立つ有効な初期化はあり  
ませんでした。
- `Sm_PolicyApi_NoSession`。 有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。 管理者には、ルールを取得する権限があり  
ません。
- `Sm_PolicyApi_InvalidOid`。 ルール OID が見つかりませんでした。

## Sm\_PolicyApi\_GetRuleByName()

`pszRuleName` によって識別されるルールの内容を取得します。

この結果が `ppstructRule` によって参照される構造で返されます。  
`Sm_PolicyApi_FreeMemoryEx()` を呼び出すことによりこの構造に割り当て  
られたメモリを解放します。

### タイプ

ルール関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetRuleByName (
    void*                pSessionHandle,
    const char*          szRealmOid,
    const char*          pszRuleName,
    Sm_PolicyApi_Rule_t** ppstructRule
);
```

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

---

パラメータ	I/O	説明
<code>szRealmOid</code>	I	既存のレルムのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<code>pszRuleName</code>	I	既存のルールの名前が含まれ、NULL で終わる文字列です。
<code>ppstructRule</code>	O	<code>Sm_PolicyApi_Rule_t</code> 構造へのポインタのアドレス。

---

### 戻り値

- `Sm_PolicyApi_Success`。 取得の操作が成功しました。
- `Sm_PolicyApi_Failure`。 `Sm_PolicyApi_Rule_t` にメモリを割り当てることができませんでした。
- `Sm_PolicyApi_InvalidHandle`。 このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。 有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。 管理者には、ルールを取得する権限がありません。
- `Sm_PolicyApi_InvalidOid`。 レルム OID が見つかりませんでした。
- `Sm_PolicyApi_NotFound`。 ルール名が見つかりませんでした。

## Sm\_PolicyApi\_GetSAMLAffiliation()

ポリシーストアでそのオブジェクト識別子によって指定された SAML アフィリエーションを取得します。

### タイプ

SAML 2.0 設定関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetSAMLAffiliation
(
    void*                pSessionHandle,
    const char*          pszAffiliationOid,
    Sm_PolicyApi_SAMLAffiliation_t** ppstructAffiliation
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAffiliationOid</i>	I	既存の SAML アフィリエーションのポリシーストアオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>ppstruct</i> アフィリエーション	O	SAML アフィリエーション構造へのポインタのアドレス。

### 戻り値

- `Sm_PolicyApi_Success`。SAML アフィリエーションは正常に取得されました。
- `Sm_PolicyApi_Failure`。SAML アフィリエーションは正常に取得されませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、SAML アフィリエーションを取得する権限がありません。
- `Sm_PolicyApi_InvalidOID`。SAML アフィリエーション OID が見つかりませんでした。

## Sm\_PolicyApi\_GetSAMLaffiliationById()

ポリシーストア内のその固有のアフィリエーション識別子（URI）によって指定される SAML アフィリエーションを取得します。

### タイプ

SAML 2.0 設定関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetSAMLaffiliationById
(
    void*                pSessionHandle,
    const char*          pszAffiliationID,
    Sm_PolicyApi_SAMLAffiliation_t** ppstructAffiliation
);
```

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

パラメータ	I/O	説明
<i>pszAffiliationID</i>	I	既存の SAML アフィリエーションの固有のアフィリエーション識別子が含まれ、NULL で終わる文字列です。 アフィリエーション識別子は SAML_KEY_AFFILIATION_ID で指定されます。
<i>ppstruct</i> アフィリエーション	O	SAML アフィリエーション構造へのポインタのアドレス。

### 戻り値

- Sm\_PolicyApi\_Success。 SAML アフィリエーションは正常に取得されました。
- Sm\_PolicyApi\_Failure。 SAML アフィリエーションは正常に取得されませんでした。
- Sm\_PolicyApi\_InvalidHandle。 このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。 有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。 管理者には、SAML アフィリエーションを取得する権限がありません。

## Sm\_PolicyApi\_GetSAMLScheme()

SAML 2.0 認証方式および関連するアイデンティティプロバイダのメタデータプロパティに関する情報を取得します。

## タイプ

SAML 2.0 設定関数、グローバル範囲。

## 構文

```
int SM_EXTERN Sm_PolicyApi_GetSAMLScheme
(
    void*                                pHandle,
    const char*                          pszSchemeOid,
    Sm_PolicyApi_Scheme_t**              ppstructScheme,
    Sm_PolicyApi_SAMLProviderProp_t**    ppProps
);
```

パラメータ	I/O	説明
<i>pHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszSchemeOid</i>	I	既存の SAML 認証方式のオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>ppstructScheme</i>	O	SAML 認証方式構造へのポインタのアドレス。
<i>ppProps</i>	O	SAML 2.0 メタデータ プロパティ構造へのポインタのアドレス。 これらのプロパティの詳細については、SAML 2.0 認証方式プロパティを参照してください。

### 戻り値

- Sm\_PolicyApi\_Success。 認証方式は正常に取得されました。
- Sm\_PolicyApi\_Failure。 認証方式は正常に取得されませんでした。
- Sm\_PolicyApi\_InvalidHandle。 このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。 有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。 管理者には、認証方式を取得する権限がありません。
- Sm\_PolicyApi\_InvalidOID。 認証方式 OID が見つかりませんでした。

## Sm\_PolicyApi\_GetSAMLServiceProvider()

ポリシーストアでそのオブジェクト識別子によって指定されたサービスプロバイダを取得します。

### タイプ

SAML 2.0 設定関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetSAMLServiceProvider
(
    void*                pSessionHandle,
    const char*          pszProviderOid,
    Sm_PolicyApi_SAMLSP_t** ppstructSAMLSP
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszProviderOid</i>	I	既存のサービスプロバイダのポリシーストアオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>ppstructSAMLSP</i>	O	サービスプロバイダ構造へのポインタのアドレス。

### 戻り値

- `Sm_PolicyApi_Success`。 サービス プロバイダは正常に取得されました。
- `Sm_PolicyApi_Failure`。 サービス プロバイダは正常に取得されませんでした。
- `Sm_PolicyApi_InvalidHandle`。 このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。 有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。 管理者には、サービス プロバイダを取得する権限がありません。
- `Sm_PolicyApi_InvalidOID`。 サービス プロバイダ `OID` が見つかりませんでした。
- `Sm_PolicyApi_DomainNotAffiliate`。 ドメインはアフィリエイト ドメインではありません。

## Sm\_PolicyApi\_GetSAMLServiceProviderById()

ポリシー ストア内のその固有のプロバイダ ID によって指定されるサーバ プロバイダを取得します。

### タイプ

SAML 2.0 設定関数、ドメイン範囲

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetSAMLServiceProvider
(
    void*                pSessionHandle,
    const char*          pszProviderId,
    Sm_PolicyApi_SAMLSP_t** pstructSAMLSP
);
```

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

パラメータ	I/O	説明
<i>pszProviderId</i>	I	既存のサービス プロバイダの一意のプロバイダ識別子が含まれ、NULL で終わる文字列です。 プロバイダ識別子は SAML_KEY_SPID で指定されます。
<i>ppstructSAMLSP</i>	O	サービス プロバイダ構造へのポインタのアドレス。

### 戻り値

- Sm\_PolicyApi\_Success。 サービス プロバイダは正常に取得されました。
- Sm\_PolicyApi\_Failure。 サービス プロバイダは正常に取得されませんでした。
- Sm\_PolicyApi\_InvalidHandle。 このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。 有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。 管理者には、サービス プロバイダを取得する権限がありません。
- Sm\_PolicyApi\_InvalidOID。 サービス プロバイダ OID が見つかりませんでした。
- Sm\_PolicyApi\_DomainNotAffiliate。 ドメインはアフィリエイト ドメインではありません。

## Sm\_PolicyApi\_GetSAMLServiceProviderUsers()

指定されたサービス プロバイダと関連付けられたユーザディレクトリ エントリを取得します。

### タイプ

SAML 2.0 設定関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetSAMLServiceProviderUsers
(
    void*                pSessionHandle,
    const char*          pszProviderOid,
    const char*          pszUserDirOid,
    Sm_PolicyApi_User_t** ppStructUsers
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszProviderOid</i>	I	既存のサービス プロバイダのポリシーストア オブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszUserDirOid</i>	I	既存のユーザのポリシーストア オブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>ppStructUsers</i>	O	ユーザ構造へのポインタのアドレス。

### 戻り値

- Sm\_PolicyApi\_Success。 ユーザは正常に取得されました。
- Sm\_PolicyApi\_Failure。 ユーザは正常に取得されませんでした。
- Sm\_PolicyApi\_InvalidHandle。 このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。 有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。 管理者には、ユーザを取得する権限がありません。
- Sm\_PolicyApi\_InvalidOid。 アフィリエイト OID が見つかりませんでした。

## Sm\_PolicyApi\_GetScheme()

*pszSchemeOid* によって識別される認証方式の内容を取得します。 この関数の結果が *ppstructScheme* によって参照される構造で返されます。

`Sm_PolicyApi_FreeMemoryEx()` を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

認証方式関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetScheme (
    void*                pSessionHandle,
    const char*          pszSchemeOid,
    Sm_PolicyApi_Scheme_t** ppstructScheme
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。 この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszSchemeOid</i>	I	既存方式のオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>ppstructScheme</i>	O	方式構造へのポインタのアドレス。

### 戻り値

- `Sm_PolicyApi_Success`。取得の操作が成功しました。
- `Sm_PolicyApi_Failure`。メモリは `Sm_PolicyApi_Scheme_t` に割り当てられ  
ることができませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はあり  
ませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、方式を取得する権限がありま  
せん。
- `Sm_PolicyApi_InvalidOid`。方式 OID が見つかりませんでした。

## Sm\_PolicyApi\_GetSchemeByName()

`pszSchemeName` によって識別される認証方式の内容を取得します。この関数の結果が `ppstructScheme` によって参照される構造で返されます。

`Sm_PolicyApi_FreeMemoryEx()` を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

認証方式関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetSchemeByName (
    void*                pSessionHandle,
    const char*          pszSchemeName,
    Sm_PolicyApi_Scheme_t** ppstructScheme
);
```

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<code>pszSchemeName</code>	I	既存方式の名前が含まれ、NULL で終わる文字列です。

パラメータ	I/O	説明
<i>ppstructScheme</i>	O	方式構造へのポインタのアドレス。

### 戻り値

- `Sm_PolicyApi_Success`。 取得の操作が成功しました。
- `Sm_PolicyApi_Failure`。 メモリは `Sm_PolicyApi_Scheme_t` に割り当てられないことができませんでした。
- `Sm_PolicyApi_InvalidHandle`。 このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。 有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。 管理者には、方式を取得する権限がありません。
- `Sm_PolicyApi_NotFound`。 方式名が見つかりませんでした。

## Sm\_PolicyApi\_GetSharedSecretPolicy()

現在の `SharedSecretPolicy` を取得します。

現在の共有秘密キー ポリシー オブジェクトを返します。 そのようなオブジェクトが常に 1 つあります。 したがって、その `OID` を指定する必要はありません。

### タイプ

エージェントの設定

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetSharedSecretPolicy (
    void*                                pSessionHandle,
    Sm_PolicyApi_SharedSecretPolicy_t** ppstructSecretPolicy
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	現在のポリシー API セッションハンドル。

---

パラメータ	I/O	説明
<i>ppstructSecret</i> ポリシー	I/O	共有秘密キー ポリシー構造へのポインタのアドレス

---

#### 戻り値

- Sm\_PolicyApi\_Success。操作は成功しました。
- Sm\_PolicyApi\_Failure。一般的な障害です。
- Sm\_PolicyApi\_InvalidHandle。セッションハンドルは無効でした。
- Sm\_PolicyApi\_NoSession。ユーザはログインされません。
- Sm\_PolicyApi\_NoPrivilege。ユーザには、オブジェクトを管理する権限がありません。

## Sm\_PolicyApi\_GetTargetConfigFromSAML1xScheme

SAML 1.x 認証方式からターゲット設定を取得します。

### 構文

Sm\_PolicyApi\_GetTargetConfigFromSAML1xScheme 関数の構文は、以下のとおりです。

```
int SM_EXTERN Sm_PolicyApi_GetTargetConfigFromSAML1xScheme(  
    void*          pHandle,  
    const char*   pszSchemeOid,  
    char**        pszDefaultTarget,  
    int*          iQPOverridesTarget  
);
```

### パラメータ

Sm\_PolicyApi\_GetTargetConfigFromSAML1xScheme 関数は、以下のパラメータを受け取ります。

#### pSessionHandle

[in] 内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

#### pszSchemeOid

[in] 更新中の認証方式のオブジェクト識別子が含まれ、NULL で終わる文字列です。

#### pszDefaultTarget

[out] ターゲット設定を指定します。呼び出しプログラムは Sm\_PolicyApi\_FreeMemory() 関数を呼び出すことによりこのメモリを解放する必要があります。

#### iQPOverridesTarget

[out] [クエリ パラメータでデフォルト ターゲットをオーバーライドする] チェック ボックスの値を指定します。呼び出しプログラムは、割り当てられたメモリのパスに関与します。

### 戻り値

Sm\_PolicyApi\_GetTargetConfigFromSAML1xScheme 関数は、以下の値のいずれかを返します。

- Sm\_PolicyApi\_Success。操作は正常に完了しました。
- Sm\_PolicyApi\_Failure。動作が失敗しました。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_SchemeNotFound。OID に対応する認証方式が見つからなかったか、SAML 1.x 認証方式ではありませんでした。
- Sm\_PolicyApi\_InvalidOid。認証方式の OID は NULL です。

## Sm\_PolicyApi\_GetTrustedHost()

既存のトラステッドホストオブジェクトを取得します。

### タイプ

エージェント設定関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetTrustedHost (
    void*                pSessionHandle,
    const char*          pszTrustedHostOid,
    Sm_PolicyApi_TrustedHost_t** ppstructTrustedHost
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszTrustedHostOid</i>	I	取得するトラステッドホストオブジェクトの一意的識別子。
<i>ppstructTrustedHost</i>	O	トラステッドホストオブジェクトを定義する構造体へのポインタのアドレス。関数によって構造体が割り当てられます。

### 戻り値

- Sm\_PolicyApi\_Success。 取得の操作が成功しました。
- Sm\_PolicyApi\_Failure。 一般的な障害です。
- Sm\_PolicyApi\_InvalidOid。 トラストド ホスト オブジェクトに一意の ID が対応しません。
- Sm\_PolicyApi\_NoPrivilege。 呼び出し元には、トラストド ホスト オブジェクトを取得する権限がありません。
- Sm\_PolicyApi\_InvalidHandle。 セッション ポインタが無効です。
- Sm\_PolicyApi\_NoSession。 API ユーザが正しくログインしていません。

## Sm\_PolicyApi\_GetTrustedHostByName()

既存のトラストド ホスト オブジェクトを取得します。

### タイプ

エージェント設定関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetTrustedHostByName (
    void*                pSessionHandle,
    const char*          pszTrustedHostName,
    Sm_PolicyApi_TrustedHost_t** ppstructTrustedHost
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszTrustedHostName</i>	I	取得するトラストド ホスト オブジェクトの一意の名前。
<i>ppstructTrustedHost</i>	O	トラストド ホスト オブジェクトを定義する構造体へのポインタのアドレス。関数によって構造が割り当てられます。

### 戻り値

- Sm\_PolicyApi\_Success。 取得の操作が成功しました。
- Sm\_PolicyApi\_Failure。 一般的な障害です。
- Sm\_PolicyApi\_NotFound。 一意の名前はトラステッド ホスト オブジェクトに相当しません。
- Sm\_PolicyApi\_NoPrivilege。 呼び出し元には、トラステッド ホスト オブジェクトを取得する権限がありません。
- Sm\_PolicyApi\_InvalidHandle。 セッション ポインタが無効です。
- Sm\_PolicyApi\_NoSession。 API ユーザが正しくログインしていません。

## Sm\_PolicyApi\_GetUseSecureAuthPropFromAffiliate()

SAML 1.x アフィリエイトから UseSecureAuthURL プロパティの値を取得します。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetUseSecureAuthPropFromAffiliate(  
    void*    pHandle,  
    char*    pszAffiliateOID,  
    bool     &bUseSecureAuthURL  
);
```

### パラメータ

pHandle

[in] 内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

pszAffiliateOid

[in] 既存の SAML 1.x アフィリエイトの OID へのポインタです。

bUseSecureAuthURL

[in] このアフィリエイトに対し安全な認証 URL を使用するかどうかを指定するブール値です。

### 戻り値

- Sm\_PolicyApi\_Success。操作は正常に完了しました。
- Sm\_PolicyApi\_Failure。動作が失敗しました。
- Sm\_PolicyApi\_NoSession。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoPrivilege。管理者に十分なアクセス権がありませんでした。

## Sm\_PolicyApi\_GetUserContext()

ポリシー管理 API の呼び出し元によるユーザ コンテキスト情報へのアクセスを可能にします。

### タイプ

ユーザ ディレクトリ関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetUserContext (
    void                *pSessionHandle,
    const char          *lpszUserDirOid,
    const char          *lpszUserPath,
    const char          *lpszSessionID,
    Sm_PolicyApi_UserContext_t **ppPolicyApiUserContext);
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>lpszUserDirOid</i>	I	<i>lpszUserPath</i> で指定したユーザのユーザ ディレクトリのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>lpszUserPath</i>	I	ユーザの識別名。
<i>lpszSessionID</i>	I	セッションの一意の識別子。 ログインに成功した後、セッション ID が <b>Sm_AgentApi_Session_t</b> 構造体の <i>lpszSessionId</i> フィールドに返されます。 セッション ID が認識されない場合は、このパラメータに空の文字列 ("") を割り当てます。
<i>ppPolicyApiUserContext</i>	O	SiteMinder が関数へ渡すユーザ コンテキスト情報。

### 注釈

この関数は、アプリケーションが基礎となるユーザディレクトリに接続する必要なしに、ユーザに関する情報にアクセスすることを可能にします。ユーザ情報を取得するには、アプリケーションは、`Sm_Api_UserContext_t` 構造で関数を呼び出し、それは `ppPolicyApiUserContext` で返されます。以下は例です。

- ユーザ属性を取得または設定するには `fGetProp` or `fSetProp` を呼び出します。
- ユーザパスワードを確認するには `fAuthenticate` を呼び出します。
- ディレクトリを検索し結果を検討し、正しいユーザが見つかったことを確認するには `fGetProp` を呼び出します。
- ログインユーザでないユーザの属性を取得するには `fGetProp` を呼び出します。

**重要:** `Sm_GetUserContext()` を呼び出した後、`Sm_PolicyApi_FreeMemoryEx()` を呼び出して割り当てられたメモリを解放します。 そうしないと、一部のポリシー サーバのパフォーマンスが低下します。

### 戻り値

- `Sm_PolicyApi_Success`。 取得の操作が成功しました。
- `Sm_PolicyApi_Failure`。 取得の操作が成功しませんでした。
- `Sm_PolicyApi_NoSession`。 有効な管理者セッションがありません。
- `Sm_PolicyApi_Invalid`。 指定されたユーザまたはディレクトリは有効ではありませんでした。
- `Sm_PolicyApi_InvalidHandle`。 このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoPrivilege`。 管理者には、ユーザディレクトリを取得する権限がありません。

## Sm\_PolicyApi\_GetUserDir()

*pszUserDirOid* によって識別されるユーザディレクトリの内容を取得します。

この関数の結果が *ppstructUserDir* によって参照される構造で返されます。  
*Sm\_PolicyApi\_FreeMemoryEx()* を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

ユーザディレクトリ関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetUserDir (
    void*                pSessionHandle,
    const char*          pszUserDirOid,
    Sm_PolicyApi_UserDir_t** ppstructUserDir
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszUserDirOid</i>	I	既存のユーザディレクトリのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>ppstructUserDir</i>	O	ユーザディレクトリ構造へのポインタのアドレス。

### 戻り値

- `Sm_PolicyApi_Success`。取得の操作が成功しました。
- `Sm_PolicyApi_Failure`。メモリは `Sm_PolicyApi_UserDir_t` に対して割り当てられることができませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoPrivilege`。管理者には、ユーザディレクトリを取得する権限がありません。
- `Sm_PolicyApi_InvalidOid`。ユーザディレクトリ `OID` が見つかりませんでした。

## Sm\_PolicyApi\_GetUserDirByName()

`pszUserDirName` によって識別されるユーザディレクトリの内容を取得します

この関数の結果が `ppstructUserDir` によって参照される構造で返されます。`Sm_PolicyApi_FreeMemoryEx()` を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

ユーザディレクトリ関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetUserDirByName (  
    void*                pSessionHandle,  
    const char*          pszUserDirName,  
    Sm_PolicyApi_UserDir_t** ppstructUserDir  
);
```

---

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

---

---

パラメータ	I/O	説明
<i>pszUserDirName</i>	I	既存ユーザディレクトリの名前が含まれ、NULL で終わる文字列です。
<i>ppstructUserDir</i>	O	ユーザディレクトリ構造へのポインタのアドレス。

---

### 戻り値

- **Sm\_PolicyApi\_Success**。取得の操作が成功しました。
- **Sm\_PolicyApi\_Failure**。メモリは **Sm\_PolicyApi\_UserDir\_t** に対して割り当てられることができませんでした。
- **Sm\_PolicyApi\_NoSession**。有効な管理者セッションがありません。
- **Sm\_PolicyApi\_InvalidHandle**。このコールに先立つ有効な初期化はありませんでした。
- **Sm\_PolicyApi\_NoPrivilege**。管理者には、ユーザディレクトリを取得する権限がありません。
- **Sm\_PolicyApi\_NotFound**。ユーザディレクトリ名が見つかりませんでした。

## Sm\_PolicyApi\_GetUserDirCapabilities()

ユーザ ディレクトリ機能を取得します。

ユーザ ディレクトリ機能は SmApi.h 内の Sm\_DirectoryCapability\_t に列挙されます。

### タイプ

ユーザ ディレクトリ関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetUserDirCapabilities (
    void*          pSessionHandle,
    const char*    pszUserDirOid,
    unsigned long* pCapabilities
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszUserDirOid</i>	I	検索を実行する必要があるユーザ ディレクトリのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pCapabilities</i>	O	ディレクトリ機能に関する情報を保持する unsigned long へのポインタのアドレス ディレクトリ機能は、ヘッダファイル SmApi.h で定義される Sm_DirectoryCapability_t に列挙されます。103 ページの図 12 を参照してください。

### 戻り値

- `Sm_PolicyApi_Success`。取得は成功しました。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoPrivilege`。管理者には、ユーザディレクトリ機能を取得する権限がありません。
- `Sm_PolicyApi_InvalidOid`。ユーザディレクトリ OID が見つかりませんでした。

## Sm\_PolicyApi\_GetUserDirSearchOrder()

`pszDomainOid` によって識別されるドメインと関連付けられるユーザディレクトリ オブジェクトの OID を取得します。

OID の取得されたリストは、`SiteMinder` によって検索された順番で `pszArray` 文字列配列に保存されます。`Sm_PolicyApi_FreeStringArray()` を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

ユーザディレクトリ関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetUserDirSearchOrder (
    void*      pSessionHandle,
    const char* pszDomainOid,
    char**     pszArray[]
);
```

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<code>pszDomainOid</code>	I	既存のドメインのオブジェクト識別子が含まれ、NULL で終わる文字列です。

パラメータ	I/O	説明
<i>pszArray</i>	O	要求されたオブジェクトのユーザディレクトリ OID の返された配列です。

### 戻り値

- `Sm_PolicyApi_Success`。関数は正常にユーザディレクトリ検索順序を返しました。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoPrivilege`。管理者には、ドメインのユーザディレクトリ検索順序を取得する権限がありません。
- `Sm_PolicyApi_InvalidOid`。ドメイン OID が見つかりませんでした。

## Sm\_PolicyApi\_GetUserGroups()

ユーザがメンバであるグループのリストを取得します。

### タイプ

ユーザとユーザ状態関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetUserGroups (
    void *          pSessionHandle,
    const char *    pszUserDirOid,
    const char *    pszUserDN,
    const bool      bRecursive,
    char**          pszGroups[]
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszUserDirOid</i>	I	ユーザのオブジェクト識別子。

パラメータ	I/O	説明
<i>pszUserDN</i>	I	ユーザの識別名。
<i>bRecursive</i>	I	検索するのは 1 レベルのみかまたはすべてのレベルかを示す値。真を指定すると、すべてのレベルが検索されます。
<i>pszGroups</i>	O	ユーザが属するグループが含まれる配列。 <b>Sm_PolicyApi_FreeStringArray()</b> を呼び出すことにより配列に割り当てられたメモリを解放します。

### 戻り値

- **Sm\_PolicyApi\_Success**。 取得の操作が成功しました。
- **Sm\_PolicyApi\_Failure**。 取得の操作が成功しませんでした。
- **Sm\_PolicyApi\_Invalid**。 指定されたユーザ **OID** および **DN** の有効なディレクトリはありませんでした。
- **Sm\_PolicyApi\_InvalidHandle**。 このコールに先立つ有効な初期化はありませんでした。
- **Sm\_PolicyApi\_InvalidOid**。 ディレクトリ **OID** が見つかりませんでした (ディレクトリ ベースのグループについて)。
- **Sm\_PolicyApi\_NoSession**。 有効な管理者セッションがありません。
- **Sm\_PolicyApi\_NoPrivilege**。 管理者には、グループを取得する権限がありません。

## Sm\_PolicyApi\_GetUserPasswordState()

Sm\_Api\_UserPasswordState\_t を返します。 Sm\_Api\_UserPasswordState\_t オブジェクトのメモリは、関数の呼び出しによって割り当てられる必要があります。

### タイプ

パスワード状態関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetUserPasswordState (
    void      *pSessionHandle,
    const char *pszUserDirOid,
    const char *pszUserDN,
    Sm_PolicyApi_UserPasswordState_t *pPasswordState
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszUserDirOid</i>	I	特定のユーザディレクトリに相当する一意のオブジェクト識別子。
<i>pszUserDN</i>	I	ユーザディレクトリ内のユーザの DN を指定します。
<i>pPasswordState</i>	O	ユーザパスワード状態オブジェクト。

### 戻り値

- `Sm_PolicyApi_Success`。 取得の操作が成功しました。
- `Sm_PolicyApi_Failure`。 一般的な障害です。
- `Sm_PolicyApi_InvalidHandle`。 セッションポインタが無効です。
- `Sm_PolicyApi_BadArgument`。 入力パスワード状態ポインタは `NULL` です。
- `Sm_PolicyApi_NoSession`。 API ユーザが正しくログインしていません。
- `Sm_PolicyApi_InvalidOid`。 一意の ID はユーザディレクトリオブジェクトに相当しません。
- `Sm_PolicyApi_UserDirNotFound`。 ポリシーストアからのユーザディレクトリオブジェクトの取得が失敗しました。
- `Sm_PolicyApi_ErrorUserDir`。 無効なユーザディレクトリオブジェクトまたは間違ったユーザ DN。
- `Sm_PolicyApi_NoPrivilege`。 呼び出し元には適切な権限がありません。

## Sm\_PolicyApi\_GetUsersFromWSFEDResourcePartner()

WS-Federation リソース パートナーと関連付けられるユーザ ディレクトリ エントリを取得します。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetUsersFromWSFEDResourcePartner (  
    void* pSessionHandle,  
    const char * pszProviderOid,  
    const char * pszUserPolicyOid,  
    Sm_PolicyApi_User_t ** ppStructUsers  
);
```

### パラメータ

pSessionHandle

[in] 内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

pszProviderOid

[in] 既存の WS フェデレーション リソース パートナーのオブジェクト識別子が含まれ、NULL で終わる文字列です。

pszUserPolicyOid

[in] 既存のポリシー ユーザのオブジェクト識別子が含まれ、NULL で終わる文字列です。

ppStructUsers

[out] ユーザ構造のリンク リストへのポインタのアドレス。

### 戻り値

- Sm\_PolicyApi\_Success。ユーザ ディレクトリは正常に追加されました。
- Sm\_PolicyApi\_Failure。ユーザ ディレクトリは正常に追加されませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、ユーザを取得する権限がありません。
- Sm\_PolicyApi\_InvalidOid。アフィリエイト OID が見つかりませんでした。

## Sm\_PolicyApi\_GetVariable()

指定された変数を取得します。

この関数の結果が *ppstructVariable* によって参照される構造で返されます。  
Sm\_PolicyApi\_FreeMemoryEx() を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

変数関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetVariable (
    void*                pSessionHandle,
    const char*          pszVariableOid,
    Sm_PolicyApi_Variable_t** ppstructVariable
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszVariableOid</i>	I	変数に相当する一意のオブジェクト識別子。
<i>ppstructVariable</i>	O	変数構造へのポインタのアドレス。

### 戻り値

- `Sm_PolicyApi_Success`。取得は正常に行われました。
- `Sm_PolicyApi_Failure`。メモリは `Sm_PolicyApi_Variable_t` に割り当てられないことができませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、変数を取得する権限がありません。
- `Sm_PolicyApi_InvalidOid`。変数のオブジェクト識別子が見つかりませんでした。
- `Sm_PolicyApi_FeatureNotSupported`。この関数を呼び出したクライアントが、`SM_POLICY_API_VERSION_6_0` より以前のバージョンで API を初期化しました。

## Sm\_PolicyApi\_GetVariableByName()

指定された変数を取得します。

この関数の結果が *ppstructVariable* によって参照される構造で返されます。  
*Sm\_PolicyApi\_FreeMemoryEx()* を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

変数関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetVariableByName (  
    void*                pSessionHandle,  
    const char*          pszVariableName,  
    Sm_PolicyApi_Variable_t** ppstructVariable  
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszDomainOid</i>	I	既存のドメインのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszVariableName</i>	I	変数に相当する一意の名前。
<i>ppstructVariable</i>	O	変数構造へのポインタのアドレス。

### 戻り値

- `Sm_PolicyApi_Success`。取得は正常に行われました。
- `Sm_PolicyApi_Failure`。メモリは `Sm_PolicyApi_Variable_t` に割り当てられないことができませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、変数を取得する権限がありません。
- `Sm_PolicyApi_InvalidOid`。ドメイン OID が見つかりませんでした。
- `Sm_PolicyApi_NotFound`。変数名が見つかりませんでした。
- `Sm_PolicyApi_FeatureNotSupported`。この関数を呼び出したクライアントが、`SM_POLICY_API_VERSION_6_0` より以前のバージョンで API を初期化しました。

## Sm\_PolicyApi\_GetVariableType()

指定された変数タイプを取得します。

この関数の結果が *ppstructVariable* によって参照される構造で返されます。  
*Sm\_PolicyApi\_FreeMemoryEx()* を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

変数関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetVariableType (  
    void*                pSessionHandle,  
    const char*          pszVariableTypeOid,  
    Sm_PolicyApi_VariableType_t** ppstructVariableType  
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszVariableOid</i>	I	変数のタイプに対応した一意のオブジェクト識別子です。
<i>ppstructVariable</i>	O	変数構造へのポインタのアドレス。

### 戻り値

- `Sm_PolicyApi_Success`。取得は正常に行われました。
- `Sm_PolicyApi_Failure`。メモリは `Sm_PolicyApi_Variable_t` に割り当てられないことができませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、変数タイプを取得する権限がありません。
- `Sm_PolicyApi_InvalidOid`。変数タイプ オブジェクト識別子が見つかりませんでした。
- `Sm_PolicyApi_FeatureNotSupported`。この関数を呼び出したクライアントが、`SM_POLICY_API_VERSION_6_0` より以前のバージョンで API を初期化しました。

## Sm\_PolicyApi\_GetVariableTypeByName()

指定された変数タイプを取得します。

この関数の結果が *ppstructVariable* によって参照される構造で返されます。  
*Sm\_PolicyApi\_FreeMemoryEx()* を呼び出すことによりこの構造に割り当てられたメモリを解放します。

### タイプ

変数関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetVariableTypeByName (  
    void*                pSessionHandle,  
    const char*          pszVariableTypeName,  
    Sm_PolicyApi_VariableType_t** ppstructVariableType  
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszVariableName</i>	I	変数タイプに相当する一意の名前。
<i>ppstructVariable</i>	O	変数構造へのポインタのアドレス。

### 戻り値

- `Sm_PolicyApi_Success`。取得は正常に行われました。
- `Sm_PolicyApi_Failure`。メモリは `Sm_PolicyApi_Variable_t` に割り当てられないことができませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、変数タイプを取得する権限がありません。
- `Sm_PolicyApi_NotFound`。変数タイプ オブジェクト識別子が見つかりませんでした。
- `Sm_PolicyApi_FeatureNotSupported`。この関数を呼び出したクライアントが、`SM_POLICY_API_VERSION_6_0` より以前のバージョンで API を初期化しました。

## Sm\_PolicyApi\_GetWSFEDResourcePartner()

既存のリソース パートナー オブジェクトを取得します。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetWSFEDResourcePartner (
    void* pSessionHandle,
    const char * pszProviderOid;
    Sm_PolicyApi_WSFEDResourcePartner_t** pstructServiceProvider
);
```

### パラメータ

#### pSessionHandle

[in] 内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

#### pszProviderOid

[in] 既存の WS フェデレーション リソース パートナーのオブジェクト識別子が含まれ、NULL で終わる文字列です。

#### pstructServiceProvider

[out] WS フェデレーション リソース パートナー構造へのポインタのアドレス。

### 戻り値

- Sm\_PolicyApi\_Success。リソース パートナーが正常に取得されました。
- Sm\_PolicyApi\_Failure。リソース パートナーが正常に取得されませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、アフィリエイトを取得する権限がありません。
- Sm\_PolicyApi\_InvalidOID。リソース パートナー OID が見つかりませんでした。
- Sm\_PolicyApi\_DomainNotAffiliate。ドメインはアフィリエイト ドメインではありません



## Sm\_PolicyApi\_GetWSFEDScheme()

既存の WSFED 認証方式オブジェクトを取得します。

### 構文

```
int SM_EXTERN Sm_PolicyApi_GetWSFEDScheme (  
    void* pSessionHandle,  
    const char * pszProviderOid,  
    Sm_PolicyApi_Scheme_t** ppstructScheme,  
    Sm_PolicyApi_WSFEDProviderProp_t** ppProps  
);
```

### パラメータ

#### pSessionHandle

[in] 内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

#### pszProviderOid

[in] 既存の WSFED 認証方式のオブジェクト識別子が含まれ、NULL で終わる文字列です。

#### ppstructScheme

[out] SiteMinder 認証方式構造へのポインタのアドレス。

#### ppProps

[out] リンク リスト WSFED プロバイダ プロパティへのポインタのアドレス。

### 戻り値

- Sm\_PolicyApi\_Success。認証方式は正常に取得されました。
- Sm\_PolicyApi\_Failure。認証方式は正常に取得されませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、アフィリエイトを取得する権限がありません。
- Sm\_PolicyApi\_InvalidOID。プロバイダ OID が見つかりませんでした。

## Sm\_PolicyApi\_Init()

SiteMinder ポリシー ストアへの接続を初期化し、*init* ハンドルを確立します。

API 動作に影響する初期化フラグを指定できます。

この関数は、API クライアントセッションにつき 1 回呼び出される必要があります。それはセッションに呼び出される最初の関数である必要があります。

### タイプ

必要な関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_Init (
    void**                ppInitHandle,
    const Sm_PolicyApi_InitFlags_t  nInitFlag
);
```

パラメータ	I/O	説明
<i>ppInitHandle</i>	O	クライアントセッション情報が含まれる内部ポリシー管理 API データ構造へのポインタ。この <i>init</i> ハンドルが成功した初期化上で返され、 <i>Sm_PolicyApi_Login()</i> および <i>Sm_PolicyApi_Release()</i> へのすべてのコールへの入力パラメータとして使用されます。
<i>nInitFlag</i>	I	API の動作に影響する値。

### 戻り値

- *Sm\_PolicyApi\_Success*。ポリシー ストア接続の初期化は成功しました。
- *Sm\_PolicyApi\_Failure*。ポリシー ストア接続の初期化が失敗しました。または、メモリが割り当てられませんでした。

## Sm\_PolicyApi\_InitEx()

SiteMinder ポリシー ストアへの接続を初期化し、指定のバージョンに基づいて init ハンドルを確立します。

### タイプ

必要な関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_InitEx (
    void**                ppInitHandle,
    const Sm_PolicyApi_InitFlags_t nInitFlag,
    const unsigned        version
)
```

パラメータ	I/O	説明
<i>ppInitHandle</i>	O	クライアントセッション情報が含まれる内部ポリシー管理 API データ構造へのポインタ。
<i>nInitFlag</i>	I	API の動作に影響する値。
<i>version</i>	I	初期化するポリシー管理 API のバージョン。

### 戻り値

- `Sm_PolicyApi_Success`。ポリシー ストア接続の初期化は成功しました。
- `Sm_PolicyApi_Failure`。ポリシー ストア接続の初期化が失敗しました。または、メモリが割り当てられませんでした。

## Sm\_PolicyApi\_IsGroup()

アイテムがグループかどうか決定します。

### タイプ

グループ関数、グローバル範囲（エージェント）またはドメイン範囲（レスポンス、ルール）。

### 構文

```
int SM_EXTERN Sm_PolicyApi_IsGroup (
    void*                pSessionHandle,
    Sm_PolicyApi_Groups_t dwGroup,
    const char*          pszOid,
    const char*          pszDomainOid,
    bool*                bIsGroup
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>dwGroup</i>	I	オブジェクトのタイプを示します。
<i>pszOid</i>	I	確認するアイテムのオブジェクト識別子が含まれ、NULLで終わる文字列です。
<i>pszDomainOid</i>	I	既存のドメインのオブジェクト識別子が含まれ、NULLで終わる文字列です。
<i>bIsGroup</i>	O	ブール値へのポインタ。 <i>pszOid</i> オブジェクトがグループである場合、真です。

### 戻り値

- `Sm_PolicyApi_Success`。OID はグループです。
- `Sm_PolicyApi_Failure`。OID はグループではありません。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、ルール、レスポンスまたはエージェント OID がグループであるか確認する権限がありません。
- `Sm_PolicyApi_InvalidOid` :
  - ドメイン OID が見つかりませんでした（ドメインベースのグループについて）。
  - グループ OID が見つかりませんでした。
  - ルール、レスポンス、またはエージェント OID またはグループ OID が見つかりませんでした。
- `Sm_PolicyApi_BadGroup`。パラメータ `dwGroup` はルール、レスポンスまたはエージェント タイプではありません。

## Sm\_PolicyApi\_Login()

初期化の後、`Sm_PolicyApi_Login()` へのコールを成功させることが、追加の関数コールを行うための前提条件です。この関数は、管理者のログイン認証情報（ユーザ名とパスワード）を確認します。API が初期化されていない、または不適当に初期化された `init` ハンドルを検出すると、エラーが生成されます。管理者が認証されると、関数は内部データ構造体とリソースを初期化します。いったん管理者がログインされると、ポリシーサーバはセッションハンドルを初期化します。このハンドルは、すべてのポリシー管理 API 関数への入力パラメータとして使用されます。

内部では、セッションハンドルには、（`init` ハンドルからの）クライアントセッションデータおよび管理者セッションデータを含む、ポリシー管理 API の操作に必要なデータ構造およびコンテキスト情報が含まれます。データ構造およびコンテキスト情報は呼び出し元には見えません。

`Sm_PolicyApi_Login()` を呼び出して、管理者の認証情報を確認したり管理者を指定することなく、セッションハンドルを初期化することができます。これらの状況のいずれかでログインするには、パラメータ `nCheckCreds` を「false」に設定します。詳細については、`nCheckCreds` パラメータの説明を参照してください。

この関数はクライアントセッション中に 2 回以上コールでき、ポリシーストア接続の成功した初期化に応じます。

### タイプ

必要な関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_Login (
    void*          pInitHandle,
    void**         ppSessionHandle,
    int            nCheckCreds,
    const char*    pszUserName,
    const char*    pszPassword,
    const char*    pszClientIP,
    char**         pszUserMsg,
    char**         pszErrMsg
);
```

---

パラメータ	I/O	説明
-------	-----	----

---

パラメータ	I/O	説明
<i>pInitHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。これは <code>Sm_PolicyApi_Init()</code> によって返された <code>init</code> ハンドルです。
<i>ppSessionHandle</i>	O	内部ポリシー管理 API データ構造体 ( <i>pInitHandle</i> からの構造体の別のインスタンス) へのポインタ。構造体には管理者セッションデータが含まれます。セッションハンドルは呼び出し元には見えません。
<i>nCheckCreds</i>	I	<p>以下のように管理者の認証情報を確認するかどうかを示すフラグ</p> <ul style="list-style-type: none"> <li>■ <i>nCheckCreds</i> が <code>false</code> で、空の文字列 ("" ) が <i>pszUserName</i> で渡される場合、認証情報の確認はされておらず、呼び出し元には管理者権限がすべて付与されます。</li> <li>■ <i>nCheckCreds</i> が <code>false</code> で、ユーザ名が <i>pszUserName</i> で渡される場合、認証情報の確認は行われていません。ただし、渡される名前は有効な SiteMinder 管理者の名前である必要があります。呼び出し元には、管理者に対して定義された権限が付与されます。</li> <li>■ <i>nCheckCreds</i> が真である場合、ユーザ名とパスワードの両方が確認され、有効な SiteMinder 管理者に一致する必要があります。呼び出し元には、管理者に対して定義された権限が付与されます。</li> </ul>
<i>pszUserName</i>	I	<p>ポリシー管理 API 管理者のユーザ名。</p> <p>空の文字列 ("" ) で渡し、<i>nCheckCreds</i> を「<code>false</code>」に設定した場合、管理者名およびパスワードは必要ありません。呼び出し元には管理者権限がすべて付与されます。</p>
<i>pszPassword</i>	I	ポリシー管理 API 管理者のパスワード。
<i>pszClientIP</i>	I	管理者がログインしている元のマシンの IP アドレス。
<i>pszUserMsg</i>	O	ポリシー管理 API によって返されたユーザメッセージです。 <code>Sm_PolicyApi_FreeString()</code> を呼び出して、この変数に割り当てられたメモリを解放します。

パラメータ	I/O	説明
<i>pszErrMsg</i>	O	ポリシー管理 API によって返されたエラー メッセージです。 <code>Sm_PolicyApi_FreeString()</code> を呼び出して、この変数に割り当てられたメモリを解放します。

### 戻り値

- `Sm_PolicyApi_Success`。 ログインは正常に行われました。
- `Sm_PolicyApi_Failure`。 メモリは割り当てることができません。
- `Sm_PolicyApi_InvalidHandle`。 このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoPrivilege`。 管理者はいくつかまたはすべてのドメインに対して権限がありません。
- `Sm_PolicyApi_ErrorLogin`。 その認証は失敗しました。

### 例

例のアプリケーション `smpolicyapiexample.cpp` で `Sm_PolicyApi_Login()` コールを参照してください。

## Sm\_PolicyApi\_Logout()

管理者セッションからログアウトします。

### タイプ

必要な関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_Logout (
    void* pSessionHandle
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	管理者のログインが成功した後に <code>Sm_PolicyApi_Login()</code> によって返されたセッションハンドル。

### 戻り値

- `Sm_PolicyApi_Success`。 ログアウトは正常に行われました。
- `Sm_PolicyApi_InvalidHandle`。 このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。 有効な管理者セッションがありません。

## Sm\_PolicyApi\_LookupDirectoryEntry()

ユーザディレクトリ内のユーザ仕様を調べます。

`pszSearchPattern` は、検索の検索パターンを保持します。 ユーザディレクトリ検索はユーザディレクトリネームスペースのタイプごとに異なります。

### タイプ

ユーザディレクトリ関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_LookupDirectoryEntry (
    void*                pSessionHandle,
    const char*          pszUserDirOid,
    const char*          pszSearchPattern,
    Sm_PolicyApi_User_t** ppStructUserSpec
);
```

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。 この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<code>pszUserDirOid</code>	I	検索が実行されるユーザディレクトリのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<code>pszSearchPattern</code>	I	検索の検索パターンが含まれ、NULL で終わる文字列です。 異なるネームスペースタイプの検索式文法に関する情報が以下のセクションに表示されます。
<code>ppStructUserSpec</code>	O	ユーザ構造へのポインタのアドレスです。

## LDAP ネームスペースの検索式文法

LDAP ユーザディレクトリの検索は、属性値ペアまたは LDAP 検索式に基づいて行うことができます。

- 属性値ペアの場合、*pszSearchPattern* の形式は次のとおりです。

```
<attribute-name>=<value>
```

たとえば、*<attribute-name>* が無効で、*<value>* が 0 である場合、LDAP 検索フィルタは次のとおりです。

```
Base:'o=airius.com',
```

```
Filter:
```

```
'(&
  (|(objectclass=organizationalPerson)
    (objectclass=inetOrgPerson)
    (objectclass=organization)
    (objectclass=organizationalUnit)
    (objectclass=groupOfNames)
    (objectclass=groupOfUniqueNames)
    (objectclass=group)
  )
  (disabled=0)
)'
```

- 検索が LDAP 検索式を使用する場合、*pszSearchPattern* は LDAP 検索式を保持します。たとえば、検索式が「uid=user1111」である場合、LDAP 検索フィルタは次のとおりです。

```
Base:'o=airius.com',
```

```
Filter:
```

```
'(&
  (|(objectclass=organizationalPerson)
    (objectclass=inetOrgPerson)
    (objectclass=organization)
    (objectclass=organizationalUnit)
    (objectclass=groupOfNames)
    (objectclass=groupOfUniqueNames)
    (objectclass=group)
  )
  (uid=user1111)
)'
```

## ODBC、WinNT およびカスタム ネームスペースの検索式文法

ODBC ユーザディレクトリでユーザ、グループまたはその両方を検索できます。その検索は属性値ペアに基づいて実行されます。

*pszSearchPattern* の形式は次のとおりです。

[ <class>= ] <value>

この形式の例において、

- <class> は空の文字列、またはユーザまたはグループです。空の文字列はユーザとグループを意味します。
- <value> はワイルドカード文字列です。

### 戻り値

- *Sm\_PolicyApi\_Success*。検索は成功しました。
- *Sm\_PolicyApi\_Failure*。ユーザ仕様検索が失敗しました。または、メモリを *Sm\_PolicyApi\_User\_t* に割り当てることができませんでした。
- *Sm\_PolicyApi\_NoSession*。有効な管理者セッションがありません。
- *Sm\_PolicyApi\_InvalidHandle*。このコールに先立つ有効な初期化はありませんでした。
- *Sm\_PolicyApi\_InvalidOid*。ユーザディレクトリ *OID* が見つかりませんでした。
- *Sm\_PolicyApi\_UserDirNotValid*。ユーザディレクトリへの接続はありません。または、ユーザディレクトリプロバイダはロードされていません。

## Sm\_PolicyApi\_ManagementCommand()

ユーザ、キーおよびリソース管理アクティビティを実行します。

ポリシー管理 API は、Sm\_PolicyApi\_ManagementCommands\_t に列挙される管理コマンドのタイプをサポートします。

### タイプ

ユーティリティ関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_ManagementCommand (
    void* pSessionHandle,
    Sm_PolicyApi_ManagementCommand_t* pstructManagementCommand
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pstructManagementCommand</i>	I	管理コマンド構造へのポインタのアドレス。

### 戻り値

- Sm\_PolicyApi\_Success。管理コマンドは正常に発行されました。
- Sm\_PolicyApi\_Failure。管理コマンドは正常に発行されませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、Sm\_PolicyApi\_ManagementCommand() を呼び出す権限がありません。
- Sm\_PolicyApi\_Invalid。無効な管理コマンドが指定されました。

## Sm\_PolicyApi\_Release()

ポリシーストアから切断し、API によって保持されたメモリおよびリソースを解放します。

この関数は API クライアントセッションによって呼び出される最後の関数である必要があります。この関数は、クライアントセッションにつき 1 回呼び出される必要があります。

注: この関数を呼び出さないとメモリリークが生じます。

### タイプ

必要な関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_Release(void* pInitHandle);
```

パラメータ	I/O	説明
<i>pInitHandle</i>	I	クライアントセッションの初期化が成功した後に Sm_PolicyApi_Init() によって返された init ハンドル。

### 戻り値

この関数は常に Sm\_PolicyApi\_Success を返します。

## Sm\_PolicyApi\_RemoveAdminFromAffiliateDomain()

アフィリエイト ドメインから管理者を削除します。

### タイプ

フェデレーション関数

### 構文

```
int SM_EXTERN Sm_PolicyApi_RemoveAdminFromAffiliateDomain (
    void*          pSessionHandle,
    const char*    pszAdminOid,
    const char*    pszAffiliateDomainOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAdminOid</i>	I	既存の管理者のオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszAffiliateDomainOid</i>	I	既存のアフィリエイト ドメインのオブジェクト識別子が含まれ、NULL で終わる文字列です。

### 戻り値

- `Sm_PolicyApi_Success`。管理者はアフィリエイト ドメインから削除されました。
- `Sm_PolicyApi_Failure`。管理者はアフィリエイト ドメインから削除されませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、アフィリエイト ドメインから管理者を削除する権限がありません。
- `Sm_PolicyApi_InvalidOID`。アフィリエイト ドメイン `OID` が更新中に見つかりませんでした。
- `Sm_PolicyApi_NotFound`。管理者オブジェクト識別子は、アフィリエイト ドメイン集合で見つけることができませんでした。

## Sm\_PolicyApi\_RemoveAdminFromDomain()

`szDomainOid` によって識別されるドメインから `pszAdminOid` によって識別される管理者オブジェクトを分離します。

### タイプ

管理者関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_RemoveAdminFromDomain (
    void*          pSessionHandle,
    const char*    pszAdminOid,
    const char*    pszDomainOid
);
```

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

パラメータ	I/O	説明
<i>pszAdminOid</i>	I	既存の管理者のオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszDomainOid</i>	I	既存のドメインのオブジェクト識別子が含まれ、NULL で終わる文字列です。

### 戻り値

- **Sm\_PolicyApi\_Success**。管理者は、ドメインから正常に削除されました。
- **Sm\_PolicyApi\_Failure**。管理者は、ドメインから正常に削除されませんでした。
- **Sm\_PolicyApi\_InvalidHandle**。このコールに先立つ有効な初期化はありませんでした。
- **Sm\_PolicyApi\_NoSession**。有効な管理者セッションがありません。
- **Sm\_PolicyApi\_NoPrivilege**。管理者には、ドメインから管理者を削除する権限がありません。
- **Sm\_PolicyApi\_InvalidOid**。管理者オブジェクト識別子またはドメインオブジェクト識別子が見つかりませんでした。
- 以下の場合には **Sm\_PolicyApi\_NotFound**。
  - ドメイン集合に管理者オブジェクト識別子はありません。
  - 管理者オブジェクト識別子はドメイン集合で見つけることができませんでした。

## Sm\_PolicyApi\_RemoveAgentConfigAssociation()

指定されたエージェント設定オブジェクトから設定パラメータ名/値ペアを削除します。

### タイプ

エージェント設定関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_RemoveAgentConfigAssociation (
    void*                pSessionHandle,
    const char*          pszAssociationOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAssociationOid</i>	I	削除すべき名前/値ペアの一意の識別子。

### 戻り値

- Sm\_PolicyApi\_Success。操作は成功しました。
- Sm\_PolicyApi\_Failure。一般的な障害です。
- Sm\_PolicyApi\_InvalidOid。エージェント設定オブジェクトに一意の ID が対応していません。
- Sm\_PolicyApi\_NoPrivilege。呼び出し元には、エージェント設定オブジェクト パラメータを削除する権限がありません。
- Sm\_PolicyApi\_InvalidHandle。セッションポインタが無効です。
- Sm\_PolicyApi\_NoSession。API ユーザが正しくログインしていません。

## Sm\_PolicyApi\_RemoveAssertionConsumerServiceFromSAMLSP()

アサーション コンシューマ サービスへの既存のインデックス付きエンドポイント参照をポリシー ストアから削除します。 インデックス、バインド方法およびアサーション コンシューマ サービス URL は、既存のアサーション コンシューマ サービスに一致する必要があります。

### タイプ

フェデレーション関数

### 構文

```
int SM_EXTERN Sm_PolicyApi_RemoveAssertionConsumerServiceToSAMLSP (
    void*                                pSessionHandle,
    const Sm_PolicyApi_SAMLSPAssertionConsumerService_t*
        pstructSAMLSPAssertionConsumerService,
    const char*                            pszSAMLSPoid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pstructSAMLAssertionConsumerService</i>	I	アサーション コンシューマ サービス構造体へのポインタです。
<i>pszSAMLSPoid</i>	I	サービス プロバイダの OID を含む文字列へのポインタです。

### 戻り値

- **Sm\_PolicyApi\_Success**。アサーション コンシューマ サービスは正常に削除されました。
- **Sm\_PolicyApi\_Failure** - アサーション コンシューマ サービスは正常に削除されませんでした。
- **Sm\_PolicyApi\_ACSIndexedEndpointNotFound** - 削除されるアサーション コンシューマ サービスに一致するアサーション コンシューマ サービスはありません。
- **Sm\_PolicyApi\_InvalidHandle** - このコールに先立つ有効な初期化はありませんでした。
- **Sm\_PolicyApi\_NoSession** - 有効な管理者セッションがありません。
- **Sm\_PolicyApi\_NoPrivilege** - 管理者には、SAML サービス プロバイダから属性を削除する権限がありません。

## Sm\_PolicyApi\_RemoveAttributeFromAffiliate()

アフィリエイトから属性を削除します。

### タイプ

フェデレーション関数

### 構文

```
int SM_EXTERN Sm_PolicyApi_RemoveAttributeFromAffiliate (
    void*                pSessionHandle,
    const Sm_PolicyApi_AffiliateAttr_t* pstructAffiliateAttr,
    const char*          pszAffiliateOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

パラメータ	I/O	説明
<i>pstructAffiliateAttr</i>	I	アフィリエイト属性構造体へのポインタ。
<i>pszAffiliateOid</i>	I	既存のアフィリエイトのオブジェクト識別子が含まれ、NULL で終わる文字列です。

### 戻り値

- `Sm_PolicyApi_Success`。アフィリエイト属性は正常に削除されました。
- `Sm_PolicyApi_Failure`。アフィリエイト属性は正常に削除されませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、アフィリエイト属性を削除する権限がありません。
- `Sm_PolicyApi_InvalidOID`。アフィリエイト OID が見つかりませんでした。

## Sm\_PolicyApi\_RemoveAttributeFromSAMLScheme()

設定された属性を SAML 認証方式から削除します。

### 構文

```
int SM_EXTERN Sm_PolicyApi_RemoveAttributeFromSAMLScheme(
    void*                pHandle,
    const Sm_PolicyApi_Scheme_t* pstructScheme,
    const Sm_PolicyApi_SAMLRequesterAttr_t* pAttr
);
```

パラメータ	I/O	説明
<i>pHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pstructScheme</i>	I	値の入力が完了している方式構造体へのポインタです。

---

パラメータ	I/O	説明
<i>pAttr</i>	I	削除される属性が含まれる Sm_PolicyApi_SAMLRequesterAttr_t 構造へのポインタ。

---

### 戻り値

- Sm\_PolicyApi\_Success。属性は正常に削除されました。
- Sm\_PolicyApi\_Failure。属性は正常に削除されませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、SAML リクエストに対する属性を削除する権限がありません。
- Sm\_PolicyApi\_SAMLIDP\_IncorrectParameters。入力された SAML 提供のプロパティが未完了か、または正しくありません。
- Sm\_PolicyApi\_NoSuchAttribute。一致する名前および NameFormat を持つ属性は存在しません。

## Sm\_PolicyApi\_RemoveAttributeFromSAMLSP()

指定された SAML 属性をサービス プロバイダから削除します。

### タイプ

SAML 2.0 設定関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_RemoveAttributeFromSAMLSP
(
    void*                pHandle,
    const Sm_PolicyApi_SAMLSPAttr_t* pstructSAMLSPAttr,
    const char*          pszSAMLSPoid
);
```

パラメータ	I/O	説明
<i>pHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pstructSAMLSPAttr</i>	I	SAML 属性構造へのポインタ
<i>pszSAMLSPoid</i>	I	ポリシーストア内のサービス プロバイダのオブジェクト識別子。

### 戻り値

- Sm\_PolicyApi\_Success。属性は正常に削除されました。
- Sm\_PolicyApi\_Failure。属性は正常に削除されませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、サービス プロバイダから SAML 属性を削除する権限がありません。
- Sm\_PolicyApi\_NoSuchAttribute。提供された名前および NameFormat を持つ属性は存在しません。

## Sm\_PolicyApi\_RemoveFromGroup()

グループからアイテムを削除します。

アイテムとグループは存在する必要がある同じタイプである必要があります。また、アイテムはグループに含まれている必要があります。

*pszDomainOid* パラメータは、ルールとレスポンス グループによって必要とされます。エージェント グループはドメインベースのオブジェクトではないので、ドメイン **OID** を必要としません。

### タイプ

グループ関数、グローバル範囲（エージェント）またはドメイン範囲（レスポンス、ルール）。

### 構文

```
int SM_EXTERN Sm_PolicyApi_RemoveFromGroup (
    void*                pSessionHandle,
    Sm_PolicyApi_Groups_t dwGroup,
    const char*          pszItemOid,
    const char*          pszGroupOid,
    const char*          pszDomainOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>dwGroup</i>	I	グループのタイプを示します。
<i>pszItemOid</i>	I	既存のアイテムのオブジェクト識別子が含まれる <b>NULL</b> で終わる文字列で、グループと同じタイプである必要があります。
<i>pszGroupOid</i>	I	グループのオブジェクト識別子が含まれ、 <b>NULL</b> で終わる文字列です。
<i>pszDomainOid</i>	I	既存のドメインのオブジェクト識別子が含まれ、 <b>NULL</b> で終わる文字列です。ルールとレスポンス グループによって必要とされます。

### 戻り値

- `Sm_PolicyApi_Success`。削除は成功しました。
- `Sm_PolicyApi_Failure`。削除は成功しませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、それぞれのグループからルール、レスポンスまたはエージェント `OID` を削除する権限がありません。
- `Sm_PolicyApi_InvalidOid` :
  - ドメイン `OID` が見つかりませんでした（ドメインベースのグループについて）。
  - グループ `OID` が見つかりませんでした。
  - ルール、レスポンス、またはエージェント `OID` またはグループ `OID` が見つかりませんでした。
- `Sm_PolicyApi_BadGroup`。パラメータ `dwGroup` はルール、レスポンスまたはエージェント タイプではありません。

## Sm\_PolicyApi\_RemovePolicyLinkFromPolicy()

*pszPolicyOid* によって識別されるポリシーから *pszPolicyLinkOid* によって識別されるポリシー リンクを削除することによって、この関数は、ポリシーからルールを効率的に削除します。

ポリシー リンク オブジェクトは、ポリシーをルールおよびオプションでレスポンスにバインドします。

注: SiteMinder v4.5 より前のリリースでは、`Sm_PolicyApi_RemovePolicyLinkFromPolicy()` によって提供される機能は、`Sm_PolicyApi_RemoveRuleFromPolicy()` によって提供されています。

### タイプ

ポリシー関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_RemovePolicyLinkFromPolicy (
    void*          pSessionHandle,
    const char*    pszPolicyLinkOid,
    const char*    pszPolicyOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszPolicyLinkOid</i>	I	<i>pszPolicyOid</i> 下の既存のポリシー リンクのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszPolicyOid</i>	I	既存のポリシーのオブジェクト識別子が含まれ、NULL で終わる文字列です。

### 戻り値

- `Sm_PolicyApi_Success`。ポリシー リンクは正常に削除されました。
- `Sm_PolicyApi_Failure`。ポリシー リンクは正常に削除されませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、ポリシー リンクを削除する権限がありません。
- `Sm_PolicyApi_InvalidOid` :
  - ポリシー OID が見つかりませんでした。
  - ポリシー リンク OID が見つかりませんでした。

## Sm\_PolicyApi\_RemoveRegularExpressionFromPasswordPolicy()

参照されたパスワード ポリシーから正規表現を削除します。セッションのバージョンが `SM_POLICY_API_VERSION_6_0` に設定される場合にのみ実装されます。

### タイプ

正規表現関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_RemoveRegularExpressionFromPasswordPolicy(
    void* pSessionHandle,
    const char* pszRegularExpressionOid,
    const char* pszPasswordPolicyOid
);
```

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<code>pszRegularExpressionOid</code>	I	削除される正規表現の OID。

パラメータ	I/O	説明
<i>pszPassword</i> <i>PolicyOid</i>	I	表現を削除するパスワードポリシーの OID。

### 戻り値

- `Sm_PolicyApi_Success`。 正規表現は正常に削除されました。
- `Sm_PolicyApi_Failure`。 正規表現は正常に削除されませんでした。
- `Sm_PolicyApi_NoSession`。 有効な管理者セッションがありません。
- `Sm_PolicyApi_InvalidHandle`。 このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoPrivilege`。 管理者には、正規表現を削除する権限がありません。
- `Sm_PolicyApi_InvalidOid`。 パスワードポリシー OID が見つかりませんでした。

## Sm\_PolicyApi\_RemoveResponseAttr()

*pszResponseAttrOid* によって識別されるレスポンスから *pstructResponseAttr* 構造によって定義されるレスポンス属性を分離します。属性名と属性値は、発生する削除の順に一致する必要があります。

### タイプ

レスポンス関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_RemoveResponseAttr (
    void*          pSessionHandle,
    const char*    pszResponseAttrOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

パラメータ	I/O	説明
<i>pszResponseAttrOid</i>	I	既存のレスポンス属性のオブジェクト識別子が含まれ、NULL で終わる文字列です。

### 戻り値

- Sm\_PolicyApi\_Success。削除操作は成功しました。
- Sm\_PolicyApi\_Failure。削除操作は成功しませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoPrivilege。管理者には、レスポンス属性を削除する権限がありません。
- Sm\_PolicyApi\_InvalidOid。レスポンス属性 OID が見つかりませんでした。

## Sm\_PolicyApi\_RemoveUserDirFromAffiliateDomain()

既存のアフィリエイト ドメインからユーザディレクトリを削除します。

### タイプ

フェデレーション関数

### 構文

```
int SM_EXTERN Sm_PolicyApi_RemoveUserDirFromAffiliateDomain(
    void*      pSessionHandle,
    const char* pszUserDirOid,
    const char* pszAffiliateDomainOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

---

パラメータ	I/O	説明
<i>pszUserDirOid</i>	I	既存のユーザディレクトリのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszAffiliate DomainOid</i>	I	既存のアフィリエイト ドメインのオブジェクト識別子が含まれ、NULL で終わる文字列です。

---

### 戻り値

- **Sm\_PolicyApi\_Success**。ユーザディレクトリは、アフィリエイト ドメインから正常に削除されました。
- **Sm\_PolicyApi\_NotFound**。ドメインにはこのユーザディレクトリがありません。
- **Sm\_PolicyApi\_InvalidHandle**。このコールに先立つ有効な初期化はありませんでした。
- **Sm\_PolicyApi\_NoSession**。有効な管理者セッションがありません。
- **Sm\_PolicyApi\_NoPrivilege**。管理者には、アフィリエイト ドメインからユーザディレクトリを削除する権限がありません。
- **Sm\_PolicyApi\_InvalidOID**。ユーザディレクトリ OID またはアフィリエイト ドメイン OID は更新中に見つかりませんでした。

## Sm\_PolicyApi\_RemoveUserDirFromDomain()

*pszDomainOid* によって識別されるドメインから *pszUserDirOid* によって識別されるユーザディレクトリを分離します。

### タイプ

ユーザディレクトリ関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_RemoveUserDirFromDomain (
    void*          pSessionHandle,
    const char*    pszUserDirOid,
    const char*    pszDomainOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszUserDirOid</i>	I	既存のユーザディレクトリのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszDomainOid</i>	I	既存のドメインのオブジェクト識別子が含まれ、NULL で終わる文字列です。

### 戻り値

- **Sm\_PolicyApi\_Success**。削除操作は成功しました。
- **Sm\_PolicyApi\_NotFound**。ドメインにはユーザディレクトリがありません。
- **Sm\_PolicyApi\_NoSession**。有効な管理者セッションがありません。
- **Sm\_PolicyApi\_InvalidHandle**。このコールに先立つ有効な初期化はありませんでした。
- **Sm\_PolicyApi\_NoPrivilege**。管理者には、ドメインからユーザディレクトリを削除する権限がありません。
- **Sm\_PolicyApi\_InvalidOid**。ユーザディレクトリまたはドメイン OID が見つかりませんでした。

## Sm\_PolicyApi\_RemoveUsersFromAffiliate()

アフィリエイトからユーザディレクトリのエントリを削除します。

### タイプ

フェデレーション関数

### 構文

```
int SM_EXTERN Sm_PolicyApi_RemoveUsersFromAffiliate (
    void*          pSessionHandle,
    const char*    pszAffiliateOid,
    const char*    pszUserPolicyOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAffiliateOid</i>	I	既存のアフィリエイトのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszUserPolicyOid</i>	I	既存のポリシー ユーザのオブジェクト識別子が含まれ、NULL で終わる文字列です。

### 戻り値

- Sm\_PolicyApi\_Success。ユーザは正常に削除されました。
- Sm\_PolicyApi\_Failure。ユーザは正常に削除されませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、ユーザを削除する権限がありません。
- Sm\_PolicyApi\_InvalidOID。アフィリエイト OID またはユーザ ポリシー OID が見つかりませんでした。

## Sm\_PolicyApi\_RemoveUsersFromPolicy()

*pszPolicyOid* によって識別されるポリシーから *pszUserPolicyOid* によって識別されるユーザを分離します。一度に削除できるユーザ仕様は 1 つのみです（集合体の場合あり）。

### タイプ

ユーザとユーザ状態関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_RemoveUsersFromPolicy (
    void*          pSessionHandle,
    const char*    pszPolicyOid,
    const char*    pszUserPolicyOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszPolicyOid</i>	I	ユーザが削除される既存のポリシーのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszUserPolicyOid</i>	I	既存のポリシーユーザのオブジェクト識別子が含まれ、NULL で終わる文字列です。

## 注釈

この関数は、Sm\_PolicyApi\_User\_t へのポインタが Sm\_PolicyApi\_GetPolicyUsers() 関数で取得された場合に限り成功します。ユーザが Sm\_PolicyApi\_LookupDirectoryEntry()、Sm\_PolicyApi\_GetDirectoryContents() または Sm\_PolicyApi\_ValidateDirectoryEntry() で取得される場合は、pszUserPolicyOid は無効になりそのコールは失敗します。

## 戻り値

- Sm\_PolicyApi\_Success。削除操作は成功しました。
- Sm\_PolicyApi\_Failure。削除操作は成功しませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、ポリシーからユーザを削除する権限がありません。
- Sm\_PolicyApi\_InvalidOid :
  - ポリシー OID が見つかりませんでした。
  - ユーザ ポリシー OID が見つかりませんでした。

## Sm\_PolicyApi\_RemoveUsersFromSAMLServiceProvider()

指定されたユーザをサービス プロバイダから削除します。

### タイプ

SAML 2.0 設定関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_AddUsersToSAMLServiceProvider
(
    void*                pSessionHandle,
    const char*          pszProviderOid,
    Sm_PolicyApi_User_t* pStructUsers,
    int                  iPolicyFlags
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszProviderOid</i>	I	サービス プロバイダのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pStructUsers</i>	I	サービス プロバイダから削除するユーザ。
<i>iPolicyFlags</i>	I	1 ビットのフィールドであり、以下を示します。 <ul style="list-style-type: none"> <li>■ SAML サービス プロバイダに対して作成されたポリシーにユーザが含まれるかどうか</li> <li>■ ポリシーが再帰的に適用されるべきかどうか</li> </ul>

### 戻り値

- `Sm_PolicyApi_Success`。 ユーザは正常に削除されました。
- `Sm_PolicyApi_Failure`。 ユーザは正常に削除されませんでした。
- `Sm_PolicyApi_InvalidHandle`。 このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。 有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。 管理者には、ユーザを削除する権限がありません。
- `Sm_PolicyApi_InvalidOid`。 サービスプロバイダ `OID` が見つかりませんでした。

## Sm\_PolicyApi\_RemoveUsersFromWSFEDResourcePartner()

WS-Federation リソース パートナーからユーザ ディレクトリ エントリを分離します。

### 構文

```
int SM_EXTERN Sm_PolicyApi_RemoveUsersFromWSFEDResourcePartner (  
    void* pSessionHandle,  
    const char * pszProviderOid,  
    const char * pszUserPolicyOid  
);
```

### パラメータ

pSessionHandle

[in] 内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

pszProviderOid

[in] 既存の WS フェデレーション リソース パートナーのオブジェクト識別子が含まれ、NULL で終わる文字列です。

pszUserPolicyOid

[in] 既存のポリシー ユーザのオブジェクト識別子が含まれ、NULL で終わる文字列です。

### 戻り値

- Sm\_PolicyApi\_Success。ユーザ ディレクトリは正常に削除されました。
- Sm\_PolicyApi\_Failure。ユーザ ディレクトリは正常に削除されませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、ユーザを削除する権限がありません。
- Sm\_PolicyApi\_InvalidOid。リソース パートナー OID が見つかりませんでした。

## Sm\_PolicyApi\_RenameObject()

ドメインまたはグローバル オブジェクトの名前を変更します。

この関数は、名前を変更するオブジェクトのオブジェクト識別子 (OID) を必要とします。オブジェクト上で取得操作を実行することによりオブジェクト識別子を取得できます。

### タイプ

一般的なオブジェクト関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_RenameObject (
    void*          pSessionHandle,
    const char*    pszOid,
    const char*    pszNewName
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszOid</i>	I	名前を変更するオブジェクトのオブジェクト識別子。
<i>pszNewName</i>	I	オブジェクトの新しい名前。

### 戻り値

- Sm\_PolicyApi\_Success。オブジェクトは正常に名前を変更されました。
- Sm\_PolicyApi\_Failure。オブジェクトは正常に名前を変更されませんでした。
- Sm\_PolicyApi\_InvalidHandle。このコールに先立つ有効な初期化はありませんでした。
- Sm\_PolicyApi\_NoSession。有効な管理者セッションがありません。
- Sm\_PolicyApi\_NoPrivilege。管理者には、オブジェクトの名前を変更する権限がありません。
- Sm\_PolicyApi\_InvalidOid。OID が見つかりませんでした。

## Sm\_PolicyApi\_SetAffiliateDomainUserDirSearchOrder()

アフィリエイト ドメインのユーザ ディレクトリ 検索順序を設定します。

### タイプ

フェデレーション関数

### 構文

```
int SM_EXTERN Sm_PolicyApi_SetAffiliateDomainUserDirSearchOrder (
    void*          pSessionHandle,
    const char*    pszAffiliateDomainOid,
    char**         pszArray[]
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszAffiliateDomainOid</i>	I	既存のアフィリエイト ドメインのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszArray</i>	I	希望する順序での、ユーザ ディレクトリ OID の配列。

### 戻り値

- **Sm\_PolicyApi\_Success**。この関数は、ユーザ ディレクトリ 検索順序を正常に設定しました。
- **Sm\_PolicyApi\_InvalidHandle**。このコールに先立つ有効な初期化はありませんでした。
- **Sm\_PolicyApi\_NoSession**。有効な管理者セッションがありません。
- **Sm\_PolicyApi\_NoPrivilege**。管理者には、アフィリエイト ドメインでユーザ ディレクトリ 検索順序を設定する権限がありません。
- **Sm\_PolicyApi\_InvalidOID**。アフィリエイト OID が見つかりませんでした。
- **Sm\_PolicyApi\_DomainNotAffiliate**。ドメインはアフィリエイト ドメインではありません。
- **Sm\_PolicyApi\_WrongNumberOfElements**。アフィリエイト ドメイン 集合内のユーザ ディレクトリの数は、配列内の要素数とは異なります。

## Sm\_PolicyApi\_SetDisabledUserState()

ユーザの無効な状態を設定します。また、この関数でユーザを有効にできます。

この関数を機能させるには、無効なユーザを追跡するための属性をユーザディレクトリ（構造 `Sm_PolicyApi_UserDir_t` の `pszDisabledAttr` フィールド）内に設定する必要があります。ポリシーサーバユーザインターフェースを使用して、属性を設定することもできます。

### タイプ

ユーザとユーザ状態関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_SetDisabledUserState (
    void*                pSessionHandle,
    const char*          pszUserDirOid,
    const char*          pszUserDN,
    const Sm_Api_DisabledReason_t nDisabledReason,
    char**               pszErrMsg
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszUserDirOid</i>	I	既存のユーザディレクトリのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszUserDN</i>	I	無効な状態を変更するユーザの識別名。
<i>nDisabledReason</i>	I	ユーザを無効または有効にする理由です。その理由は <code>Sm_Api_DisabledReason_t</code> に列挙されます。それは <code>SmApi.h</code> で定義されています。 正しい状態の設定は呼び出し元の責任です。複数の理由は同時に存在できます。ユーザが有効なとき、無効なマスク内のフラグはすべてクリアされるべきです。

---

パラメータ	I/O	説明
<i>pszErrMsg</i>	O	操作が成功しなかった場合、エラーメッセージが文字列内に保持されます。Sm_PolicyApi_FreeString() を呼び出して、この変数に割り当てられたメモリを解放します。

---

### 戻り値

- `Sm_PolicyApi_Success`。無効のユーザ状態は正常に設定されました。
- `Sm_PolicyApi_Failure` :
  - ユーザ状態は無効になりませんでした。
  - `pszErrMsg` にメモリを割り当てることができませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_NoPrivilege`。管理者には、無効のユーザ状態を設定する権限がありません。
- `Sm_PolicyApi_InvalidOid`。ユーザディレクトリ `OID` が見つかりませんでした。

### 例

- パスワードの期限切れのためユーザを無効にする方法

```
nDisabledReason = Sm_Api_Disabled_DisabledMask &  
                  Sm_Api_Disabled_PWExpired;
```

- 管理上の理由でユーザを無効にする方法

```
enum Sm_Api_DisabledReason_t nDisabledReason;
```

```
iRes = Sm_PolicyApi_GetDisabledUserState (pSessionHandle,
                                           pszUserDirOid,
                                           pszUserDN,
                                           &nDisabledReason,
                                           &pszGetErrMsg);

if (iRes != Sm_PolicyApi_Success)
{
    cout << "Error: " << pszGetErrMsg << endl;
}

// Set admin disabled reason bit.
nDisabledReason=(Sm_Api_DisabledReason_t) (nDisabledReason |
                                             Sm_Api_Disabled_AdminDisabled);

// Set Disable user state
iRes = Sm_PolicyApi_SetDisabledUserState(pSessionHandle,
                                           pszUserDirOid,
                                           pszUserDN,
                                           nDisabledReason,
                                           &pszSetErrMsg);
```

- ユーザを有効にし、すべての無効な理由ビットをクリアする方法

```
enum Sm_Api_DisabledReason_t nDisabledReason;
iRes = Sm_PolicyApi_GetDisabledUserState(pSessionHandle,
                                           pszUserDirOid,
                                           pszUserDN,
                                           &nDisabledReason,
                                           &pszGetErrMsg);

if (iRes != Sm_PolicyApi_Success)
{
    if (pszGetErrMsg)
    {
        cout << "Error: " << pszGetErrMsg << endl;
    }
}

// Clear all the disable reason bits.
nDisabledReason=(Sm_Api_DisabledReason_t)
                (nDisabledReason & (~Sm_Api_Disabled_DisabledMask));

// Set Disable user state to enable
iRes = Sm_PolicyApi_SetDisabledUserState(pSessionHandle,
                                           pszUserDirOid,
                                           pszUserDN,
                                           nDisabledReason,
                                           &pszSetErrMsg);
```

## Sm\_PolicyApi\_SetPassword()

ユーザアカウントのパスワードを変更します。また、パスワードを変更せずに、新しいパスワードまたは古いパスワードを検証するために使用することもできます。

新しいパスワードを検証するには、ディレクトリ用の **SiteMinder** パスワードサービスを設定する必要があります。また、**SiteMinder** ユーザディレクトリ内のパスワード属性を識別する必要があります。

### タイプ

ユーザとユーザ状態関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_SetPassword (
    void*          pSessionHandle,
    const char*    pszUserDirOid,
    const char*    pszUserDN,
    const char*    pszNewPassword,
    const char*    pszOldPassword,
    bool           bChangePassword,
    bool           bValidateNewPassword,
    bool           bValidateOldPassword,
    char**         pszErrMsg
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszUserDirOid</i>	I	ユーザが含まれているユーザディレクトリのオブジェクト識別子を含み、NULL で終わる文字列です。
<i>pszUserDN</i>	I	パスワードが変更され、かつ（または）新しいまたは古いパスワードが検証されるユーザの識別名。
<i>pszNewPassword</i>	I	検証または変更する新しいユーザパスワード。

パラメータ	I/O	説明
<i>pszOldPassword</i>	I	検証または変更する古いユーザパスワード。
<i>bChangePassword</i>	I	真の場合、パスワードは新しいパスワードに変更され、ユーザのパスワード履歴に記録されます。エラーが発生した場合、この関数はエラーコードを返します。
<i>bValidateNewPassword</i>	I	真の場合、新しいパスワードがパスワードポリシー要件をすべて満たしているか確認されます。新しいパスワードがパスワードポリシーに違反している場合、 <i>pszErrMsg</i> が設定され、関数はエラーコードを返します。
<i>bValidateOldPassword</i>	I	真の場合、古いパスワードはユーザを認証するために使用されます。認証が失敗した場合、関数はエラーコードを返します。
<i>pszErrMsg</i>	O	ユーザパスワードの変更または検証が成功しなかった場合のエラーメッセージが含まれる文字列。 <i>Sm_PolicyApi_FreeString()</i> を呼び出して、この変数に割り当てられたメモリを解放します。

### 戻り値

- **Sm\_PolicyApi\_Success**。変更または検証は成功しました。
- **Sm\_PolicyApi\_Failure**。変更または検証は成功しませんでした。または、メモリはエラーメッセージ文字列に割り当てられないことができませんでした。
- **Sm\_PolicyApi\_InvalidHandle**。このコールに先立つ有効な初期化はありませんでした。
- **Sm\_PolicyApi\_InvalidPasswordSyntax**。これらの条件が両方とも存在する場合に返されます。
  - *bValidateNewPassword* が真である場合。
  - 新しいパスワードが、指定されたディレクトリに対して設定されたパスワードポリシー要件を満たさなかった場合。

エラーの詳細については、**Sm\_PolicyApi\_GetPasswordMsg()** を呼び出し、それに **Sm\_PolicyApi\_SetPassword()** から返された、エンコードされたエラーメッセージ (*pszErrMsg*) を渡します。

- **Sm\_PolicyApi\_NoSession**。有効な管理者セッションがありません。
- **Sm\_PolicyApi\_NoPrivilege**。管理者には、ユーザパスワードを設定または検証する権限がありません。
- **Sm\_PolicyApi\_InvalidOid**。ユーザディレクトリ **OID** が見つかりませんでした。

## Sm\_PolicyApi\_SetPath()

*pszPolicyOid* で識別されるポリシーの *pUser* で識別されるユーザのパスを設定します。

## タイプ

ユーザとユーザ状態関数。

## 構文

```
int SM_EXTERN Sm_PolicyApi_SetPath (
    void*                pSessionHandle,
    Sm_PolicyApi_User_t* pUser
    const char*          pszNewPath,
    const char*          pszPolicyOid
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pUser</i>	I	Sm_PolicyApi_user_t 構造体へのポインタ。ユーザに関する情報が含まれます。
<i>pszNewPath</i>	I	ユーザの新しいパス。
<i>pszPolicyOid</i>	I	NULL で終わる文字列であり、ユーザのパスが設定される既存のポリシーのオブジェクト識別子が含まれます。

### 注釈

この関数は、`Sm_PolicyApi_User_t` へのポインタが `Sm_PolicyApi_GetPolicyUsers()` 関数で取得された場合に限り成功します。ユーザが `Sm_PolicyApi_LookupDirectoryEntry()`、`Sm_PolicyApi_GetDirectoryContents()` または `Sm_PolicyApi_ValidateDirectoryEntry()` で取得される場合は、`pszUserPolicyOid` は無効になりそのコールは失敗します。

### 戻り値

- `Sm_PolicyApi_Success`。パスの設定操作が成功しました。
- `Sm_PolicyApi_Failure`。パスの設定操作が成功しませんでした。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_InvalidOid` :
  - ポリシー OID が見つかりませんでした。
  - ユーザ ポリシー OID が見つかりませんでした。

## Sm\_PolicyApi\_SetResponseInPolicyLink()

*ppstructPolicyLink* 構造によって説明されたポリシー リンクを変更します。

この関数は、レスポンスまたはレスポンス グループをルールまたはルール グループに設定します。それは、ポリシー リンクからレスポンスまたはレスポンス グループを削除するために使用することもできます。ポリシー リンクからレスポンスまたはレスポンス グループを削除するには、*Sm\_PolicyApi\_PolicyLink\_t* 内の *pszResponseOid* を空の文字列に設定します。

### タイプ

レスポンス関数、ドメイン範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_SetResponseInPolicyLink (
    void*                pSessionHandle,
    const char*          pszPolicyOid,
    Sm_PolicyApi_PolicyLink_t* ppstructPolicyLink
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszPolicyOid</i>	I	既存のポリシーのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>ppstructPolicyLink</i>	I	ポリシー リンク構造のアドレス。

### 戻り値

- **Sm\_PolicyApi\_Success**。レスポンスはポリシー リンクで正常に設定されたか、または削除されました。
- **Sm\_PolicyApi\_Failure**。レスポンスはポリシー リンクで設定できないか削除できません。
- **Sm\_PolicyApi\_InvalidHandle**。このコールに先立つ有効な初期化はありませんでした。
- **Sm\_PolicyApi\_NoSession**。有効な管理者セッションがありません。
- **Sm\_PolicyApi\_NoPrivilege**。管理者には、ポリシー リンクを設定する権限がありません。
- **Sm\_PolicyApi\_InvalidOid** :
  - ポリシー OID が見つかりませんでした。
  - ルールまたはルールグループ OID が見つかりませんでした。
  - レスポンスまたはレスポンス グループ OID が見つかりませんでした。
- **Sm\_PolicyApi\_DoesNotExist**。ポリシー集合にポリシー リンクはありません。

## Sm\_PolicyApi\_SetSharedSecretPolicy()

現在の SharedSecretPolicy を設定します。そのようなオブジェクトが常に 1 つあります。したがって、bUpdate ブールフラグを提供する必要はありません。

### タイプ

エージェントの設定

### 構文

```
int SM_EXTERN Sm_PolicyApi_SetSharedSecretPolicy (
    void*                                pSessionHandle,
    Sm_PolicyApi_SharedSecretPolicy_t*  ppstructSecretPolicy
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	現在のポリシー API セッションハンドル。
<i>ppstructSecretPolicy</i> ポリシー	I/O	共有秘密キー ポリシー構造。

### 戻り値

- Sm\_PolicyApi\_Success。操作は成功しました。
- Sm\_PolicyApi\_Failure。一般的な障害です。
- Sm\_PolicyApi\_InvalidHandle。セッションハンドルは無効でした。
- Sm\_PolicyApi\_NoSession。ユーザはログインされません。
- Sm\_PolicyApi\_NoPrivilege。ユーザには、オブジェクトを管理する権限がありません。

## Sm\_PolicyApi\_SetUserDirSearchOrder()

*pszDomainOid* によって識別されるドメインと関連付けられたユーザディレクトリ オブジェクトの検索順序を並び替えます。

名前の順序付きリストは、*pszArray* 文字列配列で指定されます。この配列内のユーザディレクトリは、OID および数（順序ではない）が **Sm\_PolicyApi\_GetUserDirSearchOrder()** へのコールによって取得されたユーザディレクトリ名のリストに一致する必要があります。

### タイプ

ユーザディレクトリ関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_SetUserDirSearchOrder (
    void*          pSessionHandle,
    const char*    pszDomainOid,
    char**         pszArray[]
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszDomainOid</i>	I	既存のドメインのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszArray</i>	I	希望する順序での、ユーザディレクトリ OID の配列。

### 戻り値

- `Sm_PolicyApi_Success`。ユーザ ディレクトリ 検索順序は正常に設定されました。
- `Sm_PolicyApi_NoSession`。有効な管理者セッションがありません。
- `Sm_PolicyApi_InvalidHandle`。このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_NoPrivilege`。管理者には、ドメインにユーザ ディレクトリ 検索順序を設定する権限がありません。
- `Sm_PolicyApi_InvalidOid`。ドメイン OID が見つかりませんでした。
- `Sm_PolicyApi_WrongNumberOfElements`。ドメイン集合内のユーザ ディレクトリ 数は、配列内のエレメント数とは異なります。

## Sm\_PolicyApi\_SetUserPasswordState()

`UserPasswordState` オブジェクトを追加または更新します。

ユーザと関連付けられた `PasswordState` がなければ、新しい `PasswordState` が作成されます。そうでないと、`UserPasswordState` が更新されます。

### タイプ

パスワード状態関数。

### 構文

```
int SM_EXTERN Sm_PolicyApi_SetUserPasswordState (
    void *pSessionHandle,
    const char *pszUserDirOid,
    const char *pszUserDN,
    Sm_PolicyApi_UserPasswordState_t *pPasswordState
    bool bEmptyHistory
);
```

パラメータ	I/O	説明
<code>pSessionHandle</code>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。

パラメータ	I/O	説明
<i>pszUserDirOid</i>	I	特定のユーザディレクトリに相当する一意のオブジェクト識別子。
<i>pszUserDN</i>	I	ユーザディレクトリ内のユーザの識別名を指定します。
<i>pPasswordState</i>	I/O	ユーザパスワード状態オブジェクト。 このパラメータがパスワードが最後に変更された時のユーザディレクトリ設定を変更し、パスワードが SiteMinder の外部でリセットされた場合、パスワード再利用を防ぐパスワードポリシーが予期したように動作しない場合があります。 このパラメータが NULL に設定される場合、関数は <code>Sm_PolicyApi_Failure</code> を返します。
<i>bEmptyHistory</i>	I	この関数が既存のパスワード履歴をクリアするべきかどうかを指定します。このパラメータが真に設定される場合、構造 <code>Sm_PolicyApi_UserPasswordState_t</code> のフィールド <code>tLastPWChange</code> フィールドは暗黙的にゼロにリセットされます。

### 戻り値

- `Sm_PolicyApi_Success`。 取得の操作が成功しました。
- `Sm_PolicyApi_Failure`。 一般的な障害です。
- `Sm_PolicyApi_InvalidHandle`。 セッションポインタが無効です。
- `Sm_PolicyApi_BadArgument`。 入力パスワード状態ポインタは NULL です。
- `Sm_PolicyApi_NoSession`。 API ユーザが正しくログインしていません。
- `Sm_PolicyApi_InvalidOid`。 一意の ID はユーザディレクトリ オブジェクトに相当しません。
- `Sm_PolicyApi_UserDirNotFound`。 ポリシーストアからのユーザディレクトリ オブジェクトの取得が失敗しました。
- `Sm_PolicyApi_ErrorUserDir`。 無効なユーザディレクトリ オブジェクトまたは間違ったユーザ DN。
- `Sm_PolicyApi_NoPrivilege`。 呼び出し元には適切な権限がありません。

## Sm\_PolicyApi\_ValidateDirectoryEntry()

ユーザ ディレクトリのユーザ仕様を検証します。

### タイプ

ユーザ ディレクトリ関数、グローバル範囲。

### 構文

```
int SM_EXTERN Sm_PolicyApi_ValidateDirectoryEntry (
    void*                pSessionHandle,
    const char*          pszUserDirOid,
    const char*          pszPath,
    Sm_PolicyApi_User_t** ppUserEntry
);
```

パラメータ	I/O	説明
<i>pSessionHandle</i>	I	内部ポリシー管理 API データ構造体へのポインタ。この構造体は、管理者セッションおよびクライアントセッションに関する情報を保持します。
<i>pszUserDirOid</i>	I	ユーザ ディレクトリのオブジェクト識別子が含まれ、NULL で終わる文字列です。
<i>pszPath</i>	I	ユーザのパスが含まれ、NULL で終わる文字列です。
<i>ppUserEntry</i>	O	ユーザ構造へのポインタのアドレスです。

### 戻り値

- `Sm_PolicyApi_Success`。 検証は成功しました。
- `Sm_PolicyApi_Failure` :
  - `pszPath` 内のユーザ パスのポリシー解決はタイプ `Sm_PolicyResolution_User` ではありません。
  - ユーザ ディレクトリへの接続はありません。または、ユーザ ディレクトリ プロバイダはロードされていません。
  - メモリを `Sm_PolicyApi_User_t` に割り当てることができませんでした。
- `Sm_PolicyApi_NoSession`。 有効な管理者セッションがありません。
- `Sm_PolicyApi_InvalidHandle`。 このコールに先立つ有効な初期化はありませんでした。
- `Sm_PolicyApi_InvalidOid`。 ユーザ ディレクトリ OID が見つかりませんでした。

## 認証方式の設定

認証方式をプログラムで設定する場合、ユーザは情報を提供しますが、提供しない場合はポリシー サーバ UI の [認証方式プロパティ] ダイアログボックスを通じてその情報が提供されます。 構造 `Sm_PolicyApi_Scheme_t` 内のフィールドを介してこの情報を提供します。

**注:** 以下のカテゴリの情報は異なる認証方式で異なる目的のために使用できます。たとえば、TeleID 認証方式では、共有秘密キーは暗号化シードを提供するために使用されます。

- 方式タイプ  
`SiteMinder` は、複数の標準的な認証方式タイプ (テンプレートとしても知られている) を提供します。 認証方式タイプはそれぞれ異なる方法で設定されます。
- 説明  
認証方式の簡単な説明。
- 保護レベル  
保護レベル値の有効範囲は 1 ~ 1000 です。 値が大きいほど、その方式によって提供される保護の程度が大きくなります。

- ライブラリ

認証方式ライブラリは、関連する認証方式タイプに対して認証プロセスを実行します。事前定義済み認証方式はそれぞれデフォルトライブラリに付属しています。オプションで、デフォルトの代わりにカスタムライブラリを使用できます。

- パラメータ

HTML ログイン ページの URL など、認証方式が必要とする追加情報です。認証方式によっては、パラメータ情報は、[認証方式プロパティ] ダイアログ ボックスの [方式タイプのセットアップ] タブ内のフィールド値から構築されます。パラメータ文字列が指定された方式タイプに対してどのように構成されているか確認するには、このダイアログ ボックスを開き、適切な方式タイプを選択し、[方式タイプのセットアップ] タブ内のフィールドに値を提供し、[詳細] タブ内の構成されたパラメータを表示します。

- 共有秘密キー

認証方式とポリシー サーバの両方にとって既知の情報です。認証方式ごとに、異なる種類の秘密キーを使用します。ほとんどの方式では秘密キーを使用しません。

- テンプレートですか?

認証方式がテンプレートかどうかを指定するフラグ。

**注:** C ポリシー管理 API を使用してテンプレートとして認証方式を設定することは、SDK v6.0 SP3 では廃止されました。

- 管理者によって使用されますか?

認証方式を使用して管理者を認証できるかどうかを指定するフラグです。

- 認証情報を保存しますか?

ユーザの認証情報を保存するかどうかを指定するフラグ。

- RADIUS ですか?

方式を RADIUS エージェントと共に使用できるかどうかを指定するフラグ。

- パスワードチェックを無視しますか?

方式のパスワードポリシーが有効かどうかを指定するフラグ。true (1) の場合、パスワードポリシーは無効です。

注: 匿名認証方式の場合は、Ignore password check フラグを True に設定する必要があります。

## 匿名テンプレート

匿名方式タイプに基づいて認証方式を設定するときこの表を使用します。この表で参照される構造体フィールドは、Sm\_PolicyApi\_Scheme\_t にあります。

注: 匿名認証方式の場合は、Ignore password check フラグを True に設定する必要があります。

情報タイプ	値の割り当てと意味
方式タイプ	<i>nType=Sm_Api_SchemeType_Anonymous</i> 匿名方式タイプです。
説明	<i>pszDesc=description</i> 認証方式の説明。
保護レベル	<i>nLevel=0</i> 0 に設定します。この方式タイプには適用できません。
ライブラリ	<i>pszLib="smauthanon"</i> この方式タイプ用のデフォルトライブラリ。
パラメータ	<i>pszParam=param</i> ゲスト DN が含まれる文字列。ゲスト DN と関連付けられたポリシーを匿名のユーザに適用する必要があります。
共有秘密キー	<i>pszSecret=""</i> 空の文字列に設定します。この方式には適用できません。

情報タイプ	値の割り当てと意味
テンプレートですか?	<i>bIsTemplate=0</i> 方式がテンプレートではないことを示すには、 <b>false (0)</b> に設定します。他の値は無視されます。
管理者によって使用されますか?	<i>bIsUsedbyAdmin=0</i> <b>false (0)</b> に設定すると、方式は管理者を認証するためには使用されません。
認証情報を保存しますか?	<i>bAllowSaveCreds=0</i> ユーザ認証情報が保存されないことを示すには、 <b>false (0)</b> に設定します。
RADIUS ですか?	<i>bIsRadius=0</i> <b>false (0)</b> に設定すると、方式は RADIUS エージェントと共に使用されません。
パスワードチェックを無視しますか?	<i>bIgnorePwCheck=1</i> <b>true (1)</b> に設定すると、パスワードチェックを無視します。

## SSL を介した基本テンプレート

方式タイプ、SSL を介した基本に基づいて認証方式を設定する場合はこの表を使用します。この表で参照される構造フィールドは、`Sm_PolicyApi_Scheme_t` にあります。

情報タイプ	値の割り当てと意味
方式タイプ	<i>nType=Sm_Api_SchemeType_BasicOverSSL</i> SSL を介した基本方式タイプです。
説明	<i>pszDesc=description</i> 認証方式の説明。
保護レベル	<i>nLevel=value</i> 1 ~ 1000 の値。値が大きいほど、その方式によって提供される保護の程度が大きくなります。デフォルトは <b>10</b> です。
ライブラリ	<i>pszLib="smauthcert"</i> この方式タイプ用のデフォルトライブラリ。

情報タイプ	値の割り当てと意味
パラメータ	<p><i>pszParam=param</i></p> <p>SSL サーバのドメインまたは IP アドレス、および SSL 認証情報コレクタ (SCC) の名前が含まれる文字列。形式</p> <p><code>https://server/SCC?basic</code></p> <p>以下の例ではデフォルトの SCC を使用します。</p> <p><code>https://my.server.com/siteminderagent/nocert/smgetcred.scc?basic</code></p>
共有秘密キー	<p><i>pszSecret=""</i></p> <p>空の文字列に設定します。この方式には適用できません。</p>
テンプレートですか?	<p><i>blsTemplate=0</i></p> <p>方式がテンプレートではないことを示すには、<b>false (0)</b> に設定します。他の値は無視されます。</p>
管理者によって使用されますか?	<p><i>blsUsedbyAdmin=0</i></p> <p>この方式に対しては <b>false (0)</b> に設定します。</p>
認証情報を保存しますか?	<p><i>bAllowSaveCreds=0</i></p> <p>ユーザ認証情報が保存されないことを示すには、<b>false (0)</b> に設定します。</p>
RADIUS ですか?	<p><i>blsRadius=0</i></p> <p><b>false (0)</b> に設定すると、方式は RADIUS エージェントと共に使用されません。</p>
パスワードチェックを無視しますか?	<p><i>blgnorePwCheck=flag</i></p> <p>パスワードチェックを無視するには <b>true (1)</b>、パスワードを確認するには <b>false (0)</b> に設定します。デフォルトは <b>0</b> です。</p>

## 基本テンプレート

方式タイプ、基本に基づいて認証方式を設定する場合はこの表を使用します。この表で参照される構造体フィールドは、`Sm_PolicyApi_Scheme_t`にあります。

情報タイプ	値の割り当てと意味
方式タイプ	<code>nType=Sm_Api_SchemeType_Basic</code> 基本方式タイプです。
説明	<code>pszDesc=description</code> 認証方式の説明。
保護レベル	<code>nLevel=value</code> 1 ~ 1000 の値。値が大きいほど、その方式によって提供される保護の程度が大きくなります。デフォルトは5です。
ライブラリ	<code>pszLib="smauthdir"</code> この方式タイプ用のデフォルトライブラリ。
パラメータ	<code>pszParam=""</code> 空の文字列に設定します。この方式には適用できません。
共有秘密キー	<code>pszSecret=""</code> 空の文字列に設定します。この方式には適用できません。
テンプレートですか?	<code>bIsTemplate=0</code> 方式がテンプレートではないことを示すには、 <b>false (0)</b> に設定します。他の値は無視されます。
管理者によって使用されますか?	<code>bIsUsedbyAdmin=1</code> <b>true (1)</b> に設定すると、方式は管理者を認証するために使用できます。
認証情報を保存しますか?	<code>bAllowSaveCreds=0</code> ユーザ認証情報が保存されないことを示すには、 <b>false (0)</b> に設定します。
RADIUS ですか?	<code>bIsRadius=1</code> <b>true (1)</b> に設定すると、方式は RADIUS エージェントと共に使用できます。

情報タイプ	値の割り当てと意味
パスワードチェックを無視しますか?	<i>bIgnorePwCheck=flag</i> パスワードチェックを無視するには <b>true (1)</b> 、パスワードを確認するには <b>false (0)</b> に設定します。デフォルトは <b>0</b> です。

## カスタム テンプレート

方式タイプ、カスタムに基づいて認証方式を設定する場合はこの表を使用します。認証 API を使用して、カスタム方式を作成します。この表で参照される構造体フィールドは、`Sm_PolicyApi_Scheme_t` にあります。

情報タイプ	値の割り当てと意味
方式タイプ	<i>nType=Sm_Api_SchemeType_Custom</i> カスタム方式タイプです。
説明	<i>pszDesc=description</i> 認証方式の説明。
保護レベル	<i>nLevel=value</i> 0 ~ 1000 の値。値が大きいほど、その方式によって提供される保護の程度が大きくなります。デフォルトは <b>5</b> です。
ライブラリ	<i>pszLib=customLibName</i> 認証 API を使用して作成したカスタム共有ライブラリの名前。
パラメータ	<i>pszParam=param</i> カスタム認証方式に必要な 1 つ以上のパラメータの任意の文字列。SSL を使用するカスタム認証方式については、SSL ベースの認証に必要な SiteMinder Web エージェントライブラリを参照する URL を提供する必要があります。
共有秘密キー	<i>pszSecret=secret</i> カスタム認証方式が認証情報の暗号化に使用する共有秘密キー（存在する場合）。
テンプレートですか?	<i>bIsTemplate=0</i> 方式がテンプレートではないことを示すには、 <b>false (0)</b> に設定します。他の値は無視されます。

情報タイプ	値の割り当てと意味
管理者によって使用されますか?	<i>bIsUsedbyAdmin=flag</i> 管理者の認証に方式を使用できることを指定するには <b>true (1)</b> 、または管理者の認証に方式を使用できないことを指定するには <b>false (0)</b> を指定します。デフォルトは <b>0</b> です。
認証情報を保存しますか?	<i>bAllowSaveCreds=0</i> ユーザ認証情報が保存されないことを示すには、 <b>false (0)</b> に設定します。
RADIUS ですか?	<i>bIsRadius=0</i> <b>false (0)</b> に設定すると、方式は RADIUS エージェントと共に使用されません。
パスワードチェックを無視しますか?	<i>bIgnorePwCheck=flag</i> パスワードチェックを無視するには <b>true (1)</b> 、パスワードを確認するには <b>false (0)</b> に設定します。デフォルトは <b>0</b> です。

## HTML フォーム テンプレート

HTML フォーム方式タイプに基づいて認証方式を設定するときこのテーブルを使用します。この表で参照される構造体フィールドは、`Sm_PolicyApi_Scheme_t` にあります。

情報タイプ	値の割り当てと意味
方式タイプ	<i>nType=Sm_Api_SchemeType_HTMLForm</i> HTML フォーム方式タイプです。
説明	<i>pszDesc=description</i> 認証方式の説明。
保護レベル	<i>nLevel=value</i> 1 ~ 1000 の値。値が大きいほど、その方式によって提供される保護の程度が大きくなります。デフォルトは <b>5</b> です。
ライブラリ	<i>pszLib="smauthhtml"</i> この方式タイプ用のデフォルトライブラリ。

情報タイプ	値の割り当てと意味
パラメータ	<p><i>pszParam=param</i></p> <p>ユーザ属性リストとフォーム認証情報コレクタ (FCC) の場所が含まれる文字列。属性リストは <b>AL=</b> から始まり、リスト区切り文字としてカンマを使用する必要があります。また、セミコロンで終了する必要があります。以下は例です。</p> <p><b>AL=Password,SSN,age,zipcode;</b></p> <p>完全なパラメータ形式は次のとおりです。</p> <p><i>attr-list;https://server/fcc</i></p> <p>以下の例ではデフォルトの FCC を使用します。</p> <p><b>AL=PASSWORD,SSN,age,zipcode;</b>  <b>http://my.server.com/siteminderagent/forms/login.fcc</b></p>
共有秘密キー	<p><i>pszSecret=""</i></p> <p>空の文字列に設定します。この方式には適用できません。</p>
テンプレートですか?	<p><i>bIsTemplate=0</i></p> <p>方式がテンプレートではないことを示すには、<b>false (0)</b> に設定します。他の値は無視されます。</p>
管理者によって使用されますか?	<p><i>bIsUsedbyAdmin=0</i></p> <p><b>false (0)</b> に設定すると、方式は管理者を認証するためには使用されません。</p>
認証情報を保存しますか?	<p><i>bAllowSaveCreds=flag</i></p> <p>ユーザ認証情報が保存されることを示すには <b>true (1)</b> に、ユーザ認証情報が保存されないことを示すには <b>false (0)</b> に設定します。デフォルトは <b>0</b> です。</p>
RADIUS ですか?	<p><i>bIsRadius=0</i></p> <p><b>false (0)</b> に設定すると、方式は RADIUS エージェントと共に使用されません。</p>
パスワードチェックを無視しますか?	<p><i>bIgnorePwCheck=flag</i></p> <p>パスワードチェックを無視するには <b>true (1)</b>、パスワードを確認するには <b>false (0)</b> に設定します。デフォルトは <b>0</b> です。</p>

## インパーソネーション テンプレート

方式タイプ、インパーソネーションに基づいて認証方式を設定する場合はこの表を使用します。この表で参照される構造体フィールドは、`Sm_PolicyApi_Scheme_t`にあります。

情報タイプ	値の割り当てと意味
方式タイプ	<code>nType=Sm_Api_SchemeType_Impersonation</code> インパーソネーション方式タイプです。
説明	<code>pszDesc=description</code> 認証方式の説明。
保護レベル	<code>nLevel=value</code> 1 ~ 1000 の値。値が大きいほど、その方式によって提供される保護の程度が大きくなります。デフォルトは5です。
ライブラリ	<code>pszLib="smauthimpersonate"</code> この方式タイプ用のデフォルトライブラリ。
パラメータ	<code>pszParam=param</code> ユーザ属性リストとフォーム認証情報コレクタ (FCC) の場所が含まれる文字列。属性リストは <code>AL=</code> から始まり、リスト区切り文字としてカンマを使用する必要があります。また、セミコロンで終了する必要があります。以下は例です。 <code>AL=Password,SSN,age,zipcode;</code> 完全なパラメータ形式は次のとおりです。 <code>attr-list;https:/server/fcc</code> 以下の例ではデフォルトの FCC を使用します。 <code>AL=PASSWORD,SSN,age,zipcode;</code> <code>http://my.server.com/siteminderagent/forms/imp.fcc</code>
共有秘密キー	<code>pszSecret=""</code> 空の文字列に設定します。この方式には適用できません。
テンプレートですか?	<code>bIsTemplate=0</code> 方式がテンプレートではないことを示すには、 <code>false (0)</code> に設定します。他の値は無視されます。

情報タイプ	値の割り当てと意味
管理者によって使用されますか?	<i>bIsUsedbyAdmin=0</i> false (0) に設定すると、方式は管理者を認証するためには使用されません。
認証情報を保存しますか?	<i>bAllowSaveCreds=0</i> ユーザ認証情報が保存されないことを示すには、false (0) に設定します。
RADIUS ですか?	<i>bIsRadius=0</i> false (0) に設定すると、方式は RADIUS エージェントと共に使用されません。
パスワードチェックを無視しますか?	<i>bIgnorePwCheck=1</i> true (1) に設定すると、パスワードチェックを無視します。

## RADIUS CHAP/PAP テンプレート

RADIUS CHAP/PAP 方式タイプに基づいて認証方式を設定するときこのテーブルを使用します。この表で参照される構造体フィールドは、*Sm\_PolicyApi\_Scheme\_t* にあります。

情報タイプ	値の割り当てと意味
方式タイプ	<i>nType=Sm_Api_SchemeType_RadiusChapPap</i> RADIUS CHAP/PAP 方式タイプです。
説明	<i>pszDesc=description</i> 認証方式の説明。
保護レベル	<i>nLevel=value</i> 1 ~ 1000 の値。値が大きいほど、その方式によって提供される保護の程度が大きくなります。デフォルトは 5 です。
ライブラリ	<i>pszLib="smauthchap"</i> この方式タイプ用のデフォルトライブラリ。
パラメータ	<i>pszParam=param</i> ユーザディレクトリ属性の名前が含まれる文字列。この属性は認証用のクリアテキストパスワードとして使用されます。

情報タイプ	値の割り当てと意味
共有秘密キー	<code>pszSecret=""</code> 空の文字列に設定します。この方式には適用できません。
テンプレートですか?	<code>bIsTemplate=0</code> 方式がテンプレートではないことを示すには、 <b>false (0)</b> に設定します。他の値は無視されます。
管理者によって使用されますか?	<code>bIsUsedbyAdmin=0</code> <b>false (0)</b> に設定すると、方式は管理者を認証するためには使用されません。
認証情報を保存しますか?	<code>bAllowSaveCreds=0</code> ユーザ認証情報が保存されないことを示すには、 <b>false (0)</b> に設定します。
RADIUS ですか?	<code>bIsRadius=1</code> <b>true (1)</b> に設定すると、方式は RADIUS エージェントと共に使用できます。
パスワードチェックを無視しますか?	<code>bIgnorePwCheck=flag</code> パスワードチェックを無視するには <b>true (1)</b> 、パスワードを確認するには <b>false (0)</b> に設定します。デフォルトは <b>0</b> です。

## RADIUS サーバ テンプレート

RADIUS サーバ方式タイプに基づいて認証方式を設定するときにこのテーブルを使用します。この表で参照される構造体フィールドは、`Sm_PolicyApi_Scheme_t` にあります。

情報タイプ	値の割り当てと意味
方式タイプ	<code>nType=Sm_Api_SchemeType_RadiusServer</code> RADIUS サーバ方式タイプです。
説明	<code>pszDesc=description</code> 認証方式の説明。
保護レベル	<code>nLevel=value</code> 1 ~ 1000 の値。値が大きいほど、その方式によって提供される保護の程度が大きくなります。デフォルトは <b>5</b> です。

情報タイプ	値の割り当てと意味
ライブラリ	<i>pszLib</i> ="smauthradius" この方式タイプ用のデフォルトライブラリ。
パラメータ	<i>pszParam</i> = <i>param</i> IP アドレスおよび RADIUS サーバのポートが含まれる文字列。以下は例です。 <b>123.123.12.12:1645</b> デフォルト UDP ポートは <b>1645</b> です。
共有秘密キー	<i>pszSecret</i> = <i>secret</i> RADIUS サーバがクリア テキストパスワードとして使用するユーザ属性。
テンプレートですか?	<i>blsTemplate</i> =0 方式がテンプレートではないことを示すには、 <b>false (0)</b> に設定します。他の値は無視されます。
管理者によって使用されますか?	<i>blsUsedbyAdmin</i> =1 <b>true (1)</b> に設定すると、方式は管理者を認証するために使用できます。
認証情報を保存しますか?	<i>bAllowSaveCreds</i> =0 ユーザ認証情報が保存されないことを示すには、 <b>false (0)</b> に設定します。
RADIUS ですか?	<i>blsRadius</i> =1 <b>true (1)</b> に設定すると、方式は RADIUS エージェントと共に使用できます。
パスワードチェックを無視しますか?	<i>blgnorePwCheck</i> = <i>flag</i> パスワードチェックを無視するには <b>true (1)</b> 、パスワードを確認するには <b>false (0)</b> に設定します。デフォルトは <b>0</b> です。

## SafeWord HTML フォーム テンプレート

SafeWord HTML フォーム方式タイプに基づいて認証方式を設定するときにこのテーブルを使用します。この表で参照される構造体フィールドは、`Sm_PolicyApi_Scheme_t`にあります。

情報タイプ	値の割り当てと意味
方式タイプ	<code>nType=Sm_Api_SchemeType_SafeWordHTMLForm</code> SafeWord HTML フォーム方式タイプ。
説明	<code>pszDesc=description</code> 認証方式の説明。
保護レベル	<code>nLevel=value</code> 1 ~ 1000 の値。値が大きいほど、その方式によって提供される保護の程度が大きくなります。デフォルトは <b>10</b> です。
ライブラリ	<code>pszLib="smauthenigmahtml"</code> この方式タイプ用のデフォルトライブラリ。
パラメータ	<code>pszParam=param</code> フォーム認証情報コレクタの名前および場所が含まれている文字列。この例はデフォルトの認証情報コレクタを示しています。 <code>http://my.server.com/siteminderagent/forms/safeword.fcc</code>
共有秘密キー	<code>pszSecret=""</code> 空の文字列に設定します。この方式には適用できません。
テンプレートですか?	<code>bIsTemplate=0</code> 方式がテンプレートではないことを示すには、 <b>false (0)</b> に設定します。他の値は無視されます。
管理者によって使用されますか?	<code>bIsUsedbyAdmin=1</code> <b>true (1)</b> に設定すると、方式は管理者を認証するために使用できます。
認証情報を保存しますか?	<code>bAllowSaveCreds=0</code> ユーザ認証情報が保存されないことを示すには、 <b>false (0)</b> に設定します。

情報タイプ	値の割り当てと意味
RADIUS ですか?	<i>blsRadius=1</i> true (1) に設定すると、方式は RADIUS エージェントと共に使用できます。
パスワードチェックを無視しますか?	<i>blgnorePwCheck=1</i> true (1) に設定すると、パスワードチェックを無視します。

## SafeWord テンプレート

SafeWord 方式タイプに基づいて認証方式を設定するときにこのテーブルを使用します。この表で参照される構造体フィールドは、`Sm_PolicyApi_Scheme_t`にあります。

情報タイプ	値の割り当てと意味
方式タイプ	<i>nType=Sm_Api_SchemeType_SafeWordServer</i> SafeWord 方式タイプ。
説明	<i>pszDesc=description</i> 認証方式の説明。
保護レベル	<i>nLevel=value</i> 1 ~ 1000 の値。値が大きいほど、その方式によって提供される保護の程度が大きくなります。デフォルトは <b>10</b> です。
ライブラリ	<i>pszLib="smauthenigma"</i> この方式タイプ用のデフォルトライブラリ。
パラメータ	<i>pszParam=""</i> 空の文字列に設定します。この方式には適用できません。
共有秘密キー	<i>pszSecret=""</i> 空の文字列に設定します。この方式には適用できません。
テンプレートですか?	<i>blsTemplate=0</i> 方式がテンプレートではないことを示すには、 <b>false (0)</b> に設定します。他の値は無視されます。

情報タイプ	値の割り当てと意味
管理者によって使用されますか?	<i>bIsUsedbyAdmin=1</i> true (1) に設定すると、方式は管理者を認証するために使用できます。
認証情報を保存しますか?	<i>bAllowSaveCreds=0</i> ユーザ認証情報が保存されないことを示すには、false (0) に設定します。
RADIUS ですか?	<i>bIsRadius=1</i> true (1) に設定すると、方式は RADIUS エージェントと共に使用できます。
パスワードチェックを無視しますか?	<i>bIgnorePwCheck=1</i> true (1) に設定すると、パスワードチェックを無視します。

## SAML Artifact テンプレート

セキュリティアサーションを伝えるためのアーチファクトプロファイルタイプに基づいて SAML 認証方式を設定するときはこのテーブルを使用します。アーチファクトプロファイルタイプでは、SAML アサーションを取得するための URL がパラメータ文字列の *AssertionRetrievalURL* 部分内で参照されます。

この表で参照される構造体フィールドは、*Sm\_PolicyApi\_Scheme\_t* にあります。

情報タイプ	値の割り当てと意味
方式タイプ	<i>nType=Sm_Api_SchemeType_SAMLArtifact</i> SAML アーチファクト方式タイプ。
説明	<i>pszDesc=description</i> 認証方式の説明。
保護レベル	<i>nLevel=value</i> 1 ~ 1000 の値。値が大きいほど、その方式によって提供される保護の程度が大きくなります。デフォルトは 5 です。
ライブラリ	<i>pszLib="smauthsaml"</i> この方式タイプ用のデフォルトライブラリ。

### 情報タイプ

#### パラメータ

### 値の割り当てと意味

*pszParam=param*

必須パラメータは以下のとおりです。

- **Name**。アフィリエイトの名前です。
- **RedirectMode**。SAML 認証情報コレクタがターゲット リソースにリダイレクトする方法です。以下の数値のいずれかになります。
  0. 意味： 302 データなし。
  1. 意味： 302 Cookie データ。
  2. 意味： サーバリダイレクト。
  3. 意味： 永続属性。
- **SRCID**。SAML アサーションを作成するサイト用の 20 バイトのソース ID。ID はプロパティ ファイル `AMAssertionGenerator.properties` 内の SAML プロデューサ側にあります。
- **AssertionRetrievalURL**。SAML アサーションプロデューサ側からアサーションを取得するための URL。
- **Audience**。ポータルとアフィリエイトの間の許諾契約について説明するドキュメントの URI。この値は SAML アサーションで指定されたオーディエンス値と比較されます。
- **Issuer**。アサーションで指定された SAML 発行者。
- **AttributeXPath**。SAML アサーションに対して実行される標準の XPath クエリ。このクエリは、ユーザを検索する検索仕様で代用されるデータを取得します。以下は例です。

```
//saml:AttributeValue/SM:/SMContent  
/SM:Smlogin/SM:Username.text()
```

情報タイプ

値の割り当てと意味

このクエリは、Username エLEMENTのテキストを取得します。

- SAMLVersion。使用中の SAML バージョン：1.0 または 1.1。
- RetrievalMethod。次のいずれかの値を取ります。

0. 意味：基本認証。

1. 意味：クライアント証明書認証。

- Attribute。指定されたタイプのユーザディレクトリ内でユーザを探すための検索文字列。XPath クエリから返された値を挿入する場所を示すにはパーセント記号（%）を使用します。たとえば、属性 LDAP:uid=%s を指定し、クエリから user1 が返される場合、LDAP ディレクトリに使用される検索文字列は uid=user1 です。少なくとも 1 つの属性を指定する必要があります。

このパラメータ文字列の形式は以下のとおりです。セミコロン（;）で名前/値ペアを区切ります。この形式の例には LDAP および ODBC 属性が含まれます。

```
Name=name;RedirectMode=0|1|2;SRCID=srcid;
AssertionRetrievalURL=url;Audience=audience;
Issuer=issuer;AttributeXPath=XPathQuery;
SAMLVersion=1.0|1.1;RetrievalMethod=0|1;
attribute=LDAP:srchSpc;attribute=ODBC:srchSpc
```

共有秘密キー

*pszSecret=secret*

アフィリエイトサイトのパスワード。

テンプレートですか?

*blsTemplate=0*

方式がテンプレートではないことを示すには、false (0) に設定します。他の値は無視されます。

管理者によって使用されますか?

*blsUsedbyAdmin=0*

false (0) に設定すると、方式は管理者を認証するためには使用されません。

認証情報を保存しますか?

*bAllowSaveCreds=0*

ユーザ認証情報が保存されないことを示すには、false (0) に設定します。

RADIUS ですか?

*blsRadius=0*

false (0) に設定すると、方式は RADIUS エージェントと共に使用されません。

情報タイプ

値の割り当てと意味

パスワードチェックを  
無視しますか?

`blgnorePwCheck=1`  
`true (1)` に設定すると、パスワードチェックを無視します。

## SAML POST テンプレート

セキュリティアサーションを伝えるための *POST* プロファイルタイプに基づいて SAML 認証方式を設定するときにこのテーブルを使用します。*POST* プロファイルタイプでは、生成された SAML アサーションがパラメータ文字列の *AssertionConsumerURL* 部分内で指定された URL に *POST* 処理されます。

この表で参照される構造体フィールドは、*Sm\_PolicyApi\_Scheme\_t* にあります。

情報タイプ

値の割り当てと意味

方式タイプ

`nType=Sm_Api_SchemeType_SAMLPOST`  
SAML POST 方式タイプ。

説明

`pszDesc=description`  
認証方式の説明。

保護レベル

`nLevel=value`  
1 ~ 1000 の値。値が大きいほど、その方式によって提供される保護の程度が大きくなります。デフォルトは 5 です。

ライブラリ

`pszLib="smauthsaml"`  
この方式タイプ用のデフォルトライブラリ。

## 情報タイプ

## パラメータ

## 値の割り当てと意味

*pszParam=param*

必須パラメータは以下のとおりです。

- **Name**。アフィリエイトの名前です。
- **SAMLProfile**。プロファイルタイプ：POST。
- **SAMLVersion**。使用中の SAML バージョン。POST プロファイルにはバージョン 1.1 が必要です。
- **RedirectMode**。SAML 認証情報コレクタがターゲットリソースにリダイレクトする方法です。以下の数値のいずれかになります。
  0. 意味：302 データなし。
  1. 意味：302 Cookie データ。
  2. 意味：サーバリダイレクト。
  3. 意味：永続属性
- **AssertionConsumerURL**。生成されたアサーションの送信先 URL。
- **Audience**。ポータルとアフィリエイトの間の許諾契約について説明するドキュメントの URI。この値は SAML アサーションで指定されたオーディエンス値と比較されます。
- **Issuer**。アサーションで指定された SAML 発行者。

情報タイプ

値の割り当てと意味

パラメータ (続き)

- **AttributeXPath**。 SAML アサーションに対して実行される標準の XPath クエリ。 このクエリは、ユーザを検索する検索仕様で代用されるデータを取得します。 以下は例です。

```
//saml:AttributeValue/SM:/SMContent
/SM:Smlogin/SM:Username.text()
```

このクエリは、 **Username** エレメントのテキストを取得します。

- **attribute**。 指定されたタイプのユーザディレクトリ内でユーザを探すための検索文字列。 XPath クエリから返された値を挿入する場所を示すにはパーセント記号 (%) を使用します。 たとえば、属性 **LDAP:uid=%s** を指定し、クエリから **user1** が返される場合、LDAP ディレクトリに使用される検索文字列は **uid=user1** です。 少なくとも 1 つの属性を指定する必要があります。

このパラメータ文字列の形式は以下のとおりです。 セミコロン (;) で名前/値ペアを区切ります。 この形式の例には LDAP および ODBC 属性が含まれます。

```
Name=name;SAMLProfile=POST;
SAMLVersion=1.1;RedirectMode=0|1|2;
AssertionConsumerURL=consumerUrl;
Audience=audience;Issuer=issuer;
AttributeXPath=XPathQuery;
attribute=LDAP:srchSpc;attribute=ODBC:srchSpc
```

共有秘密キー

**pszSecret=""**

空の文字列に設定します。 この方式には適用できません。

テンプレートですか?

**blsTemplate=0**

方式がテンプレートではないことを示すには、 **false (0)** に設定します。 他の値は無視されます。

管理者によって使用されますか?

**blsUsedbyAdmin=0**

**false (0)** に設定すると、方式は管理者を認証するためには使用されません。

認証情報を保存しますか?

**bAllowSaveCreds=0**

ユーザ認証情報が保存されないことを示すには、 **false (0)** に設定します。

RADIUS ですか?

**blsRadius=0**

**false (0)** に設定すると、方式は RADIUS エージェントと共に使用されません。

情報タイプ	値の割り当てと意味
パスワードチェックを無視しますか?	<code>blgnorePwCheck=1</code> true (1) に設定すると、パスワードチェックを無視します。

## SAML 2.0 テンプレート

SAML 2.0 方式タイプに基づいて SAML 認証方式を設定する場合はこの表を使用します。サービスプロバイダでは、SAML 2.0 アサーション内の情報に基づいてユーザを透過的に検証するためにこの認証方式を使用します。この透過的な検証は、シングルサインオンやシングルログアウトなどの機能を許可します。

SAML 2.0 認証方式を設定する場合は、関連するアイデンティティプロバイダ（すなわち、サービスプロバイダにアサーションを提供するアイデンティティプロバイダ）のメタデータプロパティも定義します。

アイデンティティプロバイダのプロパティはプロパティの個別のセットとして認証方式オブジェクトを用いて保存されます。その結果、2つの構造体が SAML 2.0 認証方式を設定するために使用されます。

- 以下の表で参照される構造体フィールドは、`Sm_PolicyApi_Scheme_t` にあります。
- 関連するアイデンティティプロバイダのメタデータプロパティは `Sm_PolicyApi_SAMLProviderProp_t` によって定義されます。

情報タイプ	値の割り当てと意味
方式タイプ	<code>nType=Sm_Api_SchemeType_SAML2</code> SAML 2.0 方式タイプ。
説明	<code>pszDesc=description</code> 認証方式の説明。
保護レベル	<code>nLevel=value</code> 1 ~ 1000 の値。値が大きいくほど、その方式によって提供される保護の程度が大きくなります。デフォルトは5です。
ライブラリ	<code>pszLib="smauthsaml"</code> この方式タイプ用のデフォルトライブラリ。

情報タイプ	値の割り当てと意味
パラメータ	<p><i>pszParam=""</i></p> <p>空の文字列に設定します。 SiteMinder はパラメータ値を割り当てます。</p> <p>パラメータは、関連するアイデンティティ プロバイダの SAML 2.0 メタデータ プロパティへの参照です。 プロパティは <i>Sm_PolicyApi_SAMLProviderProp_t</i> によって定義されます。</p>
共有秘密キー	<p><i>pszSecret=""</i></p> <p>空の文字列に設定します。 この方式には適用できません。</p>
テンプレートですか?	<p><i>blsTemplate=0</i></p> <p>方式がテンプレートではないことを示すには、 <b>false (0)</b> に設定します。 他の値は無視されます。</p>
管理者によって使用されますか?	<p><i>blsUsedbyAdmin=0</i></p> <p><b>false (0)</b> に設定すると、方式は管理者を認証するためには使用されません。</p>
認証情報を保存しますか?	<p><i>bAllowSaveCreds=0</i></p> <p>ユーザ認証情報が保存されないことを示すには、 <b>false (0)</b> に設定します。</p>
RADIUS ですか?	<p><i>blsRadius=0</i></p> <p><b>false (0)</b> に設定すると、方式は RADIUS エージェントと共に使用されません。</p>
パスワードチェックを無視しますか?	<p><i>blgnorePwCheck=1</i></p> <p><b>true (1)</b> に設定すると、パスワードチェックを無視します。</p>

### 詳細情報:

[カスタム エージェントおよびシングルサインオン \(P. 49\)](#)

## SecurID HTML フォーム テンプレート

SecurID HTML フォーム方式タイプに基づいて認証方式を設定するときこのテーブルを使用します。この表で参照される構造体フィールドは、`Sm_PolicyApi_Scheme_t`にあります。

情報タイプ	値の割り当てと意味
方式タイプ	<code>nType=Sm_Api_SchemeType_ACEServerHTMLForm</code> SecurID HTML フォーム方式タイプ。
説明	<code>pszDesc=description</code> 認証方式の説明。
保護レベル	<code>nLevel=value</code> 1 ~ 1000 の値。値が大きいほど、その方式によって提供される保護の程度が大きくなります。デフォルトは 15 です。
ライブラリ	<code>pszLib="smauthacehtml"</code> この方式タイプ用のデフォルトライブラリ。
パラメータ	<code>pszParam=param</code> ACE ID が保存されている属性の名前、フォーム認証情報コレクタ (FCC) がインストールされている Web サーバ、およびフォームサポートで SecurID 認証の処理に必要なターゲット実行可能ファイルが含まれている文字列。また、SSL 接続が必要かどうかを指定します。形式 <code>attr;https://server/target</code> 注: 「https」の「s」はオプションで、SSL 接続を行うかどうかを示します。 以下の例では、フォームサポートで SecurID 認証を処理するためのデフォルトを使用します。 <code>ace_id;https://my.server.com/siteminderagent/forms/smpwservices.fcc</code> .
共有秘密キー	<code>pszSecret=""</code> 空の文字列に設定します。この方式には適用できません。
テンプレートですか?	<code>bIsTemplate=0</code> 方式がテンプレートではないことを示すには、 <code>false (0)</code> に設定します。他の値は無視されます。

情報タイプ	値の割り当てと意味
管理者によって使用されますか?	<i>bIsUsedbyAdmin=0</i> false (0) に設定すると、方式は管理者を認証するためには使用されません。
認証情報を保存しますか?	<i>bAllowSaveCreds=0</i> ユーザ認証情報が保存されないことを示すには、false (0) に設定します。
RADIUS ですか?	<i>bIsRadius=0</i> false (0) に設定すると、方式は RADIUS エージェントと共に使用されません。
パスワードチェックを無視しますか?	<i>bIgnorePwCheck=1</i> true (1) に設定すると、パスワードチェックを無視します。

## SecurID テンプレート

方式タイプ SecurID に基づいて認証方式を設定する場合はこの表を使用します。この表で参照される構造体フィールドは、Sm\_PolicyApi\_Scheme\_t にあります。

情報タイプ	値の割り当てと意味
方式タイプ	<i>nType=Sm_Api_SchemeType_ACEServer</i> SecurID 方式タイプ。
説明	<i>pszDesc=description</i> 認証方式の説明。
保護レベル	<i>nLevel=value</i> 1 ~ 1000 の値。値が大きいほど、その方式によって提供される保護の程度が大きくなります。デフォルトは 15 です。
ライブラリ	<i>pszLib="smauthace"</i> この方式タイプ用のデフォルトライブラリ。
パラメータ	<i>pszParam=param</i> ACE サーバユーザ ID を含む認証ユーザディレクトリの属性が含まれる文字列。

情報タイプ	値の割り当てと意味
共有秘密キー	<code>pszSecret=""</code> 空の文字列に設定します。この方式には適用できません。
テンプレートですか?	<code>bIsTemplate=0</code> 方式がテンプレートではないことを示すには、 <b>false (0)</b> に設定します。他の値は無視されます。
管理者によって使用されますか?	<code>bIsUsedbyAdmin=1</code> <b>true (1)</b> に設定すると、方式は管理者を認証するために使用できます。
認証情報を保存しますか?	<code>bAllowSaveCreds=0</code> ユーザ認証情報が保存されないことを示すには、 <b>false (0)</b> に設定します。
RADIUS ですか?	<code>bIsRadius=1</code> <b>true (1)</b> に設定すると、方式は RADIUS エージェントと共に使用できます。
パスワードチェックを無視しますか?	<code>bIgnorePwCheck=1</code> <b>true (1)</b> に設定すると、パスワードチェックを無視します。

## smauthetsso 認証方式

この認証方式は SiteMinder X.509 証明書方式と似ていますが、X.509 認証情報の代わりに認証情報として eSSO Cookie を使用します。

この方式が cookieorbasic または cookieorforms モードのいずれかに対して設定され、eSSO Cookie とログイン名およびパスワード認証情報の両方がそれに渡される場合、eSSO Cookie は無視され、ログイン名とパスワードは SiteMinder へのユーザを認証するために使用されます。

eSSO Cookie が唯一の認証情報である場合、認証方式では、Cookie を検証し、そこからユーザ識別名 (DN) を抽出するために、ETWAS API を使用して、設定された eSSO ポリシー サーバに接続します。

カスタム方式タイプに基づく smauthetsso 認証方式を設定する場合はこの表を使用します。この表で参照される構造体フィールドは、Sm\_PolicyApi\_Scheme\_t にあります。

情報タイプ	値の割り当てと意味
方式タイプ	<i>nType=Sm_Api_SchemeType_Custom</i> カスタム方式タイプを使用します。
説明	<i>pszDesc=description</i> 認証方式の説明。
保護レベル	<i>nLevel=value</i> 0 ~ 1000 の値。値が大きいほど、その方式によって提供される保護の程度が大きくなります。デフォルトは 5 です。
ライブラリ	<i>pszLib="smauthetsso"</i> この認証方式のライブラリの名前。

情報タイプ	値の割り当てと意味
パラメータ	<p><i>pszParam=param</i></p> <p>以下のようにセミコロンによって分離され、順序付けられたトークンのセット。</p> <p><i>&lt;Mode&gt;[; &lt;Target&gt;]; &lt;Admin&gt;; &lt;eTPS_Host&gt;</i></p> <p>文字列を読み取りやすいようにスペースを追加できます。</p> <p><i>&lt;Mode&gt;</i> は、認証方式が受け入れる認証情報のタイプを指定します。以下の値が可能です。</p> <ul style="list-style-type: none"> <li>■ <i>cookie</i> -- eTrust SSO Cookie のみが受け入れ可能です</li> <li>■ <i>cookieorbasic</i> -- eTrust SSO Cookie が提供されない場合は、ログイン名とパスワードが基本認証の使用によって要求されます。</li> <li>■ <i>cookieorforms</i> -- eTrust SSO Cookie が提供されない場合は、ログイン名とパスワードが形式認証の使用によって要求されます。</li> </ul> <p><i>&lt;Target&gt;</i> は <i>cookieorforms</i> モードでのみ有効です。これは標準の HTML フォーム認証方式用の <i>Target</i> フィールドと同一です。</p> <p><i>&lt;Admin&gt;</i> は、eTrust ポリシー サーバの管理者のログイン ID を指定します。この管理者のパスワードは [共有秘密キー] フィールドで指定されています。</p> <p><i>&lt;eTPO_Host&gt;</i> は、ポリシー サーバがインストールされているマシンの名前を指定します。</p> <p>SiteMinder は <i>&lt;eTPS_Host&gt;</i> 上の eTrust ポリシー サーバへの <i>&lt;Admin&gt;</i> としてそれ自体を認証し、SiteMinder が eTrust SSO Cookie の検証を要求できるようします。</p> <p>例：</p> <pre>pszParam="cookie; SMPS_sso; myserver.myco.com" pszParam="cookieorforms; /siteminderagent/forms/login.fcc; SMPS_sso; myserver.myco.com"</pre>
共有秘密キー	<p><i>pszSecret=secret</i></p> <p>[パラメータ] フィールドで指定される eTrust ポリシー サーバ管理者のパスワード。</p>
テンプレートですか?	<p><i>bIsTemplate=0</i></p> <p>方式がテンプレートではないことを示すには、<b>false (0)</b> に設定します。他の値は無視されます。</p>

情報タイプ	値の割り当てと意味
管理者によって使用されますか?	<i>bIsUsedbyAdmin=flag</i> 管理者の認証に方式を使用できることを指定するには <b>true (1)</b> 、または管理者の認証に方式を使用できないことを指定するには <b>false (0)</b> を指定します。デフォルトは <b>0</b> です。
認証情報を保存しますか?	<i>bAllowSaveCreds=0</i> ユーザ認証情報が保存されないことを示すには、 <b>false (0)</b> に設定します。
RADIUS ですか?	<i>bIsRadius=0</i> <b>false (0)</b> に設定すると、方式は RADIUS エージェントと共に使用されません。
パスワードチェックを無視しますか?	<i>bIgnorePwCheck=flag</i> パスワードチェックを無視するには <b>true (1)</b> 、パスワードを確認するには <b>false (0)</b> に設定します。デフォルトは <b>0</b> です。

## TeleID テンプレート

方式タイプ TeleID に基づいて認証方式を設定する場合はこの表を使用します。この表で参照される構造体フィールドは、Sm\_PolicyApi\_Scheme\_t にあります。

情報タイプ	値の割り当てと意味
方式タイプ	<i>nType=Sm_Api_SchemeType_Encotone</i> TeleID 方式タイプ。
説明	<i>pszDesc=description</i> 認証方式の説明。
保護レベル	<i>nLevel=value</i> 1 ~ 1000 の値。値が大きいほど、その方式によって提供される保護の程度が大きくなります。デフォルトは <b>15</b> です。
ライブラリ	<i>pszLib="smauthencotone"</i> この方式タイプ用のデフォルトライブラリ。
パラメータ	<i>pszParam=""</i> 空の文字列に設定します。この方式には適用できません。

情報タイプ	値の割り当てと意味
共有秘密キー	<i>pszSecret=seed</i> 暗号化シード。SiteMinder はハードウェア トークンを初期化するための暗号化シードとしてこの値を使用します。
テンプレートですか?	<i>blsTemplate=0</i> 方式がテンプレートではないことを示すには、 <b>false (0)</b> に設定します。他の値は無視されます。
管理者によって使用されますか?	<i>blsUsedbyAdmin=1</i> <b>true (1)</b> に設定すると、方式は管理者を認証するために使用できます。
認証情報を保存しますか?	<i>bAllowSaveCreds=0</i> ユーザ認証情報が保存されないことを示すには、 <b>false (0)</b> に設定します。
RADIUS ですか?	<i>blsRadius=1</i> <b>true (1)</b> に設定すると、方式は RADIUS エージェントと共に使用できます。
パスワードチェックを無視しますか?	<i>blgnorePwCheck=1</i> <b>true (1)</b> に設定すると、パスワードチェックを無視します。

## Windows 認証テンプレート

Windows 認証（以前は NTLM と呼ばれていた）方式タイプに基づいて統合 Windows 認証方式を設定する場合はこの表を使用します。この方式タイプは WinNT または Active Directory ユーザストアに対して認証を行うために使用されます。

Active Directory は混在モードまたはネイティブモードで実行されるように設定できます。混在モードで実行される場合、Active Directory は WinNT スタイル認証をサポートします。ネイティブモードでは、Active Directory は LDAP スタイル検索のみをサポートします。

この認証方式は混合モードまたはネイティブモードのいずれかをサポートします。

この表で参照される構造体フィールドは、Sm\_PolicyApi\_Scheme\_t にあります。

情報タイプ	値の割り当てと意味
方式タイプ	<i>nType</i> =Sm_Api_SchemeType_NTLM Windows 認証（NTLM）方式タイプ。
説明	<i>pszDesc</i> =description 認証方式の説明。
保護レベル	<i>nLevel</i> =value 1 ~ 1000 の値。値が大きいほど、その方式によって提供される保護の程度が大きくなります。デフォルトは 5 です。
ライブラリ	<i>pszLib</i> ="smauthntlm" この方式タイプ用のデフォルトライブラリ。

情報タイプ	値の割り当てと意味
パラメータ	<p><i>pszParam=param</i></p> <p><i>pszParam</i> の値は、この方式に対して実行する認証のスタイルを決定します。</p> <p><b>NTLM 認証</b> (混合モードで実行される WinNT または Active Directory 用)</p> <p>形式</p> <p><i>iis-web-server-url/path-to-ntc-file</i></p> <p>この形式では、<i>iis-web-server-url</i> は、リダイレクトのターゲットである IIS Web サーバの名前です。また、<i>path-to-ntc-file</i> は WinNT 認証情報を収集する .ntc ファイルの場所です。</p> <p>例 :</p> <p><code>http://myiiswebserver.mycompany.com/ siteminderagent/ntlm/creds.ntc</code></p> <p>指定されたサーバに SiteMinder Web エージェントをインストールする必要があります。デフォルトでは、Web エージェントのインストールによって、NTLM 認証情報収集用の仮想ディレクトリが作成されます。</p> <p><b>Windows 認証</b> (ネイティブ モードで実行される Active Directory 用)</p> <p>この認証スタイルでは、<i>pszParam</i> はリダイレクト URL の先頭に追加された LDAP フィルタがあります。フィルタと URL はセミコロン (;) によって区切ります。例 :</p> <p><code>cn=%{UID},ou=Users,ou=USA,dc=%{DOMAIN}, dc=mycompany,dc=com;http:// myiiswebserver.mycompany.com/ siteminderagent/ntlm/creds.ntc</code></p> <p>SiteMinder は、LDAP フィルタを使用してブラウザ/Web エージェントから受け取った認証情報を LDAP DN または検索フィルタにマップします。</p>
共有秘密キー	<p><i>pszSecret=""</i></p> <p>空の文字列に設定します。この方式には適用できません。</p>
テンプレートですか?	<p><i>bIsTemplate=0</i></p> <p>方式がテンプレートではないことを示すには、<b>false (0)</b> に設定します。他の値は無視されます。</p>

情報タイプ	値の割り当てと意味
管理者によって使用されますか?	<i>bIsUsedbyAdmin=0</i> false (0) に設定すると、方式は管理者を認証するためには使用されません。
認証情報を保存しますか?	<i>bAllowSaveCreds=0</i> ユーザ認証情報が保存されないことを示すには、false (0) に設定します。
RADIUS ですか?	<i>bIsRadius=0</i> false (0) に設定すると、方式は RADIUS エージェントと共に使用されません。
パスワードチェックを無視しますか?	<i>bIgnorePwCheck=flag</i> 混在モードで実行される WinNT および Active Directory の場合、このプロパティは true (1) (パスワードチェックを無視する) に設定する必要があります。 ネイティブ モードで Active Directory を実行している場合は、true (1) に設定してパスワード確認を無視するか、false (0) に設定してパスワードを確認します。デフォルトは 0 です。

## WS-Federation テンプレート

WSFED 方式タイプに基づいて WSFED 認証方式を設定する場合はこの表を使用します。リソース パートナーはこの認証方式を使用して、SAML 1.0 アサーション内の情報に基づいてユーザを透過的に検証します。この透過的な検証は、シングルサインオンやシングル ログアウトなどの機能を許可します。

WSFED 認証方式を設定する場合は、関連するアカウント パートナ（すなわち、リソース パートナーにアサーションを提供するアカウント パートナ）のメタデータ プロパティも定義します。

アカウント パートナの プロパティはプロパティの個別のセットとして認証方式オブジェクトを用いて保存されます。その結果、2つの構造体が WSFED 認証方式を設定するために使用されます。

- 以下の表で参照される構造体フィールドは、Sm\_PolicyApi\_Scheme\_t にあります。
- 関連するアカウント パートナのメタデータ プロパティは Sm\_PolicyApi\_WSFEDProviderProp\_t によって定義されます。

情報タイプ	値の割り当てと意味
方式タイプ	<i>nType</i> =Sm_Api_SchemeType_WSFED WSFED 方式タイプ。
説明	<i>pszDesc</i> = <i>description</i> 認証方式の説明。
保護レベル	<i>nLevel</i> = <i>value</i> 1 ~ 1000 の値。値が大きいほど、その方式によって提供される保護の程度が大きくなります。デフォルトは 5 です。
ライブラリ	<i>pszLib</i> ="smauthsaml" この方式タイプ用のデフォルト ライブラリ。

情報タイプ	値の割り当てと意味
パラメータ	<p><code>pszParam=""</code></p> <p>空の文字列に設定します。 SiteMinder はパラメータ値を割り当てません。</p> <p>パラメータは、関連するアカウント パートナの WSFED メタデータ プロパティへの参照です。 プロパティは <code>Sm_PolicyApi_WSFEDProviderProp_t</code> によって定義されます。</p>
共有秘密キー	<p><code>pszSecret=""</code></p> <p>空の文字列に設定します。 この方式には適用できません。</p>
テンプレートですか?	<p><code>blsTemplate=0</code></p> <p>方式がテンプレートではないことを示すには、 <b>false (0)</b> に設定します。 他の値は無視されます。</p>
管理者によって使用されますか?	<p><code>blsUsedbyAdmin=0</code></p> <p><b>false (0)</b> に設定すると、方式は管理者を認証するためには使用されません。</p>
認証情報を保存しますか?	<p><code>bAllowSaveCreds=0</code></p> <p>ユーザ認証情報が保存されないことを示すには、 <b>false (0)</b> に設定します。</p>
RADIUS ですか?	<p><code>blsRadius=0</code></p> <p><b>false (0)</b> に設定すると、方式は RADIUS エージェントと共に使用されません。</p>
パスワードチェックを無視しますか?	<p><code>blgnorePwCheck=1</code></p> <p><b>true (1)</b> に設定すると、パスワードチェックを無視します。</p>

詳細情報:

[WS-フェデレーション](#) (P. 143)

## X.509 クライアント証明書および基本テンプレート

X.509 クライアント証明書および基本方式タイプに基づいて認証方式を設定するときにこのテーブルを使用します。この表で参照される構造体フィールドは、Sm\_PolicyApi\_Scheme\_t にあります。

情報タイプ	値の割り当てと意味
方式タイプ	<i>nType</i> = Sm_Api_SchemeType_X509ClientCertAndBasic X.509 クライアント証明書および基本方式タイプ。
説明	<i>pszDesc</i> = <i>description</i> 認証方式の説明。
保護レベル	<i>nLevel</i> = <i>value</i> 1 ~ 1000 の値。値が大きいほど、その方式によって提供される保護の程度が大きくなります。デフォルトは 15 です。
ライブラリ	<i>pszLib</i> ="smauthcert" この方式タイプ用のデフォルトライブラリ。
パラメータ	<i>pszParam</i> = <i>param</i> SSL サーバのドメインまたは IP アドレス、および SSL 認証情報コレクタ (SCC) の名前とパスが含まれる文字列。サーバは、ユーザの X.509 証明書を SSL 接続を経由してリダイレクトします。形式 <code>https://server:port/SCC?cert+basic</code> 以下の例ではデフォルトの SCC を使用します。 <code>https://my.server.com:80/siteminderagent/cert/smgetcred.scc?cert+basic</code>
共有秘密キー	<i>pszSecret</i> ="" 空の文字列に設定します。この方式には適用できません。
テンプレートですか?	<i>bIsTemplate</i> =0 方式がテンプレートではないことを示すには、 <b>false (0)</b> に設定します。他の値は無視されます。
管理者によって使用されますか?	<i>bIsUsedbyAdmin</i> =0 <b>false (0)</b> に設定すると、方式は管理者を認証するためには使用されません。

情報タイプ	値の割り当てと意味
認証情報を保存しますか?	<i>bAllowSaveCreds=0</i> ユーザ認証情報が保存されないことを示すには、 <b>false (0)</b> に設定します。
RADIUS ですか?	<i>bIsRadius=0</i> <b>false (0)</b> に設定すると、方式は RADIUS エージェントと共に使用されません。
パスワードチェックを無視しますか?	<i>bIgnorePwCheck=flag</i> パスワードチェックを無視するには <b>true (1)</b> 、パスワードを確認するには <b>false (0)</b> に設定します。デフォルトは <b>0</b> です。

## X.509 クライアント証明書およびフォーム テンプレート

X.509 クライアント証明書およびフォーム方式タイプに基づいて認証方式を設定するときこのテーブルを使用します。この表で参照される構造体フィールドは、*Sm\_PolicyApi\_Scheme\_t* にあります。

情報タイプ	値の割り当てと意味
方式タイプ	<i>nType=</i> <i>Sm_Api_SchemeType_X509ClientCertAndForm</i> X.509 クライアント証明書および HTML フォーム方式タイプ。
説明	<i>pszDesc=description</i> 認証方式の説明。
保護レベル	<i>nLevel=value</i> 1 ~ 1000 の値。値が大きいほど、その方式によって提供される保護の程度が大きくなります。デフォルトは <b>15</b> です。
ライブラリ	<i>pszLib="smauthcert"</i> この方式タイプ用のデフォルトライブラリ。

情報タイプ	値の割り当てと意味
パラメータ	<p><i>pszParam=param</i></p> <p>SSL サーバのドメインまたは IP アドレス、およびフォーム認証情報コレクタ (FCC) の名前とパスが含まれる文字列。サーバは、ユーザの X.509 証明書を SSL 接続を経由してリダイレクトします。形式 <code>https://server:port/FCC?cert+forms</code></p> <p>以下の例ではデフォルトの FCC を使用します。</p> <p><code>https://my.server.com:80/siteminderagent/certoptional/forms/login.fcc?cert+forms</code></p>
共有秘密キー	<p><i>pszSecret=""</i></p> <p>空の文字列に設定します。この方式には適用できません。</p>
テンプレートですか?	<p><i>bIsTemplate=0</i></p> <p>方式がテンプレートではないことを示すには、<b>false (0)</b> に設定します。他の値は無視されます。</p>
管理者によって使用されますか?	<p><i>bIsUsedbyAdmin=0</i></p> <p><b>0</b> に設定し、方式が管理者を認証するために使用されないようにします。</p>
認証情報を保存しますか?	<p><i>bAllowSaveCreds=0</i></p> <p>ユーザ認証情報が保存されないことを示すには、<b>0</b> に設定します。</p>
RADIUS ですか?	<p><i>bIsRadius=0</i></p> <p><b>0</b> に設定し、方式が RADIUS エージェントと共に使用されないようにします。</p>
パスワードチェックを無視しますか?	<p><i>bIgnorePwCheck=flag</i></p> <p>パスワードチェックを無視するには <b>1</b>、パスワードを確認するには <b>0</b> に設定します。デフォルトは <b>0</b> です。</p>

## X.509 クライアント証明書または基本テンプレート

X.509 クライアント証明書または基本方式タイプに基づいて認証方式を設定するときにこのテーブルを使用します。この表で参照される構造体フィールドは、Sm\_PolicyApi\_Scheme\_t にあります。

情報タイプ	値の割り当てと意味
方式タイプ	<p><i>nType</i>= Sm_Api_SchemeType_X509ClientCertOrBasic X.509 クライアント証明書または基本方式タイプ。</p>
説明	<p><i>pszDesc</i>=<i>description</i> 認証方式の説明。</p>
保護レベル	<p><i>nLevel</i>=<i>value</i> 1 ~ 1000 の値。値が大きいほど、その方式によって提供される保護の程度が大きくなります。デフォルトは 5 です。</p>
ライブラリ	<p><i>pszLib</i>="smauthcert" この方式タイプ用のデフォルトライブラリ。</p>
パラメータ	<p><i>pszParam</i>=<i>param</i> 以下の情報が含まれる文字列。</p> <ul style="list-style-type: none"> <li>■ SSL 接続を確立するためのサーバ。このサーバは、ユーザの X.509 証明書を SSL 接続を経由してリダイレクトします。</li> <li>■ SSL 認証情報コレクタ (SSC) の名前およびパス。 SSL を介して基本認証を使用している場合は、以下の 2 種類の情報も指定します。</li> <li>■ 基本認証用の SSL 接続を確立するために使用される SSL サーバの完全修飾名。</li> <li>■ SSL 認証情報コレクタ (SSC) の名前およびパス。 https://SSLserver:port/SCC?certorbasic; [https://BasicServer/SCC]</li> </ul> <p>以下の例ではデフォルトの SCC 値を使用しています。 https://my.SSLserver.com:80/siteminderagent/ certoptional/smgetcred.scc?certorbasic; https://my.BasicServer.com/ siteminderagent/nocert/smgetcred.scc</p>

情報タイプ	値の割り当てと意味
共有秘密キー	<code>pszSecret=""</code> 空の文字列に設定します。この方式には適用できません。
テンプレートですか?	<code>bIsTemplate=0</code> 方式がテンプレートではないことを示すには、 <b>false (0)</b> に設定します。他の値は無視されます。
管理者によって使用されますか?	<code>bIsUsedbyAdmin=0</code> <b>false (0)</b> に設定すると、方式は管理者を認証するためには使用されません。
認証情報を保存しますか?	<code>bAllowSaveCreds=0</code> ユーザ認証情報が保存されないことを示すには、 <b>false (0)</b> に設定します。
RADIUS ですか?	<code>bIsRadius=0</code> <b>false (0)</b> に設定すると、方式は RADIUS エージェントと共に使用されません。
パスワードチェックを無視しますか?	<code>bIgnorePwCheck=flag</code> パスワードチェックを無視するには <b>true (1)</b> 、パスワードを確認するには <b>false (0)</b> に設定します。デフォルトは <b>0</b> です。

## X.509 クライアント証明書またはフォーム テンプレート

X.509 クライアント証明書またはフォーム方式タイプに基づいて認証方式を設定するときこのテーブルを使用します。この表で参照される構造体フィールドは、`Sm_PolicyApi_Scheme_t` にあります。

情報タイプ	値の割り当てと意味
方式タイプ	<code>nType=</code> <code>Sm_Api_SchemeType_X509ClientCertOrForm</code> X.509 クライアント証明書または HTML フォーム方式タイプ。
説明	<code>pszDesc=description</code> 認証方式の説明。
保護レベル	<code>nLevel=value</code> 1 ~ 1000 の値。値が大きいくほど、その方式によって提供される保護の程度が大きくなります。デフォルトは 5 です。

情報タイプ	値の割り当てと意味
ライブラリ	<code>pszLib="smauthcertorform"</code> この方式タイプ用のデフォルトライブラリ。
パラメータ	<code>CpszParam=param</code> 以下の情報が含まれる文字列。 <ul style="list-style-type: none"><li>■ SSL 接続を確立するためのサーバ。このサーバは、ユーザの X.509 証明書を SSL 接続を経由してリダイレクトします。</li><li>■ SSL およびフォーム認証情報コレクタ (SFCC) の名前およびパスです。</li></ul> SSL を介して代替的なフォームベースの認証を使用している場合は、以下の 2 種類の情報も指定します。 <ul style="list-style-type: none"><li>■ 認証用の SSL 接続を確立するために使用される SSL サーバの完全修飾名。</li><li>■ フォーム認証情報コレクタ (FCC) の名前およびパスです。</li></ul> <code>https://SSLserver:port/SFCC?certorform;</code> <code>[https://BasicServer/FCC]</code> 以下の例ではデフォルトの SCC 値を使用しています。 <code>https://my.SSLserver.com:80/siteminderagent/certooptional/forms/login.sfcc?certorform;</code> <code>https://my.BasicServer.com/siteminderagent/forms/login.fcc</code>
共有秘密キー	<code>pszSecret=""</code> 空の文字列に設定します。この方式には適用できません。
テンプレートですか?	<code>bIsTemplate=0</code> 方式がテンプレートではないことを示すには、 <code>false (0)</code> に設定します。他の値は無視されます。
管理者によって使用されますか?	<code>bIsUsedbyAdmin=0</code> <code>0</code> に設定し、方式が管理者を認証するために使用されないようにします。
認証情報を保存しますか?	<code>bAllowSaveCreds=0</code> ユーザ認証情報が保存されないことを示すには、 <code>0</code> に設定します。
RADIUS ですか?	<code>bIsRadius=0</code> <code>0</code> に設定し、方式が RADIUS エージェントと共に使用されないようにします。

情報タイプ	値の割り当てと意味
パスワードチェックを無視しますか?	<i>bIgnorePwCheck=flag</i> パスワードチェックを無視するには1、パスワードを確認するには0に設定します。デフォルトは0です。

## X.509 クライアント証明書テンプレート

X.509 クライアント証明書方式タイプに基づいて認証方式を設定するときにこのテーブルを使用します。この表で参照される構造体フィールドは、`Sm_PolicyApi_Scheme_t`にあります。

情報タイプ	値の割り当てと意味
方式タイプ	<i>nType=Sm_Api_SchemeType_X509ClientCert</i> X.509 クライアント証明書方式タイプ。
説明	<i>pszDesc=description</i> 認証方式の説明。
保護レベル	<i>nLevel=value</i> 1 ~ 1000 の値。値が大きいほど、その方式によって提供される保護の程度が大きくなります。デフォルトは5です。
ライブラリ	<i>pszLib="smauthcert"</i> この方式タイプ用のデフォルトライブラリ。
パラメータ	<i>pszParam=param</i> SSL 接続の確立に参与するサーバのドメインまたは IP アドレス、および SSL 認証情報コレクタ (SCC) の名前およびパスが含まれる文字列。サーバは、ユーザの X.509 証明書を SSL 接続を経由してリダイレクトします。形式 <code>https://server/SCC?cert</code> 以下の例ではデフォルトの SCC 値を使用しています。 <code>https://my.server.com/siteminderagent/cert/smgetcred.scc?cert</code>
共有秘密キー	<i>pszSecret=""</i> 空の文字列に設定します。この方式には適用できません。

情報タイプ	値の割り当てと意味
テンプレートですか?	<i>bIsTemplate=0</i> 方式がテンプレートではないことを示すには、 <b>false (0)</b> に設定します。他の値は無視されます。
管理者によって使用されますか?	<i>bIsUsedbyAdmin=0</i> <b>false (0)</b> に設定すると、方式は管理者を認証するためには使用されません。
認証情報を保存しますか?	<i>bAllowSaveCreds=0</i> ユーザ認証情報が保存されないことを示すには、 <b>false (0)</b> に設定します。
RADIUS ですか?	<i>bIsRadius=0</i> <b>false (0)</b> に設定すると、方式は RADIUS エージェントと共に使用されません。
パスワードチェックを無視しますか?	<i>bIgnorePwCheck=1</i> <b>true (1)</b> に設定すると、パスワードチェックを無視します。

# 第 5 章: ポリシー サーバのカスタマイズ

---

以下の API により、ポリシー サーバに対するカスタム拡張を開発することができます。

- 認証 API
- 許可 API
- イベント API
- ディレクトリ API

たとえば、以下を作成できます。

- 外部のビジネス ロジックに基づき動的な許可を行うアクティブなポリシー
- 外部のデータ ソースから値を返すアクティブなレスポンス
- 外部のビジネス ロジックに基づいて動的に許可を行うアクティブなルール
- カスタム認証方式
- ポリシー サーバによって発行されたイベントをキャプチャするイベントハンドラ、
- カスタム ネームスペース内のデータからのユーザ ディレクトリ

このセクションには、以下のトピックが含まれています。

[認証および認可 API での動作 \(P. 647\)](#)

[Web エージェントとの統合 \(P. 657\)](#)

## 認証および認可 API での動作

ポリシー サーバは、カスタム認証および認可タスクを実行するために使用できる複数のフックを公開します。また、フックは SiteMinder エージェントによって保護されるアプリケーションへのカスタム データの配信を有効にするために使用できます。これらのフックを使用することで、SiteMinder と既存のインフラストラクチャの間の密接な統合が可能になります。

### 認証フック

このフックを使用するには、認証 API を使用して、カスタム認証方式を構築する必要があります。カスタム認証方式は、ユーザの特定、認証およびインパーソネーション問題に対する管理を担うことができます。カスタム認証方式は、ユーザの認証試行を受理または拒否することができ、それを行うための数値的理由およびテキストメッセージを指定できます。この数値的理由およびテキストメッセージは、アプリケーションが適切な処置を講ずることができるように、アプリケーションに提供されます。これにより、アプリケーションはユーザが最後に認証されるまで、認証方式とのコミュニケーションを維持することができます。

### 認可フック

このフックを使用するには、許可 API を使用して、カスタム認可モジュールを構築する必要があります。カスタム認可モジュールは、SiteMinder ルールおよびポリシーがユーザのリソースへのアクセスを許可するかどうかを制御します。

### カスタム データ フック

このフックを使用するには、外部データ ソースからデータを取得し、このデータをポリシー サーバに返すカスタム モジュールを構築する必要があります。その後、ポリシー サーバはそのデータを SiteMinder によって保護されたアプリケーションに配信します。SiteMinder は、このデータを決して解釈しません。

### SiteMinder イベント

ポリシー サーバは、各認証および認可の後にトリガされる場合がある複数のイベントを定義します。SiteMinder ポリシーは、これらのイベントをキャッチし、適切なデータを SiteMinder エージェントによって保護されたアプリケーションに返すために使用されます。イベントは、アプリケーションワークフローを実行するために使用できます。イベントには、認証と認可の 2 つのタイプがあります。

## 認証イベント

ポリシー サーバは、認証の結果に基づいて認証イベントを開始します。認証のイベントには次のものが含まれます。

- OnAuthAccept
- OnAuthReject
- OnAuthChallenge
- OnAuthAttempt

適切な認証イベントを使用することで、ユーザが認証された直後に、レスポンスが累積され、アプリケーションに配信されます。

## 許可イベント

ポリシー サーバは、認可の結果に基づいて認可イベントを開始します。認可のイベントには次のものが含まれます。

- OnAccessAccept
- OnAccessReject

適切な許可イベントを使用することで、ユーザがリソースへのアクセスを許可された直後に、レスポンスが累積され、アプリケーションに配信されます。

## モジュール呼び出しの間のデータ保存

ポリシー サーバがカスタム モジュールへの呼び出しを行うときは、それらの間で要求特有の情報を通信することがこれらのモジュールに都合がよい場合があります。認証、許可またはディレクトリ API で構築されたカスタム モジュールは、モジュール呼び出し間の状態を保存できます。状態を保存するには、(SmApi.h で定義された) Sm\_Api\_Context\_t 構造で提供されるメモリ管理フックを使用します。状態には要求限定範囲があり、プロセスの単一のインスタンスに制限されています。すなわち、認証サーバのコンテキストで実行されるカスタム モジュールは、認可サーバのコンテキストで実行されるモジュールと状態を交換できません。

## 汎用ユーザ属性

ユーザプロフィールに保存されないユーザ プロパティを説明するデータを返すことが必要な場合があります。通常、このデータは、SiteMinder によって管理され認識されている何らかの動的な情報です。動的な情報は、汎用のまたは偽のユーザ属性を使用して提供されます。属性が標準のユーザ属性を表す場合、標準的なレスポンス メカニズムまたはカスタムモジュールはこれらの属性にアクセスできます。これらの属性は以下にリスト表示される通り SM\_ から始まります。

### SM\_USERNAME

認証されたユーザの場合、これは SiteMinder によって明確にされるユーザ DN です。認証されていないユーザの場合、これはログイン時にユーザが指定するユーザ ID です。

### SM\_USERIMPERSONATORNAME

認証方式でインパーソネーションが実行される場合、これは SiteMinder によって認証されるユーザ DN です。

### SM\_USERLOGINNAME

これはログイン時にユーザが指定するユーザ ID です。

### SM\_USERLOGINFAILURESCOUNT

失敗したログインの回数で、int 値として表されます。パスワードポリシー状態属性の 1 つです。

### SM\_USERIPADDRESS

認証時または許可時のユーザの IP アドレス。

### SM\_USERPATH

認証されたユーザの場合、これは（いずれもユーザディレクトリ定義で指定される）ディレクトリ ネームスペースとディレクトリ サーバ、および（SiteMinder によって明確にされる）ユーザ DN を表す文字列です。例：

```
"LDAP://123.123.0.1/uid=scarter,ou=people,o=airius.com"
```

認証されていないユーザの場合は、SM\_USERNAME と同じです。

### SM\_USERPREVIOUSLOGINTIME

ユーザの前のログイン時間で、GMT time\_t 値として表されます。パスワードポリシー状態属性の 1 つです。この値は読み取り専用（取得）で、設定できません。値は、ポリシーサーバシステムのエポックタイムからの経過時間を表します。エポックタイムは 1970 年 1 月 1 日 00:00:00 UTC です。（Mac OS マシンがクライアントとして実行している場合、そのエポックタイムは現地時間 1904 年 1 月 1 日 00:00:00 です。）パスワードポリシー状態属性の 1 つです。

### SM\_USERPASSWORD

これはログイン時にユーザが指定するパスワードです。値は認証時時のみ返され、許可時には返されません。

### SM\_TRANSACTIONID

エージェントによって生成されるトランザクション ID です。

### SM\_USERDISABLEDTIME

ユーザが無効になった時間で、GMT time\_t 値として表されます。パスワードポリシー状態属性の 1 つです。この値は読み取り専用（取得）で、設定できません。値は、ポリシーサーバシステムのエポックタイムからの経過時間を表します。エポックタイムは 1970 年 1 月 1 日 00:00:00 UTC です。（Mac OS マシンがクライアントとして実行している場合、そのエポックタイムは現地時間 1904 年 1 月 1 日 00:00:00 です。）パスワードポリシー状態属性の 1 つです。

### SM\_USERLASTPWCHANGETIME

ユーザが最後にパスワードを変更した時間で、GMT time\_t 値として表されます。値は、ポリシーサーバシステムのエポックタイムからの経過時間を表します。エポックタイムは 1970 年 1 月 1 日 00:00:00 UTC です。（Mac OS マシンがクライアントとして実行している場合、そのエポックタイムは現地時間 1904 年 1 月 1 日 00:00:00 です。）パスワードポリシー状態属性の 1 つです。

**注:** この値を設定し、関連するパスワードが、SiteMinder の外部でリセットされる場合、パスワード再利用を防ぐパスワードポリシーが予期したように動作しない場合があります。

### SM\_USERPASSWORDHISTORY

パスワード ポリシー状態属性の 1 つです。最大 32 エントリまでのパスワード変更情報がすべて含まれ、文字列値として表されます。暗号化および構造化された、バイナリ情報が含まれます。パスワード状態 API は、この構造に関する詳細を外部アプリケーションに公開しません。

**注:** 空の文字列を使用することでのみこの属性の値を設定し、全履歴をクリアすることができます。他の履歴変更は許可されていません。1 つの履歴エントリのみを追加したり削除することはできません。この属性に対する読み取り（取得）はありません。

### SM\_USERSESSIONSPEC

ユーザのセッション仕様。

### SM\_USERSESSIONID

すでに認証されたユーザのセッション ID。これは認証の成功時にユーザに割り当てられるセッション ID です。

### SM\_USERSESSIONIP

最初のユーザ認証時（セッションの確立時）に使用された IP アドレス。

### SM\_USERSESSIONUNIVID

ユーザのユニバーサル ID。ユーザディレクトリ定義にユニバーサル ID ディレクトリ属性の指定がない場合には、これはデフォルトでユーザの DN に設定されます。

### SM\_USERSESSIONDIRNAME

ポリシーサーバが使用するように設定されているユーザディレクトリの名前。

### SM\_USERSESSIONDIROID

ポリシーサーバが使用するように設定されているユーザディレクトリのオブジェクト ID。

### SM\_USERSESSIONTYPE

ユーザのセッションタイプ。以下のいずれか：

- 2 - セッション
- 1 - ID

### SM\_USERLASTLOGINTIME

ユーザの最後のログイン時間で、GMT `time_t` 値として表されます。適用可能なパスワードサービスポリシーがある認証時にのみ利用可能です。パスワードポリシー状態属性の 1 つです。

### SM\_USERGROUPS

ユーザが属するグループ。ユーザがネストされたグループに属する場合には、この属性には階層の最下位にあるグループが含まれます。ユーザが属するすべてのネストされたグループについては、**SM\_USERNESTEDGROUPS** を使用してください。

たとえば、ユーザ JSmith がグループ Accounting グループに含まれるグループ Accounts Payable に属する場合、**SM\_USERGROUPS** には Accounts Payable グループが含まれます。Accounting グループと Accounts Payable グループの両方を使用するには、**SM\_USERNESTEDGROUPS** を使用します。

### **SM\_USERNESTEDGROUPS**

ユーザが属するネストされたグループ。階層の最下位にあるグループについてのみ、**SM\_USERGROUPS** を使用してください。

たとえば、ユーザ JSmith が、Accounting グループに含まれる Accounts Payable グループに属する場合、**SM\_USERNESTEDGROUPS** には Accounting グループと Accounts Payable グループが含まれます。Accounting グループのみを使用する場合には、**SM\_USERGROUPS** を使用します。

### **SM\_USERSCHEMAATTRIBUTES**

DN に関連付けられたユーザ属性、またはユーザに関連付けられたプロパティ。ユーザディレクトリが SQL データベースの場合、**SM\_USERSCHEMAATTRIBUTES** には、ユーザデータが格納されているテーブルの列名が格納されます。たとえば、SmSampleUsers スキーマを使用している場合には、**SM\_USERSCHEMAATTRIBUTES** には、SmUser テーブルの列名が格納されます。

### **SM\_USERPOLICIES**

これらには、リソースに対しユーザを認可するポリシーの名前が含まれます。たとえば、商品を購入するにはユーザは Buyer ポリシーに関連付けられている必要があると仮定します。ユーザが商品の購入を正常に許可された場合、**SM\_USERPOLICIES** には Buyer が含まれます。

関連付けられたルールに対するアクションが許可イベントに設定され、指定されたイベントが OnAccessAccept であるときは、**SM\_USERPOLICIES** には値が含まれているのみです。

### **SM\_USERPRIVS**

ユーザがリソースへのアクセスを許可または認証されている場合、**SM\_USERPRIVS** 属性には、そのユーザに適用されるすべてのポリシー ドメイン内のすべてのポリシーのレスポンス属性がすべて格納されます。

### **SM\_USERREALMPRIVS**

ユーザがレルムにあるリソースへのアクセスを許可または認証されている場合、**SM\_USERREALMPRIVS** 属性には、そのレルムのすべてのルールのすべてのレスポンス属性が格納されます。

たとえば、「Equipment Purchasing」と呼ばれるレルムがあると仮定します。このレルムには、**CheckCredit** ルールがあります。さらに、この **CheckCredit** ルールには、購入者のクレジット上限 (**limit = \$15000** など) をレスポンス属性として返すレスポンスが関連付けられています。購入者が **5000** ドル相当の装置を購入しようとする、**CheckCredit** ルールが起動します。この場合、**SM\_USERREALMPRIVS** には、**Equipment Purchasing** レルムの下にあるすべてのルールのレスポンス属性がすべて格納されます。

### **SM\_USERDISABLEDSTATE**

ユーザが無効化された理由のビット マスクを表す **10** 進数です。このビットは **Sm\_Api\_DisabledReason\_t** データ構造体下の **SmApi.h** で定義されています。

たとえば、ユーザが一定の時間何も操作を行わなかった場合には、**Sm\_Api\_Disabled\_Inactivity** という理由で無効化されます。**Sm\_Api\_DisabledReason\_t** では、理由 **Sm\_Api\_Disabled\_Inactivity** は、値 **0x00000004** に対応します。したがって、この場合には **SM\_USERDISABLEDSTATE** は **4** になります。

### **NTUSERNAME**

完全修飾 WinNT 名のユーザ名の一部を返します。たとえば、**mydomain¥myname** の場合は、**myname** が返されます。

### **NTFULLUSERNAME**

完全修飾 WinNT 名を返します。

## パスワードの取得

いくつかのアプリケーションは、ユーザ認証情報についてのナレッジを必要とします。SiteMinder は既知の属性 `SM_USERPASSWORD` を返すことによりユーザパスワードをアプリケーションに利用可能にできます。この属性は、`OnAuthAccept` を通じて認証が成功した場合にのみ使用できます。

以下の例ではパスワードの取得について実証します。

```
char  lpszSmPassword(128);
int   nBytes =

lpUserContext->fGetProp(lpUserContext->lpParam,
                        "SM_USERPASSWORD",
                        sizeof (lpszSmPassword),
                        lpszSmPassword);
```

## Web エージェントとの統合

ポリシー サーバは Web エージェントと連携して動作します。Web エージェントが認証および認可している間、ポリシー サーバは操作の結果に関する情報を Web エージェントに送信します。Web エージェントはこの情報を以下のようにアプリケーションに利用可能にします。

### リダイレクトのない認証イベント

- ユーザメッセージは HTTP\_SM\_USRMSG ヘッダ変数として公開されます。
- 理由コードは HTTP\_SM\_AUTHREASON ヘッダ変数として公開されます。

### リダイレクトのある認証イベント

- ユーザのメッセージは SMUSRMSG=<text> として SMUSERMSG Cookie に保存されます。
- 理由コードは ?SMAUTHREASON=<reason code> としてリダイレクト URL に追加されます。

理由コードが以下のフォームに含まれている場合

`SMAUTHREASON=$$SMAUTHREASON$$`

値は、ユーザ名およびパスワードと共に SM\_AGENTAPI\_LOGIN 用の認証理由フィールドで使用されます。

- リダイレクトテキストは SMTEXT=<text> として SMTEXT Cookie に保存されます。

### リダイレクトのない許可イベント

- 利用可能な理由コードはありません。

### リダイレクトのある許可イベント

- 理由コードは SMREASON=<reason code> として SMREASON Cookie に保存されます。
- リダイレクトテキストは SMTEXT=<text> として SMTEXT Cookie に保存されます。



# 第 6 章: コンテキスト構造

---

このセクションには、以下のトピックが含まれています。

[Sm\\_Api\\_AppSpecificContext\\_t](#) (P. 659)

[Sm\\_Api\\_Context\\_t](#) (P. 662)

[Sm\\_Api\\_RequestContext\\_t](#) (P. 664)

[Sm\\_Api\\_TunnelContext\\_t](#) (P. 665)

[Sm\\_Api\\_UserContext\\_t](#) (P. 666)

[Sm\\_Api\\_UserCredentials\\_t](#) (P. 673)

## Sm\_Api\_AppSpecificContext\_t

この構造には共有メモリ情報が含まれます。

### 構文

この構造の構文は以下のとおりです。

```
typedef struct
{
    void* pHandle;
    Sm_Api_GetBufferFunc fGetBuffer;
    Sm_Api_AllocBufferFunc fAllocBuffer;
    Sm_Api_FreeBufferFunc fFreeBuffer;
    Sm_Api_GetBufferSizeFunc fGetBufferSize;
} Sm_Api_AppSpecificContext_t;
```

フィールド	説明
<i>pHandle</i>	渡されるハンドルで、メモリ管理機能に必要です。このハンドルは <i>Sm_Api_Context_t</i> の <i>pAppSpecific</i> なフィールドから取得されます。
<i>fGetBuffer</i>	メモリが割り当てられなかった場合、以前割り当てられた要求に固有のメモリまたは <b>NULL</b> にポインタを返す関数。 <i>pHandle</i> は関数の唯一のパラメータです。

フィールド	説明
<i>fAllocBuffer</i>	要求に固有のメモリを割り当てて、ポインタを割り当てられたバッファに返す関数。バッファがすでに割り当てられているか、メモリが利用可能でない場合、それは <b>NULL</b> を返します。パラメータは <i>pHandle</i> 、およびバッファのサイズを指定する <b>int</b> です。
<i>fFreeBuffer</i>	渡された <i>pHandle</i> 引数と関連する要求に固有のメモリを解放する関数。値は返されません。
<i>fGetBufferSize</i>	バッファのサイズを取得する関数。

---

## 関数宣言

構造 `Sm_Api_AppSpecificContext_t` では、関数 `fGetBuffer`、`fAllocBuffer`、`fFreeBuffer` および `fGetBufferSize` は、`SmApi.h` で以下のように宣言されます。

### **fGetBuffer**

```
typedef void* (SM_EXTERN *Sm_Api_GetBufferFunc)
    (const void *pHandle);
```

### **fAllocBuffer**

```
typedef void* (SM_EXTERN *Sm_Api_AllocBufferFunc)
    (const void *pHandle, const int nSize);
```

### **fFreeBuffer**

```
typedef void (SM_EXTERN *Sm_Api_FreeBufferFunc)
    (const void *pHandle);
```

### **fGetBufferSize**

```
typedef int (SM_EXTERN *Sm_Api_GetBufferSizeFunc)
    (const void *pHandle);
```

## 例

```
// Allocate 64 bytes
char* pBuffer = (char*) lpApiContext->pAppSpecific->fAllocBuffer
    (lpApiContext->pAppSpecific->pHandle, 64);
strcpy (pBuffer, "id=5");
...

// Use it somewhere else
char id[20];
char* pBuffer = (char*) lpApiContext->pAppSpecific->fGetBuffer
    (lpApiContext->pAppSpecific->pHandle);
if (pBuffer != NULL)
{
    strcpy (id, pBuffer);
}

//Free the memory when done
...
(char*) lpApiContext->pAppSpecific->fFreeBuffer
    (lpApiContext->pAppSpecific->pHandle);
```

## Sm\_Api\_Context\_t

この構造には一般的な API 情報が含まれます。

### 構文

```
typedef struct
{
    int nVersion;
    Sm_Api_LogFunc fLog;
    Sm_Api_TraceFunc fTrace;
    Sm_Api_ErrorFunc fError;
    Sm_Api_AppSpecificContext_t* pAppSpecific;
} Sm_Api_Context_t;
```

フィールド	説明
<i>nVersion</i>	使用されている API のバージョン番号。指定可能な値 <ul style="list-style-type: none"><li>■ Sm_Api_Version_V3</li><li>■ Sm_Api_Version_V4</li><li>■ Sm_Api_Version_V4_1</li></ul> バージョン定数は SmApi.h で定義されています。
<i>fLog</i>	SiteMinder ログ記録ユーティリティにアクセスするための関数。SiteMinder はログ記録が有効な状態で実行されている必要があります。 この関数の呼び出し構文は次のとおりです。 <code>char* lpszMessage = "Log this text";</code> <code>lpApiContext-&gt;fLog (lpszMessage);</code>
<i>fTrace</i>	SiteMinder トレースユーティリティにアクセスするための関数。SiteMinder はトレースが有効な状態で実行されている必要があります。 この関数の呼び出し構文は次のとおりです。 <code>char* lpszCheckpoint = "MyLib";</code> <code>char* lpszTraceMessage = "Trace text";</code> <code>lpApiContext-&gt;fTrace (lpszCheckpoint,</code> <code>lpszTraceMessage);</code>

フィールド	説明
<i>fError</i>	<p>SiteMinder エラーユーティリティにアクセスするための関数。SiteMinder はログ記録が有効な状態で実行されている必要があります。</p> <p>この関数の呼び出し構文は次のとおりです。</p> <pre>char* lpszMessage = "Log this Error text"; lpApiContext-&gt;fError (lpszMessage);</pre>
<i>pAppSpecific</i>	<p><i>pAppSpecific</i> ポインタは、Sm_Api_AppSpecificContext_t 構造内のメソッドへのアクセス権があります。これらのメソッドは要求に固有のメモリの割り当て、削除、および解放を行います。このメモリは単一のエージェント要求の全処理中も存続し、認証デーモンや認可デーモンなどの各サーバプロセスに局所的です。</p> <p>メモリがモジュールによって割り当てられ解放される回数に関して制限はありません。いったん要求が処理されれば、メモリは自動的に解放されます。</p>

## 関数宣言

構造 Sm\_Api\_Context\_t では、関数 fLog、fTrace および fError は、SmApi.h で以下のように宣言されます。

### fLog

```
/* string to log (null-terminated) */
typedef void (SM_EXTERN *Sm_Api_LogFunc) (const char* lpszBuffer);
```

### fTrace

```
/* string to log (null-terminated) */
typedef void (SM_EXTERN *Sm_Api_TraceFunc)
    (const char* lpszCheckpoint, const char* lpszBuffer);
```

### fError

```
/* string to log (null-terminated) */
typedef void (SM_EXTERN *Sm_Api_ErrorFunc)
    (const char* lpszBuffer);
```

## Sm\_Api\_RequestContext\_t

実行中の要求に関する情報を含みます。

### 構文

```
typedef struct
{
    char* lpszServer;
    char* lpszResource;
    char* lpszAction;
} Sm_Api_RequestContext_t;
```

---

フィールド	説明
<i>lpszServer</i>	ユーザアクセス要求のサーバ部分。たとえば、ユーザが <code>http://www.server.com/index.html</code> にアクセスする場合、サーバは <code>www.server.com</code> です。
<i>lpszResource</i>	ユーザアクセス要求のリソース部分。たとえば、ユーザが <code>http://www.server.com/index.html</code> にアクセスする場合、リソースは <code>/index.html</code> です。
<i>lpszAction</i>	ユーザによって要求されるようなアクション。たとえば、Web リソースにアクセスするときの標準的なアクションは <code>GET</code> です。

---

## Sm\_Api\_TunnelContext\_t

トンネル エージェントから渡された情報を保持します。

### 構文

```
typedef struct
{
    char* lpszClientIp;
    char* lpszTransactionId;
    char* lpszParameter;
} Sm_Api_TunnelContext_t;
```

フィールド	説明
<i>lpszClientIP</i>	トンネル コールを行うトンネル エージェントの IP アドレス。
<i>lpszTransactionId</i>	(オプション) エージェントがセキュリティ アクティビティをアプリケーション アクティビティに関連付けるために使用する ID です。ポリシー サーバはこの ID をログ記録します。
<i>lpszParameter</i>	トンネル エージェントから渡された任意の文字列パラメータ。トンネル エージェント側で、この情報は <code>Sm_AgentApi_TunnelServiceRequest_t</code> 内の <i>lpszParameter</i> に渡されます。

## Sm\_Api\_UserContext\_t

ユーザに関する情報が含まれます。

### 構文

```
typedef struct
{
    unsigned char  bIsUserContext;
    char* lpszUserName;
    char* lpszUserPath;
    char* lpszDirPath;
    void* lpReserved1;
    char* lpszDirServer;
    char* lpszDirNamespace;
    char* lpszSessionId;
    Sm_Api_GetDnProp fGetDnProp;
    Sm_Api_SetDnProp fSetDnProp;
    void* lpParam;
    Sm_Api_GetUserProp fGetProp;
    Sm_Api_SetUserProp fSetProp;
    Sm_Api_AuthenticateUser fAuthenticate;
} Sm_Api_UserContext_t;
```

フィールド	説明
<i>bIsUserContext</i>	SiteMinder がユーザの ID を確立し、ユーザ コンテキストが利用可能であることを示すフラグ。このフラグが設定されると、 <i>fGetProp</i> 、 <i>fSetProp</i> 、 <i>fGetDnProp</i> および <i>fSetDnProp</i> ユーザ属性関数が利用可能になります。
<i>lpszUserName</i>	ユーザの完全識別名。
<i>lpszUserPath</i>	以下の形式のユーザ パス。 <i>directory-namespace + server + / + user-DN</i> 例： ldap://server.company.com/ uid=user1,ou=people,o=company.com
<i>lpszDirPath</i>	ユーザ コンテキストが確立された SiteMinder ユーザ ディレクトリのディレクトリ パスで、以下の形式になります。 <i>directory-namespace + server</i> 例： ldap://server.company.com

フィールド	説明
<i>lpReserved1</i>	内部使用向けです。
<i>lpzDirServer</i>	ユーザ コンテキストが確立された SiteMinder ユーザ ディレクトリのディレクトリ サーバ。
<i>lpzDirNamespace</i>	LDAP:、AD:、WinNT:、または ODBC: などのディレクトリ ネームスペース。
<i>lpzSessionId</i>	ユーザのセッションに割り当て済みまたは割り当てられる予定のセッション ID。セッションがすでに確立されているかどうかによって決まります。
<i>fGetDnProp</i>	<p>ディレクトリ エントリの属性を返す関数。ユーザ コンテキスト フラグ <i>blsUserContext</i> が設定される場合、開発者はこの関数を呼び出し、ディレクトリのコンテキストでユーザが関連付けられる DN の既知の属性を取得することができます (たとえば、ユーザはグループのメンバです)。</p> <p>この関数の呼び出し構文は次のとおりです。</p> <pre>if (lpUserContext-&gt;blsUserContext) { char lpzDN[]="cn=group,ou=org unit,o=org"; char lpzCommonName[100]; int nBytes = lpUserContext-&gt;fGetDnProp( lpUserContext-&gt;lpParam, lpzDN, "accesslevel", sizeof (lpzCommonName), lpzCommonName); }</pre> <p>エラーが発生しない場合、この関数は要求された属性の値を <b>NULL</b> で終わる出力バッファに配置し、その長さを返します。それ以外の場合、関数は <b>-1</b> を返します。</p> <p>この関数から返された属性は <i>nBytesValueBuf</i> 引数で指定された最大バッファ サイズより小さくなければなりません。それより大きい属性は <i>nBytesValueBuf</i> に切り捨てられます。</p>

フィールド	説明
<i>fSetDnProp</i>	<p>ディレクトリ エントリの属性を設定する関数。ユーザ コンテキスト フラグ <i>bIsUserContext</i> が設定される場合、開発者はこの関数を呼び出し、ディレクトリのコンテキストでユーザが関連付けられる DN の既知の属性を設定することができます（たとえば、ユーザはグループのメンバです）。この時、タイプ「文字列」の属性のみがサポートされます。</p> <p>この関数の呼び出し構文は次のとおりです。</p> <pre>if (lpUserContext-&gt;bIsUserContext) { char lpszDN[]="cn=group,ou=org unit,o=org"; char lpszTimestamp[] = "&lt;timestamp&gt;"; int nErr = lpUserContext-&gt;fSetDnProp (     lpUserContext-&gt;lpParam,     lpszDN,     "lastaccess",     sizeof (lpszTimestamp),     lpszTimestamp); }</pre>
<i>lpParam</i>	<p><i>fGetProp</i>、<i>fSetProp</i>、<i>fGetDnProp</i>、および <i>fSetDnProp</i> 関数に渡されるパラメータへのポインタ。</p>

---

---

フィールド	説明
<i>fGetProp</i>	<p>ユーザ属性を返す関数。ユーザ コンテキスト フラグ <code>bIsUserContext</code> が設定される場合、開発者はこの関数を呼び出し、既知のユーザ属性を取得することができます。</p> <p>この関数の呼び出し構文は次のとおりです。</p> <pre>if (lpUserContext-&gt;bIsUserContext) { char lpszCommonName[100]; int nBytes = lpUserContext-&gt;fGetProp (     lpUserContext-&gt;lpParam,     "cn",     sizeof (lpszCommonName),     lpszCommonName); }</pre> <p>エラーが発生しない場合、この関数は要求された属性の値を <code>NULL</code> で終わる出力バッファに配置し、その長さを返します。それ以外の場合、関数は <code>-1</code> を返します。</p> <p>この関数から返された属性は <code>nBytesValueBuf</code> 引数で指定された最大バッファ サイズより小さくなければなりません。それより大きい属性は <code>nBytesValueBuf</code> に切り捨てられます。</p>
<i>fSetProp</i>	<p>ユーザ属性を設定する関数。ユーザ コンテキスト フラグ <code>bIsUserContext</code> が設定される場合、開発者はこの関数を呼び出し、既知のユーザ属性を設定することができます。この時、タイプ「文字列」の属性のみがサポートされます。</p> <p>この関数の呼び出し構文は次のとおりです。</p> <pre>if (lpUserContext-&gt;bIsUserContext) { char lpszCommonName[] = "John Smith"; int nErr = lpUserContext-&gt;fSetProp (     lpUserContext-&gt;lpParam,     "cn",     sizeof (lpszCommonName),     lpszCommonName); }</pre> <p>エラーが発生しない場合、関数は <code>0</code> を返します。それ以外の場合、関数は <code>-1</code> を返します。</p>

---

## 関数宣言

構造体 Sm\_Api\_UserContext\_t で、関数 fGetDnProp、fSetDnProp、fGetProp、fSetProp および fAuthenticate は、SmApi.h で以下のように宣言されます。

### fGetDnProp

```
typedef int (SM_EXTERN *Sm_Api_GetDnProp)
(
    const void* lpParam,          /* The function parameter */
    const char* lpDn,            /* The DN of a directory object */
    const char* lpszPropName,    /* User property name (null-term) */
    const int nBytesValueBuf,    /* Max size of user property buffer */
    char* lpszValueBuf /* Output buffer to hold the user property */
);
```

### fSetDnProp

```
typedef int (SM_EXTERN *Sm_Api_SetDnProp)
(
    const void* lpParam,          /* The function parameter */
    const char* lpDn,            /* The DN of a directory object */
    const char* lpszPropName,    /* User property name (null-term) */
    const int nBytesValueBuf,    /* Size of user property buffer */
    const char* lpszValueBuf /* The user property buffer */
);
```

### fGetProp

```
typedef int (SM_EXTERN *Sm_Api_GetUserProp)
(
    const void* lpParam,          /* The function parameter */
    const char* lpszPropName,    /* User property name (null-term) */
    const int nBytesValueBuf,    /* Max size of user property buffer */
    char* lpszValueBuf /* Output buffer to hold the user property */
);
```

### fSetProp

```
typedef int (SM_EXTERN *Sm_Api_SetUserProp)
(
    const void* lpParam,          /* The function parameter */
    const char* lpszPropName,    /* User property name (null-term) */
    const int nBytesValueBuf,    /* Size of user property buffer */
    const char* lpszValueBuf /* The user property buffer */
);
```

### fAuthenticate

```
typedef int (SM_EXTERN *Sm_Api_AuthenticateUser)
```

```
(  
const void* lpParam,      /* The function parameter */  
const char* lpszPassword, /* User password (null-terminated) */  
const int nBytesUserMsg,  /* Max size of user message buffer */  
char* lpszUserMsg,       /* Output buffer to hold the user message */  
const int nBytesErrMsg,  /* Maximum size of the error buffer */  
char* lpszErrMsg        /* Output buffer to hold the error message */  
);
```

詳細情報:

[LDAP の複数値属性](#) (P. 672)

## LDAP の複数値属性

LDAP ユーザストアで複数値属性を設定または取得する場合、その値は単一の文字列で示され、キャラット文字 (^) によって区切られます。

たとえば、3 つの異なる電話番号を以下のように設定することができます。

```
char lpszTemp[] = "111-1234^111-5678^111-0000";

int getResult = lpUserContext->fSetProp (lpUserContext->lpParam,
                                         "telephonenumber",
                                         strlen(lpszTemp),
                                         lpszTemp);
```

複数値属性を設定または取得するカスタム コードは、予期される複数値文字列形式をサポートする必要があります。

**注:** ODBC ユーザストアは複数値属性設定をサポートしません。

### ポリシー サーババージョンのサポート

4.61 またはそれ以降のポリシー サーバでは、上記の例における電話番号は、以下のように LDAP ユーザストア内のユーザの電話番号属性に入れられます。

```
111-1234
111-5678
111-0000
```

4.61 ポリシー サーバより前に、上記のコードは以下のように電話番号属性を設定します。

```
111-1234^111-5678^111-0000
```

## Sm\_Api\_UserCredentials\_t

ユーザ認証情報に関する情報が含まれます。

SiteMinder は利用可能な情報を指定します。通常は、ユーザ名とパスワードフィールドが指定されます。認証方式はそれぞれ、可能な認証情報のサブセットを予期します。

この構造内のフィールドは、要求された認証情報に基づいて入力されます。

### 構文

```
typedef struct
{
    char* lpszUsername;
    char* lpszPassword;
    int nBytesChapPassword;
    char* lpszChapPassword;
    int nBytesChapChallenge;
    char* lpszChapChallenge;
    char* lpReserved1;
    char* lpszCertUserDN;
    char* lpszCertIssuerDN;
    int nCertBinLen;
    void* lpCertBinary;
    char* lpszDirPath;
    char* lpszDirServer;
    char* lpszDirNamespace;
    char* lpszNewPassword;
    char* lpszPasswordToken;
} Sm_Api_UserCredentials_t;
```

フィールド	説明
<i>lpszUsername</i>	ユーザによって指定されたユーザ名から SiteMinder によって明確にされるユーザの DN。
<i>lpszPassword</i>	ユーザによって指定されるユーザパスワード。
<i>nBytesChapPassword</i>	CHAP パスワードの長さ。
<i>lpszChapPassword</i>	CHAP パスワード。
<i>nBytesChapChallenge</i>	CHAP チャレンジの長さ。
<i>lpszChapChallenge</i>	CHAP チャレンジ。

フィールド	説明
<i>lpReserved1</i>	内部使用向けです。
<i>lpszCertUserDN</i>	X.509 ユーザの証明書のユーザ DN 部分。
<i>lpszCertIssuerDN</i>	X.509 ユーザの証明書の発行者 DN 部分。
<i>nCertBinLen</i>	ユーザのバイナリ X.509 証明書の長さ。
<i>lpCertBinary</i>	ユーザのバイナリ X.509 証明書。
<i>lpszDirPath</i>	ユーザ コンテキストが確立された SiteMinder ユーザ ディレクトリの SiteMinder 表記法におけるディレクトリ パス。
<i>lpszDirServer</i>	ユーザ コンテキストが確立された SiteMinder ユーザ ディレクトリのディレクトリ サーバ。
<i>lpszDirNamespace</i>	ユーザ コンテキストが確立された SiteMinder ユーザ ディレクトリのディレクトリ ネームスペース。
<i>lpszNewPassword</i>	ユーザの新しいパスワード。
<i>lpszPasswordToken</i>	パスワード サービスからのパスワード トークン。

# 第 7 章: C 用の認証 API

---

このセクションには、以下のトピックが含まれています。

[認証 API の概要 \(P. 675\)](#)

[カスタム認証方式ライブラリの作成 \(P. 679\)](#)

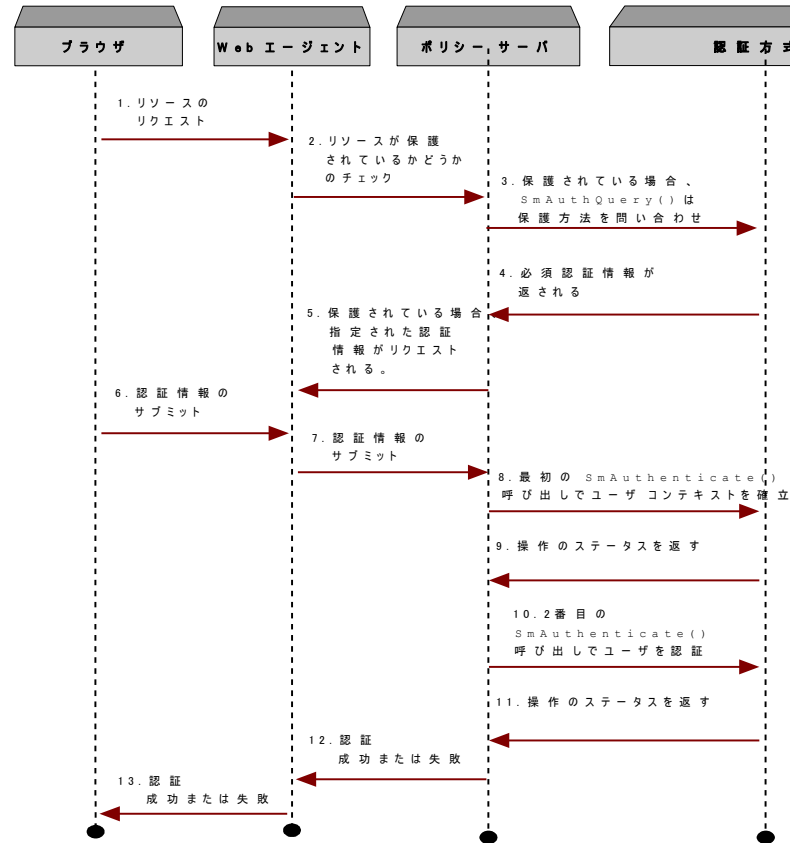
## 認証 API の概要

SiteMinder 認証方式はそれぞれ、認証 API プロバイダ インターフェースをサポートする共有ライブラリのインスタンスです。

通常、Administrative UI で認証方式を定義すると、使用する認証方式タイプのデフォルトライブラリを受理します。たとえば、HTML フォーム テンプレートに基づいて認証方式を使用する場合、そのデフォルトライブラリ、`smauthhtml` を受理します。

認証方式にカスタム認証機能をサポートさせる場合、認証方式の新しいライブラリを構築します。認証 API を使用して、認証方式ライブラリを構築します。

以下の図では、認証方式ライブラリが認証処理でどのように使用されるかを示しています。



## 認証方式ライブラリのインストール

特注のライブラリを以下のいずれかのデフォルトの場所にインストールします。

- UNIX プラットフォームでは、SiteMinder lib ディレクトリ
- Windows プラットフォームでは、SiteMinder bin ディレクトリ

## 認証方式のロード

認証方式ライブラリは認証サーバおよび認可サーバの両方によってロードされます。両方のサーバは、この方式を使用して必要な認証情報を取得します。

方式がロードされた直後に、以下の操作が発生します。

1. `SmAuthQuery()` 関数が呼び出され、認証 API バージョン番号および説明が取得されます。
2. `SmAuthQuery()` が呼び出された後、`SmAuthInit()` が認証サーバによって呼び出されます。`SmAuthInit()` 関数は、許可サーバによって呼び出されません。

## ユーザ コンテキスト

認証方式は、`SiteMinder` によって渡されたユーザ認証情報を確認し、認証結果を返します。認証方式はこれらのモードで作動します。

- **ユーザ コンテキストは不明です** -- ユーザは、まだユーザストアに位置する必要があります。`SiteMinder` または認証方式のいずれかがユーザを調べ、ユーザの保存された認証情報がログイン時に提供された認証情報と比較できるようにします。

ユーザストアでのユーザの検索は、「ユーザの特定」と呼ばれます。

- **ユーザ コンテキストは既知です** -- ユーザはユーザストアに位置しています。これでカスタム認証方式はユーザの認証情報を確認できます。

## 認証イベント

認証結果は `SiteMinder` のイベントに結び付けられます。認証イベントがユーザが認証されているレلمで有効な場合、`SiteMinder` は `OnAuthAccept`、`OnAuthReject`、`OnAuthAttempt`、および `OnAuthChallenge` ルールに結び付けられたオプションのポリシーを評価します。これらのポリシーを、ユーザのアイデンティティに基づいてカスタム レスポンスを返すように設定するには、認証結果に基づいてユーザを別の場所にリダイレクトするか、外部データベース内でユーザデータを更新します。

### リダイレクト

認証方式は、ポリシー サーバにリダイレクトを実行するようエージェントに命じるよう指示することができます。リダイレクト機能を提供する認証方式を構築するには、`lpszErrMsg` パラメータに URL を配置し、理由コード `Sm_Api_Reason_ErrorMessageIsRedirect` が含まれるステータス コードを返します。

例：

```
strcpy (lpszErrMsg, "https://12.12.1.1/display/user.cgi?dn=");
strcat (lpszErrMsg, lpUserContext->lpszUserName);
return SM_MAKEAUTH_STATUSVALUE (Sm_Api_Accept,
                                Sm_Api_Reason_ErrorMessageIsRedirect);
```

この機能は、標準的なエージェントを使用して、**Web** アプリケーションのワークフローをカスタマイズするときに役立ちます。ただし、リダイレクトの設定もまたカスタム エージェントを使用するときに役立ちます。

### サポートされた認証情報

認証 API は、以下のタイプの認証情報に基づいてユーザ認証をサポートします。

- ユーザ名/パスワード
- X.509 証明書
- カスタム ユーザ属性

詳細情報：

[SmAuthQuery\(\)](#) (P. 684)

## カスタム認証方式ライブラリの作成

カスタム認証方式ライブラリを作成する方法

以下のように `SmApi.h` ファイルを含めます。

```
#include "SmApi.h"
```

以下の関数を外部的に表示できるようにします。

関数	説明
<a href="#">SmAuthenticate()</a> (P. 680)	ユーザの認証を実行します。この方式はユーザ認証情報を認証し、結果を返します。
<a href="#">SmAuthInit()</a> (P. 683)	方式を初期化します。この方式は、現時点で必要とするリソースすべてを初期化する必要があります。
<a href="#">SmAuthQuery()</a> (P. 684)	方式情報を要求します。
<a href="#">SmAuthRelease()</a> (P. 688)	SiteMinder がシャットダウンしているときに限り、方式を解放します。この方式は現時点でのそのリソースを解放する必要があります。

共有ライブラリ内のエントリ ポイントはそれぞれ指定された構文に従って定義される必要があります。

**注:** Microsoft Visual Studio を使用している場合は、関数アドレスをモジュール定義ファイル (.DEF ファイル) にエクスポートします。関数アドレスをエクスポートするには、.DEF ファイルを作成し、.DEF ファイルのエクスポートセクションに、先の表に記載されたすべての認証方式関数をリストアップします。.DEF ファイルを作成したら、Microsoft Visual Studio プロジェクトにそれを追加します。

DLL または共有ライブラリにコードをコンパイルします。Administrative UI で認証方式を定義するときは、認証方式プロパティ ダイアログ ボックスでこのライブラリ名を指定します。

カスタム認証方式ライブラリを書き込みコンパイルした後、カスタムライブラリを使用する認証方式を定義します。Administrative UI を使用して行います。

### SmAuthenticate()

SmAuthenticate() 関数はユーザ クレデンシヤルを認証します。

#### 構文

この関数の構文は、以下のようになります。

```
Sm_Api_Status_t SM_EXTERN SmAuthenticate (  
    const Sm_Api_Context_t*          lpApiContext,  
    const Sm_Api_UserContext_t*      lpUserContext,  
    const Sm_Api_UserCredentials_t*  lpUserCredentials,  
    const Sm_Api_Reason_t            nChallengeReason,  
    const char*                       lpszParam,  
    const char*                       lpszSharedSecret,  
    const int                          nBytesUserMsg,  
    char*                              lpszUserMsg,  
    const int                          nBytesErrMsg,  
    char*                              lpszErrMsg  
);
```

#### パラメータ

この関数には以下のパラメータがあります。

##### lpApiContext

[in] API コンテキスト構造体へのポインタを示します。

##### lpUserContext

[in] ユーザ コンテキスト構造体へのポインタを示します。

##### lpUserCredentials

[in] ユーザ クレデンシヤル コンテキスト構造体へのポインタを示します。

##### nChallengeReason

[in] 元のチャレンジの理由を示します。そうでない場合はゼロに設定します。

##### lpszParam

[in] 認証方式に対して指定される NULL で終わるパラメータ文字列が含まれるバッファへのポインタを示します。

##### lpszSharedSecret

[in] 認証方式に対して指定される NULL で終わる共有秘密キー文字列が含まれるバッファへのポインタを示します。

### nBytesUserMsg

[in] ユーザメッセージを受信する `lpszUserMsg` バッファの最大サイズを示します（文字列の終了文字を含む 4097 バイト）。

### lpszUserMsg

[out] ユーザメッセージを受信する出力バッファへのポインタを示します。このメッセージは、この方式の開発者が `SiteMinder` の外部のメカニズムによってユーザに提示したいチャレンジテキストまたは他のメッセージにすることができます。エージェントは `HTTP_SM_USERMSG` HTTP 変数にこのメッセージを保存します。`RADIUS` 認証については、ユーザメッセージが `REPLY-MESSAGE` レスポンス属性で返されます。

### nBytesErrMsg

[in] `lpszErrMsg` エラー バッファの最大サイズ（文字列の終端文字を含む 4097 バイト）を示します。

### lpszErrMsg

[out] エラー テキストを受信する出力バッファへのポインタを示します。`SiteMinder` にエラーメッセージを返すにはこのバッファを使用します。

## 戻り値

この関数は、`SM_MAKEAUTH_STATUSVALUE` マクロを使用して戻り値を作成します。このマクロは `SmApi.h` で定義されています。コマンドの構文は以下のとおりです。

```
SM_MAKEAUTH_STATUSVALUE(status, reason)
```

このマクロには次の 2 つのパラメータがあります。

- *status*—An `Sm_AuthApi_Status_t` enumeration. `SmAuthenticate()` が特定時に呼び出される場合、および認証中に呼び出される場合、異なるステータスコードを以下の通り返すことができます。

不明瞭解消フェーズ中：

- `Sm_AuthApi_NoUserContext`
- `Sm_AuthApi_Success`
- `Sm_AuthApi_SuccessUserDN`
- `Sm_AuthApi_SuccessUserFilter`
- `Sm_AuthApi_Attempt`
- `Sm_AuthApi_Failure`

認証フェーズ中：

- `Sm_AuthApi_Accept`
- `Sm_AuthApi_Reject`
- `Sm_AuthApi_Challenge`
- `Sm_AuthApi_Failure`

- *理由* — `Sm_Api_Reason_t` 列挙：

- 値 0 ～ 31999 は SiteMinder による使用のため予約されています。
- 値 32000 ～ 32767 はユーザ定義の理由で利用可能です。

## SmAuthInit()

SmAuthInit() 関数を使用すると、認証方式で独自の初期化処理を実行できます。このコールは方式が最初にロードされるときに 1 回行われます。

### 構文

この関数の構文は、以下のようになります。

```
Sm_Api_Status_t SM_EXTERN SmAuthInit (  
    const char*    lpszParam,  
    const char*    lpszSharedSecret  
);
```

### パラメータ

この関数には以下のパラメータがあります。

#### lpszParam

[in] 認証方式に対して指定される NULL で終わるパラメータ文字列が含まれるバッファへのポインタを示します。

#### lpszSharedSecret

[in] 認証方式に対して指定される NULL で終わる共有秘密キー文字列が含まれるバッファへのポインタを示します。

### 戻り値

この関数は以下のいずれかの値を返します。

#### Sm\_Api\_Success。

関数が正常に完了したことを示します。

#### Sm\_Api\_Failure。

関数が失敗したことを示します。この方式はロードされません。

### SmAuthQuery()

SmAuthQuery() 関数は、認証方式に関する情報を返します。

#### 構文

この関数の構文は、以下のようになります。

```
Sm_Api_Status_t SM_EXTERN SmAuthQuery (  
    const char*          lpszParam,  
    const char*          lpszSharedSecret,  
    const Sm_Api_QueryCode_t code,  
    char*                lpszStatusBuffer,  
    int*                 lpnStatusCode  
);
```

#### パラメータ

この関数には以下のパラメータがあります。

##### lpszParam

[in] 認証方式に対して指定される NULL で終わるパラメータ文字列が含まれるバッファへのポインタを示します。

##### lpszSharedSecret

[in] 認証方式に対して指定される NULL で終わる共有秘密キー文字列が含まれるバッファへのポインタを示します。

##### コード

[in] enum タイプ Sm\_Api\_QueryCode\_t によって定義されるような要求コード。以下の値が含まれます。

- Sm\_Api\_QueryDescription。方式の説明を要求します。この方式は、lpszStatusBuffer によって説明を、および lpnStatusCode によって認証 API バージョン番号を返します。
- Sm\_Api\_QueryCredentialsReq。方式に必要な認証情報とそれらを取得する場所を要求します。この方式は lpnStatusCode によってビットマスクを返す必要があります。これは、認証に必要な認証情報を表します。

個別のフラグは enum タイプ Sm\_Api\_Credentials\_t によって定義されます。この方式では、lpszStatusBuffer パラメータを使用して認証情報を取得できる場所 (HTTP URL) を返すことができます。

##### lpszStatusBuffer

文字列レスポンスを受信します。最大 `Sm_Api_StatusBufSize` 文字まで返すことができます。

### lpnStatusCode

このコールが方式の説明を要求する場合は、認証 API のバージョン番号を示す数値レスポンスを受信します。サポートされているバージョンは `Sm_Api_Version_V4` と `Sm_Api_Version_V4_1` です。

このコールが方式に必要な認証情報を要求する場合、この方式は 1 つ以上の認証情報を返します。「備考」を参照してください。これらの定数は `SmApi.h` で定義されています。

### 戻り値

この関数は以下のいずれかの値を返します。

#### `Sm_Api_Success`

関数が正常に完了したことを示します。

#### `Sm_Api_Failure`

呼び出し元が無効なリクエストコードを指定したことを示します。

### 注釈

方式は列挙タイプ `Sm_Api_QueryCode_t` によって指定されるような要求コードをサポートします。呼び出し元は方式に問い合わせを行い、認証に必要な認証情報の種類およびそれらを取得する場所を確認します。

認証情報は `enum` タイプ `Sm_Api_Credentials_t` によって定義され、複数の認証情報を要求するために組み合わせることができます。呼び出し元は要求された認証情報を収集し、`Sm_Api_UserCredentials_t` コンテキスト構造にそれらを置き、`SmAuthenticate()` 関数を呼び出します。

`Sm_Api_Credentials_t` 内の個別のフラグは以下のとおりです。

- `Sm_Api_Cred_None` -- 認証情報は必要ありません。
- `Sm_Api_Cred_Basic` -- ユーザ名とパスワードが必要です。
- `Sm_Api_Cred_Digest` -- 必要なユーザ名とパスワードはダイジェストプロトコルを使用して交換されます。
- `Sm_Api_Cred_X509Cert` -- 完全な X.509 クライアント証明書が必要です。 `Sm_Api_Cred_SSLRequired` が指定される必要があります。

- `Sm_AuthApi_Cred_X509CertUserDN` -- X.509 クライアント証明書からのユーザ DN が必要です。 `Sm_AuthApi_Cred_SSLRequired` が指定される必要があります。
- `Sm_AuthApi_Cred_X509CertIssuerDN` -- X.509 クライアント証明書からの発行者 DN が必要です。 `Sm_AuthApi_Cred_SSLRequired` が指定される必要があります。
- `Sm_AuthApi_Cred_CertOrBasic` -- 証明書またはユーザ名とパスワードが必要です。
- `Sm_AuthApi_Cred_CertOrForm` -- 証明書またはフォーム ベースのユーザ名およびパスワードが必要です。
- `Sm_AuthApi_Cred_SSLRequired` -- SSL 接続が必要です。 `https URL` へのリダイレクトが指定される必要があります。
- `Sm_AuthApi_Cred_NTChalResp` -- 必要なユーザ名およびパスワードは NT チャレンジレスポンス プロトコルを使用して交換されます。
- `Sm_AuthApi_Cred_AllowSaveCreds` -- 認証情報がユーザによって 30 日間保存できるかどうかを示します。ユーザが認証情報を保存する場合、保護されているリソースにアクセスするごとに、ユーザ名やパスワードなどの認証情報を入力する必要はありません。
- `Sm_AuthApi_Cred_FormRequired` -- フォーム ベースのユーザ名とパスワードが必要です。フォームが含まれる URL へのリダイレクトが指定される必要があります。
- `Sm_AuthApi_Cred_PreserveSessionId` -- 現在のセッションがまだ有効な場合、セッション ID を保存する必要があります。
- `Sm_AuthApi_Cred_DoNotChallenge` -- 認証情報の要求を行いません。
- `Sm_AuthAPI_Cred_AllowAnonymous` -- 匿名の ID の検証を許可します。

### 例: ステータスコードの設定

- 匿名方式はこれらの認証情報フラグを組み合わせます。

```
*lpnStatusCode = Sm_AuthApi_Cred_None|Sm_AuthApi_Cred_AllowAnonymous;
```
- ユーザ名およびパスワードを収集します。エージェントは、HTTP 基本認証を使用してユーザに要求を行います。

```
*lpnStatusCode = Sm_AuthApi_Cred_Basic;
```

- ユーザ名およびパスワードを収集します。エージェントは、HTTP 基本認証を使用して、SSL を介してユーザに要求を行います。

```
*lpnStatusCode = Sm AuthApi_Cred_Basic |
                    Sm AuthApi_Cred_SSLRequired;
strcpy(lpszStatusBuffer,
        "https://xxx.yyy.zzz/secure.ssc?basic")
```

- ユーザ名およびパスワードを収集します。エージェントは、HTML フォームを使用して、SSL を介してユーザに要求を行います。

```
*lpnStatusCode = Sm AuthApi_Cred_Basic |
                    Sm AuthApi_Cred_FormRequired;
strcpy(lpszStatusBuffer,
        "https://xxx.yyy.zzz/getcredentials.fcc")
```

- X.509 証明書を収集します。エージェントは、証明書の入力をユーザに要求します。

```
*lpnStatusCode = Sm AuthApi_Cred_SSLRequired |
                    Sm AuthApi_Cred_X509Cert;
strcpy(lpszStatusBuffer,
        "https://xxx.yyy.zzz/getcert.scc?cert")
```

- X.509 証明書、ユーザ名およびパスワードを収集します。エージェントは、HTTP 基本認証を使用して、SSL を介してユーザに要求を行います。

```
*lpnStatusCode = Sm AuthApi_Cred_Basic |
                    Sm AuthApi_Cred_SSLRequired |
                    Sm AuthApi_Cred_X509Cert;
strcpy(lpszStatusBuffer,
        "https://xxx.yyy.zzz/getcert.scc?cert+basic")
```

### SmAuthRelease()

**SmAuthRelease()** 関数を使用すると、認証方式で独自の要約手順を実行できます。呼び出し元は、**SiteMinder** のシャットダウン時にこの呼び出しを 1 回行います。

#### 構文

この関数の構文は、以下のようになります。

```
Sm_Api_Status_t SM_EXTERN SmAuthRelease (  
    const char*    lpszParam  
    const char*    lpszSharedSecret  
);
```

#### パラメータ

この関数には以下のパラメータがあります。

##### lpszParam

[in] 認証方式に対して指定される **NULL** で終わるパラメータ文字列が含まれるバッファへのポインタを示します。

##### lpszSharedSecret

[in] 認証方式に対して指定される **NULL** で終わる共有秘密キー文字列が含まれるバッファへのポインタを示します。

#### 戻り値

この関数は以下のいずれかの値を返します。

##### Sm\_Api\_Success。

関数が正常に完了したことを示します。

##### Sm\_Api\_Failure。

関数が失敗したことを示します。この方式はロードされません。

# 第 8 章: C 用の許可 API

---

このセクションには、以下のトピックが含まれています。

[認可 API の概要 \(P. 689\)](#)

[アクティブな式 \(P. 691\)](#)

[認可関数宣言 \(P. 695\)](#)

[アクティブな式の例 \(P. 697\)](#)

## 認可 API の概要

許可 API を使用すると、カスタム アクセス制御機能を実装できます。カスタム アクセス制御機能を実装するには、以下を行う必要があります。

1. 許可 API をサポートし、必要なカスタム機能を提供する共有ライブラリを開発します。

共有ライブラリには、エクスポート可能な記号として定義された 1 つ以上の関数が含まれる必要があります。**SmApi.h** はカスタム ポリシー、ルールおよびレスポンス プラグインを作成するのに必要なデータ構造をすべて定義します。

2. 共有ライブラリを以下のいずれかのデフォルトの場所にインストールします。

- UNIX プラットフォームでは、**SiteMinder lib** ディレクトリ
- Windows プラットフォームでは、**SiteMinder bin** ディレクトリ

3. Administrative UI で以下の 1 つ以上を定義します。

- アクティブ ポリシー -- 外部のビジネス ロジックに基づいて動的な認可を提供するポリシー。

たとえば、ユーザがアクティブなポリシー式のパラメータ (*param*) フィールドで定義されるような LDAP ディレクトリ内の特定の組織単位 (*ou*) に属する場合に真を返すアクティブなポリシーを定義することができます。

- アクティブ レスポンス -- 共有ライブラリから返されたカスタム レスポンス。 アクティブ レスポンスの使用は、ユーザに固有の権限情報を定義できる 1 つの方法です。

たとえば、ユーザがアクティブなレスポンス表現の *param* フィールドで指定された *ou* に属する場合、ユーザの共通名 (*cn*) を返すアクティブなレスポンスを定義することができます。

- アクティブ ルール -- 外部のビジネス ロジックに基づいて動的な認可を提供するルール。

たとえば、ユーザがレルムを表示する権限があるディレクトリ管理者など、グループのメンバである場合に真を返すアクティブなルールを定義することができます。

## ファイルを含める

認可 API を拡張する場合、`SmApi.h` ヘッダ ファイルを含めます。

```
#include "SmApi.h"
```

## アクティブな式

アクティブな式は、アクティブなポリシー、ルールまたはレスポンスを含む変数定義の文字列です。アクティブな式は以下の構文を使用して、Administrative UI で作成されます。

```
<@ lib=<lib-spec> func=<func-spec> param=<func-params>@>
```

この構文例において、

- *lib-spec* はカスタム共有ライブラリへのパスです。この節は必須です。ライブラリをデフォルトの場所に配置する場合、指定する必要があるのはライブラリファイル名のみでパスは不要です。また、拡張子 *.dll* または *.so* はオプションです。
- *func-spec* は、共有ライブラリで定義されたユーザ定義の、外部的に表示できる関数の名前です。この節は必須です。
- *func-params* は関数に渡されるパラメータ文字列です。この節はオプションです。

SiteMinder は、[アクティブルールエディタ]、[アクティブポリシーエディタ] または [アクティブレスポンス属性エディタ] ダイアログボックスで提供される情報からアクティブな式を組み立てます。

### SiteMinder がアクティブな式を解釈する方法

アプリケーションのアクティブな式は、以下のタスクを開始します。

- アクティブな式で指定された共有ライブラリをロードします。
- アクティブな式で指定されたユーザ定義関数を呼び出します。
- ユーザ定義関数に、オプションのパラメータ文字列とコンテキスト情報、すなわち API コンテキスト (*Sm\_Api\_Context\_t*)、要求コンテキスト (*Sm\_Api\_RequestContext\_t*) およびユーザコンテキスト (*Sm\_Api\_UserContext\_t*) を渡します。

共有ライブラリ内の指定されたユーザ定義関数は、結果を *lpszOutBuf* パラメータ内の **SiteMinder** に返します。 **SiteMinder** は、アクティブな式のタイプに従って以下のようにこの結果を解釈します。

- アクティブ ポリシー -- 関数の呼び出しが失敗した場合や、 *lpszOutBuf* で返された結果が空の場合、認可が拒否されます。

*lpszOutBuf* で返された結果が次の文字列（大文字と小文字を区別しない）のいずれかに一致する場合、ポリシーは開始しません。 **FALSE**、**F** または **0**。

他の結果値の場合ポリシーが開始します。

- アクティブ ルール -- 関数の呼び出しが失敗した場合や、 *lpszOutBuf* で返された結果が空の場合、以下の動作が発生します。

- アクセス許可ルールでは、ルールが起動しません。

- アクセス拒否ルールでは、ルールが起動します。

これ以外の場合、動作はアクティブ ポリシーと同じです。

- アクティブ レスポンス -- 結果はレスポンス属性値を表す文字列です。 **SiteMinder** がどのようにこの値を使用するかは **Administrative UI** で指定されたレスポンス属性によって決定されます。例：

- **WebAgent-OnReject-Redirect**。この属性を与えられると、 **SiteMinder** は、レスポンス値がリソースへのアクセスを拒否されたユーザをリダイレクトするために **URL** など場所を指定すると予測します。

たとえば、アクティブな式のオプションの *param* 変数でグループ名を指定し、次に、パスを戻す **URL** を決定するために関数内のグループ名の検査を行うことができます。

- **WebAgent-HTTP-Cookie-Variable**。この属性を与えられると、 **SiteMinder** は、ユーザの共通名などのレスポンス値が **Cookie** 変数に割り当てられることになっていると予期します。フォームをパーソナライズするためにユーザの共通名を表示するなどのように、レスポンス値を好きなように使用できます。

**SiteMinder** レスポンス属性エディタで **Cookie** 名を指定します。

## アクティブ ルールの定義

アクティブ ルールは、アクティブ ルール エディタ ダイアログ ボックスを使用して、**Administrative UI** で定義されます。 [ルール プロパティ] ダイアログ ボックスからこのエディタにアクセスするには、 [詳細] グループ ボックス内の [アクティブ ルール] タブを選択し、次に、 [編集] をクリックします。

## アクティブ レスポンスの定義

アクティブ レスポンスは、レスポンス属性エディタ ダイアログ ボックスを使用して、**Administrative UI** で定義されます。

[レスポンス プロパティ] ダイアログ ボックスから、 [作成] をクリックしてエディタにアクセスし、 [属性のセットアップ] タブ上の [属性の種類] グループ ボックス内の [アクティブ レスポンス] ボタンを選択します。

## アクティブ ポリシーの定義

アクティブ ポリシーは、アクティブ ポリシー エディタ ダイアログ ボックスを使用して、**Administrative UI** で定義されます。

[ポリシー プロパティ] ダイアログ ボックスから、 [詳細] タブを選択し [編集] をクリックしてこのエディタにアクセスします。

## ポリシー サーバへの HTTP ヘッダおよび Cookie のパス

任意のカスタム キー/値ペアを現在のセッションに追加して、HTTP ヘッダ および Cookie をポリシー サーバに渡すことができます。これらのキー/値ペアはセッションストアに保持され、有効期間はセッションと同じです。名前/値ペアは、セッションストア データベース内の Expiry Data テーブルに保存されます。1つのセッションに対し最大5つのエントリが可能です。アクティブ プラグイン コードは、UserContext 構造

(Sm\_Api\_UserContext\_t\* lpUserContext) を使用して fSetProp と fGetProp をそれぞれ呼び出すことによりこれらの名前/値ペアを設定し取得することができます。値を設定するには、fSetProp は値として

SM\_SESSIONVAR(<name>) および lpszValueBuf として lpszPropName で呼び出されます。値を取得するには、fGetProp は SM\_SESSIONVAR(<name>)/SM\_SESSIONVAR として lpszPropName で呼び出され、値/値は lpszValueBuf で返されます。

以下の点に注意してください。

- Expiry Data テーブルで名前/値ペアを設定するには、名前は SM\_SESSIONVAR(<name>) 形式のみで渡される必要があります。名前は最大 32 文字の長さで、文字またはアンダースコアで始まることができ、文字、数字またはアンダースコアのみが使用できます。設定できる値の長さは最大 4000 文字です。
- 名前の値を取得するには、名前は SM\_SESSIONVAR(<name>) 形式で渡される必要があります。<name> に対応する Expiry Data テーブル内の値が返されます。SM\_SESSIONVAR のみが渡される場合（名前は含まれない）、そのセッションの名前がすべてキャレットで区切られた形式で返されます。
- Expiry Data テーブル内の名前/値ペアを削除する場合、名前は SM\_SESSIONVAR(<name>) として渡されます。また値は空白として渡される必要があります。

## 認可関数宣言

共有ライブラリには、適切なエントリ ポイントが必要です。共有ライブラリ内のエントリ ポイントはそれぞれ 1 つ以上のアクティブな式を表し、指定された構文に従って定義される必要があります。

**注:** Microsoft Visual Studio を使用している場合は、関数アドレスをモジュール定義ファイル (.DEF ファイル) にエクスポートします。関数アドレスをエクスポートするには、.DEF ファイルを作成し、.DEF ファイルのエクスポートセクションに、アクティブ ルールまたはアクティブ ポリシーから呼び出す関数をリストアップします。.DEF ファイルを作成した後、Microsoft Visual Studio プロジェクトにそれを追加します。

## ユーザ定義関数

ポリシー サーバは、ユーザ定義関数を呼び出しカスタム ポリシー、ルールまたはレスポンス操作を実行します。

任意の名前の関数を割り当てることができます。Administrative UI で定義するアクティブな式によって、SiteMinder に関数名および関数が存在する共有ライブラリの名前を知らせます。

### 構文

```
int SM_EXTERN <func-spec> (
    const Sm_Api_Context_t*      lpApiContext,
    const Sm_Api_UserContext_t*  lpUserContext,
    const Sm_Api_RequestContext_t* lpReqContext,
    const char*                  lpszParam,
    const int                     nBytesOutBuf,
    char*                         lpszOutBuf,
    const int                     nBytesErrBuf,
    char*                         lpszErrBuf
);
```

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。
<i>lpUserContext</i>	I	ユーザ コンテキスト構造へのポインタ。
<i>lpReqContext</i>	I	要求コンテキスト構造へのポインタ。

パラメータ	I/O	説明
<i>lpszParam</i>	I	<Param-String> で指定された NULL で終わるパラメータ文字列が含まれるバッファへのポインタ。
<i>nBytesOutBuf</i>	I	出力結果バッファ（4097 バイト）の最大サイズ。
<i>lpszOutBuf</i>	O	SiteMinder に送信する結果の出力バッファ。
<i>nBytesErrBuf</i>	I	出力エラー バッファ（4097 バイト）の最大サイズ。
<i>lpszErrBuf</i>	O	エラー テキストを受信するための出力バッファ。 SiteMinder は、デバッグ ログ ファイル内のエラー テキストを表示します。

### 戻り値

- 実行が成功すると、関数は 0 より大きい値、すなわち結果バッファ *lpszOutBuf* 内のバイトの総数を返します。
- エラーが発生した場合、関数は -1 を返し、エラー バッファ *lpszErrBuf* に適用可能なエラー メッセージを保存します。
- アクティブ ルールに対して使用される場合、0 の戻り値（空の文字列は、*lpszOutbuf* に戻されました）は、以下のいずれかのアクションをもたらします。
  - アクセス許可ルールでは、ルールが起動しません。
  - アクセス拒否ルールでは、ルールが起動します。

## SmQueryVersion()

ポリシー サーバは、この関数を呼び出しカスタム ライブラリが準拠している許可 API のバージョンを決定します。

### 構文

```
int SmQueryVersion (
    const Sm_Api_Context_t* lpApiContext
);
```

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。

### 戻り値

許可 API のバージョン番号を返します。現在、サポートされたバージョンは `Sm_Api_Version_V3` です。この定数は `SmApi.h` で定義されています。

## アクティブな式の例

以下のセクションで提供されるこのサンプルは以下にあります。

```
sdk¥samples¥smazapi¥smazapi.cpp
```

以下に示すような各サンプルに先行する構文は、

```
<@ lib="SmAzAPI" func="activeRule" param="" @>
```

[SiteMinder アクティブ ルール エディタ]、[アクティブ ポリシー エディタ] または [アクティブ レスポンス属性エディタ] ダイアログ ボックスにおいてダイアログ ボックスでユーザが提供する情報から SiteMinder が作成する生成されたスクリプトの例です。

UNIX 用のサンプルのアクティブな式を作るには、以下にある `makefile` を使用します。

```
<install_path>¥sdk¥samples¥smazapi¥makefile
```

## アクティブ ルールの例

以下の例では、ユーザがレルムを表示する特別なアクセス許可がある場合、真を返します。ユーザにディレクトリ管理者権限がある場合、ユーザはレルムを表示できます。

```
<@ lib="SmAzAPI" func="activeRule" param="" @>
*****
int SM_EXTERN activeRule(
const Sm_Api_Context_t* lpApiContext,
// the structure that provides API context
const Sm_Api_UserContext_t* lpUserContext,
// the structure that provides user context
const Sm_Api_RequestContext_t* lpReqContext,
// the structure that provides request context
const char* lpszParam,
// the parameter string (null-terminated)
const int nBytesOutBuf,
// the maximum size of the output buffer
char* lpszOutBuf,
// the output buffer to hold the null-terminated result
const int nBytesErrBuf,
// the maximum size of the error message buffer
char* lpszErrBuf)
// the output buffer to hold the null-terminated error message
{
/* User Context is required to use the functions like fGetProp, fSetProp.. */
if(!lpUserContext->bIsUserContext)
{
strncpy (lpszErrBuf, "No User Context ", nBytesErrBuf);
lpszErrBuf[nBytesErrBuf-1] = '\0';
return -1;
}
/*
// The DN to look for the attribute "uniquemember"
// If the user is listed as the member of the above attribute,
// it has directory manager privileges.
*/
char lpszDn[] = "cn=Directory Administrators,ou=Groups,o=airius.com";
char lpszDnvalue[256];
memset(lpszDnvalue, 0, sizeof(lpszDnvalue));
/*
// fGetDnProp function is used to retrieve an attribute value
// in a directory entry.
*/
```

```
int getResult = lpUserContext->fGetDnProp(
    lpUserContext->lpParam,
    lpszDn,
    "uniquemember",
    sizeof(lpszDnvalue),
    lpszDnvalue);
/*
// If no error occurs, fGenDnProp will return the length of the
// buffer lpszDnvalue. Otherwise the function returns 0.
*/
if(getResult > 0)
{
    /* Check to see if the user is present in the list. */
    if(strpbrk(lpszDnvalue, lpUserContext->lpszUserName) != NULL)
    {
        /* The result "true" is placed in the output buffer. */
        strncpy(lpszOutBuf, "true", nBytesOutBuf);
        lpszOutBuf[nBytesOutBuf-1] = '\0';
        return strlen(lpszOutBuf);
    }

    else

    {
        strncpy(lpszOutBuf, "false", nBytesOutBuf);
        lpszOutBuf[nBytesOutBuf-1] = '\0';
        return strlen(lpszOutBuf);
    }
}

else

{
    strncpy(lpszErrBuf, "Failed to get attribute value for the DN ",
              nBytesErrBuf);
    strcat( (lpszErrBuf + strlen(lpszErrBuf)), lpszDn,
            (nBytesErrBuf-strlen(lpszErrBuf)));
    lpszErrBuf[nBytesErrBuf-1] = '\0';
    return -1;
}

/* everything failed.... */

return 0;
}
```

## アクティブ レスポンスの例

このアクティブ レスポンスは、ユーザがアクティブなレスポンス表現のパラメータ (param) フィールドで指定された組織単位に属する場合、ユーザの共通名 (cn) を返します。

```
<@ lib="SmAzAPI" func="activeResponse" param="Human Resources" @>
*****
int SM_EXTERN activeResponse(
const Sm_Api_Context_t* lpApiContext,
/* the structure that provides API context */
const Sm_Api_UserContext_t* lpUserContext,
/* the structure that provides user context */
const Sm_Api_RequestContext_t* lpReqContext,
/* the structure that provides request context */
const char* lpszParam,
/* the parameter string (null-terminated) */
const int nBytesOutBuf,
/* the maximum size of the output buffer */
char* lpszOutBuf,
/* the output buffer to hold the null-terminated attribute value */
const int nBytesErrBuf,
/* the maximum size of the error message buffer */
char* lpszErrBuf)
/* the output buffer to hold the null-terminated error message */
{
memset(lpszOutBuf, 0, sizeof(lpszOutBuf));
if(!lpUserContext->bIsUserContext)
{
strncpy (lpszErrBuf, "No User Context ", nBytesErrBuf);
lpszErrBuf[nBytesErrBuf-1] = '\0';
return -1;
}
/* Store all the organizational units to which the user belongs. */
char lpszOrgUnit[30];
memset(lpszOrgUnit, 0, sizeof(lpszOrgUnit));
/* store the common name of the user. */
char lpszCN[30];
memset(lpszCN, 0, sizeof(lpszCN));
/* Check to see if a parameter is requested. */
if(lpszParam == NULL || strlen(lpszParam) == 0)
{
strncpy (lpszErrBuf, "Organizational unit is not entered ",
nBytesErrBuf);
lpszErrBuf[nBytesErrBuf-1] = '\0';
return -1;
}
/* Get all the organization units to which the user belongs. */
```

```
int getResult = lpUserContext->fGetProp (
    lpUserContext->lpParam,
    "ou",          /* Attribute name */
    sizeof (lpszOrgUnit), lpszOrgUnit);
if (getResult < 0)
{
    strncpy (lpszErrBuf,
        "Failed to get organization unit for the user's profile ",
        nBytesErrBuf);
    strncat( (lpszErrBuf + strlen(lpszErrBuf)),
        lpUserContext->lpszUserName,
        (nBytesErrBuf-strlen(lpszErrBuf)));
    lpszErrBuf[nBytesErrBuf-1] = '\0';
    return -1;
}
else
{
    /* Check if the user belongs to the organization unit that is requested. */
    if(strstr(lpszOrgUnit, lpszParam) != NULL)
    {
        if((lpUserContext->fGetProp(lpUserContext->lpParam,
            "cn",sizeof(lpszCN),lpszCN)) > 0)
        {
            strncpy(lpszOutBuf, lpszCN, nBytesOutBuf);
            lpszOutBuf[nBytesOutBuf-1] = '\0';
            return strlen(lpszOutBuf);
        } /* end of fGetProp */
        else
        {
            strncpy (lpszErrBuf,
                "Failed to get user common name from user's profile attribute ",
                nBytesErrBuf);
            strncat( (lpszErrBuf + strlen(lpszErrBuf)),
                lpUserContext->lpszUserName,
                (nBytesErrBuf-strlen(lpszErrBuf)));
            lpszErrBuf[nBytesErrBuf-1] = '\0';
            return -1;
        }
    } /* end of strstr */
    else
    {
        strncpy (lpszErrBuf,
            "The user does not belong to the requested organizational unit ",
            nBytesErrBuf);
        lpszErrBuf[nBytesErrBuf-1] = '\0';
        return -1;
    }
}
}
```

```
        /* everything failed.... */  
        return 0;  
    }  
#ifndef _WIN32  
}  
  
#endif
```

# 第 9 章: トンネル サービス API ガイド

---

このセクションには、以下のトピックが含まれています。

[トンネル サービス API の概要 \(P. 703\)](#)

[カスタム トンネル サービスの開発 \(P. 704\)](#)

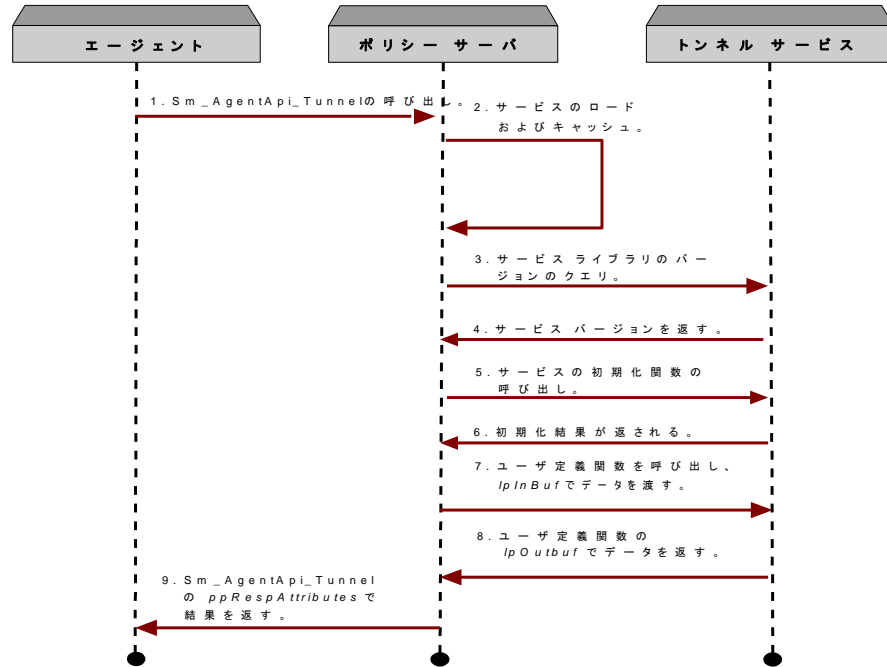
## トンネル サービス API の概要

トンネル サービス API は、エージェントとトンネル サービスをサポートする共有ライブラリ間のデータの安全な転送を提供します。

エージェントがトンネル要求をポリシー サーバに送信する場合、要求には次のものが含まれます。

- サービス ライブラリ の名前
- サービス ライブラリ で呼び出される関数
- 関数に渡されるデータ

ポリシー サーバは適切なサービスを初期化し、要求された関数を呼び出し、データを関数へ渡します。サービスがそのタスクを実行すると、ポリシー サーバはエージェントに結果を返します。以下の図は、トンネル サービス プロセスを示しています。



## カスタムトンネル サービスの開発

トンネル サービスはそれぞれ、トンネル サービス API をサポートする共有ライブラリのインスタンスです。トンネル サービスをサポートするには、新しい共有ライブラリを構築する必要があります。

以下のいずれかの場所に共有ライブラリをインストールします。

- UNIX プラットフォームでは、SiteMinder lib ディレクトリ
- Windows プラットフォームでは、SiteMinder bin ディレクトリ

## ファイルを含める

トンネル サービスを開発するには、SmApi.h ヘッダ ファイルを含めます。

```
#include "SmApi.h"
```

## トンネル サービス API リファレンス

共有ライブラリは、外部的に表示できるエントリ ポイントとして以下の関数を提供する必要があります。

関数	説明
SmQueryVersion()	カスタム ライブラリが準拠しているトンネル サービス API のバージョンを要求します。
SmTunnelInit()	トンネル サービスを初期化します。
SmTunnelRelease()	トンネル サービスを解放します。
ユーザ定義関数	トンネル エージェントが要求している関数を呼び出します。

共有ライブラリ内のエントリ ポイントはそれぞれ指定された構文に従って定義される必要があります。

**注:** Microsoft Visual Studio を使用している場合は、関数アドレスをモジュール定義ファイル (.DEF ファイル) にエクスポートします。関数アドレスをエクスポートするには、.DEF ファイルを作成し、.DEF ファイルのエクスポートセクションに、先の表に記載されたすべてのトンネル サービス関数をリストアップします。.DEF ファイルを作成したら、Microsoft Visual Studio プロジェクトにそれを追加します。

### SmQueryVersion()

SiteMinder は、この関数を呼び出しカスタム ライブラリが準拠しているトンネル サービス API のバージョンを要求します。

#### 構文

```
int SM_EXTERN SmQueryVersion (
    const Sm_Api_Context_t* lpApiContext
);
```

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。

### 戻り値

トンネル サービス API のバージョン番号を返します。現在サポートされるバージョンは `Sm_Api_Version_V4` と `Sm_Api_Version_V4_1` です。バージョン定数は `SmApi.h` で定義されています。

## SmTunnelInit()

SiteMinder はこの関数を呼び出し、トンネル サービスがそれ自身の初期化処理を実行できるようにします。このコールは、トンネル サービスが初めてロードされるたびに 1 回行われます。情報は次に使用するためキャッシュされます。

### 構文

```
int SM_EXTERN SmTunnelInit (  
    void**                ppServiceHandle,  
    const Sm_Api_Context_t* lpApiContext,  
    const int              nBytesStatusBuf,  
    char*                  lpszStatusBuf  
);
```

パラメータ	I/O	説明
<i>ppServiceHandle</i>	I	トンネル サービス ライブラリによって使用される内部ポインタ。
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。
<i>nBytesStatusBuf</i>	O	出力ステータス バッファの最大サイズ。
<i>lpszStatusBuf</i>	O	出力バッファは、トンネル サービスからステータスメッセージを受信します。

### 戻り値

成功の場合は 0、失敗の場合は -1 を返します。

## SmTunnelRelease()

トンネル サービスは自身のランダウン手順を実行できます。この呼び出しは、SiteMinder がシャットダウンしているときに 1 回実行されます。

### 構文

```
void SM_EXTERN SmTunnelRelease (
    void*                pServiceHandle,
    const Sm_Api_Context_t* lpApiContext
);
```

パラメータ	I/O	説明
<i>pServiceHandle</i>	I	トンネル サービス ライブラリによって使用される内部ポインタ。
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。

## ユーザ定義関数

トンネル エージェントは、この関数をリクエストします。

### 構文

```
int SM_EXTERN <func-spec> (
    void*                pServiceHandle,
    const Sm_Api_Context_t* lpApiContext,
    const Sm_Api_RequestContext_t* lpReqContext,
    const Sm_Api_TunnelContext_t* lpTunnelContext,
    const int            nBytesInBuf,
    void*                lpInBuf,
    const int            nBytesOutBuf,
    void*                lpOutBuf,
    const int            nBytesStatusBuf,
    char*                lpszStatusBuf
);
```

パラメータ	I/O	説明
<i>pServiceHandle</i>	I	トンネル サービス ライブラリによって使用される内部ポインタ。
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。

パラメータ	I/O	説明
lpReqContext	I	API 要求構造へのポインタ。
lpTunnelContext	I	API トンネル コンテキストへのポインタ。
nBytesInBuf	I	入力バッファ内のバイト数。
lpInBuf	I	リモート エージェントから送信される情報が含まれる入力バッファ。
nBytesOutBuf	O	出力結果バッファの最大サイズ。
lpOutBuf	O	結果を受信する出力バッファ。
nBytesStatusBuf	O	ステータス バッファの最大サイズ。
lpSzStatusBuf	O	ステータスを受信するステータス バッファ。

# 第 10 章: イベント API ガイド

---

このセクションには、以下のトピックが含まれています。

[イベント API 概要 \(P. 709\)](#)

[イベント API セットアップ \(P. 710\)](#)

[イベント プロバイダ 構造 \(P. 710\)](#)

[イベント API リファレンス \(P. 715\)](#)

[イベント 関数宣言 \(P. 762\)](#)

[アクティブ ポリシーの例 \(P. 765\)](#)

[イベント ハンドラ に対する ポリシー サーバ の設定 \(P. 767\)](#)

## イベント API 概要

SiteMinder イベント API を使用すると、カスタム イベント ハンドラを作成できます。

イベント API によって、SiteMinder は外部のソース、プロバイダ、またはアプリケーションを使用してイベントをログ記録できます。その後、これらの他のソース、プロバイダ、またはアプリケーションを通してログ記録された情報にアクセスできます。

イベント API を使用すると、SiteMinder アクティビティの管理者にアラートするためのアプリケーションを構築できます。たとえば、監査サーバが開始するか、誰かが新規ポリシーを作成したときにイベントハンドラは管理者に電子メールを送信できます。

## イベント API セットアップ

各 イベント ハンドラは、イベント API プロバイダ インターフェースをサポートする共有ライブラリのインスタンスです。カスタム イベント ハンドラをサポートするには、共有ライブラリを構築する必要があります。

以下のいずれかの場所に共有ライブラリをインストールします。

- UNIX プラットフォームでは、SiteMinder lib ディレクトリ
- Windows プラットフォームでは、SiteMinder bin ディレクトリ

共有ライブラリは以下のエントリ ポイントをエクスポートする必要があります。

- SmEventInit()
- SmEventRecord()
- SmEventRelease()

イベント ハンドラを構築するには、SmEventApi.h ヘッダ ファイルを含めます。

```
#include "SmEventApi.h"
```

## イベント プロバイダ 構造

以下の表は、イベント プロバイダ API 関数と共に使用される構造定義のリストを示しています。

構造	説明
<a href="#">SmLog_Access_t</a> (P. 711)	アクセス イベントに関する情報が含まれます。
<a href="#">SmLog_EMS_t</a> (P. 713)	Entitlement Management Services (EMS) イベントに関する情報が含まれます。EMS イベントは、ディレクトリ オブジェクト上で実行されたアクションから生成されます。
<a href="#">SmLog_Obj_t</a> (P. 714)	オブジェクト イベントに関する情報が含まれます。
<a href="#">SmLog_System_t</a> (P. 715)	システム イベントに関する情報が含まれます。

## SmLog\_Access\_t

アクセス イベントに関する情報が含まれます。

### 構文

```
typedef struct SmLog_Access_s
{
    long nVersion;
    long nCurrentTime;
    Sm_Api_Reason_t nReason;
    char* szAgentName;
    char* szSessionId;
    char* szClientIp;
    char* szUserName;
    char* szDomainOid;
    char* szRealmName;
    char* szRealmOid;
    char* szAuthDirName;
    char* szAuthDirServer;
    char* szAuthDirNamespace;
    char* szServer;
    char* szResource;
    char* szAction;
    char* szTransactionId;
    char* szStatusMsg;
    char* szDomainName;
    char* szImpersonatorName;
    char* szImpersonatorDirName;
} SmLog_Access_t;
```

フィールド	説明
<i>nVersion</i>	ポリシー サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生したポリシー サーバ時間 (GMT) 。
<i>nReason</i>	理由識別子が含まれます。
<i>szAgentName</i>	エージェントの名前です。
<i>szSessionId</i>	セッション ID。
<i>szClientIp</i>	クライアントの IP アドレス。
<i>szUserName</i>	ユーザまたは管理者の完全な識別名。
<i>szDomainOid</i>	ドメイン オブジェクトのオブジェクト識別子。

フィールド	説明
<i>szRealmName</i>	リソースが保護されているレルムの名前。
<i>szRealmOid</i>	レルム オブジェクトに対するオブジェクト識別子。
<i>szAuthDirName</i>	ディレクトリの名前。
<i>szAuthDirServer</i>	ユーザ コンテキストが確立された SiteMinder ユーザ ディレクトリのディレクトリ サーバ。
<i>szAuthDirNamespace</i>	LDAP、WinNT または ODBC などのディレクトリ ネームスペース。
<i>szServer</i>	リソースを保持するサーバの名前。
<i>szResource</i>	リソースの名前。
<i>szAction</i>	リソース上で実行されるアクションのタイプ。Web リソースにアクセスする場合の典型的なアクションは GET です。
<i>szTransactionId</i>	<p>エージェントとポリシー サーバ間のトランザクションの識別子。エージェントはこの ID を設定し、ポリシー サーバがそれをログ記録します。</p> <p>エージェントがポリシー サーバに対して要求を発行する場合、ID を要求と関連付けます。エージェントは、このような要求を多く発行する可能性があるためポリシー サーバからの情報を元の要求と一致するために ID を使用します。</p>
<i>szStatusMsg</i>	ステータス メッセージ。
<i>szImpersonatorName</i>	<p>インパーソネータの名前を指定します。</p> <p>進行中のインパーソネートされたセッションがない場合、NULL に設定されます。</p>
<i>szDomainName</i>	ドメインの名前を指定します。
<i>szImpersonatorDirName</i>	<p>インパーソネータを認証するのに使用されるユーザ ディレクトリの名前を指定します。</p> <p>進行中のインパーソネートされたセッションがない場合、NULL に設定されます。</p>

詳細情報:

[Sm\\_Api\\_Reason\\_t \(P. 868\)](#)

## SmLog\_EMS\_t

Entitlement Management Services (EMS) イベントに関する情報が含まれます。EMS イベントは、ディレクトリ オブジェクト上で実行されたアクションから生成されます。

## 構文

```
typedef struct SmLog_EMS_s
{
    long nVersion;
    long nCurrentTime;
    char* szUserName;
    char* szSessionId;
    char* szDirName;
    char* szObjName;
    char* szObjPath;
    char* szObjClass;
    char* szOrgName;
    char* szRoleName;
    int szFieldDesc;
    char* szStatusMsg;
} SmLog_EMS_t;
```

フィールド	説明
<i>nVersion</i>	ポリシー サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生したポリシー サーバ時間 (GMT)。
<i>szUserName</i>	ユーザが管理者の場合、EMS イベントを開始した 管理者の ID。 イベント がエンドユーザイベントの場合、ユーザ名は <b>Registration</b> です。
<i>szSessionID</i>	EMS サービス セッション ID。
<i>szDirName</i>	EMS のイベントによって影響を受ける SiteMinder ディレクトリ の名前。
<i>szObjName</i>	このイベントによってターゲットにされるオブジェクトの名前。
<i>szObjPath</i>	オブジェクトの完全な識別名。
<i>szObjClass</i>	オブジェクトのクラス名。
<i>szOrgName</i>	オブジェクトの組織の名前。

フィールド	説明
<i>szRoleName</i>	オブジェクトが関連付けられるロールの名前 (ロールが含まれるイベントに対してのみ)。
<i>szFieldDesc</i>	イベントの説明。
<i>szStatusMsg</i>	ステータス メッセージ。

## SmLog\_Obj\_t

オブジェクト イベントに関する情報が含まれます。

### 構文

```
typedef struct SmLog_Obj_s
{
    long nVersion;
    long nCurrentTime;
    char* szUserName;
    char* szSessionId;
    char* szDomainOid;
    char* szObjName;
    char* szObjOid;
    char* szFieldDesc;
    char* szStatusMsg;
} SmLog_Obj_t;
```

フィールド	説明
<i>nVersion</i>	ポリシー サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生したポリシー サーバ時間 (GMT)。
<i>szUserName</i>	イベントをトリガしたユーザの名前。
<i>szSessionId</i>	セッション ID。
<i>szDomainOid</i>	ドメイン オブジェクトに対するオブジェクト識別子。
<i>szObjName</i>	オブジェクトの名前。
<i>szObjOid</i>	オブジェクトのオブジェクト識別子。
<i>szFieldDesc</i>	ユーザ DN。
<i>szStatusMsg</i>	ステータス メッセージ。

## SmLog\_System\_t

システム イベントに関する情報が含まれます。

### 構文

```
typedef struct SmLog_System_s
{
    long nVersion;
    long nCurrentTime;
    char* szName;
    char* szIpAddress;
    int nIpPort;
    char* szMsg;
} SmLog_System_t;
```

フィールド	説明
<i>nVersion</i>	ポリシー サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szName</i>	クライアント、データベースまたはディレクトリの名前。
<i>szIpAddress</i>	クライアントの IP アドレス。
<i>nIpPort</i>	クライアントが通信するポート番号。
<i>szMsg</i>	イベントに対する文字列メッセージを受信するバッファ。

## イベント API リファレンス

SiteMinder は 4 つのタイプのイベントをログ記録します。イベント分類にはすべて詳細に定義されたデータがあります。イベントのタイプは以下のとおりです。

- アクセス イベント（構造 SmLog\_Access\_t に関連する）。
- EMS イベント（構造 SmLog\_EMS\_t に関連する）。
- オブジェクト イベント（構造 SmLog\_Obj\_t に関連する）。
- システム イベント（構造 SmLog\_Obj\_t に関連する）。

## アクセス イベント タイプ

アクセス イベントはユーザ関連アクティビティの結果として発生します。これらのイベントは、認証、許可、管理、およびアフィリエイトアクティビティのコンテキストでコールされます。

アクセス イベントには4つのカテゴリがあります。これらのイベントカテゴリのそれぞれが自身のイベントのセットで応答します。以下の表は、アクセス イベント カテゴリおよびそれらの関連するレスポンス イベント のリストを示しています。

イベント カテゴリ	SiteMinder アクティビティ	SiteMinder イベント
認証	<ul style="list-style-type: none"> <li>■ ユーザ認証が受け入れられました</li> <li>■ ユーザ認証が拒否されました</li> <li>■ ユーザ認証が試行されました</li> <li>■ ユーザ認証がチャレンジされました</li> <li>■ ユーザセッションが検証されました</li> </ul>	<p>SmLogAccessEvent_AuthAccept SmLogAccessEvent_AuthReject</p> <p>SmLogAccessEvent_AuthAttempt SmLogAccessEvent_AuthChallenge</p> <p>SmLogAccessEvent_ValidateAccept SmLogAccessEvent_ValidateReject</p> <p>SmLogAccessEvent_AuthLogout</p>
許可	<ul style="list-style-type: none"> <li>■ ユーザ許可は受け入れられました</li> <li>■ ユーザ許可が拒否されました</li> </ul>	<p>SmLogAccessEvent_AzAccept SmLogAccessEvent_AzReject</p> <p>SmLogAccessEvent_AzUnresolved</p>
管理	<ul style="list-style-type: none"> <li>■ 管理者ログイン</li> <li>■ 管理者が拒否されました</li> <li>■ 管理者ログアウト</li> </ul>	<p>SmLogAccessEvent_AdminLogin</p> <p>SmLogAccessEvent_AdminLogout SmLogAccessEvent_AdminReject</p>
アフィリエイト	—	SmLogAccessEvent_Visit

## フィルタ アクセス イベント

SiteMinder v5.x 以降では、[ポリシー サーバ管理] コンソール上の [監査] タブを使用して、監査およびログ記録を実行したいアクセス イベントの種類をフィルタできます。たとえば、4 つのイベントカテゴリのそれぞれについて、[すべてのイベントのログ取得] または [イベントのログ取得なし] を選択できます。

加えて、認証、監査、および認証カテゴリに対して、[拒否イベントのみログ取得] を選択できます。たとえば、このオプションが認証カテゴリに選択された場合、SmLogAccessEvent\_AuthReject イベントはログ記録されませんが、SmLogAccessEvent\_AuthAccept のイベントはログ記録されません。また、[拒否イベントのみログ取得] が選択されている場合、以下の動作に注意してください。

- SmLogAccessEvent\_AuthAttempt イベントはログ記録されません。

受け入れられる認証が生成されないログイン試行は失敗とみなされません。ただし、認証が実際に拒否されなかったため、[拒否イベントのみログ取得] が選択されている場合はイベントはログされません。

侵入検知のために SmLogAccessEvent\_AuthAttempt イベントを使用することもできます。

- SmLogAccessEvent\_AuthChallenge イベントがログ記録されます。

チャレンジは失敗とはみなされません。それは、単に追加の認証情報の必要を示します。ただし、チャレンジに拒否された認証が含まれるため、[拒否イベントのみログ取得] が選択されている場合、イベントがログ記録されます。

## EMS イベント タイプ

EMS イベントは、ディレクトリ オブジェクト上で実行されたアクションから生成されます。

SiteMinder は以下の場合に EMS のイベントを呼び出します。

- ディレクトリ オブジェクトが作成、更新、または削除された場合
- メンバシップなどの関係がオブジェクト間で形成された場合

イベントを呼び出した後、SiteMinder はオブジェクトへのセッションアクティビティをログ記録します。分散代行管理サービス (DMS) などの EMS ベースのアプリケーションが EMS にログインする場合、新しいセッションが作成されます。EMS はログインセッションを検証し、適切なイベントをレポートします。

これらのディレクトリ オブジェクトは EMS イベントと関連付けられます。

- ユーザ
- ロール
- 組織
- 一般的なまたはユーザ定義のディレクトリ オブジェクト

上記の各オブジェクトは、以下のオブジェクト イベントと関連付けられます。

- 作成
- 削除
- 変更

EMS のイベントはカテゴリに基づいて分類されます。

- **管理上のイベント**は、ディレクトリ内のオブジェクトを変更するのに十分な権限を持つユーザによって生成されます。
- **セッションイベント**は、セッションが初期化されるか終了するときに生成されます。
- **エンドユーザ イベント**は、自己登録を実行するまたは自分のユーザ プロファイルを変更するユーザによって生成されます。

- ワークフロープリプロセス イベントは、ワークフロープリプロセスが完了したときに生成されます。
- ワークフローポストプロセス イベントは、ワークフローポストプロセスが完了したときに生成されます。

イベント カテゴリ	EMS イベント
SmLogEmsCat_DirectoryAdmin	SmLogEmsEvent_CreateUser SmLogEmsEvent_DeleteUser SmLogEmsEvent_ModifyUser SmLogEmsEvent_AssignUserRole SmLogEmsEvent_RemoveUserRole SmLogEmsEvent_EnableUser SmLogEmsEvent_DisableUser SmLogEmsEvent_CreateOrg SmLogEmsEvent_DeleteOrg SmLogEmsEvent_ModifyOrg SmLogEmsEvent_CreateRole SmLogEmsEvent_DeleteRole SmLogEmsEvent_ModifyRole SmLogEmsEvent_PasswordModify SmLogEmsEvent_CreateObject SmLogEmsEvent_DeleteObject SmLogEmsEvent_ModifyObject
SmLogEmsCat_DirectoryUser	SmLogEmsEvent_CreateUser SmLogEmsEvent_ModifyUser SmLogEmsEvent_PasswordModify
SmLogEmsCat_DirectorySession	SmLogEmsEvent_Login SmLogEmsEvent_Logout SmLogEmsEvent_LoginFail SmLogEmsEvent_SessionTimeout SmLogEmsEvent_AuthFail

## ログ記録ワークフロー イベント

プリプロセスまたはポストプロセス イベントが処理される場合、イベントがログ記録されます。この場合、ログ内のカテゴリは `SmLogEmsCat_EventPreprocess` または `SmLogEmsCat_EventPostprocess` のいずれかです。イベント ID は元のイベント ID（たとえば `SmLogEmsEvent_CreateUser`）です。

さらに、`SmLog_EMS_t` の以下のフィールドは、プリプロセスおよびポストプロセス イベントに適用されます。

フィールド	説明
<code>nVersion</code>	SiteMinder サーバのバージョン番号。
<code>nCurrentTime</code>	イベントが発生した時間。
<code>szUserName</code>	元のイベントを生成した管理者の名前。
<code>szSessionId</code>	EMS セッション ID。
<code>szDirName</code>	ターゲット オブジェクトが検出される SiteMinder ディレクトリの名前。
<code>szObjPath</code>	このイベントによってターゲットにされたオブジェクトの完全識別名。
<code>szFieldDesc</code>	イベントを実行したワークフロー ライブラリの名前。
<code>szStatusMsg</code>	<p>これにはワークフロー完了コードおよびオプションのエラーメッセージが含まれます。以下のワークフロー完了コードが <code>SmApi.h</code> で定義されています。</p> <ul style="list-style-type: none"> <li>■ <code>Sm_DmsWorkflowApi_Success</code></li> <li>■ <code>Sm_DmsWorkflowApi_Ignore</code></li> <li>■ <code>Sm_DmsWorkflowApi_Failure</code></li> <li>■ <code>Sm_DmsWorkflowApi_Reject</code></li> <li>■ <code>Sm_DmsWorkflowApi_NoUserContext</code></li> <li>■ <code>Sm_DmsWorkflowApi_SkipSuccess</code></li> </ul>

詳細情報:

[SmLog\\_EMS\\_t \(P. 713\)](#)

## オブジェクト イベント タイプ

SiteMinder 環境には、ドメイン、ポリシー、レルムおよびユーザディレクトリなどオブジェクトと呼ばれるエレメントが含まれます。これらの永続的なオブジェクトは集散的にオブジェクトストアを形成します。

オブジェクトが作成、更新、または削除されると、SiteMinder はオブジェクト イベントを呼び出します。オブジェクト イベントは次のとおりです。

- オブジェクト作成
- オブジェクト更新
- オブジェクト削除

以下の SiteMinder オブジェクトがオブジェクト イベントと関連付けられます。

エージェント	エージェント グループ
エージェント タイプ	エージェント タイプ属性
エージェント キー	キー管理
ドメイン	管理者
ポリシー	ポリシー リンク
パスワード ポリシー	登録
ユーザ ポリシー	ユーザディレクトリ
レルム	管理コマンド
レスポンス	レスポンス グループ
レスポンス属性	証明書マッピング
ルール	ルール グループ
認証	認証と許可のマッピング

認証方式	ODBC クエリ
ルート	ルート設定

---

オブジェクトイベントを呼び出した後、SiteMinder は、オブジェクトへのセッションアクティビティをログ記録します。アプリケーションがオブジェクトストアにログインするとき、新しいセッションが作成されます。SiteMinder はログインセッションを検証し、適切なイベントをレポートします。

### 認証イベント

認証は以下のイベントをレポートします。

- オブジェクトの作成、変更、または更新のためのアプリケーションまたはユーザによるログイン
- アプリケーションまたはユーザによるログアウト
- ログイン拒否

### 管理コマンド イベント

管理コマンドは永続的ではありません。これらは、キャッシュのクリアやキーの変更などの管理関数に関する情報をログ記録します。管理コマンドは以下のイベントと関連付けられます。

- すべてのキャッシュのクリア
- すべてのユーザキャッシュのクリア
- キャッシュからの単一ユーザのクリア
- リソースのクリア
- ダイナミック キーの変更
- 永続的なキーの変更

## システム イベント タイプ

システム イベントは、システムおよびサーバに関連するアクティビティを示します。以下の表は、システム イベントのリストを示しています。

アクティビティタイプ	SiteMinder イベント
サーバアクティビティ	SmLogSystemEvent_ServerInit SmLogSystemEvent_ServerInitFail SmLogSystemEvent_ServerUp SmLogSystemEvent_ServerDown SmLogSystemEvent_LogFileOpenFail SmLogSystemEvent_ServerHeartBeat
システム アクティビティ	SmLogSystemEvent_AgentInfo SmLogSystemEvent_AgentConnectionStart SmLogSystemEvent_AgentConnectionFail SmLogSystemEvent_AgentConnectionEnd SmLogSystemEvent_DbConnect SmLogSystemEvent_DbConnectFail SmLogSystemEvent_LdapConnect SmLogSystemEvent_LdapConnectFail SmLogSystemEvent_AmbiguousResourceMatch SmLogSystemEvent_AmbiguousRADIUSMatch SmSystemEvent_AgentHeartBeat

## SmLogAccessEvent\_AuthAccept

このイベントは、ユーザが認証されるときに呼び出されます。

以下の表は、関連する [SmLog\_Access\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szAgentName</i>	エージェントの名前です。

フィールド	説明
<i>szSessionId</i>	認証されるユーザのセッション識別子が含まれます。
<i>szClientIp</i>	エージェントの IP アドレスが含まれます。
<i>szUserName</i>	ユーザの名前。
<i>SzDomainOid</i>	リソースがアクセスされたドメインのオブジェクト識別子が含まれています。
<i>szRealmName</i>	アクセスされているレルムの名前が含まれます。
<i>szRealmOid</i>	レルムのオブジェクト識別子。
<i>szAuthDirName</i>	ユーザが認証されたディレクトリの名前が含まれています。
<i>szAuthDirServer</i>	認証ディレクトリ データ ソースが含まれます。
<i>szAuthDirNamespace</i>	認証ディレクトリのネームスペースが含まれます。
<i>szResource</i>	ユーザがアクセスしているリソースの名前が含まれます。
<i>szAction</i>	リソースと関連付けられたアクションが含まれます。
<i>szTransactionId</i>	<p>新しく作成されたセッションに関する情報が以下の形式で含まれます。</p> <p><b>idletime=N;maxtime=N;authlevel=N</b></p> <p>この形式の例において、</p> <ul style="list-style-type: none"> <li>■ <b>idletime</b> は、新規作成されたセッションの秒単位のアイドルタイムアウトです</li> <li>■ <b>maxtime</b> は、新規作成されたセッションの秒単位の最大タイムアウトです</li> <li>■ <b>authlevel</b> は新規作成されたセッションのセキュリティレベルです</li> </ul>

## SmLogAccessEvent\_AuthReject

このイベントは、ユーザが認証されないときに呼び出されます。

以下の表は、関連する [SmLog\_Access\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szAgentName</i>	エージェントの名前です。
<i>szClientIp</i>	エージェントの IP アドレスが含まれます。
<i>szUserName</i>	ユーザの名前。
<i>SzDomainOid</i>	リソースがアクセスされたドメインのオブジェクト識別子が含まれています。
<i>szRealmName</i>	アクセスされているレルムの名前が含まれます。
<i>szRealmOid</i>	レルムのオブジェクト識別子。
<i>szAuthDirName</i>	ユーザが認証されたディレクトリの名前が含まれています。
<i>szAuthDirServer</i>	認証ディレクトリ データ ソースが含まれます。
<i>szAuthDirNamespace</i>	認証ディレクトリのネームスペースが含まれます。
<i>szResource</i>	ユーザがアクセスしているリソースの名前が含まれます。
<i>szAction</i>	リソースと関連付けられたアクションが含まれます。

## SmLogAccessEvent\_AuthAttempt

このイベントは、認証の試行が失敗したときに呼び出されます。

以下の表は、関連する [SmLog\_Access\_t] フィールドのリストを示しています

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szAgentName</i>	エージェントの名前です。
<i>szClientIp</i>	エージェントの IP アドレスが含まれます。
<i>szUserName</i>	ユーザの名前。
<i>SzDomainOid</i>	リソースがアクセスされたドメインのオブジェクト識別子が含まれています。
<i>szRealmName</i>	アクセスされているレルムの名前が含まれます。
<i>szRealmOid</i>	レルムのオブジェクト識別子。
<i>szResource</i>	ユーザがアクセスしているリソースの名前が含まれます。
<i>szAction</i>	リソースと関連付けられたアクションが含まれます。

## SmLogAccessEvent\_AuthChallenge

このイベントは、認証がチャレンジされる場合に呼び出されます。

以下の表は、関連する [SmLog\_Access\_t] フィールドのリストを示しています

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szAgentName</i>	エージェントの名前です。
<i>szClientIp</i>	エージェントの IP アドレスが含まれます。

フィールド	説明
<i>szUserName</i>	ユーザの名前。
<i>SzDomainOid</i>	リソースがアクセスされたドメインのオブジェクト識別子が含まれています。
<i>szRealmName</i>	アクセスされているレルムの名前が含まれます。
<i>szRealmOid</i>	レルムのオブジェクト識別子。
<i>szResource</i>	ユーザがアクセスしているリソースの名前が含まれます。
<i>szAction</i>	リソースと関連付けられたアクションが含まれます。

### SmLogAccessEvent\_AzAccept

このイベントは、ユーザにリソースにアクセスする権限が与えられたときに呼び出されます。

以下の表は、関連する [SmLog\_Access\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szAgentName</i>	エージェントの名前です。
<i>szSessionId</i>	許可が受け入れられたユーザのセッション識別子が含まれます。
<i>szClientIp</i>	エージェントの IP アドレスが含まれます。
<i>szUserName</i>	ユーザの名前。
<i>SzDomainOid</i>	リソースがアクセスされたドメインのオブジェクト識別子が含まれています。
<i>szRealmName</i>	アクセスされているレルムの名前が含まれます。
<i>szRealmOid</i>	レルムのオブジェクト識別子。
<i>szResource</i>	ユーザがアクセスしているリソースの名前が含まれます。

フィールド	説明
<i>szAction</i>	リソースと関連付けられたアクションが含まれます。
<i>szTransactionId</i>	ポリシー サーバから返された情報を追跡するためにエージェントによって設定されるトランザクション識別子が含まれます。ポリシー サーバはこの ID をログ記録します。

## SmLogAccessEvent\_AzReject

このイベントは、ユーザにリソースにアクセスする権限が与えられていないときに呼び出されます。

以下の表は、関連する [SmLog\_Access\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szAgentName</i>	エージェントの名前です。
<i>szSessionId</i>	許可が拒否されたユーザのセッション識別子が含まれます。
<i>szClientIp</i>	エージェントの IP アドレスが含まれます。
<i>szUserName</i>	ユーザの名前。
<i>SzDomainOid</i>	リソースがアクセスされたドメインのオブジェクト識別子が含まれています。
<i>szRealmName</i>	アクセスされているレルムの名前が含まれます。
<i>szRealmOid</i>	レルムのオブジェクト識別子。
<i>szResource</i>	ユーザがアクセスしているリソースの名前が含まれます。
<i>szAction</i>	リソースと関連付けられたアクションが含まれます。

## SmLogAccessEvent\_AdminLogin

このイベントは、管理者ログインが成功するときに呼び出されます。

以下の表は、関連する [SmLog\_Access\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szSessionId</i>	ログインが成功した管理者のセッション識別子が含まれます。
<i>szClientIp</i>	エージェントの IP アドレスが含まれます。
<i>szUserName</i>	管理者の名前。

## SmLogAccessEvent\_AdminLogout

このイベントは、管理者がログアウトするときに呼び出されます。

以下の表は、関連する [SmLog\_Access\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szSessionId</i>	ログアウトした管理者のセッション識別子が含まれます。
<i>szClientIp</i>	エージェントの IP アドレスが含まれます。
<i>szUserName</i>	管理者の名前。

## SmLogAccessEvent\_AdminReject

このイベントは、管理者ログインが拒否されるときに呼び出されます。

以下の表は、関連する [SmLog\_Access\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>nReason</i>	理由識別子が含まれます。
<i>szClientIp</i>	エージェントの IP アドレスが含まれます。
<i>szUserName</i>	管理者の名前。
<i>szStatusMsg</i>	ログイン拒否の理由が含まれます。

## SmLogAccessEvent\_AuthLogout

このイベントは、認証サーバがセッションをログアウトするときに呼び出されます。

以下の表は、関連する [SmLog\_Access\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>nReason</i>	タイプ Sm_Api_Reason_t の理由識別子。 .

## SmLogAccessEvent\_ValidateAccept

このイベントは、セッション検証が成功するときに呼び出されます。

以下の表は、関連する [SmLog\_Access\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。

## SmLogAccessEvent\_ValidateReject

このイベントは、セッション検証が拒否されるときに呼び出されます。

以下の表は、関連する [SmLog\_Access\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。

## SmLogEmsEvent\_CreateUser、SmLogEmsEvent\_DeleteUser、 SmLogEmsEvent\_ModifyUser

ユーザ オブジェクトの作成、削除、または変更の結果として EMS イベントが発生します。ユーザ オブジェクトは、クラス `inetorgperson` の LDAP エンティティまたは `inetorgperson` を拡張するクラスのエンティティです。

以下の表は、関連する [SmLog\_Ems\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	これが管理上のイベントである場合、ユーザ オブジェクトを作成、変更、または削除している管理者の名前。これがエンドユーザ イベントの場合、ユーザ名は <b>Registration</b> です。
<i>szSessionId</i>	EMS セッション ID。
<i>szDirName</i>	ユーザが置かれる SiteMinder ディレクトリの名前。
<i>szObjName</i>	このイベントが対象とするユーザ オブジェクトの名前 (ユーザ ID)。
<i>szObjPath</i>	このイベントが対象とするユーザ オブジェクトの完全識別名。
<i>szObjClass</i>	ユーザ オブジェクトのクラス名。
<i>szOrgName</i>	ユーザ オブジェクトの組織の名前。
<i>szFieldDesc</i>	イベントの説明。
<i>szStatusMsg</i>	ステータス メッセージ。イベントが正常に処理されなかった場合にのみ使用されます。

## SmLogEmsEvent\_PasswordModify

ユーザ パスワード変更の結果として EMS イベントが発生します。

以下の表は、関連する [SmLog\_Ems\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	これが管理上のイベントである場合、パスワードを変更している管理者の名前。これがエンドユーザ イベントの場合、ユーザ名は <b>Registration</b> です。
<i>szSessionId</i>	EMS セッション ID。
<i>szDirName</i>	ユーザが置かれる SiteMinder ディレクトリの名前。
<i>szObjName</i>	このイベントが対象とするユーザ オブジェクトの名前 (ユーザ ID)。
<i>szObjPath</i>	このイベントが対象とするユーザ オブジェクトの完全識別名。
<i>szObjClass</i>	ユーザ オブジェクトのクラス名。
<i>szOrgName</i>	ユーザ オブジェクトの組織の名前。
<i>szFieldDesc</i>	イベントの説明。
<i>szStatusMsg</i>	ステータス メッセージ。イベントが正常に処理されなかった場合にのみ使用されます。

## SmLogEmsEvent\_AssignUserRole、SmLogEmsEvent\_RemoveUserRole

ユーザのロールへの割り当て、またはロールからのユーザの削除の結果として EMS イベントが発生します。ロールはクラス `groupofuniquenames` の LDAP エンティティ、または `groupofuniquenames` を拡張するエンティティです。

以下の表は、関連する [SmLog\_Ems\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	ロールを割り当てているか削除している管理者の名前。
<i>szSessionId</i>	EMS セッション ID。
<i>szDirName</i>	ロールが置かれる SiteMinder ディレクトリの名前。
<i>szObjName</i>	このイベントが対象とするユーザ オブジェクトの名前 (ユーザ ID)。
<i>szObjPath</i>	このイベントが対象とするユーザ オブジェクトの完全識別名。
<i>szObjClass</i>	ユーザ オブジェクトのクラス名。
<i>szOrgName</i>	ユーザの組織の名前。
<i>szRoleName</i>	割り当てられているか削除されるロールの名前。
<i>szFieldDesc</i>	イベントの説明。
<i>szStatusMsg</i>	ステータス メッセージ。イベントが正常に処理されなかった場合にのみ使用されます。

## SmLogEmsEvent\_EnableUser、SmLogEmsEvent\_DisableUser

ユーザのアクセス権限が有効または無効になるとき、これらのイベントが呼び出されます。SiteMinder 保護の下にあるリソースを取得するユーザの能力に関するアクセス権限。

以下の表は、関連する [SmLog\_Ems\_t] フィールドのリストを示しています。

フィールド	説明
nVersion	SiteMinder サーバのバージョン番号。
nCurrentTime	イベントが発生した時間。
szUserName	ユーザのアクセス権限を有効または無効にしている管理者の名前。
szSessionId	EMS セッション ID。
szDirName	ユーザが置かれる SiteMinder ディレクトリの名前。
szObjName	このイベントが対象とするユーザ オブジェクトの名前 (ユーザ ID)。
szObjPath	このイベントが対象とするユーザ オブジェクトの完全識別名。
szObjClass	ユーザ オブジェクトのクラス名。
szOrgName	ユーザの組織の名前。
szFieldDesc	イベントの説明。
szStatusMsg	ステータス メッセージ。ユーザのアクセス権限が有効な場合は 0、ユーザのアクセス権限が無効な場合は 0 以外の値になります。

## SmLogEmsEvent\_CreateOrg、SmLogEmsEvent\_DeleteOrg、SmLogEmsEvent\_ModifyOrg

これらのイベントは、組織オブジェクトが作成、削除、または変更される場合に呼び出されます。組織オブジェクトは、クラス `organizationalunit` の LDAP エンティティまたは `organizationalunit` を拡張するクラスのエンティティです。

以下の表は、関連する [SmLog\_Ems\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	組織を作成、変更、または削除している管理者の名前。
<i>szSessionId</i>	EMS セッション ID。
<i>szDirName</i>	組織が置かれる SiteMinder ディレクトリの名前。
<i>szObjName</i>	このイベントが対象とする組織オブジェクトの名前（ユーザ ID）。
<i>szObjPath</i>	このイベントが対象とする組織オブジェクトの完全識別名。
<i>szObjClass</i>	組織オブジェクトのクラス名。
<i>szFieldDesc</i>	イベントの説明。
<i>szStatusMsg</i>	ステータス メッセージ。イベントが正常に処理されなかった場合にのみ使用されます。

## SmLogEmsEvent\_CreateRole、SmLogEmsEvent\_DeleteRole、SmLogEmsEvent\_ModifyRole

これらのイベントは、ロールオブジェクトが作成、削除、または変更される場合に呼び出されます。ロールオブジェクトは、クラス `groupofuniquenames` の LDAP エンティティまたは `groupofuniquenames` を拡張するクラスのエンティティです。

以下の表は、関連する [SmLog\_Ems\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	グループを作成、変更、または削除している管理者の名前。
<i>szSessionId</i>	EMS セッション ID。
<i>szDirName</i>	グループが置かれる SiteMinder ディレクトリの名前。
<i>szObjName</i>	このイベントが対象とするグループの名前。
<i>szObjPath</i>	このイベントが対象とするグループ オブジェクトの完全識別名。
<i>szObjClass</i>	グループ オブジェクトのクラス名。
<i>szOrgName</i>	ユーザ グループの組織の名前。
<i>szFieldDesc</i>	イベントの説明。
<i>szStatusMsg</i>	ステータス メッセージ。イベントが正常に処理されなかった場合にのみ使用されます。

## SmLogEmsEvent\_CreateObject、SmLogEmsEvent\_DeleteObject、SmLogEmsEvent\_ModifyObject

これらのイベントは、**top** オブジェクトクラスを持つ LDAP エントリが作成、削除、または変更される場合に呼び出されます。**top** オブジェクトクラスは、LDAP ディレクトリ内の他のすべてのオブジェクトクラスのスーパークラスです。これらのイベントは、属性オブジェクトクラスを使用するあらゆる LDAP エントリに適用されます。

以下の表は、関連する [SmLog\_Ems\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	オブジェクトを作成、変更、または削除している管理者の名前。
<i>szSessionId</i>	EMS セッション ID。
<i>szDirName</i>	オブジェクトが置かれる SiteMinder ディレクトリの名前。
<i>szObjName</i>	このイベントが対象とするオブジェクトの名前。
<i>szObjPath</i>	このイベントによってターゲットにされたオブジェクトの完全識別名。
<i>szObjClass</i>	オブジェクトのクラス名。
<i>szFieldDesc</i>	イベントの説明。
<i>szStatusMsg</i>	ステータス メッセージ。イベントが正常に処理されなかった場合にのみ使用されます。

## SmLogEmsEvent\_Login, SmLogEmsEvent\_Logout

これらのイベントは、管理者がログインまたはログアウトする場合に呼び出されます。

以下の表は、関連する [SmLog\_Ems\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	管理者の名前。
<i>szSessionId</i>	EMS セッション ID。
<i>szDirName</i>	管理者が置かれる SiteMinder ディレクトリの名前。
<i>szObjName</i>	管理者の名前 (UID) 。
<i>szObjPath</i>	管理者の完全な識別名。
<i>szOrgName</i>	ユーザ管理者の組織の名前。
<i>szFieldDesc</i>	イベントの説明。
<i>szStatusMsg</i>	ステータス メッセージ。 イベントが正常に処理されなかった場合にのみ使用されます。

## SmLogEmsEvent\_AuthFail

このイベントは、管理認証が失敗するときに呼び出されます。

以下の表は、関連する [SmLog\_Ems\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	管理者の名前。

フィールド	説明
<i>szSessionId</i>	EMS セッション ID。これは存在しない可能性があります。
<i>szFieldDesc</i>	イベントの説明。
<i>szStatusMsg</i>	ステータス メッセージ。

### SmLogEmsEvent\_SessionTimeout

このイベントは、EMS サーバセッションがタイムアウトになるときに呼び出されます。

以下の表は、関連する [SmLog\_Ems\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	管理者の名前。
<i>szSessionId</i>	EMS セッション ID。
<i>szOrgName</i>	管理者の組織の名前。
<i>szFieldDesc</i>	イベントの説明。
<i>szStatusMsg</i>	ステータス メッセージ。

## SmLogObjEvent\_Create

このイベントは、システムまたはドメイン オブジェクトが作成される時に呼び出されます。

以下の表は、関連する [SmLog\_Obj\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	オブジェクトを生成している管理者の名前。
<i>szSessionId</i>	オブジェクトを生成している管理者のセッション ID。
<i>szDomainOid</i>	作成されているオブジェクトがドメイン オブジェクトである場合、この値はドメイン オブジェクト識別子に設定されます。
<i>szObjName</i>	作成されているオブジェクトの名前。
<i>szObjOid</i>	作成されているオブジェクトのオブジェクト識別子。

## SmLogObjEvent\_Update

このイベントは、システムまたはドメイン オブジェクトが更新される時に作成されます。

以下の表は、関連する [SmLog\_Obj\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	オブジェクトを更新している管理者の名前。
<i>szSessionId</i>	オブジェクトを更新している管理者のセッション ID。

フィールド	説明
<i>szDomainOid</i>	作成されているオブジェクトがドメイン オブジェクトである場合、この値はオブジェクトが更新されるドメイン オブジェクト識別子に設定されます。
<i>szObjName</i>	更新されているオブジェクトの名前。
<i>szObjOid</i>	更新されているオブジェクトのオブジェクト識別子。
<i>szFieldDesc</i>	管理者の DN。

### SmLogObjEvent\_Delete

このイベントは、システムまたはドメイン オブジェクトが削除されるときに呼び出されます。

以下の表は、関連する [SmLog\_Obj\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	オブジェクトを削除している管理者の名前。
<i>szSessionId</i>	オブジェクトを削除している管理者のセッション ID。
<i>szDomainOid</i>	作成されているオブジェクトがドメイン オブジェクトである場合、この値はオブジェクトが削除されるドメイン オブジェクト識別子に設定されます。
<i>szObjName</i>	削除されているオブジェクトの名前。
<i>szObjOid</i>	削除されているオブジェクトのオブジェクト識別子。

## SmLogObjEvent\_Login

このイベントは、管理者またはアプリケーション（smobjimport など）がポリシーストアにログインするときに呼び出されます。

以下の表は、関連する [SmLog\_Obj\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	ポリシーストアにログインしている管理者またはアプリケーションの名前。
<i>szSessionId</i>	ポリシーストアにログインしている管理者またはアプリケーションのセッション ID。

## SmLogObjEvent\_FailedLoginAttemptsCount

このイベントは、ユーザログインが失敗し、適用されるパスワードポリシーがある場合に呼び出されます。

以下の表は、関連する [SmLog\_Obj\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	ログイン試行が失敗したユーザ。
<i>szSessionId</i>	ユーザのセッション ID。
<i>szObjName</i>	ユーザが見つかったディレクトリの名前。
<i>szFieldDesc</i>	ユーザの DN。
<i>szStatusMsg</i>	ログインが試行された回数。この数は、アカウントが無効になる試行回数より大きくはなりません。

## SmLogObjEvent\_Logout

このイベントは、管理者またはアプリケーション（smobjimport など）がポリシーストアからログアウトするときに呼び出されます。

以下の表は、関連する [SmLog\_Obj\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	ポリシーストアからログアウトしている管理者またはアプリケーションの名前。
<i>szSessionId</i>	ポリシーストアからログアウトしている管理者またはアプリケーションのセッション ID。

## SmLogObjEvent\_LoginReject

このイベントは、ポリシーストアへの管理者またはアプリケーションのログインが拒否される場合に呼び出されます。

以下の表は、関連する [SmLog\_Obj\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	拒否された管理者またはアプリケーションの名前。

## SmLogObjEvent\_FlushAll

このイベントは、ユーザセッション、リソース情報、およびユーザディレクトリ キャッシュを含むすべての SiteMinder キャッシュがクリアされる場合に呼び出されます。

以下の表は、関連する [SmLog\_Obj\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	キャッシュをクリアしている管理者の名前。
<i>szSessionId</i>	キャッシュをクリアしている管理者のセッション ID。

## SmLogObjEvent\_FlushUser

このイベントは、ユーザがユーザセッション キャッシュからクリアされる場合に呼び出されます。

以下の表は、関連する [SmLog\_Obj\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	ユーザをクリアしている管理者の名前。
<i>szSessionId</i>	ユーザをクリアしている管理者のセッション ID。
<i>szObjName</i>	ユーザが定義されるディレクトリの名前。
<i>szObjOid</i>	ユーザが定義されるディレクトリのオブジェクト識別子。
<i>szFieldDesc</i>	クリアされるユーザの DN。

## SmLogObjEvent\_FlushUsers

このイベントは、すべてのユーザユーザセッション キャッシュからクリアされる場合に呼び出されます。

以下の表は、関連する [SmLog\_Obj\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	ユーザをクリアしている管理者の名前。
<i>szSessionId</i>	ユーザをクリアしている管理者のセッション ID。

## SmLogObjEvent\_FlushRealms

このイベントは、すべてのレルムがすべてリソース情報キャッシュからクリアされる場合に呼び出されます。

以下の表は、関連する [SmLog\_Obj\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	レルムをクリアしている管理者の名前。
<i>szSessionId</i>	レルムをクリアしている管理者のセッション ID。

## SmLogObjEvent\_ChangeDynamicKeys

このイベントは、ダイナミック キーが変更されるときに呼び出されます。

以下の表は、関連する [SmLog\_Obj\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	キーを変更している管理者の名前。
<i>szSessionId</i>	キーを変更している管理者のセッション ID。

## SmLogObjEvent\_ChangePersistentKey

このイベントは、永続的なキーが変更されるときに呼び出されます。

以下の表は、関連する [SmLog\_Obj\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	キーを変更している管理者の名前。
<i>szSessionId</i>	キーを変更している管理者のセッション ID。

## SmLogObjEvent\_ChangeSessionKey

このイベントは、セッション キーが変更されるときに呼び出されます。

以下の表は、関連する [SmLog\_Obj\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。

## SmLogObjEvent\_ChangeUserPassword

このイベントは、管理者がユーザ パスワードを変更するとき常に呼び出されます。

以下の表は、関連する [SmLog\_Obj\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	パスワードを変更している管理者の名前。
<i>szSessionId</i>	パスワードを変更している管理者のセッション ID。
<i>szObjName</i>	ユーザが定義されるディレクトリの名前。
<i>szObjOid</i>	ユーザが定義されるディレクトリのオブジェクト識別子。
<i>szFieldDesc</i>	パスワードが変更されているユーザの DN。

## SmLogObjEvent\_CreateUserSuccess

このイベントは、新規ユーザが作成されるときに呼び出されます。

以下の表は、関連する [SmLog\_Obj\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	ユーザ アカウントを作成している管理者の名前。
<i>szSessionId</i>	ユーザ アカウントを生成している管理者のセッション ID。
<i>szObjName</i>	ユーザが定義されるディレクトリの名前。
<i>szFieldDesc</i>	アカウントを作成されているユーザの DN。

## SmLogObjEvent\_DeleteUserSuccess

このイベントは、ユーザ アカウントがディレクトリで削除されるときに呼び出されます。

以下の表は、関連する [SmLog\_Obj\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	ユーザ アカウントを削除している管理者の名前。
<i>szSessionId</i>	ユーザ アカウントを削除している管理者のセッション ID。
<i>szObjName</i>	ユーザが定義されるディレクトリの名前。

フィールド	説明
<i>szFieldDesc</i>	アカウントが削除されているユーザの DN。

## SmLogObjEvent\_ChangeDisabledUserState

このイベントは、以下のような条件で呼び出されます。

- Admin がユーザ アカウントを有効または無効にするとき。
- Administrative UI または SDK を通じてアカウントを有効にするとき。
- アカウントが非アクティブのためにユーザが無効になるとき。
- ログイン失敗の数が超過したためにユーザが無効になるとき。
- パスワード期限切れのためにユーザが無効になるとき。
- Administrative UI で、[ユーザは次回ログイン時にパスワード変更が必要] を設定または設定解除するとき。
- パスワードの変更により、ユーザが新規パスワードを入力する必要があるため、ユーザ パスワードを変更するとき。

以下の表は、関連する [SmLog\_Obj\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	アカウントの状態を変更している管理者の名前。
<i>szSessionId</i>	アカウントの状態を変更している管理者のセッション ID。
<i>szObjName</i>	ユーザが定義されるディレクトリの名前。
<i>szFieldDesc</i>	アカウントが変更されているユーザの DN。
<i>szStatusMsg</i>	アカウントの状態を変更する理由。

## SmLogObjEvent\_ModifyUserSuccess

このイベントは、ユーザアカウントが変更されるときに呼び出されます。

以下の表は、関連する [SmLog\_Obj\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	ユーザアカウントを修正している管理者の名前。
<i>szSessionId</i>	ユーザアカウントを修正している管理者のセッション ID。
<i>szObjName</i>	ユーザが定義されるディレクトリの名前。
<i>szFieldDesc</i>	アカウントが修正されているユーザの DN。

## SmLogObjEvent\_CreateUserFail

このイベントは、ユーザアカウントを作成できないときに呼び出されます。

以下の表は、関連する [SmLog\_Obj\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	ユーザアカウントを作成することを試みている管理者の名前。
<i>szSessionId</i>	ユーザアカウントを作成することを試みている管理者のセッション ID。
<i>szObjName</i>	ユーザが定義されるディレクトリの名前。

フィールド	説明
<i>szFieldDesc</i>	アカウントが作成できなかったユーザの DN。

### SmLogObjEvent\_DeleteUserFail

このイベントは、ユーザアカウントを削除できないときに呼び出されます。

以下の表は、関連する [SmLog\_Obj\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	ユーザアカウントを削除することを試みている管理者の名前。
<i>szSessionId</i>	ユーザアカウントを削除することを試みている管理者のセッション ID。
<i>szObjName</i>	ユーザが定義されるディレクトリの名前。
<i>szFieldDesc</i>	アカウントが削除できなかったユーザの DN。

### SmLogObjEvent\_ModifyUserFail

このイベントは、ユーザアカウントを修正できないときに呼び出されます。

以下の表は、関連する [SmLog\_Obj\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。

フィールド	説明
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szUserName</i>	ユーザアカウントを修正することを試みている管理者の名前。
<i>szSessionId</i>	ユーザアカウントを修正することを試みている管理者のセッション ID。
<i>szObjName</i>	ユーザが定義されるディレクトリの名前。
<i>szFieldDesc</i>	アカウントが修正できなかったユーザの DN。

## SmLogSystemEvent\_ServerInit

サーバは初期化しています。

以下の表は、関連する [SmLog\_System\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szName</i>	サーバの名前。

フィールド	説明
<code>szMsg</code>	<p>以下の形式でのサーバに関する情報。 <code>Product=%,Platform=%,Version=%,Update=%, Label=%,Crypto=%,UTC=%,TZ=%</code></p> <p>この形式の例において、 <code>Product</code> はサーバの名前です。 <code>Platform</code> はサーバソフトウェアのサポートされているプラットフォームです。 <code>Version</code> は SiteMinder サーバソフトウェアのバージョン番号です。 <code>Update</code> はサーバソフトウェアの更新番号です。 <code>Label</code> はサーバソフトウェアのビルド番号です。 <code>Crypto</code> は暗号の強さです。 <code>UTC</code> は協定世界時の時間です。 <code>TZ</code> はタイムゾーンです。 <code>Unknown</code> 節は空白のままです。</p> <p>例： <code>Product=smservacct,Platform=Windows NT, Version=4.0,Update=None,Label=C144, Crypto=56,UTC=949621705,TZ=5</code></p>

## SmLogSystemEvent\_ServerInitFail

イベント呼び出しでシステム カテゴリに基づいてどのサーバ初期化が失敗したかを見つけるためにイベントハンドラによって呼び出されます。

このイベントには関連する [SmLog\_System\_t] フィールドがありません。

## SmLogSystemEvent\_ServerUp

イベント呼び出しでシステム カテゴリに基づいてどのサーバが実行されているかを見つけるためにイベントハンドラによって呼び出されます。

このイベントには関連する [SmLog\_System\_t] フィールドがありません。

## SmLogSystemEvent\_ServerDown

イベント呼び出しでシステム カテゴリに基づいてどのサーバがダウンしているかを見つけるためにイベント ハンドラによって呼び出されます。

このイベントには関連する [SmLog\_System\_t] フィールドがありません。

## SmLogSystemEvent\_LogFileOpenFail

このイベントは、監査ログに使用されたテキスト ファイルが開けなかった場合に呼び出されます。

以下の表は、関連する [SmLog\_System\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szMsg</i>	開けなかった監査ログ ファイルの名前。

## SmLogSystemEvent\_ServerHeartbeat

このイベントは、サーバに関する定期情報を提供します。それは 30 秒ごとに呼び出されます。

以下の表は、関連する [SmLog\_System\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szName</i>	サーバの名前。
<i>szIpAddress</i>	サーバが実行されているホストの名前。

フィールド	説明
<i>nIpPort</i>	サーバがリスンするポート。

## SmLogSystemEvent\_AgentInfo

エージェントに関する情報が含まれます。

以下の表は、関連する [SmLog\_System\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szName</i>	エージェントの名前です。
<i>szMsg</i>	<p>以下の形式でのエージェントに関する情報。            Product=%,Platform=%,Version=%,Update=%,            Label=%,Crypto=%,UTC=%,TZ=%</p> <p>この形式の例において、            Product はエージェント ソフトウェアの名前です。            Platform はエージェント ソフトウェアのサポートされているプラットフォーム形式です。            Version はエージェント ソフトウェアのバージョン番号です。            Update はエージェント ソフトウェアの更新番号です。            Label はエージェント ソフトウェアのビルド番号です。            Crypto は暗号の強さです。            UTC は協定世界時の時間です。            TZ はタイムゾーンです。            Unknown 節は空白のままです。            たとえば、Web エージェントは以下を発行します。            Product=WebAgent,Platform=NT/ISAPI,            Version=4.0,Update=SP1,Label=C134,            Crypto=56,UTC=949621705,TZ=5</p>

## SmLogSystemEvent\_AgentConnectionStart

このイベントは、エージェントがポリシー サーバに接続する場合に呼び出されます。

以下の表は、関連する [SmLog\_System\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szName</i>	エージェントの名前です。
<i>szIpAddress</i>	エージェント マシンの IP アドレス。
<i>nIpPort</i>	エージェントが接続される ポリシー サーバ マシン上のポート。

## SmLogSystemEvent\_AgentConnectionFail

このイベントは、エージェント接続が失敗すると呼び出されます。

以下の表は、関連する [SmLog\_System\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szName</i>	エージェントの名前です。
<i>szIpAddress</i>	エージェント マシンの IP アドレス。
<i>nIpPort</i>	エージェントが接続を試行するポリシー サーバ マシン上のポート。

## SmLogSystemEvent\_AgentConnectionEnd

このイベントは、エージェント接続が終了すると呼び出されます。

以下の表は、関連する [SmLog\_System\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szName</i>	エージェントの名前です。
<i>szIpAddress</i>	エージェントマシンの IP アドレス。
<i>nIpPort</i>	エージェントが接続されていた ポリシー サーバマシン上のポート。

## SmLogSystemEvent\_DbConnect

このイベントは、ポリシーサーバがデータベースに接続するとき呼び出されます。

以下の表は、関連する [SmLog\_System\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szName</i>	データソースの名前。

## SmLogSystemEvent\_DbConnectFail

このイベントは、ポリシー サーバのデータベースへの接続が失敗するときに呼び出されます。

以下の表は、関連する [SmLog\_System\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szName</i>	データ ソースの名前。
<i>szMsg</i>	エラーメッセージ。

## SmLogSystemEvent\_LdapConnect

このイベントは、ポリシー サーバが LDAP ディレクトリに接続するときに呼び出されます。

以下の表は、関連する [SmLog\_System\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szName</i>	LDAP ディレクトリの名前。
<i>szIpAddress</i>	LDAP ディレクトリの IP アドレス。
<i>nIpPort</i>	LDAP ディレクトリのポート。

## SmLogSystemEvent\_LdapConnectFail

このイベントは、ポリシー サーバの LDAP ディレクトリへの接続が失敗するときに呼び出されます。

以下の表は、関連する [SmLog\_System\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szName</i>	LDAP ディレクトリの名前。
<i>szIpAddress</i>	LDAP ディレクトリの IP アドレス。
<i>nIpPort</i>	LDAP ディレクトリのポート。
<i>szMsg</i>	エラーメッセージ。

## SmLogSystemEvent\_AmbiguousResourceMatch

このイベントは、不明瞭なリソース一致があるときに呼び出されます。

以下の表は、関連する [SmLog\_System\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szName</i>	リソースを要求するマシンの名前。
<i>szMsg</i>	要求されたリソース。

## SmLogSystemEvent\_AmbiguousRadiusMatch

このイベントは、不明瞭な RADIUS 一致があるときに呼び出されます。

以下の表は、関連する [SmLog\_System\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szName</i>	RADIUS デバイスの名前。

## SmLogSystemEvent\_AgentHeartbeat

このイベントは、エージェントが DoManagement 要求を発行する場合に常に呼び出されます。

以下の表は、関連する [SmLog\_System\_t] フィールドのリストを示しています。

フィールド	説明
<i>nVersion</i>	SiteMinder サーバのバージョン番号。
<i>nCurrentTime</i>	イベントが発生した時間。
<i>szName</i>	エージェントの名前です。
<i>szIpAddress</i>	エージェントの IP アドレス。
<i>nIpPort</i>	エージェントが接続されるポリシー サーバマシン上のポート。

# イベント関数宣言

以下の表は、イベント API で使用される関数のリストを示しています。共有ライブラリはこれらのエントリ ポイントをエクスポートする必要があります。

関数定義	説明
<a href="#">SmEventInit()</a> (P. 762)	イベント ハンドラが自身の初期化手順を実行できるように、ポリシー サーバによって呼び出されます。
<a href="#">SmEventRecord()</a> (P. 763)	イベントが示される場合にポリシー サーバによって呼び出されます。
<a href="#">SmEventRelease()</a> (P. 764)	イベント ハンドラが自身のランダウン手順を実行できるように、ポリシー サーバによって呼び出されます。

## SmEventInit()

この関数は、イベント プロバイダが自身の初期化手順を実行するように、ポリシー サーバによって呼び出されます。この呼び出しは、プロバイダが最初にロードされるときに一度実行されます。

### 構文

```
int SM_EXTERN SmEventInit();
```

### 戻り値

成功する場合は 1、失敗した場合は 0 を返します。

## SmEventRecord()

この関数は、イベントが示される場合にポリシーサーバによって呼び出されます。

## 構文

```
void SM_EXTERN SmEventRecord (
    const int    nCategoryType,
    const int    nCategory,
    const int    nEventId,
    void*       pData
);
```

パラメータ	I/O	説明
<i>nCategoryType</i>	I	ログ記録されているイベントのタイプ。以下のいずれかの有効な値です。 <ul style="list-style-type: none"> <li>■ SmLog_Access</li> <li>■ SmLog_EMS</li> <li>■ SmLog_Obj</li> <li>■ SmLog_System</li> </ul>
<i>nCategory</i>	I	イベントカテゴリ。このパラメータ表の後に有効なイベントカテゴリのリストが示されます。
<i>nEventID</i>	I	イベント。
<i>pData</i>	I	プロパティインスタンスへのポインタ。構造は以下のいずれかです。 <ul style="list-style-type: none"> <li>■ <a href="#">SmLog_Access_t</a> (P. 711)</li> <li>■ <a href="#">SmLog_EMS_t</a> (P. 713)</li> <li>■ <a href="#">SmLog_Obj_t</a> (P. 714)</li> <li>■ <a href="#">SmLog_System_t</a> (P. 715)</li> </ul>

*nCategory* パラメータに対する有効な値は以下のとおりです。

---

SmLogAccessCat_Admin	SmLogObjCat_PropertyCollection
SmLogAccessCat_Affiliate	SmLogObjCat_PropertySection
SmLogAccessCat_Auth	SmLogObjCat_Realm
SmLogAccessCat_Az	SmLogObjCat_Response
SmLogEmsCat_DirectoryAdmin	SmLogObjCat_ResponseAttr
SmLogEmsCat_DirectorySession	SmLogObjCat_ResponseGroup
SmLogEmsCat_DirectoryUser	SmLogObjCat_Root
SmLogEmsCat_EventPostprocess	SmLogObjCat_RootConfig
SmLogEmsCat_EventPreprocess	SmLogObjCat_Rule
SmLogObjCat_ActiveExpr	SmLogObjCat_RuleGroup
SmLogObjCat_Admin	SmLogObjCat_Scheme
SmLogObjCat_Agent	SmLogObjCat_SelfReg
SmLogObjCat_AgentGroup	SmLogObjCat_TaggedString
SmLogObjCat_AgentKey	SmLogObjCat_UserDirectory
SmLogObjCat_Auth	SmLogObjCat_UserPolicy
SmLogObjCat_AuthAzMap	SmLogObjCat_Variable
SmLogObjCat_CertMap	SmLogObjCat_VariableType
SmLogObjCat_Domain	SmLogObjCat_Vendor
SmLogObjCat_KeyManagement	SmLogObjCat_VendorAttr
SmLogObjCat_ManagementCommand	SmLogSystemCat_Acct
SmLogObjCat_ODBCQuery	SmLogSystemCat_Admin
SmLogObjCat_PasswordPolicy	SmLogSystemCat_Auth
SmLogObjCat_Policy	SmLogSystemCat_Az
SmLogObjCat_PolicyLink	SmLogSystemCat_Combined
SmLogObjCat_Property	SmLogSystemCat_System

---

## SmEventRelease()

この関数は、イベントハンドラが自身のランダウン手順を実行するように、ポリシーサーバによって呼び出されます。このコールは SiteMinder がシャットダウンしているときに 1 回行われます。

### 構文

```
void SM_EXTERN SmEventRelease();
```

## アクティブ ポリシーの例

ユーザがアクティブなポリシー式のパラメータ (*param*) フィールドで指定された組織単位に属する場合、この関数は **true** を返します。

```
<@ lib="SmAzAPI" func="activePolicy" param="Accounting" @>
*****
int SM_EXTERN activePolicy(
const Sm_Api_Context_t* lpApiContext,
// the structure that provides API context
const Sm_Api_UserContext_t* lpUserContext,
// the structure that provides user context
const Sm_Api_RequestContext_t* lpReqContext,
// the structure that provides request context
const char* lpszParam,
// the parameter string (null-terminated)
const int nBytesOutBuf,
// the maximum size of the output buffer
char* lpszOutBuf,
// the output buffer to hold the null-terminated attribute value
const int nBytesErrBuf,
// the maximum size of the error message buffer
char* lpszErrBuf)
// the output buffer to hold the null-terminated error message
{
/* User Context is required to use the functions like fGetProp, fSetProp.. */
if(!lpUserContext->bIsUserContext) {
    strncpy (lpszErrBuf, "No User Context ", nBytesErrBuf);
    lpszErrBuf[nBytesErrBuf-1] = '\0';
    return -1;
}
/* Buffer to store all the organizational units user belongs to. */
char lpszOrgUnit[30];
memset(lpszOrgUnit, 0, sizeof(lpszOrgUnit));
/*
// Check to see if an organizational unit has been
// entered in the parameter.
*/
if(lpszParam == NULL || strlen(lpszParam) == 0)
{
    strncpy (lpszErrBuf, "Organizational unit is not entered ",
            nBytesErrBuf);
    lpszErrBuf[nBytesErrBuf-1] = '\0';
    return -1;
}
/* Get all the organizational units to which the user belongs. */
```

```
int getResult = lpUserContext->fGetProp (
lpUserContext->lpParam,
"ou",          /* Attribute name */
sizeof (lpszOrgUnit),lpszOrgUnit);

if (getResult < 0)
{
strncpy (lpszErrBuf,
"Failed to get user password from user's profile attribute ",
nBytesErrBuf);
lpszErrBuf[nBytesErrBuf-1] = '\0';
return -1;
}
else
{
/* Check if the user belongs to the organization unit that is
requested. */
if(strstr(lpszOrgUnit, lpszParam) != NULL)
{
/*
// Yes the user belongs to the organization unit
// mentioned in the parameter field of active policy.
*/
strncpy(lpszOutBuf, "true", nBytesOutBuf);
lpszOutBuf[nBytesOutBuf-1] = '\0';
return strlen(lpszOutBuf);
}
else
{
strncpy (lpszErrBuf,
"The user does not belong to the requested organizational unit ",
nBytesErrBuf);
lpszErrBuf[nBytesErrBuf-1] = '\0';
return -1;
}
}
/* everything failed.... */
return 0;
}
```

## イベントハンドラに対するポリシー サーバの設定

SiteMinder ポリシー サーバにはイベントハンドラ ライブラリを追加できます。

**注:** SiteMinder バイナリ ファイル (XPS.dll、libXPS.so、libXPS.sl) への書き込みアクセス許可がユーザにない場合は、管理者が **Administrative UI** または **XPSSecurity** ツールを使用して、関連する **XPS** コマンドライン ツールを使用する権限を付与する必要があります。

### イベントハンドラライブラリを追加する方法

1. ポリシー サーバでコマンドラインを開き、以下のコマンドを入力します。

```
xpsconfig
```

ツールが起動し、このセッションのログファイルの名前が表示されます。また、選択項目のメニューが開きます。

2. 以下のコマンドを入力します。

```
xps
```

オプションのリストが表示されます。

3. 以下のコマンドを入力します。

```
5 (AuditSMHandlers)
```

イベントハンドラ ライブラリの設定が表示されます。

4. 「C」と入力し、追加するイベントハンドラライブラリのパスとファイル名を入力します。ライブラリの場所が複数ある場合はカンマで区切ります。

イベントハンドラライブラリの設定が表示されます。追加した値は「保留中の値」として設定の下に表示されます。

5. 以下の手順を実行します。
  - a. 2回「Q」と入力します。
  - b. 「L」と入力します。
  - c. 「Q」と入力して XPS セッションを終了します。  
変更が保存され、コマンドプロンプトが表示されます。

# 第 11 章: DMS ワークフロー API

---

このセクションには、以下のトピックが含まれています。

[DMS ワークフロー API ガイド \(P. 769\)](#)

[DMS ワークフロー API リファレンス \(P. 773\)](#)

## DMS ワークフロー API ガイド

DMS ワークフロー API を使用すると、DMS イベントに対する単純なプリプロセスおよびポストプロセス ワークフローを追加できます。これらのプロセスを実装するには、以下を実行する必要があります。

1. DMS ワークフロー API をサポートし、必要なカスタム機能を提供する共有ライブラリを開発します。共有ライブラリには、エクスポート可能な記号として定義されたワークフロー関数が含まれる必要があります。

`SmApi.h` はカスタム ワークフローを作成するのに必要なすべてのデータ構造体を定義します。`SmEventApi.h` はすべてのワークフロー イベントを定義します。

2. 以下のいずれかの場所に共有ライブラリをインストールします。
  - UNIX プラットフォームでは、`SiteMinder lib` ディレクトリ
  - Windows プラットフォームでは、`SiteMinder bin` ディレクトリ
3. Administrative UI でワークフロー ライブラリを定義します。すべての DMS 管理レールおよびエンドユーザが自身を登録するすべてのリソース レールは、登録方式に関連付けられています。登録方式プロパティ UI には、DMS ワークフローを使用するための共有ライブラリの名前を入力できる、テキストフィールドが含まれます。

**注:** DMS API (Java のみで利用可能) の機能は、DMS Workflow API (C/C++ のみで利用可能) とは異なっています。DMS API を使用すると、SiteMinder DMS 製品と同様の操作を実行するディレクトリ管理アプリケーションを開発できます。DMS Workflow API は、DMS と共に動作し、特定のプリプロセスおよびポストプロセス DMS イベントが発生した場合に起動します。これにより、これらのイベントの前または後に追加の機能を実行するアプリケーションを開発できます。

## DMS セッション

DMS アプリケーションを使用するユーザには、ユーザの管理権限に基づいて、アプリケーション コンテキストを維持するためのセッションが割り当てられます。

DMS ユーザセッションには以下の 2 種類があります。

- **管理セッション**は、DMS アプリケーション内で管理権限を持つユーザに対して作成されます。DMS セッションを開始するには、ユーザはアプリケーションを使用するために認証および許可される必要があります。管理上のセッションは、管理上のイベントとして分類されるイベントを生成します。このようなイベントはワークフロー ライブラリにユーザ コンテキストを提供します。
- **登録セッション**は、ユーザがディレクトリに自身を追加する登録レلمとして作成されます。登録セッションは、レلم内のリソースに対する自己登録の最初の要求のときに、そのレلمに対して作成されます。セッションは、SiteMinder がシャットダウンするまで DMS サーバ上で有効です。登録セッションは、登録イベントとして分類されるイベントを生成します。このようなイベントはワークフロー ライブラリにユーザ コンテキストを提供しません。登録イベントは、ユーザ エントリの作成およびユーザ エントリの自己変更に制限されています。

## DMS のワークフロー

自己登録を要求するまたは DMS 管理アプリケーションを入力することによりユーザが DMS セッションを開始すると、以下の SiteMinder オブジェクトが検出されます。

- 要求されたリソースのレلم。自己登録に対しては、これはユーザが手続きをしているページが含まれるレلمです。DMS 管理者に対しては、レلمは DMS 管理 サーブレットの場所です。レلمは、ユーザがアクセスを要求する場合に SiteMinder Web エージェントによって取得されます。
- レلمと関連付けられる登録方式。登録方式には、ワークフロー ライブラリの名前が存在する場合はそれが含まれます。

ワークフローはレلمと関連付けられるので、特定のリソースに対して必要な DMS イベント処理に応じて、複数のワークフロー ライブラリをセットアップすることが可能です。たとえば、別個のサイトからドキュメントを受け取るために自己登録が必要だが、各サイトに対するプリプロセスおよびポストプロセスユーザのルールが異なるとします。

たとえば、それぞれ別個のワークフロー ライブラリを持つ、一連の自己登録サイトをセットアップすることが可能です。

登録方式にワークフロー ライブラリ名が含まれる場合、ライブラリがロードされます。ローダは、必要な関数がすべてライブラリによってエクスポートされることを確認します。ライブラリをロードできない、または関数が見つからない場合、セッション初期化は失敗し、そのレلمの DMS 要求は処理されません。この問題を解決するには、以下を実行します。

1. DMS ワークフロー API 概要で指定されるように、ライブラリが正しい場所にインストールされたことを確認します。
2. 必要な関数がライブラリによってエクスポートされることを確認します。
3. 必要に応じて、Administrative UI を使用して、登録方式からライブラリ名を削除することによりワークフローを無効にします。

ワークフロー ライブラリがロードされる場合、セッション中にユーザディレクトリでのエンティティの追加、変更、および削除に関連するすべての DMS イベントに対して呼び出されます。

## ワークフロー イベント

ワークフロー イベントは「EMS イベント タイプ」でドキュメント化されています。これらのイベントはワークフローだけでなくログ記録にも使用されます。イベントは以下のカテゴリに分類されます。

- **SmLogEmsCat\_DirectoryUser**。自己登録や変更などエンドユーザーによって生成されたイベントに割り当てられます。
- **SmLogEmsCat\_DirectoryAdmin**。DMS 管理者によって生成されたイベントに割り当てられます。
- **SmLogEmsCat\_DirectorySession**。DMS セッション管理と関連付けられたイベントに割り当てられます。
- **SmLogEmsCat\_EventPreprocess**。ワークフロー プリプロセス手順によって生成されたイベントに割り当てられ、ワークフロー イベントの結果をログ記録するのに使用されます。
- **SmLogEmsCat\_EventPostprocess**。ワークフロー ポストプロセス手順によって生成されたイベントに割り当てられ、ワークフロー イベントの結果をログ記録するのに使用されます。

詳細情報:

[EMS イベント タイプ \(P. 717\)](#)

## プリプロセス イベント

プリプロセス イベントは、DMS 要求が処理される前に発生します。ワークフロー ライブラリは要求を評価し、それを受け入れるか拒否するかを決定する必要があります。受け入れられる場合は要求が実行されます。拒否される場合は、要求は実行されず、プリプロセス エラーが DMS アプリケーションに返されます。オプションで、プリプロセス関数は DMS でユーザーに表示するエラーメッセージを設定することができます。

ワークフロー ライブラリはカテゴリ **SmLogEmsCat\_DirectoryUser** および **SmLogEmsCat\_DirectoryAdmin** 内のすべてのイベントを前処理するために呼び出されます。

## ポストプロセス イベント

ポストプロセス イベントは、DMS 要求が正常に処理された後に発生します。ワークフロー ライブラリは、要求を評価し、ビジネス プロセスによって指示されるすべてのアクションを実行し、成功または失敗のステータスを返します。オプションで、ポストプロセス関数は DMS でユーザに表示するエラー メッセージを設定することができます。ポストプロセスの失敗はトランザクションのロールバックにつながらないことを念頭に置いてください。受け入れられない要求はプリプロセスで検出されます。

ワークフロー ライブラリは、SmLogEmsCat\_DirectorySession に加えて、SmLogEmsCat\_DirectoryUser および SmLogEmsCat\_DirectoryAdmin カテゴリ内のすべてのイベントを後処理するために呼び出されます。最後のカテゴリはセッション ステータスのみを提供します。

## DMS ワークフロー API リファレンス

カスタム ワークフロー ライブラリを作成するには以下を実行します。

1. 以下のように SmApi.h ファイルを含めます。  

```
#include "SmApi.h"
```
2. 以下のように SmEventApi.h ファイルを含めます。  

```
#include " SmEventApi.h"
```
3. 以下の関数が外部的に表示されることを確認してください。

関数	説明
<a href="#">SmDmsWorkflowInit()</a> (P. 778)	ワークフロー ライブラリを初期化します。ライブラリは、この時点で必要なリソースを初期化する必要があります。
<a href="#">SmDmsWorkflowPostprocess()</a> (P. 779)	ワークフロー ポストプロセス手順を実行します。
<a href="#">SmDmsWorkflowPreprocess()</a> (P. 780)	ワークフロー プリプロセス手順を実行します。
<a href="#">SmDmsWorkflowRelease()</a> (P. 783)	ライブラリは、この時点で初期化によって作成されたあらゆるリソースを解放する必要があります。

共有ライブラリ内のエントリ ポイントはそれぞれ指定された構文に従って定義される必要があります。

**注:** **Microsoft Visual Studio** を使用している場合は、関数アドレスをモジュール定義ファイル (.DEF ファイル) にエクスポートします。関数アドレスをエクスポートするには、.DEF ファイルを作成します。ファイルのエクスポートセクションで、前述の表内のすべての関数をリストします。.DEF ファイルを作成したら、**Microsoft Visual Studio** プロジェクトにそれを追加します。

DLL または共有ライブラリにコードをコンパイルします。このファイルの名前は、[SiteMinder 登録のプロパティ] ダイアログ ボックス内でライブラリとして指定されます。

この章の最後で提供される例のようなカスタム ワークフローを記述した場合、**Administrative UI** を使用して [登録のプロパティ] ダイアログ ボックスでワークフローを定義する必要があります。

## Sm\_Api\_DmsContext\_t

この構造は DMS 固有のコンテキストを提供します。

## 構文

```
typedef struct
{
    unsigned char    bIsUserContext;
    char*            lpszAdminUserName
    char*            lpszAdminUserPath;
    char*            lpszDirPath;
    void             lpReserved1;
    char*            lpszDirServer;
    char*            lpszDirNamespace;
    char*            lpszSessionId;
    void*            lpDirParam;
    void*            lpAdminParam;
    Sm_Api_GetUserProp fGetAdminProp;
    Sm_Api_GetDmsDnProp fGetDnProp;
} Sm_Api_DmsContext_t;
```

フィールド	説明
<i>bIsUserContext</i>	SiteMinder がユーザの識別情報を確立しており <i>lpszAdminUserName</i> に保存したことを示すフラグ。このフラグが設定される場合、 <i>fGetAdminProp</i> で参照された関数を、このユーザのプロパティを取得するために使用できます。
<i>lpszAdminUserName</i>	DMS 管理者ユーザ名。これは管理者イベントの完全識別名です。これ以外の場合は空の文字列となります。
<i>lpszAdminUserPath</i>	管理者イベントのみを対象とする、SiteMinder 表記法に基づく DMS 管理者ユーザパス。
<i>lpszDirPath</i>	登録方式によって指定された SiteMinder ユーザディレクトリの SiteMinder 表記法のディレクトリパス。
<i>lpReserved1</i>	内部使用向けです。
<i>lpszDirServer</i>	登録方式によって指定された SiteMinder ユーザディレクトリのディレクトリサーバ。
<i>lpszDirNamespace</i>	ディレクトリネームスペース。
<i>lpszSessionId</i>	DMS セッション ID。

フィールド	説明
<i>lpDirParam</i>	<i>fGetDnProp</i> に渡されるパラメータへのポインタ。
<i>lpAdminParam</i>	<i>fGetAdminProp</i> に渡されるパラメータへのポインタ。
<i>fGetAdminProp</i>	<p>管理者ユーザの属性を返す関数。この関数の呼び出し構文は次のとおりです。</p> <pre>int nMaxBufSize = 256; char lpszResult[256]; int nBuflen = lpDmsContext-&gt;fGetProp (     lpDmsContext-&gt;lpAdminParam,     "mail",     nMaxBufSize,     lpszResult);</pre> <p>関数が成功する場合、戻り値は出力バッファ内のバイト数です。関数が失敗する場合、戻り値は -1 です。</p>
<i>fGetDnProp</i>	<p>識別名が指定された場合に属性を返す関数。ディレクトリサーバから取得可能なかぎり、どのオブジェクトも検討されます。この関数の呼び出し構文は次のとおりです。</p> <pre>int nMaxBufSize = 256; char lpszResult[256]; char lpszDn[] =     "uid=jsmith,ou=People,o=dms.com"; int nBuflen = lpDmsContext-&gt;fGetProp (     lpDmsContext-&gt;lpDirParam,     lpszDn,     "mail",     nMaxBufSize,     lpszResult);</pre> <p>関数が成功する場合、戻り値は出力バッファ内のバイト数です。関数が失敗する場合、戻り値は -1 です。</p>

---

## 関数宣言

構造 `Sm_Api_DmsContext_t` で、関数 `fGetAdminProp` および `fGetDnProp` は、`SmApi.h` で以下のように宣言されます。

### fGetAdminProp

```
typedef int (SM_EXTERN *Sm_Api_GetUserProp)
(
    const void* lpParam,          /* The function parameter */
    const char* lpszPropName,     /* User property name (null-term) */
    const int nBytesValueBuf,     /* Max size of user property buffer */
    char* lpszValueBuf           /* Output buffer to hold the user property */
);
```

### fGetDnProp

```
typedef int (SM_EXTERN *Sm_Api_GetDmsDnProp)
(
    const void* lpParam,          /* The function parameter */
    const char* lpszDN,           /* The DN of the object */
    const char* lpszPropName,     /* Object property name (null-term) */
    const int nBytesValueBuf,     /* Max size of object property buffer */
    char* lpszValueBuf           /* Output buffer to hold the object property */
);
```

## Sm\_DmsWorkflow\_Attribute\_t

この構造は、属性値の文字列表現で要求またはレスポンス属性名をペアにします。これらの構造の配列は、ディレクトリ オブジェクトを作成または変更する DMS 要求を表すために使用されます。

### 構文

```
typedef struct
{
    const char* lpszAttrName;
    const char* lpszAttrValue;
} Sm_DmsWorkflow_Attribute_t;
```

フィールド	説明
<i>lpszAttrName</i>	要求またはレスポンス属性の名前。

フィールド	説明
<i>lpszAttrValue</i>	属性値を表す文字列。

## SmDmsWorkflowInit()

SiteMinder は、ワークフロー ライブラリが自身の初期化手順を実行できるようにこの関数を呼び出します。この呼び出しは、ライブラリが最初にロードされるときに 1 度実行されます。

### 構文

```
Sm_DmsWorkflowApi_Status_t SM_EXTERN SmDmsWorkflowInit();
```

### 戻り値

- `Sm_DmsWorkflowApi_Success`。関数は成功しました。
- `Sm_DmsWorkflowApi_Failure`。関数は成功せず、ワークフローはロードされません。ワークフローが正常にロードされない場合、ワークフローを指定した登録方式に依存する DMS イベントは以下が該当するまで失敗します。
  - ワークフロー ライブラリの正常なロードが可能になる
  - ワークフロー ライブラリが登録方式から削除される

## SmDmsWorkflowPostprocess()

SiteMinder は、ワークフロー ライブラリが DMS イベントを後処理できるようにこの関数を呼び出します。

## 構文

```
Sm_DmsWorkflowApi_Status_t SM_EXTERN SmDmsWorkflowPostprocess(
    const Sm_Api_Context_t*          lpApiContext,
    const Sm_Api_DmsContext_t*      lpDmsContext,
    const char *                     lpszWorkflowOid,
    const int                        nCategoryId,
    const int                        nEventId,
    const char *                     lpszObjectPath,
    const int                        nAttributes,
    const Sm_DmsWorkflow_Attribute_t* pAttributes,
    const int                        nBytesErrMsg,
    char *                            lpszErrMsg
);
```

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。
<i>lpDmsContext</i>	I	DMS コンテキスト構造へのポインタ。
<i>lpszWorkflowOid</i>	I	一意のワークフロー識別子。
<i>nCategoryId</i>	I	DMS イベント カテゴリ。
<i>nEventId</i>	I	DMS イベント識別子
<i>lpszObjectPath</i>	I	このイベントのターゲットであるオブジェクトの識別名 (存在する場合)。ターゲット オブジェクト DN はディレクトリ オブジェクトを作成、削除、および変更する要求に対して利用可能になります。
<i>nAttributes</i>	I	レスポンス属性の配列内の属性の数。
<i>pAttributes</i>	I	DMS レスポンス属性/値ペアの配列。
<i>nBytesErrMsg</i>	I	出力エラー バッファの最大サイズ。
<i>lpszErrMsg</i>	O	エラー テキストを受信する出力バッファ。DMS アプリケーションにエラー メッセージを返すためにこのバッファを使用します。

### 戻り値

- `Sm_DmsWorkflowApi_Success`。関数は成功しました。
- `Sm_DmsWorkflowApi_Failure`。関数は成功しませんでした。
- `Sm_DmsWorkflowApi_Ignore`。イベントはこの手順に関連性がありませんでした。

### 注釈

イベントはすでに処理されています。`Sm_DmsWorkflowApi_Success` または `Sm_DmsWorkflowApi_Ignore` の戻り値により、アプリケーションに成功のステータスが返されます。`Sm_DmsWorkflowApi_Failure` によってアプリケーションにエラーコードが返されます。エラーコードは、DMS ポストプロセス手順に失敗があったことを示しています。

文字配列 `lpzErrMsg` は、エラーメッセージをアプリケーションに送り返すためにポストプロセス関数によって使用される場合があります。文字配列の最大サイズは `nBytesErrMsg` で指定されます。

## SmDmsWorkflowPreprocess()

SiteMinder は、ワークフロー ライブラリが DMS イベントを前処理できるようにこの関数を呼び出します。

### 構文

```
Sm_DmsWorkflowApi_Status_t SM_EXTERN SmDmsWorkflowPreprocess(
    const Sm_Api_Context_t*          lpApiContext,
    const Sm_Api_DmsContext_t*       lpDmsContext,
    const char *                      lpzWorkflowOid,
    const int                         nCategoryId,
    const int                         nEventId,
    const char *                      lpzObjectPath,
    const int                         nAttributes,
    const Sm_DmsWorkflow_Attribute_t* pAttributes,
    const int                         nBytesErrMsg,
    char *                            lpzErrMsg
);
```

パラメータ	I/O	説明
<code>lpApiContext</code>	I	API コンテキスト構造へのポインタ。

パラメータ	I/O	説明
<i>lpDmsContext</i>	I	DMS コンテキスト構造へのポインタ。
<i>lpszWorkflowOid</i>	I	一意のワークフロー識別子。
<i>nCategoryId</i>	I	DMS イベント タイプ。
<i>nEventId</i>	I	DMS イベント識別子
<i>lpszObjectPath</i>	I	このイベントのターゲットであるオブジェクトの識別名 (存在する場合)。ターゲット オブジェクト DN はディレクトリ オブジェクトを作成、削除、および変更する要求に対して利用可能になります。
<i>nAttributes</i>	I	要求属性の配列内の属性の数。
<i>pAttributes</i>	I	DMS 要求属性/値ペアの配列。
<i>nBytesErrMsg</i>	I	出力エラー バッファの最大サイズ。
<i>lpszErrMsg</i>	O	エラー テキストを受信する出力バッファ。DMS アプリケーションにエラー メッセージを返すためにこのバッファを使用します。

### 戻り値

- `Sm_DmsWorkflowApi_Success`。関数は成功しました。
- `Sm_DmsWorkflowApi_Failure`。関数は成功しませんでした。
- `Sm_DmsWorkflowApi_NoUserContext`。管理者ユーザ イベントが生成され、ユーザ コンテキストを判断できませんでした。
- `Sm_DmsWorkflowApi_SkipSuccess`。イベントはプリプロセス関数によって処理され、さらなる処理は必要ありません。
- `Sm_DmsWorkflowApi_Ignore`。イベントはこの手順に関連性はありませんでした。

### 注釈

`Sm_DmsWorkflowApi_Success` または `Sm_DmsWorkflowApi_Ignore` が返されると処理は継続します。その他が返された場合は処理が停止し、エラーコードがアプリケーションに返されます。エラーコードは、DMS プリプロセス手順に失敗があったことを示しています。

`Sm_DmsWorkflowApi_SkipSuccess` により処理が停止しますが、成功ステータスがアプリケーションに返されます。必要な目標を達成することは、プリプロセス関数の実装に依存しています。関数が失敗すると `Sm_DmsWorkflowApi_SkipSuccess` ではなく失敗コードが返されます。

文字配列 `lpszErrMsg` は、エラーメッセージをアプリケーションに送り返すためにプリプロセス関数によって使用される場合があります。文字配列の最大サイズは `nBytesErrMsg` で指定されます。

## SmDmsWorkflowRelease()

SiteMinder は、ワークフロー ライブラリが自身のランダウン手順を実行できるようにこの関数を呼び出します。この呼び出しは、SiteMinder がシャットダウンしているときに 1 回実行されます。

### 構文

```
Sm_DmsWorkflowApi_Status_t SM_EXTERN SmDmsWorkflowRelease();
```

### 戻り値

- Sm\_DmsWorkflowApi\_Success。関数は成功しました。
- Sm\_DmsWorkflowApi\_Failure。関数は成功しませんでした。



# 第 12 章: ディレクトリ API ガイド

---

このセクションには、以下のトピックが含まれています。

[ディレクトリ API の目的 \(P. 785\)](#)

[ディレクトリ API を使用する前に \(P. 786\)](#)

[ディレクトリ API を使用する方法 \(P. 787\)](#)

[ディレクトリ API リファレンス \(P. 799\)](#)

[サンプルディレクトリ アプリケーションで使用される構造 \(P. 860\)](#)

## ディレクトリ API の目的

ディレクトリ API は、SiteMinder がサポートしないタイプのデータベースまたはディレクトリに保存されたデータにアクセスします。ディレクトリ API では以下が実行できます。

- SiteMinder ポリシーへのサポートされないディレクトリ エントリ (ユーザ) の追加
- サポートされないディレクトリ エントリの管理

SiteMinder は、ユーザディレクトリに対する以下のネームスペースをサポートします。

- LDAP
- ODBC
- Microsoft Windows NT
- カスタム

LDAP、ODBC および NT ネームスペースはディレクトリ API なしで使用できます。別のタイプのユーザディレクトリにアクセスするには、ディレクトリ API でディレクトリのインターフェースを作成します。

## ディレクトリ API を使用する前に

ディレクトリ API を使用する前に、SiteMinder Administrative UI に移動し、カスタム ネームスペースでユーザディレクトリ オブジェクトを作成および設定します。

ディレクトリ API を使用するには、以下が必要です。

- 以下のタスクのいずれかの経験がある、1人以上のアプリケーション開発者
  - C プログラミング言語でのコーディング
  - 共有ライブラリの構築
  - 必要なディレクトリ タイプのベンダーによって提供される、関連 API の使用
- 共有ライブラリを構築するためのサポートされた開発環境。

リンクが必要ないので、サポートされる開発環境の制限リストはありません。たとえば、GNU Compiler Collection (GCC) を使用できます。ディレクトリ API は以下の開発環境でテスト済みです。

  - UNIX プラットフォーム : Sun Visual WorkShop C++ 5.0
  - Windows プラットフォーム : Microsoft Visual C++6.0
- SiteMinder ポリシー サーバ

SiteMinder ポリシー サーバはディレクトリ API アプリケーションの構築には必要ありませんが、アプリケーションの使用には必要です。SiteMinder ポリシー サーバはディレクトリ API を呼び出します。

## ディレクトリ API を使用方法

前提条件を満たしたら、以下の手順に従います。

1. ディレクトリ API で提供されたサンプルを確認します。
2. ディレクトリ API を実装するためにソース コードを記述します。
3. ソース コードから共有ライブラリを構築します。
4. 以下のデフォルトの場所に共有ライブラリを配置します。

- UNIX プラットフォームでは、SiteMinder lib ディレクトリ
- Windows プラットフォームでは、SiteMinder bin ディレクトリ

オプションで、SiteMinder ポリシー サーバがアクセスできることを条件として別の場所に共有ライブラリを配置できます。別の場所を使用する場合は、[SiteMinder ユーザディレクトリ] ダイアログ ボックスの [ライブラリ] フィールドで共有ライブラリへの完全修飾パスを指定します。

## ディレクトリ アプリケーションの構築

ディレクトリ API アプリケーションを構築する場合、SmAPI.h ファイルを含めます。ソース コードには以下のステートメントが含まれる必要があります。

```
#include "SmApi.h"
```

ディレクトリ API を実装する共有ライブラリを構築する場合、他の共有ライブラリまたはインポートライブラリにもリンクしている必要はありません。ディレクトリ API は、インクルードファイル SmApi.h で定義されたエクスポート可能な関数を持つ共有ライブラリとして構築されます。

## エクスポートされた列挙

SmApi.h には、ディレクトリ API によって使用される以下の列挙が含まれます。

- Sm\_DirApi\_Capability\_t (ディレクトリ機能)
- Sm\_PolicyResolution\_t (ポリシー解決)

### ディレクトリ機能

Sm\_DirApi\_Capability\_t は、カスタム ディレクトリに対して設定できる機能を列挙します。

以下の表は、Sm\_DirApi\_Capability\_t で列挙されたディレクトリ機能のリストを示しています。表の後に各機能の説明が示されます。

名前	値
Sm_DirApi_Capability_ForceResetUserPassword	0x00000001
Sm_DirApi_Capability_ChangeUserPassword	0x00000002
Sm_DirApi_Capability_DisableUser	0x00000004
Sm_DirApi_Capability_SetUserAttributes	0x00000008
Sm_DirApi_Capability_Recursive	0x00000010

カスタム ディレクトリが特定の機能を持つためには、その機能に対して必須ユーザ属性を定義する必要があります。たとえば、ユーザパスワードを変更するように SiteMinder を有効にするには、パスワード属性を識別する必要があります。その後、SiteMinder はユーザパスワードを取得および設定するためにその属性を使用します。

- **Sm\_DirApi\_Capability\_ForceResetUserPassword**。カスタム ディレクトリはユーザパスワードリセットを強制できます。

SiteMinder でパスワードのリセットの強制を有効にするには、以下のユーザ属性を定義します。

- **パスワード属性**。ユーザパスワードを取得および設定するために SiteMinder が使用できる属性。SiteMinder Administrative UI で、[ユーザディレクトリ] ダイアログ ボックス上の [ユーザ属性] タブの [パスワード属性] フィールドにこの属性名を入力します。サンプル内では、属性名は **password** です。
- **無効フラグ**。ユーザの無効状態を取得および設定するために SiteMinder が使用できる属性。SiteMinder Administrative UI で、[ユーザディレクトリ] ダイアログ ボックス上の [ユーザ属性] タブの [無効フラグ] フィールドにこの属性名を入力します。サンプル内では、属性名は **Disabled** です。

ユーザがパスワードの変更を強制される場合、ポリシー サーバは **SmDirSetUserDisabledState()** を呼び出します。サンプルコードで、ユーザの [無効フラグ] は **Sm\_Api\_Disabled\_PWMustChange** (無効の理由) に設定されます。

- **Sm\_DirApi\_Capability\_ChangeUserPassword**。カスタム ディレクトリはユーザパスワードを変更できます。パスワードを変更するには、パスワード属性を識別する必要があります。これは、ユーザパスワードを取得および設定するために SiteMinder が使用できる属性です。SiteMinder Administrative UI で、[ユーザディレクトリ] ダイアログ ボックス上の [ユーザ属性] タブの [パスワード属性] フィールドにこの属性名を入力します。

SiteMinder ポリシー サーバは、**SmDirChangeUserPassword()** を呼び出して、カスタム ディレクトリ内の [パスワード] フィールドの値を変更できるようにします。

- **Sm\_DirApi\_Capability\_DisableUser**。カスタムディレクトリはユーザアカウントを無効にできます。ユーザを無効にするには、無効フラグを識別する必要があります。これは、ユーザの無効状態を取得および設定するために SiteMinder が使用できる属性です。SiteMinder Administrative UI で、[ユーザディレクトリ] ダイアログボックス上の [ユーザ属性] タブの [無効フラグ] フィールドにこの属性名を入力します。

管理者がユーザアカウントを無効または有効にするために SiteMinder Administrative UI を使用する場合、またはパスワードサービスがユーザアカウントを無効にする場合、SiteMinder ポリシー サーバは `SmDirSetUserDisabledState()` を呼び出します。

SiteMinder で、ユーザアカウントは多くの理由で無効になり、これらの理由はデータ構造 `Sm_Api_DisabledReason_t` のメンバによって表されます。

- **Sm\_DirApi\_Capability\_SetUserAttributes**。SiteMinder はカスタムディレクトリでユーザ属性を設定できます。

SiteMinder ポリシー サーバは、`SmDirSetUserAttr()` を呼び出して、ユーザが SiteMinder を使用してカスタムディレクトリでユーザ属性を設定できるようにします。

- **Sm\_DirApi\_Capability\_Recursive**。カスタムディレクトリには再帰をサポートする機能があります。たとえば、カスタムディレクトリはネストされたグループをサポートする場合があります。

ディレクトリ API 内の以下の関数が、再帰的なフラグを保持するためのパラメータを持っています。

- `SmDirGetUserGroups()`
- `SmDirValidateUserPolicyRelationship()`

SiteMinder ポリシー サーバにディレクトリ機能に関する情報を送信するには、関数 `SmDirQueryVersion()` を実装します。`Sm_DirApi_Capability_t` に列挙された 1 つ以上の値を渡すには、機能パラメータ (`pnCapabilites`) を使用します。その後、SiteMinder はこれらの機能をチェックします。

たとえば、ユーザがパスワードの変更を試みる場合、SiteMinder ポリシーサーバは、`SmDirQueryVersion()` を呼び出して機能

`Sm_DirApi_Capability_ChangeUserPassword` を確認します。カスタムディレクトリにその機能がない場合、ユーザはエラーメッセージを受信します。

ディレクトリ機能を設定する例は、サンプルコードに示されています。まず、*\*pnCapabilities* をゼロに初期化し、次に、以下のように *\*pnCapabilities* を設定します。

```
*pnCapabilities =  
    *pnCapabilities | Sm_DirApi_Capability_<supported_capability>;
```

例：

```
*pnCapabilities =  
    *pnCapabilities | Sm_DirApi_Capability_ChangeUserPassword;  
*pnCapabilities =  
    *pnCapabilities | Sm_DirApi_Capability_DisableUser;
```

SiteMinder による使用を対象としたフィールドで他のアプリケーションがデータを変更することがないようにしてください。たとえば、SiteMinder の無効な状態を保持するフィールドのデータを他のアプリケーションが変更することがないようにしてください。

### ポリシー解決

*Sm\_PolicyResolution\_t* は、*SmApi.h* で定義され、2つのポリシー オブジェクト間の関係を説明する値を列挙します。以下のディレクトリ API 関数が *Sm\_PolicyResolution\_t* を使用します。

- *SmDirAddEntry()*
- *SmDirGetDirObjInfo()*
- *SmDirRemoveEntry()*
- *SmDirValidateUserPolicyRelationship()*

## 一般的なデータ型と構造

データ型と構造はディレクトリ API で使用されますが、他の SiteMinder API によって使用される場合もあります。

*Sm\_Api\_DisabledReason\_t* は、ユーザ アカウントが無効になる理由を列挙します。

以下のディレクトリ API 関数は *Sm\_PolicyResolution\_t* を使用します。

- *SmDirGetUserDisabledState()*
- *SmDirSetUserDisabledState()*

ユーザアカウントが有効または無効になる場合、SiteMinder ポリシー サーバは `SmDirSetUserDisabledState()` を呼び出します。この呼び出しは、`Sm_Api_DisabledReason_t` で列挙されるとおり、カスタムディレクトリ内で 1 つ以上の無効の理由に対して無効フラグを設定する機会を与えます。ユーザアカウントが無効または有効にされる場合、`SmDirGetUserDisabledState()` は無効の理由を返します。`SmDirGetUserDisabledState()` を実装するとき、カスタムディレクトリが無効なフラグをサポートしない場合は、`Sm_Api_Disabled_Enabled` を返します。

**注:** ユーザのアカウントが複数の理由で無効になる場合があります。たとえば、[ユーザは次回ログイン時にパスワード変更が必要] チェックボックスが選択され、管理者が [無効] をクリックする場合、`nDisabledReason` は `Sm_Api_Disabled_PWMustChange` ビットおよび `Sm_Api_Disabled_AdminDisabled` ビットの両方を保持します。

無効なフラグは SiteMinder ユーザ属性です。SiteMinder Administrative UI で、[ユーザディレクトリ] ダイアログボックスの [ユーザ属性] タブ上で、[無効フラグ] フィールド内に属性の名前を入力します。サンプル内では、属性名は `Disabled` です。

構造 `Sm_Api_Context_t` は、SiteMinder ログ記録ユーティリティ、トレースユーティリティおよびエラー ユーティリティに関数ポインタを与えます。

`Sm_Api_Reason_t` は、認証失敗などアクセス イベントの理由を列挙します。ユーザが認証に対する認証情報を提供する場合、ユーザ名と DN を検証する SiteMinder ポリシー サーバは `SmDirAuthenticateUser()` を呼び出します。この呼び出しは、アクセス イベントに関する情報を返す機会を与えます。

## 初期化およびリリース関数

オブジェクトを初期化するには、SiteMinder ポリシー サーバは、以下の表内の関数を呼び出します。

関数名	SiteMinder ポリシー サーバによって初期化されるオブジェクト
<a href="#">SmDirInit()</a> (P. 830)	ディレクトリ プロバイダ。プロバイダ ハンドルを設定します。
<a href="#">SmDirInitDirInstance()</a> (P. 832)	ディレクトリ インスタンス。ディレクトリ インスタンス ハンドルを設定します。

---

関数名	SiteMinder ポリシー サーバによって初期化されるオブジェクト
<a href="#">SmDirInitUserInstance()</a> (P. 835)	ディレクトリ エントリ (ユーザ) インスタンス。ユーザ エントリ インスタンス ハンドルを設定します。

---

### ディレクトリ プロバイダの初期化

SiteMinder ポリシー サーバの開始後に最初にカスタム ディレクトリ プロバイダが必要になるとき、ポリシー サーバは `SmDirInit()` を呼び出してディレクトリ プロバイダを初期化します。この時点で、サンプルコードで示されるようにプロバイダ ハンドルを設定します。SiteMinder ポリシー サーバは、ポリシー サーバ サービスのいずれかが開始 (または再開) されるまで `SmDirInit()` を再度呼び出すことはしません。

`SmDirInit()` は、各カスタム ディレクトリ プロバイダ ライブラリ (.dll または .so) に対して 1 度呼び出されます。

### ディレクトリ インスタンスの初期化

ポリシー サーバは、ディレクトリ インスタンスを初期化するために `SmDirInitDirInstance()` を呼び出します。サンプルコードで示されるようにディレクトリ インスタンス ハンドルを設定します。

このディレクトリ プロバイダ ライブラリを使用して、`SmDirInitDirInstance()` がディレクトリ インスタンスにつき 1 回呼び出されます。SiteMinder は、(検索または取得プロパティなどの操作を実行するために) ディレクトリ コンテキストを必要とする場合に、認証または許可要求を処理する間に関数を呼び出します。この関数は、通常要求の初めに呼び出されます。

### ディレクトリ エントリ(ユーザ) インスタンスの初期化

SiteMinder ポリシー サーバは `SmDirInitUserInstance()` を呼び出すことによりユーザ インスタンスを初期化します。 サンプル コードで示されるようにディレクトリ エントリ (ユーザ) インスタンス ハンドルを設定します。

オブジェクトを解放するには、以下の表の関数を使用します。

関数名	SiteMinder ポリシー サーバによって解放されるオブジェクト
<a href="#">SmDirRelease()</a> (P. 840)	ディレクトリ プロバイダ。 プロバイダ ハンドルの削除
<a href="#">SmDirReleaseInstance()</a> (P. 840)	ディレクトリまたはエントリ インスタンス。どのハンドルが渡されるかを決定します。

### ユーザ インスタンスの解放

SiteMinder ポリシー サーバは、必要に応じてユーザ インスタンス ハンドルを解放できるように、`SmDirReleaseInstance()` を呼び出します。 渡されるハンドルがディレクトリ インスタンス ハンドルではなくユーザ インスタンス ハンドルであることを確認します。

### ディレクトリ インスタンスのリリース

SiteMinder ポリシー サーバは、必要に応じてディレクトリ インスタンス ハンドルを解放できるように、`SmDirReleaseInstance()` を呼び出します。 渡されるハンドルがユーザ インスタンス ハンドルではなくディレクトリ インスタンス ハンドルであることを確認します。

ディレクトリ コンテキストが必要なくなった後、SiteMinder は `SmDirInitDirInstance()` の呼出しごとに 1 回 `SmDirReleaseInstance()` を呼び出します。 これは通常要求の最後に呼び出されます。

### ディレクトリ プロバイダの解放

管理者が SiteMinder ポリシー サーバのシャットダウンを開始するとき、SiteMinder ポリシー サーバは `SmDirRelease()` を呼び出してディレクトリ プロバイダを解放します。

詳細情報:

[ハンドルタイプを区別する方法 \(P. 864\)](#)

## ユーティリティ関数

これらの関数は、ディレクトリ操作のシーケンス、またはディレクトリ エントリ (ユーザ) 操作のシーケンス内で呼び出すことができます。関数がパラメータを通してインスタンス ハンドルを受信する場合、それがディレクトリ インスタンス ハンドルかディレクトリ エントリ (ユーザ) インスタンス ハンドルかを判断します。

関数名	説明
<a href="#">SmDirFreeString()</a> (P. 812)	文字列に割り当てられた空きメモリ。
<a href="#">SmDirFreeStringArray()</a> (P. 812)	文字列配列に割り当てられた空きメモリ。
<a href="#">SmDirQueryVersion()</a> (P. 839)	ディレクトリ API バージョン、およびサポートするディレクトリ機能を確認します。

### 文字列および文字列配列の解放

SiteMinder ポリシー サーバが文字列パラメータをとる操作関数を呼び出した後、SiteMinder ポリシー サーバは、`SmDirFreeString()` または `SmDirFreeStringArray()` を呼び出して割り当てられたメモリを解放します。呼び出しを繰り返して複数の文字列を解放できます。

たとえば、SiteMinder 管理者は、ユーザ Mikel の検索を実行するために SiteMinderAdministrative UI を使用できます。SiteMinder 管理者は、まず [検索] ドロップダウン リスト ボックスから文字列 User を選択し、次に、[検索式] フィールドに文字列 Mikel を入力します。SiteMinder は `SmDirLookup()` を呼び出し、文字列 (User = Mikel" の形式) を `IpszPattern` パラメータに渡します。その後、SiteMinder は `SmDirFreeStringArray()` を 2 回呼び出します。最初の呼び出しで、SiteMinder は文字列配列 Mikel を渡します。2 番目の呼び出しで、SiteMinder は文字列配列 User を渡します。

### クエリおよび検証

SiteMinder ポリシー サーバは、`SmDirQueryVersion()`、そして `SmDirValidateInstance()` を頻繁に呼び出します。このシーケンスは複数回繰り返される場合があります。

## ディレクトリ上の操作

ポリシー サーバは、ディレクトリ操作関数を呼び出して、カスタム ディレクトリに対するディレクトリ操作を定義できるようにします。たとえば、ユーザがユーザを検索するために SiteMinderAdministrative UI を使用している場合、ポリシー サーバは `SmDirLookup()` を呼び出します。

関数名	以下に対してこの関数を使用します。
<a href="#">SmDirAddEntry()</a> (P. 802)	カスタム ディレクトリにディレクトリ エントリ (ユーザ) を挿入します。
<a href="#">SmDirAddMemberToGroup()</a> (P. 804)	既存のグループにユーザまたはグループを追加します。
<a href="#">SmDirAddMemberToRole()</a> (P. 805)	ユーザ、またはグループにロールを割り当てます。
<a href="#">SmDirEnumerate()</a> (P. 811)	ディレクトリ エントリおよび対応するクラス名のリストを取得します。

関数名	以下に対してこの関数を使用します。
<a href="#">SmDirGetDirConnection()</a> (P. 813)	ディレクトリに接続ハンドルを取得します。
<a href="#">SmDirGetDirObjInfo()</a> (P. 814)	オブジェクト パラメータで指定されたオブジェクトに関する情報を取得します。
<a href="#">SmDirGetGroupMembers()</a> (P. 816)	ユーザ グループのメンバを取得します。
<a href="#">SmDirGetLastErrMsg()</a> (P. 817)	どのインスタンス ハンドルが渡されるか決定し、関連するエラーメッセージを返します。また、ディレクトリ エントリ (ユーザ) 操作と共に使用されます。
<a href="#">SmDirGetRoleMembers()</a> (P. 818)	ロールに割り当てられたディレクトリ エントリを取得します。
<a href="#">SmDirLookup()</a> (P. 837)	ディレクトリのパターンを検索します。
<a href="#">SmDirRemoveEntry()</a> (P. 842)	カスタムディレクトリからディレクトリ エントリ (ユーザ) を削除します。
<a href="#">SmDirRemoveMemberFromGroup()</a> (P. 844)	既存グループからユーザまたはグループを削除します。
<a href="#">SmDirRemoveMemberFromRole()</a> (P. 845)	割り当てられたロールからユーザまたはグループを削除します。
<a href="#">SmDirSearch()</a> (P. 846)	検索フィルタ パラメータで指定された基準で検索します。
<a href="#">SmDirSearchCount()</a> (P. 848)	パラメータで指定された基準を満たすエントリの数を返します。
<a href="#">SmDirValidateInstance()</a> (P. 855)	どのインスタンス ハンドルが渡されるか決定し、そのインスタンスを検証します。また、ディレクトリ エントリ (ユーザ) 操作と共に使用されます。
<a href="#">SmDirValidateUserDN()</a> (P. 856)	ユーザ ID 上であらゆる必要な検証を実行します。
<a href="#">SmDirValidateUsername()</a> (P. 857)	ユーザによって提示される認証情報をカスタムディレクトリに対するユーザ ID に変換します。

## ディレクトリ エントリ(ユーザ) 上の操作

このセクションで扱われる操作は、ユーザ、グループ、およびロールなどのディレクトリ エントリに適用されます。

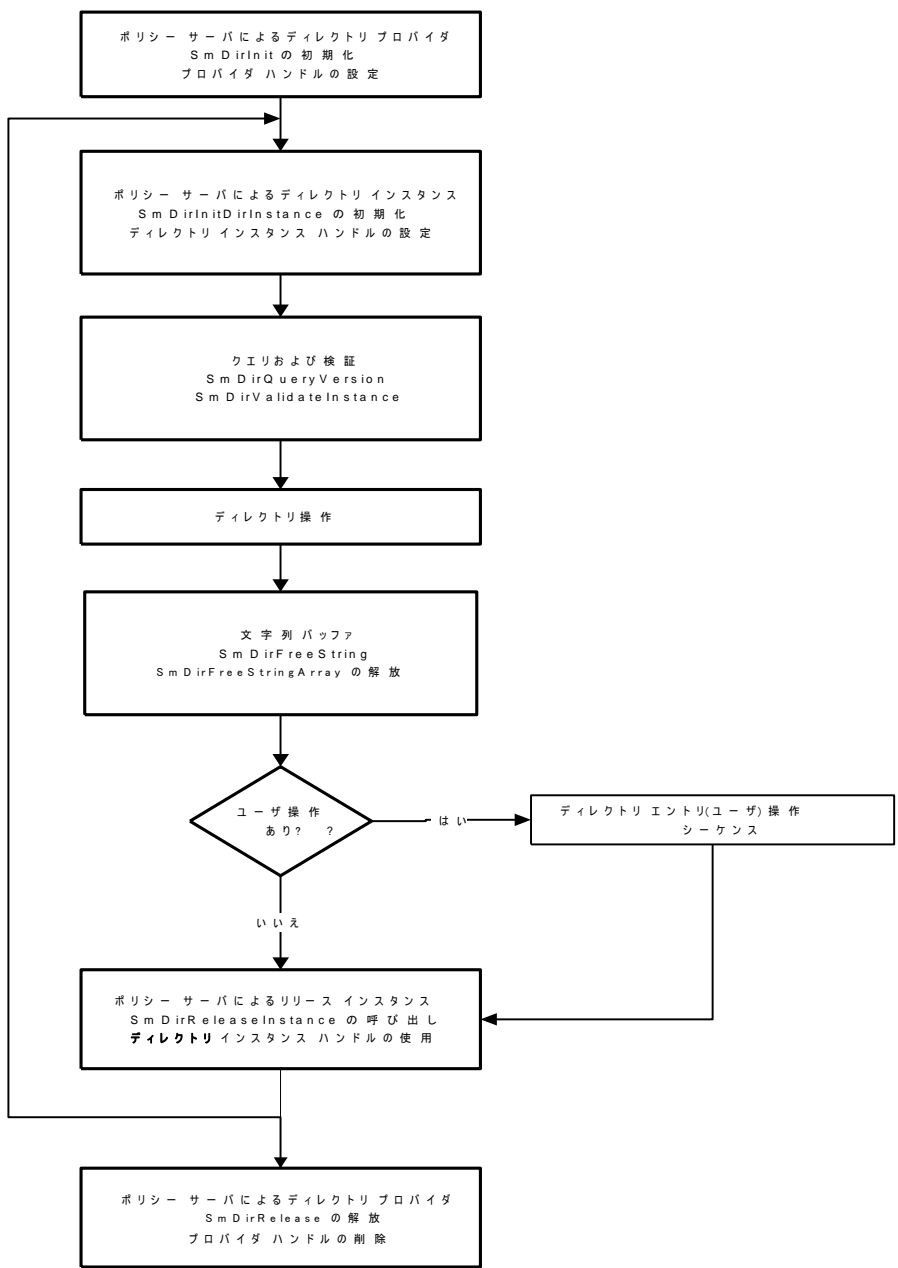
SiteMinder ポリシー サーバは、実行される操作に関連する、ディレクトリ エントリ (ユーザ) 操作関数を呼び出します。たとえば、ユーザがユーザ アカウントを無効にするために SiteMinder Administrative UI を使用している場合、SiteMinder ポリシー サーバは `SmDirSetUserDisabledState()` を呼び出します。

関数名	以下に対してこの関数を使用します。
<a href="#">SmDirAuthenticateUser()</a> (P. 806)	提供されるユーザ名およびパスワードに対するディレクトリを確認します。
<a href="#">SmDirChangeUserPassword()</a> (P. 809)	指定されたユーザに対するパスワードフィールドの値を変更します。
<a href="#">SmDirGetLastErrMsg()</a> (P. 817)	どのインスタンス ハンドルが渡されるか決定し、関連するエラー メッセージを返します。また、ディレクトリ操作と共に使用されます。
<a href="#">SmDirGetUserAttr()</a> (P. 819)	指定されたユーザ属性の値を取得します。
<a href="#">SmDirGetUserAttrMulti()</a> (P. 820)	単一の属性に対する値の配列を取得します。
<a href="#">SmDirGetUserClasses()</a> (P. 822)	指定された DN に対するオブジェクト クラスを取得します。
<a href="#">SmDirGetUserDisabledState()</a> (P. 823)	無効なユーザ アカウントに対して、アカウントが無効にされた理由を返します。これ以外の場合は有効を返します。
<a href="#">SmDirGetUserGroups()</a> (P. 826)	ユーザが属するグループを取得します。
<a href="#">SmDirGetUserProperties()</a> (P. 827)	すべてのユーザ属性の名前または必要なユーザ属性のみの名前を返します。
<a href="#">SmDirGetUserRoles()</a> (P. 828)	ユーザが属するロールを取得します。
<a href="#">SmDirSetUserAttr()</a> (P. 850)	ユーザ属性に対する値を設定します。
<a href="#">SmDirSetUserAttrMulti()</a> (P. 851)	単一の属性に対する値の配列を設定します。

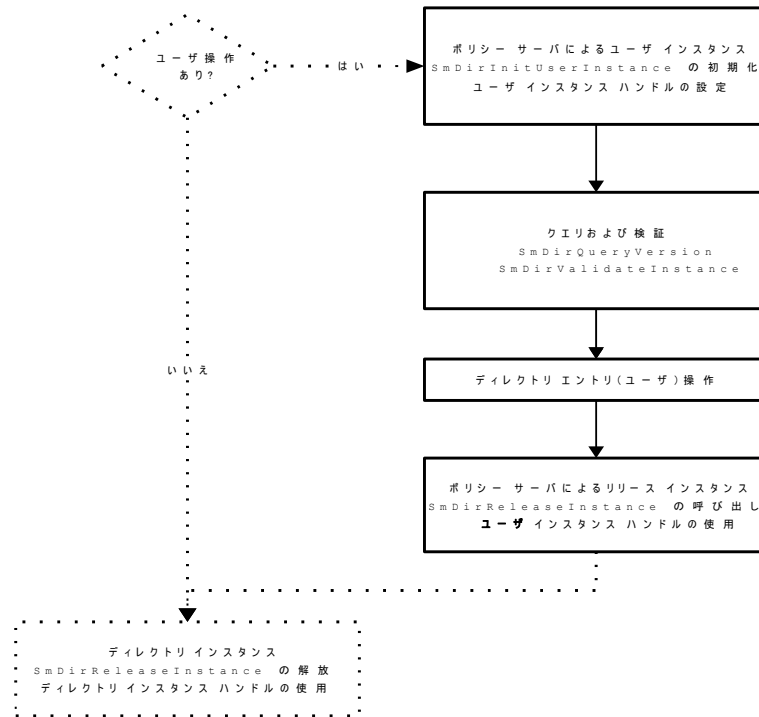
関数名	以下に対してこの関数を使用します。
<a href="#">SmDirSetUserDisabledState()</a> (P. 852)	ユーザ アカウントを有効または無効にします。
<a href="#">SmDirValidateInstance()</a> (P. 855)	どのインスタンス ハンドルが渡されるか決定し、そのインスタンスを検証します。また、ディレクトリ操作と共に使用されます。
<a href="#">SmDirValidateUserPolicy Relationship()</a> (P. 858)	ポリシー オブジェクト値の関係を検証します。

## ディレクトリ API リファレンス

以下の図は、ディレクトリ上で操作を実行する、そして場合によってはディレクトリ エントリ (ユーザ) 上で操作を実行する手順の関数呼び出しの順番の概要を示しています。たとえば、**SiteMinder Administrative UI** を使用してカスタム ディレクトリ内で特定のユーザを検索するには、ディレクトリ上の操作およびディレクトリ エントリ (ユーザ) 上の操作の両方が必要です。手順によっては、ディレクトリ上の操作のみが含まれます。たとえば、カスタム ユーザ ディレクトリのプロパティを表示するためだけに **SiteMinder Administrative UI** を使用する場合は、ディレクトリ上の操作のみが必要です。



この 2 番目の図は、ディレクトリ エントリ (ユーザ) 操作が発生する場合にのみ発生する追加のシーケンスを示しています。



ユーザを認証するには、SiteMinder ポリシー サーバはユーザにユーザ名を要求します。ユーザ提供のユーザ名を、ディレクトリによってユーザデータへのプライマリ キーとして使用される内部のユーザ ID キーに変換するために `SmDirValidateUsername()` が呼び出されます。認証情報からのユーザ名は `IpszUsername` パラメータで提供されます。`SmDirValidateUsername()` が実装されない場合、ユーザ提供のユーザ名が `IpszUserDN` に渡されます。

`SmDirValidateUsername()` が実装される場合、`IpszNewUsername` パラメータ内にユーザ ID を返します。`IpszNewUsername` によって返された値は `IpszUserDN` パラメータ値になります。

`IpszUserDN` パラメータ値は、`SmDirValidateUserDN()` および `SmDirAuthenticateUser()` など他の多くの関数へ渡されます。

## SmDirAddEntry()

SiteMinder ポリシー サーバは、ユーザがカスタム ディレクトリにディレクトリ エントリ (ユーザ) を挿入できるように、`SmDirAddEntry()` を呼び出します。ディレクトリ エントリの例はユーザ、グループおよびロールです。たとえば、SQL データベースを使用しており、グループを追加する必要がある場合、データベースに対するグループ テーブル (およびすべての関連するテーブル) にレコードを挿入するために `SmDirAddEntry()` を使用することができます。

階層ディレクトリにエントリを追加するとき、LDAP 内のオブジェクト クラスなどエントリに渡された属性を参照すると役立つ場合があります。

### 構文

```
int SM_EXTERN SmDirAddEntry (
    const Sm_Api_Context_t*    lpApiContext,
    void*                       pHandle,
    void*                       pInstanceHandle,
    const Sm_EntryType_t      nPolicyResolution,
    const char*                 lpszEntryDN,
    const char**               lpszAttrNames,
    const char**               lpszAttrValues
);
```

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。
<i>pHandle</i>	I	<code>SmDirInit()</code> によって返されたポインタのアドレス。
<i>pInstanceHandle</i>	I	ディレクトリ インスタンス ハンドルへのポインタのアドレス。

パラメータ	I/O	説明
<i>nEntryType</i>	I	エントリのポリシー解決。ポリシー解決は <code>SmApi.h</code> で定義される <code>Sm_PolicyResolution_t</code> に列挙されます。 <code>Sm_PolicyResolution_t</code> の以下のエレメントは有効なエントリタイプです。 <ul style="list-style-type: none"><li>■ <code>Sm_PolicyResolution_Unknown</code></li><li>■ <code>Sm_PolicyResolution_User</code></li><li>■ <code>Sm_PolicyResolution_UserGroup</code></li><li>■ <code>Sm_PolicyResolution_UserRole</code></li><li>■ <code>Sm_PolicyResolution_Org</code></li></ul>
<i>lpszEntryDN</i>	I	追加されているエントリに対する識別名が含まれるバッファ。
<i>lpszAttrNames</i>	I	エントリ属性の名前が含まれるバッファ。
<i>lpszAttrValues</i>	I	エントリ属性の値が含まれるバッファ。

### 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

### 注釈

分散代行管理サービスがユーザまたはロールを含むディレクトリ エントリを作成するために使用されるとき、この関数が呼び出されます。

## SmDirAddMemberToGroup()

SiteMinder ポリシー サーバは、既存グループにユーザまたはグループを追加できるように `SmDirAddMemberToGroup()` を呼び出します。

ロールにユーザまたはグループを追加する場合は、`SmDirAddMemberToRole()` を使用します。グループとロールの間の相違はカスタム ディレクトリ プロバイダによって定義されます。一部のプロバイダにおいては違いはありません。

### 構文

```
int SM_EXTERN SmDirAddMemberToGroup (  
    const Sm_Api_Context_t* lpApiContext,  
    void* pHandle,  
    void* pInstanceHandle,  
    const char* lpszMemberDN,  
    const char* lpszGroupDN  
);
```

パラメータ	I/O	説明
<code>lpApiContext</code>	I	API コンテキスト構造へのポインタ。
<code>pHandle</code>	I	<code>SmDirInit()</code> によって返されたポインタのアドレス。
<code>pInstanceHandle</code>	I	ディレクトリ インスタンス ハンドルへのポインタのアドレス。
<code>lpszMemberDN</code>	I	既存のグループに追加されるユーザまたはグループの識別名が含まれるバッファ。
<code>lpszGroupDN</code>	I	メンバが追加されているグループに対する識別名が含まれるバッファ。

### 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

### 注釈

この関数は、既存のグループへのユーザまたはグループの割り当てに分散代行管理サービスが使用される場合に呼び出されます。

## SmDirAddMemberToRole()

SiteMinder ポリシー サーバは、ロールをユーザまたはグループに割り当てることができるようにするために `SmDirAddMemberToRole()` を呼び出します。

たとえば、Oracle では、ロールは、ユーザに付与できる一連のオブジェクトまたはシステム権限です。グループは、一連のユーザです。コレクションを実行する者すべてが AR テーブルを更新し、CUSTOMER テーブルから選択できるようにするには、COLLECTIONS という名前のロールを作成できます。その後、コレクションを実行する各ユーザ、または Accounts Receivable などのグループに COLLECTIONS ロールを割り当てることができます。

既存のグループにユーザまたはグループを割り当てたい場合、`SmDirAddMemberToGroup()` を使用します。グループとロールの違いはカスタム ディレクトリのプロバイダによって定義されます。

プロバイダによっては、ロールとグループの間に相違はありません。

### 構文

```
int SM_EXTERN SmDirAddMemberToRole (
    const Sm_Api_Context_t* lpApiContext,
    void*                    pHandle,
    void*                    pInstanceHandle,
    const char*              lpszMemberDN,
    const char*              lpszRoleDN
);
```

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。
<i>pHandle</i>	I	SmDirInit() によって返されたポインタのアドレス。
<i>pInstanceHandle</i>	I	ディレクトリ インスタンス ハンドルへのポインタのアドレス。
<i>lpszMemberDN</i>	I	既存のロールに追加されるユーザまたはグループの識別名が含まれるバッファ。

パラメータ	I/O	説明
<i>lpszRoleDN</i>	I	メンバが追加されているロールに対する識別名が含まれるバッファ。

### 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

### 注釈

この関数は、分散代行管理サービスがユーザまたはグループにロールを割り当てるために使用されるときに呼び出されます。

## SmDirAuthenticateUser()

提供されるユーザ名およびパスワード用のディレクトリを確認するには `SmDirAuthenticateUser()` 関数を使用します。

`SmDirAuthenticateUser()` を呼び出した後、`SiteMinder` は `SmDirFreeString()` を呼び出して *lpszUserMsg* バッファを解放し、次に、`SmDirFreeString()` を再度呼び出して *lpszErrMsg* バッファを解放します。

### 構文

```
int SM_EXTERN SmDirAuthenticateUser (
    const Sm_Api_Context_t* lpApiContext,
    void* pHandle,
    void* pInstanceHandle,
    const char* lpzUserDN,
    const char* lpzPassword,
    Sm_Api_Reason_t* pnReason,
    char** lpzUserMsg,
    char** lpzErrMsg
);
```

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。
<i>Handle</i>	I	<code>SmDirInit()</code> によって返されたポインタのアドレス。

パラメータ	I/O	説明
<i>pInstanceHandle</i>	I	ユーザ インスタンス ハンドルへのポインタのアドレス。
<i>lpszUserDN</i>	I	<p>認証される必要があるユーザ DN が含まれるバッファ。</p> <p><b>SmDirValidateUsername()</b> が実装されない場合、ユーザ提供のユーザ名が <i>lpszUserDN</i> に渡されます。</p> <p><b>SmDirValidateUsername()</b> は、実装される場合 <i>lpszNewUsername</i> パラメータ内にユーザ ID を返します。<i>lpszNewUsername</i> によって返された値は <i>lpszUserDN</i> パラメータ値になります。</p>
<i>lpszPassword</i>	I	認証される必要があるパスワードが含まれるバッファ。
<i>pnReason</i>	O	<b>Sm_Api_Reason_t</b> で列挙された理由を使用する、認証イベントの結果の理由へのポインタ。
<i>lpszUserMsg</i>	O	<p>ユーザに対するメッセージを受信する出力バッファ。このメッセージは、チャレンジテキスト、または認証方式開発者が <b>SiteMinder</b> 外部の機構によってユーザに提示することを希望する他のメッセージです。サンプルで、偽のユーザ名が示される場合、認証は失敗し、認証に失敗した文字列が <i>lpszUserMsg</i> にコピーされます。</p> <p><b>Web</b> エージェントは HTTP 変数 <b>HTTP_SM_USERMSG</b> にこのメッセージを保存します。<b>RADIUS</b> 認証については、ユーザメッセージが <b>REPLY-MESSAGE</b> レスポンス属性で返されます。</p> <p><b>SiteMinder</b> ポリシー サーバは <i>lpszUserMsg</i> で <b>SiteMinder</b> 認証ログにエラーメッセージを書き込みます。</p>

パラメータ	I/O	説明
<i>lpszErrMsg</i>	0	<p>エラーメッセージを受信する出力バッファ。SiteMinder にエラーメッセージを返すにはこのバッファを使用します。サンプルで、偽のユーザ名が示される場合、認証は失敗し、認証に失敗した文字列が <i>lpszErrMsg</i> にコピーされます。</p> <p>SiteMinder ポリシー サーバは <i>lpszErrMsg</i> で SiteMinder 認証ログにエラーメッセージを書き込みます。 <i>lpszErrMsg</i> 内の文字列がログ内の [Not Authenticated] に続きます。 <i>lpszErrMsg</i> 内の文字列が <i>lpszErrMsg</i> 内の文字列に続きます。たとえば、チャレンジされたユーザが偽のユーザ名を示す場合、SiteMinder ポリシー サーバは以下のステータスメッセージをログに書き込みます。 'impostor' Not Authenticated. ErrMsg. UserMsg</p>

### 戻り値

認証が成功する場合、0 を返し、処理にエラーがあるまたはユーザ提供の認証情報が無効の場合 -1 を返します。

認証が失敗する場合は、出力パラメータ *pnReason* を使用して理由を示し、-1 を返します。

### 注釈

この関数は、カスタム ディレクトリ内のユーザに対して *IsAuthenticated* を実行するために SiteMinder テストツールを使用する場合に呼び出されません。

## SmDirChangeUserPassword()

SiteMinder ポリシー サーバは、**SmDirChangeUserPassword()** を呼び出して、カスタム ディレクトリ内の [パスワード] フィールドの値を変更できるようにします。

**SmDirChangeUserPassword()** を実装するには、カスタム ディレクトリで [パスワード] フィールドの名前を指定する必要があります。SiteMinder Administrative UI で、[ユーザディレクトリ] ダイアログ ボックス上の [ユーザ属性] タブの [パスワード属性] フィールドにこの属性名を入力します。

以下の情報でユーザの識別名が渡されます。

- 古いパスワード
- 新しいパスワード
- カスタム ディレクトリの [パスワード] フィールドの名前
- 古いパスワードが必要かどうか

### 構文

```
int SM_EXTERN SmDirChangeUserPassword (
    const Sm_Api_Context_t* lpApiContext,
    void*                    pHandle,
    void*                    pInstanceHandle,
    const char*              lpzUserDN,
    const char*              lpzOldPassword,
    const char*              lpzNewPassword,
    const char*              lpzPasswordAttr,
    const int                bDoNotRequireOldPassword
);
```

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。
<i>pHandle</i>	I	SmDirInit() によって返されたポインタのアドレス。
<i>pInstanceHandle</i>	I	ユーザ インスタンス ハンドルへのポインタのアドレス。

パラメータ	I/O	説明
<i>lpszUserDN</i>	I	パスワードを変更する必要があるユーザ DN が含まれるバッファ。
<i>lpszOldPassword</i>	I	ユーザ DN の古いパスワードが含まれるバッファ。
<i>lpszNewPassword</i>	I	ユーザ DN の新しいパスワードが含まれるバッファ。
<i>lpszPasswordAttr</i>	I	ユーザのパスワードが保存されるディレクトリ属性。ユーザパスワードを変更するためにこの属性を使用します。 Netscape LDAP ディレクトリ内のこの属性名のデフォルト値は <i>userpassword</i> です。
<i>bDoNotRequireOldPassword</i>	I	パスワード変更を実行するためにユーザが古いパスワードを指定する必要があるかどうかを示す値。値 1 は、古いパスワードが必要ないことを示し、0 は、古いパスワードが必要であることを示します。 管理者は古いパスワードを指定する必要がない場合がありますが、エンドユーザは指定する必要があります。

### 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

### サンプルコード情報

ユーザパスワードを変更している場合、SiteMinder は *lpszPasswordAttr* を通してパスワード属性の名前を渡します。この情報を保持するディレクトリ属性の名前を示すには以下を実行します。

1. SiteMinderAdministrative UI で、[ユーザディレクトリ] ダイアログボックスに移動します
2. [ユーザ属性] タブで、[パスワード属性] フィールドに入力します。サンプルを使用するには、このフィールドでパスワードを入力します。

## SmDirEnumerate()

SiteMinder ポリシー サーバは、ユーザディレクトリで識別名のリストおよびその対応するクラス名（User または Group）を取得するために SmDirEnumerate() を呼び出します。

SiteMinder ポリシー サーバは、SmDirFreeStringArray() を呼び出して *lpszDNs* と *lpszClasses* の配列を解放します。

### 構文

```
int SM_EXTERN SmDirEnumerate (
    const Sm_Api_Context_t* lpApiContext,
    void*                    pHandle,
    void*                    pInstanceHandle,
    char***                  lpszDNs,
    char***                  lpszClasses
);
```

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。
<i>pHandle</i>	I	SmDirInit() によって返されたポインタのアドレス。
<i>pInstanceHandle</i>	I	ディレクトリ インスタンス ハンドルへのポインタのアドレス。
<i>lpszDNs</i>	O	ディレクトリの中にあるユーザ識別名のリスト。
<i>lpszClasses</i>	O	ユーザ識別名用の対応するクラス情報のリスト。

### 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

### 注釈

この関数は、[SiteMinder ユーザディレクトリ]ダイアログ ボックスで[内容の表示] ボタンをクリックすると呼び出されます。

## SmDirFreeString()

SiteMinder ポリシー サーバは *lpszString* の指定された文字列に割り当てられたメモリを解放するために `SmDirFreeString()` を呼び出します。

### 構文

```
void SM_EXTERN SmDirFreeString (  
    char* lpszString  
);
```

---

パラメータ	I/O	説明
<i>lpszString</i>	I	割り当てられたメモリが解放されることになっている文字列。

---

## SmDirFreeStringArray()

SiteMinder ポリシー サーバは *lpszStringArray* の指定された文字列配列に割り当てられたメモリを解放するために `SmDirFreeStringArray()` を呼び出します。たとえば、`SmDirEnumerate()` を呼び出した後、SiteMinder ポリシーサーバはパラメータ *lpszDNs* に割り当てられたメモリを解放するために `SmDirFreeStringArray()` を呼び出し、次に、パラメータ *lpszClasses* に割り当てられたメモリを解放するために `SmDirFreeStringArray()` を再度呼び出します。

### 構文

```
void SM_EXTERN SmDirFreeStringArray (  
    char* lpszStringArray  
);
```

---

パラメータ	I/O	説明
<i>lpszStringArray</i>	I	割り当てられたメモリが解放されることになっている文字列配列。

---

## SmDirGetDirConnection()

この関数は、ディレクトリに接続ハンドルを取得するために呼び出されます。

アクティブなルール、ポリシーまたはレスポンスを使用している場合、SiteMinder ポリシー サーバは、認証方式ライブラリまたはアクティブなライブラリを呼び出すときに `SmDirGetDirConnection()` を呼び出します。SiteMinder ポリシー サーバは認証の前に `SmDirGetDirConnection()` を呼び出します。

`SmDirGetDirConnection()` を実装するとき、NULL を返します。

### 構文

```
void* SM_EXTERN SmDirGetDirConnection (
    const Sm_Api_Context_t* lpApiContext,
    void* pHandle,
    void* pInstanceHandle
);
```

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。
<i>pHandle</i>	I	<code>SmDirInit()</code> によって返されたポインタのアドレス。
<i>pInstanceHandle</i>	I	ディレクトリ インスタンス ハンドルへのポインタのアドレス。

### 戻り値

`SmDirGetDirConnection()` を実装するとき、サンプルで示されるように、NULL を返します。

## SmDirGetDirObjInfo()

SiteMinder ポリシー サーバは、*lpszObject* バッファで指定されたオブジェクトに関する情報を取得するために `SmDirGetDirObjInfo()` を呼び出します。この関数は、*lpszObject* として渡されたオブジェクトに関する以下の情報を取得するために呼び出します。

- 識別名 (*lpszDN*)
- クラス (*lpszClass*)
- ポリシー解決 (*pnSmPolicyResolution*)

SiteMinder ポリシー サーバは、*lpszDN* と *lpszClass* のバッファを解放するために `SmDirFreeString()` を呼び出します。

## 構文

```
int SM_EXTERN SmDirGetDirObjInfo (
    const Sm_Api_Context_t* lpApiContext,
    void*                    pHandle,
    void*                    pInstanceHandle,
    const char*              lpszObject,
    char**                   lpszDN,
    char**                   lpszClass,
    int*                     pnSmPolicyResolution
);
```

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。
<i>pHandle</i>	I	<code>SmDirInit()</code> によって返されたポインタのアドレス。
<i>pInstanceHandle</i>	I	ディレクトリ インスタンス ハンドルへのポインタのアドレス。
<i>lpszObject</i>	I	ユーザ名の NULL で終わる文字列が含まれる文字列。
<i>lpszDN</i>	O	<i>lpszObject</i> から解決されたユーザ識別名に入力します。
<i>lpszClass</i>	O	解決されたユーザ識別名のクラスに入力します。

パラメータ	I/O	説明
<i>pnSmPolicyResolution</i>	0	解決されたユーザ識別名のポリシー解決に入力します。ポリシー解決は <i>Sm_PolicyResolution_t</i> で列挙されます。

### 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

### 注釈

この関数は以下の状況で呼び出されます。

- ユーザの検索
- ユーザアカウントの有効化または無効化

### サンプルコード情報

ユーザアカウント *Mikel* を無効にしている場合、*Mikel* が *IpszObject* として渡されます。サンプルコードを使用して、以下の値が返されます。

パラメータ	値
<i>IpszDN</i>	<i>Mikel</i>
<i>IpszClass</i>	<i>User</i>
<i>pnSmPolicyResolution</i>	1

## SmDirGetGroupMembers()

SiteMinder ポリシー サーバは、ユーザ グループのメンバを取得できるように、`SmDirGetGroupMembers()` を呼び出します。この関数は、分散代行管理サービスをサポートするように設計されています。

### 構文

```
int SM_EXTERN SmDirGetGroupMembers (  
    const Sm_Api_Context_t* lpApiContext,  
    void* pHandle,  
    void* pInstanceHandle,  
    const char* lpszGroupDN,  
    char*** lpszMembers,  
);
```

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。
<i>pHandle</i>	I	<code>SmDirInit()</code> によって返されたポインタのアドレス。
<i>pInstanceHandle</i>	I	ディレクトリ インスタンス ハンドルへのポインタのアドレス。
<i>lpszGroupDN</i>	I	メンバが取得されているグループに対する識別名が含まれるバッファ。
<i>lpszMembers</i>	O	グループのメンバが含まれるバッファ。

### 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

## SmDirGetLastErrMsg()

**SmDirGetLastErrMsg()** 関数を呼び出すと、ディレクトリ API 呼び出しからの最後のエラーメッセージが取得されます。

ユーザ操作またはディレクトリ操作が失敗する場合、インスタンスハンドル構造の 3 番目のフィールドである *szErrMsg* にエラーメッセージを保存します。その後、エラーメッセージを取得するために **SmDirGetLastErrMsg()** を呼び出すことができます。操作が失敗した場合、SiteMinder ポリシー サーバは **SmDirGetLastErrMsg()** も呼び出します。

この関数呼び出しは、ディレクトリ インスタンスおよびユーザインスタンスの両方に対して作成されます。ディレクトリ インスタンスハンドルまたはユーザ インスタンスハンドルのいずれかを *pInstanceHandle* を通して渡すことができます。コードは、どのハンドルが渡されるか決定し、適切なエラーメッセージを返す必要があります。例については、サンプルコードを参照してください。

**SmDirGetLastErrMsg()** を呼び出した後、SiteMinder ポリシー サーバは、エラーバッファを解放するために **SmDirFreeString()** を呼び出します。

### 構文

```
char* SM_EXTERN SmDirGetLastErrMsg (
    const Sm_Api_Context_t* lpApiContext,
    void*                    pHandle,
    void*                    pInstanceHandle
);
```

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。
<i>pHandle</i>	I	<b>SmDirInit()</b> によって返されたポインタのアドレス。
<i>pInstanceHandle</i>	I	ユーザ インスタンスハンドルまたはディレクトリ インスタンスハンドルへのポインタのアドレス。

## 戻り値

最後のエラーメッセージがディレクトリ操作またはユーザ操作中に発生しました。

## 注釈

この関数は、ユーザまたはグループを検索しているときに呼び出され、検索 (`SmDirLookup()`) は失敗します。

## サンプルコード情報

`SmDirValidateUserPolicyRelationship()` に対するサンプルコードで、ポリシー解決が `UserGroup` の条件で、関数 `chk_grp()` への呼び出しがあります。`chk_grp()` の 3 つ目のパラメータは、ユーザハンドル内に `szErrMsg` を返す出力パラメータです。その後、`SmDirGetLastErrMsg()` を呼び出し、ユーザインスタンスハンドルを渡した場合、`szErrMsg` に値を保存します。

## SmDirGetRoleMembers()

SiteMinder ポリシー サーバは、ユーザがロールに割り当てられたディレクトリ エントリを取得できるように、`SmDirGetRoleMembers()` を呼び出します。この関数は、分散代行管理サービスをサポートするように設計されています。

## 構文

```
int SM_EXTERN SmDirGetRoleMembers (  
    const Sm_Api_Context_t* lpApiContext,  
    void* pHandle,  
    void* pInstanceHandle,  
    const char* lpszRoleDN,  
    char*** lpszMembers  
);
```

パラメータ	I/O	説明
<code>lpApiContext</code>	I	API コンテキスト構造へのポインタ。
<code>pHandle</code>	I	<code>SmDirInit()</code> によって返されたポインタのアドレス。
<code>pInstanceHandle</code>	I	ディレクトリ インスタンス ハンドルへのポインタのアドレス。

パラメータ	I/O	説明
<i>lpszRoleDN</i>	I	メンバが取得されているロールに対する識別名が含まれるバッファ。
<i>lpszMembers</i>	O	ロールのメンバが含まれるバッファ。

### 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

## SmDirGetUserAttr()

SiteMinder ポリシー サーバは、ユーザがカスタム ディレクトリのユーザ属性の値を取得できるように `SmDirGetUserAttr()` を呼び出します。たとえば、ユーザの最後の名前を取得する必要がある場合があります。

SiteMinder は、*lpszAttrData* バッファを解放するために `SmDirFreeString()` を呼び出します。

### 構文

```
int SM_EXTERN SmDirGetUserAttr (
    const Sm_Api_Context_t* lpApiContext,
    void*                    pHandle,
    void*                    pInstanceHandle,
    const char*              lpszUserDN,
    const char*              lpszAttrName,
    char**                   lpszAttrData
);
```

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。
<i>pHandle</i>	I	SmDirInit() によって返されたポインタのアドレス。
<i>pInstanceHandle</i>	I	ユーザ インスタンス ハンドルへのポインタのアドレス。
<i>lpszUserDN</i>	I	ユーザ属性を取得する必要があるユーザ DN が含まれるバッファ。

パラメータ	I/O	説明
<i>lpszAttrName</i>	I	値を取得しているユーザ属性の名前が含まれるバッファ。
<i>lpszAttrData</i>	O	要求された属性の取得された値が含まれるバッファ。 <i>lpszAttrName</i> で指定された属性の値を検索して、 <i>lpszAttrData</i> に値を返します。

### 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

## SmDirGetUserAttrMulti()

SiteMinder ポリシー サーバは、ユーザが単一の属性に対する値の配列を取得できるように、**SmDirGetUserAttrMulti()** を呼び出します。プロバイダは、この関数が呼び出され、属性が単一の値のみを持つ場合に処理を実行する必要があります。

SiteMinder は、*lpszAttrData* バッファを解放するために **SmDirFreeStringArray()** を呼び出します。

### 構文

```
int SM_EXTERN SmDirGetUserAttrMulti (
    const Sm_Api_Context_t* lpApiContext,
    void*                    pHandle,
    void*                    pInstanceHandle,
    const char*              lpszUserDN,
    const char*              lpszAttrName,
    char***                  lpszAttrData
);
```

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。
<i>pHandle</i>	I	<b>SmDirInit()</b> によって返されたポインタのアドレス。
<i>pInstanceHandle</i>	I	ユーザ インスタンス ハンドルへのポインタのアドレス。

---

パラメータ	I/O	説明
<i>lpszUserDN</i>	I	ユーザ属性を取得する必要があるユーザ DN が含まれるバッファ。
<i>lpszAttrName</i>	I	ユーザ属性の名前が含まれるバッファ。
<i>lpszAttrData</i>	O	ユーザ属性の値が含まれるバッファ。 <i>lpszAttrName</i> で指定された属性の値を検索して、 <i>lpszAttrData</i> に値を返します。

---

### 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

## SmDirGetUserClasses()

SiteMinder ポリシー サーバは、ユーザが指定された識別名 (DN) のオブジェクト クラスを取得できるように、`SmDirGetUserClasses()` を呼び出します。この関数は、分散代行管理サービスをサポートするように設計されています。

カスタム ディレクトリは階層的かフラットである場合があります。LDAP ディレクトリのようにディレクトリが階層的である場合、DN は複数のオブジェクト クラスに属する場合があります。SQL データベースのようにディレクトリがフラットな場合、ユーザ DN は `User` または `Group` など単一のクラスに属します。

ユーザのコードは、渡された DN を判断し、適切に処理する必要があります。たとえば、グループの名前が渡された場合、グループであることを判断し、`Group` を返すことができます。

SiteMinder は、`lpszAttrData` バッファを解放するために `SmDirFreeStringArray()` を呼び出します。

### 構文

```
int SM_EXTERN SmDirGetUserClasses (  
    const Sm_Api_Context_t* lpApiContext,  
    void* pHandle,  
    void* pInstanceHandle,  
    const char* lpszUserDN,  
    char*** lpszClasses  
);
```

パラメータ	I/O	説明
<code>lpApiContext</code>	I	API コンテキスト構造へのポインタ。
<code>pHandle</code>	I	<code>SmDirInit()</code> によって返されたポインタのアドレス。
<code>pInstanceHandle</code>	I	ユーザ インスタンス ハンドルへのポインタのアドレス。
<code>lpszUserDN</code>	I	クラスを取得する必要がある識別名が含まれるバッファ。
<code>lpszClasses</code>	O	指定された識別名のクラスが含まれるバッファ。

## 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

この関数を実装しない場合は -1 を返します。

## SmDirGetUserDisabledState()

SiteMinder ポリシー サーバは、ユーザ アカウントが無効かどうかに関する情報を取得するために `SmDirGetUserDisabledState()` を呼び出します。

ユーザ アカウントが無効な場合、`SmDirGetUserDisabledState()` はアカウントがどのように無効にされたかに関する情報を `pnDisabledReason` 内に返します。

ユーザが無効になる理由は、`SmApi.h` で定義される

`Sm_Api_DisabledReason_t` で列挙されます。SiteMinder ポリシー サーバが `SmDirSetUserDisabledState()` を呼び出す場合、無効な理由が設定されます。

カスタム ディレクトリが無効なフラグをサポートしない場合、ユーザが常に有効にされることを示すために以下のコードを使用します。

```
*pnDisabledReason = Sm_Api_Disabled_Enabled;
return 0;
```

## 構文

```
int SM_EXTERN SmDirGetUserDisabledState (
    const Sm_Api_Context_t* lpApiContext,
    void* pHandle,
    void* pInstanceHandle,
    const char* lpszUserDN,
    const char* lpszDisabledAttr,
    Sm_Api_DisabledReason_t* pnDisabledReason
);
```

パラメータ	I/O	説明
<code>lpApiContext</code>	I	API コンテキスト構造へのポインタ。
<code>pHandle</code>	I	<code>SmDirInit()</code> によって返されたポインタのアドレス。

---

パラメータ	I/O	説明
<i>pInstanceHandle</i>	I	ユーザ インスタンス ハンドルへのポインタのアドレス。
<i>pszUserDN</i>	I	無効な状態を取得する必要があるユーザ DN が含まれるバッファ。
<i>lpszDisabledAttr</i>	I	無効なユーザを追跡するために SiteMinder によって使用されるユーザ ディレクトリ属性。無効な状態を取得するためにこの属性を使用します。
<i>pnDisabledReason</i>	O	ユーザの無効な状態が含まれる属性。 <i>lpszDisabledAttr</i> で指定されたユーザ ディレクトリ属性から取得される、ユーザの無効な状態を保存します。

---

## 戻り値

戻り値は、関数がユーザの無効状態を正常に判断したかどうかを示しています。

- 関数がユーザの無効状態を正常に判断する場合は **0** を返します。
- 関数がユーザの無効状態の判断に失敗する場合は **-1** を返します。

無効な状態はパラメータ *pnDisabledReason* に保存され、戻り値によっては示されません。

## 注釈

この関数は以下の状況で呼び出されます。

- ユーザを検索する場合、検索を実行するために `SmDirLookup()` を呼び出した後。
- ユーザの無効な状態を変更するために `SmDirSetUserDisabledState()` を呼び出した後。
- ユーザを認証するために `SmDirAuthenticateUser()` を呼び出した後。無効なフラグがユーザに対して設定される場合、`SiteMinder` はたとえユーザが有効な認証情報を提供したとしてもユーザを認証しません。

## サンプルコード情報

ユーザを管理している場合、ユーザを検索するとき、`SiteMinder` は *lpszDisabledAttr* をチェックすることでそのユーザの無効状態を確認します。この情報を持つディレクトリ属性の名前を示すには、`SiteMinder Administrative UI` の [ユーザ ディレクトリ] ダイアログ ボックスの [ユーザ属性] タブで、[無効フラグ] フィールドに入力します。サンプルを使用するには、このフィールドに `Disabled` と入力します。

ユーザが有効な場合、サンプルコードを実行すると、結果として *pnDisabledReason* に対して `Sm_Api_Disabled_Enabled` の値がもたらされます。`SiteMinder` 管理者がユーザを無効にするために [無効] ボタンを使用している場合、サンプルコードを実行すると、結果として *pnDisabledReason* に対して `Sm_Api_Disabled_AdminDisabled` の値がもたらされます。

## SmDirGetUserGroups()

SiteMinder ポリシー サーバは、ユーザが属するグループを取得できるように、SmDirGetUserGroups を呼び出します。

SiteMinder は、*lpszGroups* 配列を解放するために SmDirFreeStringArray() を呼び出します。

## 構文

```
int SM_EXTERN SmDirGetUserGroups (
    const Sm_Api_Context_t* lpApiContext,
    void* pHandle,
    void* pInstanceHandle,
    const char* lpszUserDN,
    const int bRecursive,
    char*** lpszGroups
);
```

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。
<i>pHandle</i>	I	SmDirInit() によって返されたポインタのアドレス。
<i>pInstanceHandle</i>	I	ユーザ インスタンス ハンドルへのポインタのアドレス。
<i>lpszUserDN</i>	I	ユーザ DN が含まれるバッファ。
<i>bRecursive</i>	I	カスタム ディレクトリが再帰 (たとえば、ネストされたグループ) をサポートするかどうかを指定する値。値 1 は再帰サポートを示し、0 はサポートがないことを示します。  カスタム ディレクトリが再帰をサポートする場合、ユーザを検索するためにグループの階層を下方方向に検索する必要があります。ユーザ DN の値が <b>Bill Collector</b> であるとし、 <b>Bill Collector</b> は <b>AR</b> と呼ばれるグループに属し、 <b>AR</b> は <b>Accounting</b> と呼ばれるグループに属するとし、
<i>lpszGroups</i>	O	<i>lpszUserDN</i> と関連するグループのリスト。

## 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

## SmDirGetUserProperties()

SiteMinder はすべてのユーザ属性または必要なユーザ属性のみの名前を返すことができるように、`SmDirGetUserProperties()` を呼び出します。

渡された名前に対する値を返すように設計されている `SmDirGetUserAttr()` とは異なり、`SmDirGetUserProperties()` は属性名のみを返すように設計されています。

SiteMinder は、`lpszProperties` 配列を解放するために `SmDirFreeStringArray()` を呼び出します。

## 構文

```
int SM_EXTERN SmDirGetUserProperties (
    const Sm_Api_Context_t* lpApiContext,
    void*                    pHandle,
    void*                    pInstanceHandle,
    const char*              lpszUserDN,
    const int                bMandatory,
    char***                  lpszProperties
);
```

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。
<i>pHandle</i>	I	<code>SmDirInit()</code> によって返されたポインタのアドレス。
<i>pInstanceHandle</i>	I	ユーザ インスタンス ハンドルへのポインタのアドレス。
<i>lpszUserDN</i>	I	ユーザ属性の名前を取得する必要があるユーザ DN が含まれるバッファ。

パラメータ	I/O	説明
<i>bMandatory</i>	I	必須属性名のみを返すべきかどうかを指定する値。値 1 は、必須属性名のみが返されることを示します。0 は、属性名がすべて返されることを示します。 ディレクトリの中にはオブジェクトの必須属性およびオプションの属性を区別するインターフェースを持つものがあります。たとえば、Microsoft Active Directory の IAD はこのような区別を実行します。
<i>pszProperties</i>	O	ユーザ属性名のリストが含まれる配列。

### 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

## SmDirGetUserRoles()

SiteMinder ポリシー サーバは、ユーザが属するロールを取得できるように、SmDirGetUserRoles() を呼び出します。

SiteMinder は、*lpzRoles* 配列を解放するために SmDirFreeStringArray() を呼び出します。

### 構文

```
int SM_EXTERN SmDirGetUserRoles (
    const Sm_Api_Context_t* lpApiContext,
    void*                    pHandle,
    void*                    pInstanceHandle,
    const char*              lpzUserDN,
    char***                  lpzRoles
);
```

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。
<i>pHandle</i>	I	SmDirInit() によって返されたポインタのアドレス。

---

パラメータ	I/O	説明
<i>pInstanceHandle</i>	I	ユーザ インスタンス ハンドルへのポインタのアドレス。
<i>lpszUserDN</i>	I	ロールを取得する必要があるユーザ DN が含まれるバッファ。
<i>lpszRoles</i>	O	<i>lpszUserDN</i> でユーザと関連付けられるロールのリスト。

---

### 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

## SmDirInit()

**SmDirInit()** は、**SiteMinder** がカスタム ネームスペースに対するディレクトリ サービス プロバイダを初期化する場合に呼び出されます。

この関数は、他のディレクトリ API 関数が呼び出される前に 1 度呼び出されます。関数は、ポインタのアドレスをディレクトリのハンドルに返します。ハンドルはすべての後続の関数呼び出しで渡されます。

一度管理プロセスが開始すると、**SiteMinder** ポリシー サーバは、ユーザーがカスタム ディレクトリ プロバイダが必要なタスクを最初に実行したときに **SmDirInit()** を呼び出します。

たとえば、ポリシー サーバを開始し、その後即時にカスタム ネームスペースを持つユーザーディレクトリのプロパティを表示した場合、**SiteMinder** ポリシー サーバは **SmDirInit()** を呼び出し、次に **SmDirInitDirInstance()** を呼び出します。

## 構文

```
int SM_EXTERN SmDirInit (
    const Sm_Api_Context_t* lpApiContext,
    void** ppHandle,
    const char* lpszParameter
);
```

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。
<i>ppHandle</i>	O	ディレクトリ API のハンドルへのポインタのアドレス。このパラメータは <b>SmDirInit()</b> への呼び出しで初期化され、すべての後続の関数呼び出しに渡されます。 たとえば、ディレクトリ側に共有ライブラリをロードするために <b>SmDirInit()</b> を使用している場合、その共有ライブラリ内のすべての関数に対する関数ポインタを保存するために <i>ppHandle</i> を使用できます。
<i>lpszParameter</i>	I	[SiteMinder ユーザーディレクトリ] ダイアログボックスの [パラメータ] フィールドで指定される、NULL で終わる文字列。

## 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

## 注釈

`SmDirInit()` が呼び出されるときに `ProviderHandle_t` をインスタンス化します。その後、このハンドルはほぼすべての後続の関数に渡されます。全プロセスを通して同じ値を使用する必要はありません。値は変更することが許可されています。

`SmDirRelease()` が呼び出される場合に `ProviderHandle_t` を解放します。

別の共有ライブラリをロードするために `SmDirInit()` を使用できます。ユーザのデータが含まれるディレクトリのベンダーは、共有ライブラリの構築により実装できるインターフェースを提供する場合があります。共有ライブラリへのパスを [ユーザディレクトリ] ダイアログボックスの [ディレクトリのセットアップ] タブ上の [パラメータ] フィールドに置くことで、`SmDirInit()` を使用してその共有ライブラリをロードできます。 [パラメータ] フィールドに入力された文字列は、`SmDirInit()` および `SmDirInitDirInstance()` の呼び出しで *lpszParameter* に渡されます。

注: [パラメータ] フィールドに入力された文字列も、`SmDirSearch()` および `SmDirSearchCount()` への呼び出しで *lpszSearchRoot* に渡されます。`SmDirInit()` に対するコードが *lpszParameter* を使用する必要があり、検索関数に対するコードが検索ルートが必要とする場合、 [パラメータ] フィールドから文字列を解析する必要があります。

## SmDirInitDirInstance()

SiteMinder は、指定されたディレクトリ インスタンス上のディレクトリ関数を呼び出す前に、このディレクトリ インスタンス初期化関数を呼び出します。この関数は、ユーザにカスタム ディレクトリに接続する機会を提供します。

SiteMinder は、ディレクトリ関数呼び出しを完了した後、SmDirReleaseInstance() を呼び出します。この関数は複数回呼び出されます。SiteMinder は、初期化と解放の呼び出しが実行されるたびにディレクトリ接続を作成および切断することは要求しません。

## 構文

```
int SM_EXTERN SmDirInitDirInstance (
    const Sm_Api_Context_t* lpApiContext,
    void* pHandle,
    void** ppInstanceHandle,
    const char* lpzUniqueKey,
    const char* lpzParameter,
    const char* lpzUsername,
    const char* lpzPassword,
    const int bRequireCredentials,
    const int bSecureConnection,
    const int nSearchResults,
    const int nSearchTimeout
);
```

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。
<i>pHandle</i>	I	SmDirInit() によって返されたポインタのアドレス。
<i>ppInstanceHandle</i>	O	ディレクトリ インスタンスのハンドルを設定するためにこのパラメータを使用します。たとえば、ディレクトリへの接続に関する情報を保持するために <i>ppInstanceHandle</i> を使用できます。 ディレクトリ インスタンス ライフサイクルの最後に、SiteMinder ポリシー サーバは SmDirReleaseInstance() を呼び出し、ユーザがハンドルを削除できるようにディレクトリ インスタンスハンドルを渡します。

パラメータ	I/O	説明
<i>lpszUniqueKey</i>	I	ディレクトリ インスタンス セッションの一意の識別子。この一意のキーは、カスタムディレクトリ オブジェクトのオブジェクト識別子 (OID) を保持します。
<i>lpszParameter</i>	I	[SiteMinder ユーザディレクトリ] ダイアログボックスの [パラメータ] フィールドで指定される、NULL で終わる文字列。値は <code>SmDirInit()</code> の呼び出しのときと同じです。
<i>lpszUsername</i>	I	[SiteMinder ユーザディレクトリ] ダイアログボックスの [認証情報と接続] タブで指定されるユーザ名が含まれる文字列。  <i>bRequireCredentials</i> が <code>true</code> に設定されている場合、 <i>lpszUsername</i> は [管理者認証情報] グループボックス内の [ユーザ名] フィールドの値を保持します。
<i>lpszPassword</i>	I	[SiteMinder ユーザディレクトリ] ダイアログボックスの [認証情報と接続] タブで指定されるパスワードが含まれる文字列。  <i>bRequireCredentials</i> が <code>true</code> に設定されている場合、 <i>lpszPassword</i> は [管理者認証情報] グループボックス内の [パスワード] フィールドの値を保持します。
<i>bRequireCredentials</i>	I	このブール値は、ユーザディレクトリ アクセスに認証情報が必要かどうかを示しています。  SiteMinderAdministrative UI で、[管理者認証情報] グループボックスの [認証情報と接続] タブに [認証情報が必要] チェックボックスがあります。このチェックボックスをオンにすると、 <i>bRequireCredentials</i> が <code>true (1)</code> に設定されます。  <i>bRequireCredentials</i> が <code>true</code> に設定されると、 <i>lpszUsername</i> と <i>lpszPassword</i> は [管理者認証情報] グループボックスの [ユーザ名] と [パスワード] のフィールドで値を保持します。SiteMinder ポリシー サーバは、ディレクトリにアクセスするためにこれらの認証情報を使用します。

パラメータ	I/O	説明
<i>bSecureConnection</i>	I	このブール値は、ユーザディレクトリへのアクセスに SSL 接続が必要かどうかを示します。
<i>nSearchResults</i>	I	このパラメータは、ディレクトリ API による単一の検索の結果セットとして返すレコードの最大数を示します。 [カスタム ネームスペース] グループボックスの [ディレクトリのセットアップ] タブ上の [SiteMinder ユーザディレクトリ] ダイアログボックスに、 [検索の上限] フィールドがあります。 <i>nSearchResults</i> パラメータは、 [検索の上限] フィールドの値を保持します。
<i>nSearchTimeout</i>	I	このパラメータは、ディレクトリ API が結果に対してユーザディレクトリの検索を停止する時間を秒単位で示します。 [カスタム ネームスペース] グループボックスの [ディレクトリのセットアップ] タブ上の [SiteMinder ユーザディレクトリ] ダイアログボックスに、 [最大時間] フィールドがあります。 <i>nSearchResults</i> パラメータは、 [最大時間] フィールドの値を保持します。

### 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

### 注釈

`SmDirInitDirInstance()` が呼び出されるときに `DirHandle_t` をインスタンス化します。ディレクトリ インスタンス ハンドルとユーザ インスタンス ハンドルを区別するために `nTag` を 0 に設定します。

`DirHandle_t` によって参照されたハンドルは、後続のディレクトリ操作関数に渡されます。ハンドル値を変更できます。

`SmDirReleaseInstance()` が呼び出される場合に `DirHandle_t` を解放します。ディレクトリ インスタンス ハンドルとユーザ インスタンス ハンドルを区別するために `nTag` の値を使用します。

詳細情報:

[ディレクトリ上の操作 \(P. 796\)](#)

## SmDirInitUserInstance()

SiteMinder ポリシー サーバは、特定のディレクトリ インスタンス上のディレクトリ エントリ (ユーザ) 操作関数のいずれかを呼び出す前に `SmDirInitUserInstance()` を呼び出します。

`SmDirInitUserInstance()` は複数回呼び出すことができます。

### 構文

```
int SM_EXTERN SmDirInitUserInstance (
    const Sm_Api_Context_t* lpApiContext,
    void*                    pHandle,
    void**                   ppInstanceHandle,
    void*                    pDirInstanceHandle,
    const char*              lpszUserDN
);
```

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。
<i>pHandle</i>	I	<code>SmDirInit()</code> によって返されたポインタのアドレス。
<i>ppInstanceHandle</i>	O	ユーザ インスタンス セッションのハンドルを保持するポインタのアドレス。
<i>pDirInstanceHandle</i>	I	ディレクトリインスタンス セッションのポインタ ハンドルのアドレス。この値は、 <code>SmDirInitDirInstance()</code> から渡されます。
<i>lpszUserDN</i>	I	ユーザ識別名の NULL で終わる文字列が含まれる文字列。

### 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

### 注釈

`SmDirInitUserInstance()` が呼び出される時、`UserHandle_t` をインスタンス化します。ユーザ インスタンス ハンドルとディレクトリ インスタンス ハンドルを区別するために `nTag` を 1 に設定します。

`UserHandle_t` によって参照されたハンドルは、後続のディレクトリ エントリ (ユーザ) 操作関数に渡されます。ハンドル値を変更できます。

`SiteMinder` が関連するディレクトリ エントリ (ユーザ) 操作関数の呼び出しを完了した後、`SmDirReleaseInstance()` を呼び出します。この呼び出しが実行される場合に `UserHandle_t` を解放します。ユーザ インスタンス ハンドルとディレクトリ インスタンス ハンドルを区別するために `nTag` の値を使用します。

## SmDirLookup()

SiteMinder は、ディレクトリのパターンを調べるために `SmDirLookup()` を呼び出します。

検索パターンに対して以下の検索式文法を使用してください。

```
[<class> =] <value>
```

検索パターン形式内

- `<class> = empty-string | user | group` (empty-string implies user & group)
- `<value> = wildcard-string`

コードは、文字列 `user=` で開始する検索パターンを解釈する必要があります。例：

```
...
CStringArray Paths, Classes;
CString szPattern = CString ("user=") + UserDir.m_szUniversalIDAttr
    + CString ("=") + Session.m_szUnivId;
if (m_pDsDir->Lookup (szPattern, Paths, Classes))
....
```

SiteMinder は、`lpszDNs` および `lpszClasses` の配列を解放するために `SmDirFreeStringArray()` を呼び出します。

### 構文

```
int SM_EXTERN SmDirLookup (
    const Sm_Api_Context_t* lpApiContext,
    void* pHandle,
    void* pInstanceHandle,
    const char* lpszPattern,
    char*** lpszDNs,
    char*** lpszClasses
);
```

パラメータ	I/O	説明
<code>lpApiContext</code>	I	API コンテキスト構造へのポインタ。
<code>pHandle</code>	I	<code>SmDirInit()</code> によって返されたポインタのアドレス。
<code>pInstanceHandle</code>	I	ディレクトリ インスタンス ハンドルへのポインタのアドレス。

パラメータ	I/O	説明
<i>lpszPattern</i>	I	ディレクトリで検索するパターンが含まれるバッファ。
<i>lpszDNs</i>	O	パターンに一致するユーザ識別名のリスト。
<i>lpszClasses</i>	O	ユーザ識別名に対応するクラス情報のリスト。

### 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

### 注釈

ユーザ ディレクトリ検索を実行するときこの関数が呼び出されます。

### サンプルコード情報

サンプルコードを試行する場合、SiteMinder Administrative UI でディレクトリ検索用のダイアログボックスに移動し、以下のように検索条件を入力します。

- [検索] ドロップダウンリストボックスで、[ユーザ] を選択します。
- [検索式] フィールドで、Mikel などのサンプルユーザ名のいずれかを正確に入力します。ワイルドカードは使用しないでください。

## SmDirQueryVersion()

SmDirQueryVersion() は、自身のバージョンとディレクトリ機能を調べるためにディレクトリ API にクエリを発行します。サポートされた機能は、Sm\_DirApi\_Capability\_t に列挙されます。

### 構文

```
int SM_EXTERN SmDirQueryVersion (
    const Sm_Api_Context_t* lpApiContext,
    void* pHandle,
    unsigned long* pnCapabilities
);
```

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。
<i>pHandle</i>	I	SmDirInit() によって返されたポインタのアドレス。
<i>pnCapabilities</i>	O	このパラメータはカスタム ディレクトリの機能を渡すために使用されます。カスタム ディレクトリの機能は、SmApi.h で定義される Sm_DirApi_Capability_t に列挙されています。Sm_DirApi_Capability_t の詳細については、「ディレクトリ機能」を参照してください。

### 戻り値

カスタム ライブラリが準拠するディレクトリ API のバージョン番号を返します。現在サポートされるバージョンは Sm\_Api\_Version\_V4 と Sm\_Api\_Version\_V4\_1 です。バージョン定数は SmApi.h で定義されています。

## SmDirRelease()

ポリシー サーバは、カスタム ネームスペースに対するディレクトリ サービス プロバイダが必要なくなったときに **SmDirRelease()** を呼び出します。

### 構文

```
void SM_EXTERN SmDirRelease (  
    const Sm_Api_Context_t* lpApiContext,  
    void*                    pHandle  
);
```

---

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。
<i>pHandle</i>	I	SmDirInit() によって返されたポインタのアドレス。

---

### 注釈

この関数は、ユーザが ポリシー サーバ サービスのいずれかを停止する場合に呼び出されます。

## SmDirReleaseInstance()

SiteMinder ポリシー サーバは、必要に応じてユーザ インスタンス またはディレクトリ インスタンスを解放できるように、**SmDirReleaseInstance()** を呼び出します。

### 構文

```
void SM_EXTERN SmDirReleaseInstance (  
    const Sm_Api_Context_t* lpApiContext,  
    void*                    pHandle,  
    void*                    pInstanceHandle  
);
```

---

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。

---

パラメータ	I/O	説明
<i>pHandle</i>	I	SmDirInit() によって返されたポインタのアドレス。
<i>pInstanceHandle</i>	I	ユーザまたはディレクトリ インスタンス ハンドルへのポインタのアドレス。ユーザ インスタンスを解放するには、 <i>pInstanceHandle</i> にユーザ インスタンス ハンドルが含まれることを確認します。ディレクトリ インスタンスを解放するには、 <i>pInstanceHandle</i> にディレクトリ インスタンス ハンドルが含まれることを確認します。

### 注釈

ユーザまたはディレクトリの、渡されるインスタンス ハンドルタイプで機能するようにリリース コードを記述する必要があります。

プロセス中に、SiteMinder ポリシー サーバはまず SmDirInitDirInstance() を呼び出し、次に SmDirInitUserInstance() を呼び出す場合があります。この場合、SiteMinder ポリシー サーバは、プロセスの最後に 2 回 SmDirReleaseInstance() を呼び出します。最初の呼び出しでは、ユーザ インスタンスを解放できるように、ユーザ インスタンス ハンドルが (*pInstanceHandle* として) 渡されます。2 番目の呼び出しでは、ディレクトリ インスタンスを解放できるように、ディレクトリ インスタンス ハンドルが (*pInstanceHandle* として) 渡されます。

ユーザ ハンドルとディレクトリ ハンドルを区別するために *nTag* の値を以下のように使用します。

- ユーザ ハンドルでは、構造 DirHandle\_t の *nTag* は 0 です。
- ディレクトリ ハンドルでは、構造 UserHandle\_t の *nTag* は 1 です。

## SmDirRemoveEntry()

SiteMinder ポリシー サーバは、ユーザがカスタム ディレクトリからディレクトリ エントリ (ユーザ) を削除できるように、`SmDirRemoveEntry()` を呼び出します。

ディレクトリ エントリの例はユーザ、グループおよびロールです。たとえば、SQL データベースを使用しており、グループを削除する必要がある場合、`SmDirRemoveEntry()` を使用して、データベースのグループ テーブル (およびすべての関連するテーブル) から関連するレコードを削除できます。

LDAP ディレクトリのように、ディレクトリが階層になっている場合、関連するデータを削除するプロセスを階層の別のレベルで処理する必要があります。LDAP 内のオブジェクト クラスなどエントリの属性を確認することが役立つ場合があります。

### 構文

```
int SM_EXTERN SmDirRemoveEntry (
    const Sm_Api_Context_t*    lpApiContext,
    void*                      pHandle,
    void*                      pInstanceHandle,
    const Sm_EntryType_t      nPolicyResolution,
    const char*                lpszEntryDN
);
```

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。
<i>pHandle</i>	I	<code>SmDirInit()</code> によって返されたポインタのアドレス。
<i>pInstanceHandle</i>	I	ディレクトリ インスタンス ハンドルへのポインタのアドレス。

---

パラメータ	I/O	説明
<i>nEntryType</i>	I	エントリのポリシー解決。ポリシー解決は <code>SmApi.h</code> で定義される <code>Sm_PolicyResolution_t</code> に列挙されます。 以下の <code>Sm_PolicyResolution_t</code> エレメントは有効なエントリタイプです。 <ul style="list-style-type: none"><li>■ <code>Sm_PolicyResolution_Unknown</code></li><li>■ <code>Sm_PolicyResolution_User</code></li><li>■ <code>Sm_PolicyResolution_UserGroup</code></li><li>■ <code>Sm_PolicyResolution_UserRole</code></li><li>■ <code>Sm_PolicyResolution_Org</code></li></ul>
<i>lpszEntryDN</i>	I	削除されているエントリに対する識別名が含まれるバッファ。

---

### 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

### 注釈

分散代行管理サービスがグループまたはロールを含むディレクトリ エントリを削除するために使用されるとき、この関数が呼び出されます。

## SmDirRemoveMemberFromGroup()

SiteMinder ポリシー サーバは、既存グループからユーザまたはグループを削除できるように `SmDirRemoveMemberFromGroup()` を呼び出します。

### 構文

```
int SM_EXTERN SmDirRemoveMemberFromGroup (  
    const Sm_Api_Context_t* lpApiContext,  
    void*                    pHandle,  
    void*                    pInstanceHandle,  
    const char*              lpszMemberDN,  
    const char*              lpszGroupDN  
);
```

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。
<i>pHandle</i>	I	SmDirInit() によって返されたポインタのアドレス。
<i>pInstanceHandle</i>	I	ディレクトリ インスタンス ハンドルへのポインタのアドレス。
<i>lpszMemberDN</i>	I	親グループから削除されるユーザまたはグループの識別名が含まれるバッファ。
<i>lpszGroupDN</i>	I	メンバが削除されているグループに対する識別名が含まれるバッファ。

### 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

### 注釈

この関数は、グループからユーザまたはグループを削除するために分散代行管理サービスが使用される場合に呼び出されます。

## SmDirRemoveMemberFromRole()

SiteMinder ポリシー サーバは、割り当てられたロールからユーザまたはグループを削除できるように `SmDirRemoveMemberFromRole()` を呼び出します。

### 構文

```
int SM_EXTERN SmDirRemoveMemberFromRole (
    const Sm_Api_Context_t* lpApiContext,
    void* pHandle,
    void* pInstanceHandle,
    const char* lpszMemberDN,
    const char* lpszRoleDN
);
```

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。
<i>pHandle</i>	I	<code>SmDirInit()</code> によって返されたポインタのアドレス。
<i>pInstanceHandle</i>	I	ディレクトリ インスタンス ハンドルへのポインタのアドレス。
<i>lpszMemberDN</i>	I	割り当てられたロールから削除されるユーザまたはグループの識別名が含まれるバッファ。
<i>lpszRoleDN</i>	I	メンバが削除されているロールに対する識別名が含まれるバッファ。

### 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

### 注釈

この関数は、割り当てられたロールからユーザまたはグループを削除するために分散代行管理サービスが使用される場合に呼び出されます。

## SmDirSearch()

SiteMinder は、検索フィルタ *lpszSearchFilter* で指定された基準で検索するために *SmDirSearch()* を呼び出します。SQL select などのクエリをカスタムディレクトリ上で実行するために *SmDirSearch()* を使用できます。

検索フィルタに加えて、関数 *SmDirSearch()* はディレクトリ検索パラメータを渡します。SiteMinder Administrative UI の [ディレクトリのセットアップ] タブで、以下のフィールドのパラメータを指定できます。

- 最大時間
- 検索の上限

*SmDirSearch()* の *nSearchTimeout* と *nSearchResults* のパラメータは、これらのフィールドに入力された情報を渡します。

SiteMinder は、*lpszDNs* 配列を解放するために *SmDirFreeStringArray()* を呼び出します。

## 構文

```
int SM_EXTERN SmDirSearch (
    const Sm_Api_Context_t* lpApiContext,
    void*                    pHandle,
    void*                    pInstanceHandle,
    char***                  lpszDNs,
    const char*              lpszSearchFilter,
    const char*              lpszSearchRoot,
    const int                nSearchResults,
    const int                nSearchTimeout,
    const int                nSearchScope
);
```

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。
<i>pHandle</i>	I	<i>SmDirInit()</i> によって返されたポインタのアドレス。
<i>pInstanceHandle</i>	I	ディレクトリ インスタンス ハンドルへのポインタのアドレス。
<i>lpszDNs</i>	O	検索の結果として検出されたユーザの識別名。
<i>lpszSearchFilter</i>	I	この検索表現が含まれるバッファ。

パラメータ	I/O	説明
<i>lpszSearchRoot</i>	I	<p>このパラメータは、検索ルート、つまり検索の開始点を保持するように設計されています。階層ディレクトリで、ユーザを認証する場合、<b>SiteMinder</b> はルートから開始し、ツリーを下方方向に移動します。</p> <p><b>SiteMinder Administrative UI</b> の [ディレクトリのセットアップ] タブの [カスタム ネームスペース] グループボックスで、[パラメータ] フィールドに入力された値は <i>lpszSearchRoot</i> を通して渡されます。</p> <p>注: [パラメータ] フィールドに入力された文字列も、<b>SmDirInit()</b> および <b>SmDirInitDirInstance()</b> での使用のために <i>lpszParameter</i> に渡されます。<b>SmDirInit()</b> に対するコードが <i>lpszParameter</i> を使用する必要があり、<b>SmDirSearch()</b> に対するコードが検索ルートを必要とする場合、[パラメータ] フィールドから文字列を解析する必要があります。</p>
<i>nSearchResults</i>	I	<p>このパラメータは、1 回の LDAP ユーザ ディレクトリの検索で返される最大レコード数を保持します。</p>
<i>nSearchTimeout</i>	I	<p>このパラメータは、API がディレクトリで結果を検索する最大時間を秒単位で保持します。</p>
<i>nSearchScope</i>	I	<p>このパラメータは、一致を検出するために API がルート (<i>lpszSearchRoot</i>) のどれだけ下までディレクトリをクエリするかを示します。<i>nSearchScope</i> 内の値に応じて、検索は 1 つ下のレベルまで実行されることも、サブツリー全体に対して実行されることも可能です。デフォルト値は 2 です。</p>

### 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

## SmDirSearchCount()

ディレクトリ検索パラメータで指定される制限に基づいて、*lpszSearchFilter* で指定される検索条件を通して取得されたエントリの数を返すのに関数 **SmDirSearchCount()** を使用します。詳細については、「注釈」セクションを参照してください。

## 構文

```
int SM_EXTERN SmDirSearchCount (
    const Sm_Api_Context_t* lpApiContext,
    void* pHandle,
    void* pInstanceHandle,
    int* pnCount,
    const char* lpszSearchFilter,
    const char* lpszSearchRoot,
    const int nSearchResults,
    const int nSearchTimeout,
    const int nSearchScope
);
```

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。
<i>pHandle</i>	I	<b>SmDirInit()</b> によって返されたポインタのアドレス。
<i>pInstanceHandle</i>	I	ディレクトリ インスタンス ハンドルへのポインタのアドレス。
<i>pnCount</i>	O	検索によって返されたエントリの数
<i>lpszSearchFilter</i>	I	検索表現が含まれるバッファ。
<i>lpszSearchRoot</i>	I	このパラメータは、検索ルート、つまり検索の開始点を保持するように設計されています。階層ディレクトリで、ユーザを認証する場合、 <b>SiteMinder</b> はルートから開始し、ツリーを下方向に移動します。
<i>nSearchResults</i>	I	このパラメータは、1 回の LDAP ユーザ ディレクトリの検索で返される最大レコード数を保持します。
<i>nSearchTimeout</i>	I	このパラメータは、API がディレクトリで結果を検索する最大時間を秒単位で保持します。

---

パラメータ	I/O	説明
<i>nSearchScope</i>	I	このパラメータは、一致を検出するために API がルート ( <i>IpszSearchRoot</i> ) のどれだけ下までディレクトリをクエリするかを示します。 <i>nSearchScope</i> 内の値に応じて、検索は 1 つ下のレベルまで実行されることも、サブツリー全体に対して実行されることも可能です。デフォルト値は 2 です。

---

### 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

### 注釈

SiteMinder ポリシー サーバユーザ インターフェースの [ディレクトリのセットアップ] タブで、パラメータを以下のフィールドで指定できます。

- パラメータ -- ルートを検索します。値はパラメータ *IpszSearchRoot* を通して渡されます。
- 最大時間 -- 秒単位の最大検索時間。値はパラメータ *nSearchTimeout* を通して渡されます。
- 検索の上限 -- 返す結果の最大数。値はパラメータ *nSearchResults* を通して渡されます。

注: [パラメータ] フィールドに入力された文字列も、*SmDirInit()* および *SmDirInitDirInstance()* での使用のために *IpszParameter* に渡されます。*SmDirInit()* に対するコードが *IpszParameter* を使用する必要があり、*SmDirSearchCount()* に対するコードが検索ルートが必要とする場合、[パラメータ] フィールドから文字列を解析する必要があります。

## SmDirSetUserAttr()

SiteMinder ポリシー サーバは、**SmDirSetUserAttr()** を呼び出して、ユーザが SiteMinder を使用してカスタム ディレクトリでユーザ属性を設定できるようにします。たとえば、ユーザの最後の名前を変更する必要がある場合があります。

### 構文

```
int SM_EXTERN SmDirSetUserAttr (  
    const Sm_Api_Context_t*  lpApiContext,  
    void*                    pHandle,  
    void*                    pInstanceHandle,  
    const char*              lpszUserDN,  
    const char*              lpszAttrName,  
    const char*              lpszAttrData  
);
```

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。
<i>pHandle</i>	I	<b>SmDirInit()</b> によって返されたポインタのアドレス。
<i>pInstanceHandle</i>	I	ユーザ インスタンス ハンドルへのポインタのアドレス。
<i>lpszUserDN</i>	I	ユーザ属性を設定する必要があるユーザ DN が含まれるバッファ。
<i>lpszAttrName</i>	I	ユーザ属性の名前が含まれるバッファ。
<i>lpszAttrData</i>	I	ユーザ属性の値が含まれるバッファ。

### 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

## SmDirSetUserAttrMulti()

SiteMinder ポリシー サーバは、ユーザがカスタム ディレクトリで単一の属性に対して値の配列を設定できるように SmDirSetUserAttrMulti() を呼び出します。

### 構文

```
int SM_EXTERN SmDirSetUserAttrMulti (
    const Sm_Api_Context_t* lpApiContext,
    void* pHandle,
    void* pInstanceHandle,
    const char* lpzUserDN,
    const char* lpzAttrName,
    const char** lpzAttrData
);
```

パラメータ	I/O	説明
<i>lpApiContext</i>	I	API コンテキスト構造へのポインタ。
<i>pHandle</i>	I	SmDirInit() によって返されたポインタのアドレス。
<i>pInstanceHandle</i>	I	ユーザ インスタンス ハンドルへのポインタのアドレス。
<i>lpzUserDN</i>	I	ユーザ属性を設定する必要があるユーザ DN が含まれるバッファ。
<i>lpzAttrName</i>	I	ユーザ属性の名前が含まれるバッファ。
<i>lpzAttrData</i>	I	ユーザ属性に対する値が含まれるバッファ。

### 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

## SmDirSetUserDisabledState()

管理者がユーザを無効または有効にするために **SiteMinder** ポリシー サーバユーザ インターフェースを使用する場合、またはパスワード サービスがユーザを無効にする場合、**SiteMinder** ポリシー サーバは **SmDirSetUserDisabledState()** を呼び出します。

この呼び出しによって、カスタム ディレクトリ内の無効なフラグを、**nDisabledReason** を通して渡された無効理由に設定する機会が与えられます。

**SmDirSetUserDisabledState()** を実装する場合、カスタム ディレクトリ内のどのフィールド（または属性）が無効理由を保持するかを確実に指定するようにしてください。**SiteMinder** ポリシー サーバユーザ インターフェースで、[ユーザディレクトリ] ダイアログ ボックス上の [ユーザ属性] タブの [無効フラグ] フィールドにこの属性名を入力します。この属性は **lpszDisabledAttr** を通して渡されます。

## 構文

```
int SM_EXTERN SmDirSetUserDisabledState (
    const Sm_Api_Context_t*      lpApiContext,
    void*                         pHandle,
    void*                         pInstanceHandle,
    const char*                   lpszUserDN,
    const char*                   lpszDisabledAttr,
    const Sm_Api_DisabledReason_t nDisabledReason
);
```

パラメータ	I/O	説明
lpApiContext	I	API コンテキスト構造へのポインタ。
pHandle	I	<b>SmDirInit()</b> によって返されたポインタのアドレス。
pInstanceHandle	I	ユーザ インスタンス ハンドルへのポインタのアドレス。
lpszUserDN	I	無効な状態を変更する必要があるユーザの識別名 (DN) が含まれるバッファ。

パラメータ	I/O	説明
<code>lpszDisabledAttr</code>	I	ユーザの無効な状態を保持するユーザディレクトリ属性。 <code>SiteMinder</code> ポリシー サーバは、ユーザが有効にされるかまたは無効になるかを確認するためにこの属性を確認します。ユーザが無効にされる場合、この属性は特定の理由も保持します。ユーザの無効な状態を変更するためにこの属性を使用します。
<code>nDisabledReason</code>	I	ユーザが無効または有効にされた理由。可能な理由が <code>Sm_Api_DisabledReason_t</code> に列挙されます。 <code>lpszDisabledAttr</code> で指定されるユーザディレクトリ属性にユーザの無効な状態を保存します。

**注:** ユーザ アカウントは複数の理由で無効になる場合があります。必ず無効な理由を保持し、これらのビットを上書きしないようにしてください。たとえば、[ユーザは次回ログイン時にパスワード変更が必要] チェックボックスが選択され、管理者が [無効] をクリックする場合、この `nDisabledReason` パラメータは `Sm_Api_Disabled_PWMustChange` ビットおよび `Sm_Api_Disabled_AdminDisabled` ビットの両方を保持します。ユーザ アカウントが有効なとき、必ず無効なビットをすべてクリアしてください。

### 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

### 注釈

ポリシー サーバユーザ インターフェイスでユーザ アカウントを有効にするには以下を実行します。

1. [ユーザ管理] ダイアログ ボックスに移動します。
2. アカウントを有効にしているユーザを選択します。
3. [有効] をクリックします。

ポリシー サーバユーザ インターフェイスでユーザ アカウントを無効にするには以下を実行します。

1. [ユーザ管理] ダイアログ ボックスに移動します。
2. アカウントを無効にしているユーザを選択します。
3. [無効] をクリックします。

有効なユーザアカウントもパスワードサービスを使用して無効にすることができます。たとえば、以下の条件で、ユーザアカウントを無効にするようにパスワードサービスを設定できます。

- 非アクティブ状態 -- ユーザがログインした後から、特定の期間が経過しました。
- ログイン試行の失敗 -- ユーザは連続したログイン試行失敗の指定数に達しました。
- 期限切れパスワード -- ユーザは必要なパスワード変更の実行に失敗しました。
- 強制的なパスワード変更 -- ユーザは、パスワードの変更を強制されています。 [ユーザ管理] ダイアログ ボックスで、管理者は [ユーザは次回ログイン時にパスワード変更が必要] をオンにしました。

### サンプルコード情報

サンプルを使用する場合、無効な属性を **Disabled** として指定します。この属性はパラメータ `IpszDisabledAttr` を使用して渡されます。 [ユーザディレクトリ] ダイアログ ボックスの [ユーザ属性] タブ上の [無効フラグ] フィールドで「**Disabled**」と入力することで属性を指定できます。

SiteMinder ポリシー サーバユーザ インターフェースの [ユーザ管理] ダイアログ ボックスで、管理者が有効なユーザ、**Mikel** を選択し [無効] をクリックするとします。 サンプルコードでは、ユーザ DN (`IpszUserDN`) は **Mikel** です。また、無効属性 (`IpszDisabledAttr`) は **Disabled** です。無効理由 (`nDisabledReason`) は `Sm_Api_Disabled_AdminDisabled` です。 SiteMinder ポリシー サーバユーザ インターフェースの [ユーザ管理] ダイアログ ボックスで、**Mikel** の [現在の設定] が [ユーザは有効です。] から [無効 - 管理者用です。] へ変更されたことを示します。

## SmDirValidateInstance()

ハンドルが渡されるインスタンスをディレクトリ API が検証できるように、SiteMinder は `SmDirValidateInstance()` を呼び出します。

ディレクトリ インスタンス ハンドルまたはユーザ インスタンス ハンドルのいずれかを `pInstanceHandle` を通して渡すことができます。ユーザのコードはどのハンドルが渡されるかを決定する必要があります。ハンドルが有効かどうかを確認し、対応するレスポンスを返します。

### 構文

```
int SM_EXTERN SmDirValidateInstance (  
    const Sm_Api_Context_t* lpApiContext,  
    void*                    pHandle,  
    void*                    pInstanceHandle  
);
```

パラメータ	I/O	説明
<code>lpApiContext</code>	I	API コンテキスト構造へのポインタ。
<code>pHandle</code>	I	<code>SmDirInit()</code> によって返されたポインタのアドレス。
<code>pInstanceHandle</code>	I	ユーザ インスタンス ハンドルまたはディレクトリ インスタンス ハンドルへのポインタのアドレス。

### 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

## SmDirValidateUserDN()

`lpszUserDN` を通して渡されたユーザ ID 上で必要な検証を実行する場合にこの関数を使用します。

`SmDirValidateUserDN()` を実装する必要がない場合は、サンプルコードで表示されるように 0 を返します。

### 構文

```
int SM_EXTERN SmDirValidateUserDN (  
    const Sm_Api_Context_t* lpApiContext,  
    void* pHandle,  
    void* pInstanceHandle,  
    const char* lpszUserDN  
);
```

パラメータ	I/O	説明
<code>lpApiContext</code>	I	API コンテキスト構造へのポインタ。
<code>pHandle</code>	I	<code>SmDirInit()</code> によって返されたポインタのアドレス。
<code>pInstanceHandle</code>	I	ディレクトリ インスタンス ハンドルへのポインタのアドレス。
<code>lpszUserDN</code>	I	検証される必要があるユーザ DN が含まれるバッファ。

### 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

### 注釈

この関数は、カスタム ディレクトリ内のユーザに対して `IsAuthenticated` を実行するために SiteMinder テストツールを使用する場合に呼び出されます。

## SmDirValidateUsername()

SiteMinder は、`lpszUsername` を通して渡されたユーザ名を検証するためにこの関数を呼び出します。

ユーザ名を解決し、ユーザ ID を `lpszNewUsername` バッファにコピーするためにこの関数を使用できます。`SmDirValidateUsername()` を実装しないことを選択する場合、サンプルで示されるように、`lpszNewUsername` を NULL に設定します。

SiteMinder は、`lpszNewUsername` バッファを解放するために `SmDirFreeString()` を呼び出します。

### 構文

```
int SM_EXTERN SmDirValidateUsername (  
    const Sm_Api_Context_t* lpApiContext,  
    void*                    pHandle,  
    void*                    pInstanceHandle,  
    const char*              lpszUsername,  
    char**                   lpszNewUsername  
);
```

パラメータ	I/O	説明
<code>lpApiContext</code>	I	API コンテキスト構造へのポインタ。
<code>pHandle</code>	I	<code>SmDirInit()</code> によって返されたポインタのアドレス。
<code>pInstanceHandle</code>	I	ディレクトリ インスタンス ハンドルへのポインタのアドレス。
<code>lpszUsername</code>	I	検証される必要があるユーザ名が含まれるバッファ。
<code>lpszNewUsername</code>	O	検証されたユーザ名が含まれるバッファ。

### 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

### 注釈

この関数は、カスタム ディレクトリ内のユーザに対して `IsAuthenticated` を実行するために `SiteMinder` テストツールを使用する場合に呼び出されます。

## SmDirValidateUserPolicyRelationship()

ポリシー オブジェクトの関係を検証するために `SmDirValidateUserPolicyRelationship()` を使用します。ユーザ識別名がポリシー識別名との間に指定された関係があるかどうか判断します。関係は `nPolicyResolution` を通して渡されます。

例 :

- `nPolicyResolution` が `Sm_PolicyResolution_User` である場合、`IpszUserDN` を通して渡されたユーザが `IpszPolicyDN` を通して渡されるユーザと同じかどうか判断します。
- `nPolicyResolution` が `Sm_PolicyResolution_UserGroup` である場合、`IpszUserDN` を通して渡されたユーザが `IpszPolicyDN` を通して渡されたグループのメンバかどうか決定します。

## 構文

```
int SM_EXTERN SmDirValidateUserPolicyRelationship (
    const Sm_Api_Context_t*    lpApiContext,
    void*                      pHandle,
    void*                      pInstanceHandle,
    const char*                lpszUserDN,
    const Sm_PolicyResolution_t nPolicyResolution,
    const int                  bRecursive,
    const char*                lpszPolicyDN,
    const char*                lpszPolicyClass
);
```

パラメータ	I/O	説明
lpApiContext	I	API コンテキスト構造へのポインタ。
pHandle	I	SmDirInit() によって返されたポインタのアドレス。
pInstanceHandle	I	ユーザ インスタンス ハンドルへのポインタのアドレス。
lpszUserDN	I	関係を検証する必要があるユーザ DN が含まれるバッファ。
nPolicyResolution	I	ユーザ識別名とポリシー識別名の関係は、 <b>nPolicyResolution</b> で指定されるものに一致するべきです。特定のポリシー解決は <b>Sm_PolicyResolution_t</b> で列挙されています。 <b>Sm_PolicyResolution_t</b> の詳細については、 <b>Sm_PolicyResolution_t</b> を参照してください。
bRecursive	I	ディレクトリが再帰 (たとえばネストされたグループ) をサポートするかどうか。 <b>nPolicyResolution</b> の値が <b>Sm_PolicyResolution_UserGroup</b> で、ユーザ DN が <b>Bill Collector</b> 、また、ポリシー DN が <b>Accounting</b> であるとします。カスタムディレクトリが再帰をサポートする場合、ユーザを検索するためにグループの階層を下方方向に検索する必要があります。 <b>Bill Collector</b> は AR と呼ばれるグループに存在し、AR はグループ <b>Accounting</b> に存在する場合があります。

パラメータ	I/O	説明
lpszPolicyDN	I	ポリシーにバインドされたユーザ、グループまたはロールなどのオブジェクトの識別名。
lpszPolicyClass	I	lpszPolicyDN で指定されたオブジェクトのクラス。たとえば、クラスは Group の可能性があります。

### 戻り値

成功した場合は 0、失敗した場合は -1 を返します。

## サンプル ディレクトリ アプリケーションで使用される構造

サンプル ディレクトリ API アプリケーションは以下に存在します。

```
<install_path>%sdk%samples%smidirapi%smidirapi.cpp
```

ディレクトリ API にはディレクトリ インスタンス ハンドル、ディレクトリ プロバイダ ハンドルおよびディレクトリ エントリ (ユーザ) インスタンス ハンドルが含まれます。これらのハンドルは、「初期化および解放関数」にリストされた初期化関数から返されます。

サンプル コードは、これらのハンドルを管理するために以下の構造を使用します。

ハンドル タイプ	データ構造
ディレクトリ インスタンス	DirHandle_t
ディレクトリ プロバイダ	ProviderHandle_t
ディレクトリ エントリ (ユーザ) インスタンス	UserHandle_t

## ディレクトリ インスタンス ハンドル

`SmDirInitDirInstance()` が呼び出される場合、サンプルは `DirHandle_t` をインスタンス化します。その後、ハンドルはディレクトリ操作関数に渡されます。

全プロセスを通して同じ値を使用する必要はありません。値は変更することが許可されています。

`DirHandle_t` の定義は以下のとおりです。

```
typedef struct DirHandle_s
{
    char nTag;
    bool bValid;
    char szErrMsg[ERRMSG_SIZE];
    char* pszUniqueKey;
    char* pszParameter;
    char* pszUsername;
    char* pszPassword;
    int bRequireCredentials;
    int bSecureConnection;
    int nSearchResults;
    int nSearchTimeout;
} DirHandle_t;
```

`SmDirReleaseInstance()` が呼び出される時、サンプルは `DirHandle_t` を解放します。ディレクトリ インスタンス ハンドルとユーザ インスタンス ハンドルを区別するために `nTag` の値が使用されます。

詳細情報:

[ハンドルタイプを区別する方法 \(P. 864\)](#)

## ディレクトリプロバイダ ハンドル

サンプルは、SiteMinder ポリシー サーバとディレクトリ API の間の架け橋として機能するためにプロバイダ ハンドル構造 `ProviderHandle_t` を定義します。プロバイダ ハンドルは、SiteMinder がライブラリをロードするときからのデータを SiteMinder ポリシー サーバがシャットダウンするまで保存するために使用できます。

`SmDirInit()` が呼び出される場合、サンプルは `ProviderHandle_t` をインスタンス化します。その後、このハンドルはほぼすべての後続の関数に渡されます。全プロセスを通して同じ値を使用する必要はありません。値は変更することが許可されています。

`SmDirRelease()` が呼び出される時、サンプルは `ProviderHandle_t` を解放します。

`ProviderHandle_t` の例については、サンプルコードで関数 `SmDirInit()` を参照してください。この例に従うことができますが、必須ではありません。`ProviderHandle_t` には、ユーザのパスワードなど、プロセスの始めに設定し、使用したい情報を含めることができます。

## ディレクトリ エントリ(ユーザ) インスタンス ハンドル

サンプルは、`SmDirInitUserInstance()` が呼び出される場合に `UserHandle_t` をインスタンス化します。その後、このハンドルはディレクトリ エントリ (ユーザ) 操作関数に渡されます。これらの関数のリストについては、「ディレクトリ エントリ (ユーザ)」上の「操作」を参照してください。

全プロセスを通して同じ値を使用する必要はありません。値は変更することが許可されています。

`UserHandle_t` の定義は以下のとおりです。

```
typedef struct UserHandle_s
{
    char nTag;
    bool bValid;
    char szErrMsg[ERRMSG_SIZE];
    DirHandle_t* phDir;
    char* pszUserDn;
} UserHandle_t;
```

`SmDirReleaseInstance()` が呼び出されると、サンプルは `UserHandle_t` を解放します。

## ハンドルタイプを区別する方法

`SmDirReleaseInstance()` などの一部の関数は、ディレクトリ インスタンスハンドルまたはディレクトリ エントリ (ユーザ) インスタンスハンドルのいずれかを渡される場合があります。サンプルコードは、ディレクトリ インスタンスハンドルとディレクトリ エントリ (ユーザ) インスタンスハンドルを区別する方法を提供します。

`nTag` が `[DirHandle_t]` および `[UserHandle_t]` の両方の最初のフィールドであることに注目してください。 `SmDirInitDirInstance()` が呼び出される場合、`nTag` は `DirHandle_t` で 0 に設定されます。 `SmDirInitUserInstance` が呼び出される場合、`nTag` は `UserHandle_t` で 1 に設定されます。

いずれかのタイプのハンドルを受け入れる関数が呼び出される場合、どのタイプのハンドルが渡されたかを判断するために `nTag` の値が確認されます。

# 第 13 章：共通データ型および構造

---

このセクションには、以下のトピックが含まれています。

[エクスポートされた列挙](#) (P. 865)

[共通構造](#) (P. 873)

## エクスポートされた列挙

SmApi.h 内の以下の列挙が複数の SiteMinder API によって使用されます。

- [Sm\\_Api\\_DisabledReason\\_t](#) (P. 865)
- [Sm\\_Api\\_Reason\\_t](#) (P. 868)
- [Sm\\_PolicyResolution\\_t](#) (P. 870)

### Sm\_Api\_DisabledReason\_t

ユーザアカウントが無効になる理由を列挙します。

以下の API が Sm\_Api\_DisabledReason\_t を使用します。

- ポリシー管理 API
- ディレクトリ API

**注:** ユーザアカウントは複数の理由で無効になる場合があります。たとえば、[ユーザは次回ログイン時にパスワード変更が必要] チェックボックスがオンになっていて、管理者が [無効] をクリックする場合、*nDisabledReason* パラメータは Sm\_Api\_Disabled\_PWMustChange ビットおよび Sm\_Api\_Disabled\_AdminDisabled ビットの両方を保持します。

Sm\_Api\_DisabledReason\_t を使用する例については、Sm\_PolicyApi\_SetDisabledUserState() 下の例を参照してください。

以下の表は、無効理由に対して設定できるビットについて説明しています。表に続けて、理由タイプによって組織された各理由の簡潔な説明が記されています。

無効理由	タイプ	値
Sm_Api_Disabled_DisabledMask	マスク	0x00ffffff
Sm_Api_Disabled_Enabled	マスク	0
Sm_Api_Disabled_AdminDisabled	ビット	0x00000001
Sm_Api_Disabled_MaxLoginFail	ビット	0x00000002
Sm_Api_Disabled_Inactivity	ビット	0x00000004
Sm_Api_Disabled_PWExpired	ビット	0x00000008
Sm_Api_Disabled_DirNativeDisabled	ビット	0x00000010
Sm_Api_Disabled_PWMustChange	修飾子	0x01000000

### 無効マスク

- Sm\_Api\_Disabled\_DisabledMask

以下の 2 つのケースを区別するために無効ビット マスクが使用されます。

- 無効ビットが設定される場合、結果の 16 進数値は 0x00 で開始します。
- ユーザがパスワード変更を強制される場合、結果の 16 進数値は 0x01 で開始します。

- Sm\_Api\_Disabled\_Enabled

ユーザアカウントが有効な場合、値は 0 以外の値を持つ他のケースに相反して 0 となります。

## 無効ビット

ユーザアカウントは以下の理由のいずれかによって無効になる場合があります。

- **Sm\_Api\_Disabled\_AdminDisabled**  
管理者によって無効になりました。
- **Sm\_Api\_Disabled\_MaxLoginFail**  
ログイン失敗の最大数に基づいて無効になりました。
- **Sm\_Api\_Disabled\_Inactivity**  
一定期間の非アクティブ状態に基づいて無効になりました。
- **Sm\_Api\_Disabled\_PWExpired**  
パスワード期限切れに基づいて無効になりました。
- **Sm\_Api\_Disabled\_DirNativeDisabled**  
ユーザリポジトリによって無効になり、**SiteMinder** で有効になりません。たとえば、**accountExpires** 時間が経過したためにアクティブディレクトリユーザオブジェクトが期限切れになる場合、**SiteMinder** はこの無効理由を返します。

## 修飾子

- **Sm\_Api\_Disabled\_PWMustChange**  
次のログイン中にパスワードを変更することをユーザに強制します。ユーザがパスワード変更を強制される場合、結果の16進数値は0x00ではなく0x01で開始します。

## Sm\_Api\_Reason\_t

認証失敗やセッションログアウトなどのアクセス イベントの理由を列挙します。

認証方式が認証に失敗する場合、理由ステータス コードを返す場合があります。 認証関数から返されたステータス コードは **SM\_MAKEAUTH\_STATUSVALUE** マクロを使用して構築されます (10 ページの「戻り値」を参照してください)。

ポリシー サーバはエージェントに理由を送信します。 **SiteMinder Web** エージェントは、**Web** アプリケーションがロジックで使用できるように理由を明示します。

以下の API が **Sm\_Api\_Reason\_t** を使用します。

- 認証 API
- イベント API
- ディレクトリ API

以下の表は、アクセス イベントに対して設定できるビットについて説明しています。

注: 0 から 31999 の値は **SiteMinder** での使用のために予約されています。 32000 から 32767 の値は、ユーザ定義の理由に対して使用可能です。

認証理由	値
Sm_Api_Reason_None	0
Sm_Api_Reason_PwMustChange	1
Sm_Api_Reason_InvalidSession	2
Sm_Api_Reason_RevokedSession	3
Sm_Api_Reason_ExpiredSession	4
Sm_Api_Reason_AuthLevelTooLow	5
Sm_Api_Reason_UnknownUser	6
Sm_Api_Reason_UserDisabled	7
Sm_Api_Reason_InvalidSessionId	8
Sm_Api_Reason_InvalidSessionIp	9

認証理由	値
Sm_Api_Reason_CertificateRevoked	10
Sm_Api_Reason_CRLOutOfDate	11
Sm_Api_Reason_CertRevokedKeyCompromised	12
Sm_Api_Reason_CertRevokedAffiliationChange	13
Sm_Api_Reason_CertOnHold	14
Sm_Api_Reason_TokenCardChallenge	15
Sm_Api_Reason_ImpersonatedUserNotInDir	16
Sm_Api_Reason_Anonymous	17
Sm_Api_Reason_PwWillExpire	18
Sm_Api_Reason_PwExpired	19
Sm_Api_Reason_ImmedPWChangeRequired	20
Sm_Api_Reason_PWChangeFailed	21
Sm_Api_Reason_BadPWChange	22
Sm_Api_Reason_PWChangeAccepted	23
Sm_Api_Reason_ExcessiveFailedLoginAttempts	24
Sm_Api_Reason_AccountInactivity	25
Sm_Api_Reason_NoRedirectConfigured	26
Sm_Api_Reason_ErrorMessageIsRedirect	27
Sm_Api_Reason_Next_Tokencode	28
Sm_Api_Reason_New_PIN_Select	29
Sm_Api_Reason_New_PIN_Sys_Tokencode	30
Sm_Api_Reason_New_User_PIN_Tokencode	31
Sm_Api_Reason_New_PIN_Accepted	32
Sm_Api_Reason_Guest	33
Sm_Api_Reason_PWSelfChange	34
Sm_Api_Reason_ServerException	35
Sm_Api_Reason_UnknownScheme	36
Sm_Api_Reason_UnsupportedScheme	37

認証理由	値
Sm_Api_Reason_Misconfigured	38
Sm_Api_Reason_BufferOverflow	39
Sm_Api_Reason_SetPersistentSessionFailed	40
Sm_Api_Reason_UserLogout	41
Sm_Api_Reason_IdleSession	42
Sm_Api_Reason_PolicyServerEnforcedTimeout	43
Sm_Api_Reason_PolicyServerEnforcedIdle	44
Sm_Api_Reason_ImpersonationNotAllowed	45
Sm_Api_Reason_ImpersonationNotAllowedUser	46
Sm_Api_Reason_FederationNoLoginID	47
Sm_Api_Reason_FederationUserNotInDir	48
Sm_Api_Reason_FederationInvalidMessage	49
Sm_Api_Reason_FederationUnacceptedMessage	50

## Sm\_PolicyResolution\_t

SmApi.h で定義された Sm\_PolicyResolution\_t は、ユーザなどの 1 つのポリシー オブジェクトがグループなど別のポリシー オブジェクトにどのように関連付けられるかを説明する値を列挙します。

以下の API が列挙 Sm\_PolicyResolution\_t を使用します。

- ポリシー管理 API
- ディレクトリ API

以下の表は、サポートされているポリシー解決を示しています。表の後に各解決の簡単な説明が続きます。

ポリシー解決	値
Sm_PolicyResolution_Unknown	0
Sm_PolicyResolution_User	1
Sm_PolicyResolution_UserGroup	2

---

ポリシー解決	値
Sm_PolicyResolution_UserProp	3
Sm_PolicyResolution_UserRole	4
Sm_PolicyResolution_Org	5
Sm_PolicyResolution_Query	6
Sm_PolicyResolution_All	7
Sm_PolicyResolution_GroupProp	8
Sm_PolicyResolution_OrgProp	9
Sm_PolicyResolution_DnProp	10

---

注: 別のポリシー解決 (`Sm_PolicyResolution_Max`) は内部使用のみを対象としています。

- `Sm_PolicyResolution_Unknown`。ポリシー オブジェクトは不明です。
- `Sm_PolicyResolution_User`。ポリシー オブジェクトは指定されたユーザー オブジェクトです。
- `Sm_PolicyResolution_UserGroup`。ポリシー オブジェクトは、指定されたユーザー グループ オブジェクトのメンバ (直接、または別のグループを通して間接的に) です。
- `Sm_PolicyResolution_UserProp`。ポリシー オブジェクトは指定されたフィルタ (ユーザー スcope) に一致します。
- `Sm_PolicyResolution_UserRole`。ポリシー オブジェクトは指定されたロールを持っています。
- `Sm_PolicyResolution_Org`。ポリシー オブジェクトは、(組織および組織単位をサポートする) 指定された組織 オブジェクトの (直接、または別のグループを通じた間接的な) メンバです。
- `Sm_PolicyResolution_Query`。ポリシー オブジェクトはディレクトリ固有のクエリの結果セットに含まれています。
- `Sm_PolicyResolution_All`。ポリシー オブジェクトは指定されたディレクトリに置かれています。
- `Sm_PolicyResolution_GroupProp`。ポリシー オブジェクトは、特定のフィルタに一致する DN のような、「グループ」に関連するポリシーです。
- `Sm_PolicyResolution_OrgProp`。ポリシー オブジェクトは、特定のフィルタに一致する DN のような、「組織」に関連するポリシーです。
- `Sm_PolicyResolution_DnProp`。ポリシー オブジェクトは、特定のフィルタに一致する DN のような、「DN」に関連するポリシーです。

詳細情報:

[SmDirValidateUserPolicyRelationship\(\)](#) (P. 858)

## 共通構造

Sm\_Api\_Context\_t 構造は複数の SiteMinder API によって使用されます。

この構造は、SiteMinder ログ記録ユーティリティ、トレースユーティリティおよびエラーユーティリティに関数ポインタを与えます。

以下の API が Sm\_Api\_Context\_t を使用します。

- 認証 API
- 許可 API
- トンネル サービス API
- DMS ワークフロー API
- ディレクトリ API

詳細情報:

[Sm\\_Api\\_Context\\_t](#) (P. 662)

## 第 14 章: イベント ログ形式

---

### アクセス イベント

アクセス イベントはユーザ関連アクティビティを示します。これらのイベントは、認証、許可、および管理アクティビティのコンテキストで呼び出されます。

テキスト ログ内のアクセス イベントに対する形式は、イベント カテゴリ ID に依存しています。

## 認証形式と許可形式

イベント カテゴリ ID が認証 (SmLogAccessCat\_Auth) または許可 (SmLogAccessCat\_Az) である場合、形式は以下のとおりです。

```
lpszEvent lpszHostName lpszTimeString "szClientIp szUserName" "szAgentName szAction  
szResource" [szTransactionId] [nReason] szStatusMsg
```

上記の形式例内のエレメントは以下のように説明されます。

- lpszEvent。 アクセス イベントの名前 (タイプ)。
  - SmLogAccessEvent\_AuthAccept : lpszEvent = "AuthAccept"
  - SmLogAccessEvent\_AuthReject : lpszEvent = "AuthReject"
  - SmLogAccessEvent\_AuthAttempt : lpszEvent = "AuthAttempt"
  - SmLogAccessEvent\_AuthChallenge : lpszEvent = "AuthChallenge"
  - SmLogAccessEvent\_AzAccept : lpszEvent = "AzAccept"
  - SmLogAccessEvent\_AzReject : lpszEvent = "AzReject"
  - SmLogAccessEvent\_AdminLogin : lpszEvent = "AdminLogin"
  - SmLogAccessEvent\_AdminLogout : lpszEvent = "AdminLogout"
  - SmLogAccessEvent\_AdminReject : lpszEvent = "AdminReject"
  - SmLogAccessEvent\_AuthLogout : lpszEvent = "AuthLogout"
  - SmLogAccessEvent\_ValidateAccept : lpszEvent = "ValidateAccept"
  - SmLogAccessEvent\_ValidateReject : lpszEvent = "ValidateReject"
- lpszHostName。 ホストの名前。
- lpszTimeString。 イベント発生の日付スタンプの形式は、  
[<date>/<month>/<year>:<hour>:<minute>:<second>  
<difference from GMT>] です。 例 : [27/Jun/2000:11:27:29 -0500]
- szClientIp。 クライアント マシンの IP アドレス。
- szUserName。 ユーザの名前。
- szAgentName。 エージェントの名前。
- szAction。 リソースと関連付けられたアクション。
- szResource。 アクセスされたリソース。
- [szTransactionId]。 idletime=<value> が含まれる文字列。

- [nReason]。 イベントと関連する理由。 理由は SmApi.h 内の Sm\_Api\_Reason\_t で列挙されます。
- szStatusMsg。 イベントに関連するメッセージ。 以下の表内に示されるように、メッセージはイベントタイプに依存しています。

イベント	szStatusMsg のロール
SmLogAccessEvent_AdminLogin	認証方式によって返された UserMsg を保持します。
SmLogAccessEvent_AdminReject	認証方式によって返された ErrorMsg を保持します。
SmLogAccessEvent_AuthAccept	UserMsg を保持します。
SmLogAccessEvent_AuthReject	UserMsg と ErrorMsg の連結文字列を保持します。
SmLogAccessEvent_AuthAttempt	UserMsg と ErrorMsg の連結文字列を保持します。
SmLogAccessEvent_AuthChallenge	UserMsg を保持します。
SmLogAccessEvent_ValidateAccept	空の文字列です。
SmLogAccessEvent_ValidateReject	有効な拒否に対する理由を含むエラーメッセージを保持します。 例： 「無効なセッション トークン」 ("Invalid session token") 「無効なセッション ID」 ("Invalid session id") 「無効なセッション IP」 (Invalid session ip) 「無効なユーザ DN」 ("Invalid user DN") 「セッションは期限切れになりました」 ("Session has expired") 「無効なキーが使用されています」 ("Invalid key in use") 「無効なエラー状態」 ("Invalid error status")
SmLogAccessEvent_AuthLogout	空の文字列。

イベント	szStatusMsg のロール
SmLogAccessEvent_AzAccept	空の文字列。
SmLogAccessEvent_AzReject	az 拒否のタイプに応じて、拒否の理由について説明する文字列。 例 「アフィリエイト エージェントに対する セッションの種類が無効です」 「セッションタイプが無効です」 「このセキュリティ レベルに対して セッションが許可されませんでした」

例：

```
AuthAccept testbox [27/Jun/2000:11:27:29 -0500] "190.158.4.90
uid=scarter,ou=people,o=airius.com" "testagent GET /test/index.html"
[idletime=3600;maxtime=7200;authlevel=5;] [0]
```

この例では以下が当てはまります。

- lpszEvent は AuthAccept です
- lpszHostName は testbox です
- lpszTimeString は [27/Jun/2000:11:27:29 -0500] です
- szClientIp は 190.158.4.90 です
- szUserName は、uid=scarter、ou=people、o=airius.com です
- szAgentName は testagent です
- szAction は GET です
- szResource は /test/index.html です
- [szTransactionId] は [idletime=3600;maxtime=7200;authlevel=5;]
- [nReason] は [0] です
- szStatusMsg は指定されません。

## 管理形式

イベント カテゴリ ID が SmLogAccessCat\_Admin である場合、形式は以下のとおりです。

```
lpszEvent lpszHostName lpszTimeString "szClientIp szUserName"      szStatusMsg
```

上記の形式例内のエレメントは以下のように説明されます。

- lpszEvent はアクセス イベントの名前 (タイプ) です。  
 SmLogAccessEvent\_AdminLogin:lpszEvent = "AdminLogin"  
 SmLogAccessEvent\_AdminLogout:lpszEvent = "AdminLogout"  
 SmLogAccessEvent\_AdminReject:lpszEvent = "AdminReject"
- lpszHostName はホストの名前です。
- lpszTimeString は、[<date>/<month>/<year>:<hour>:<minute>:<second>  
 <difference from GMT>] の形式の、イベント発生のタイムスタンプです。  
 例：  
 [27/Jun/2000:11:27:29 -0500]
- szClientIp はクライアント マシンの IP アドレスです。
- szUserName はユーザの名前です。
- szStatusMsg はイベントと関連付けられたメッセージです。以下の表に示されるように、メッセージはイベントタイプに依存しています。

イベント	szStatusMsg のロール
SmLogAccessEvent_AdminLogin	認証方式によって返された UserMsg を保持します。
SmLogAccessEvent_AdminReject	認証方式によって返された ErrorMsg を保持します。

例 :

```
AdminLogin textbox [27/Jun/2000:11:26:50 -0500]
    "190.158.4.90 siteminder"
```

この例では以下が当てはまります。

- lpszEvent は AdminLogin です
- lpszHostName は textbox です
- lpszTimeString is [27/Jun/2000:11:26:50 -0500]
- szClientIp は 190.158.4.90 です
- szUserName は siteminder です
- szStatusMsg は指定されません。

## オブジェクト イベント

オブジェクト イベントは、

- SiteMinder オブジェクトが作成、更新、または削除された場合に呼び出されます。
- アプリケーションまたはユーザはオブジェクトストアにログインします。

オブジェクトのイベントは、認証、SiteMinder オブジェクト変更および管理アクティビティのコンテキストで呼び出されます。

テキストログ内のオブジェクトイベントに対する形式は、オブジェクトイベントカテゴリ ID に依存しています。

## AdminChange 形式

管理者がオブジェクトを追加、更新、または削除する場合に AdminChange イベントが発生します。形式は以下のとおりです。

```
AdminChange <Hostname> <Time String> "-" <Username> " <Event> <Category>
'<ObjectName>'
```

上記の形式例内のエレメントは以下のように説明されます。

- <HostName> はホストの名前です。
- <Time String> は、[<date>/<month>/<year>:<hour>:<minute>:<second> <difference from GMT>] の形式の、イベント発生のタイムスタンプです。  
例：

```
[27/Jun/2000:11:27:29 -0500]
```

- <UserName> はイベントを生成したユーザの名前です。
- <Event> はオブジェクト イベントの名前です。以下のようになります。

```
作成  
削除  
Update  
UpdateField
```

- <Category> は、たとえば Rule または UserDirectory の、イベントのターゲットとなるオブジェクトです。
- <ObjectName> はオブジェクトのユーザ定義の名前です。一部のオブジェクトカテゴリ (RootConfig など) には ObjectName が関連付けられていません。

以下に、管理者 JLewis がルール MyNewRule を作成した場合にログ記録された AdminChange イベント形式の例を示します。

```
AdminChange MyHost [20/Jul/2001:10:26:15 -0500] "-" JLewis" Create Rule 'MyNewRule'
```

### 管理コマンド形式

オブジェクト カテゴリ ID が管理コマンド (SmLogObjCat\_ManagementCommand) である場合、形式は以下のとおりです。

```
ManagementCommand <Hostname> <Time String> "- <Username> " <Event> '<Description>'
```

上記の形式例内のエレメントは以下のように説明されます。

- <HostName> はホストの名前です。
- <Time String> は、[<date>/<month>/<year>:<hour>:<minute>:<second> <difference from GMT>] の形式の、イベント発生のタイムスタンプです。  
例：  
[27/Jun/2000:11:27:29 -0500]
- <UserName> はイベントを生成したユーザの名前です。
- <Event> は管理コマンド イベントの名前です。以下のようになります。

FlushAll	ChangeDynamicKeys
FlushUser	ChangePersistentKey
FlushAllUsers	ChangeDisabledUserState
FlushAllRealms	ChangeUserPassword
- <Description> は、クリア ユーザ コマンド、無効ユーザ状態変更、およびパスワード変更などのユーザが関連する管理コマンドに対するユーザ DN です。

以下は、管理者 JLewis が BRoy に対するユーザ キャッシュをクリアした場合にログ記録された、管理コマンド イベント形式の例を示しています。

```
ManagementCommand MyHost [20/Jul/2001:13:26:23 -0500]
"- JLewis" FlushUser 'uid=BRoy,ou=HR,o=security.com'
```

## EMS イベント

EMS イベントには以下が含まれます。

- 以下のいずれかのオブジェクトの作成、削除、または修正。
  - ユーザ
  - 組織
  - ユーザ ロール
  - 管理ロール
  - リソース
  - トップ オブジェクト クラスの一般的なオブジェクト
- ユーザの有効化または無効化
- ユーザ ロールを割り当てまたは削除すること
- ユーザ パスワードの変更
- 管理者としてのログインまたはログアウト
- 認証の失敗
- EMS サーバセッション タイムアウト

## EMS ログ形式

SiteMinder は、以下の形式を使用して、テキスト ファイルに EMS イベントをログ記録します。形式例で、リテラル文字列は**太字**タイプで表示されます。

```
lpszTimeString: Category lpszCat (nCategoryId),  
                Event lpszEvent (nEventId)  
                Username szUserName, SessionId szSessionId  
                DirectoryName szDirName  
                ObjectName szObjName, ObjectClass szObjClass,  
                ObjectPath szObjPath  
                Organization szOrgName, Role szRoleName  
                Description: szFieldDesc  
                Status: szStatusMsg
```

前述の形式例内のエレメントは以下のように説明されます。

- `lpszTimeString` はイベント発生のタイムスタンプで、形式は [`<date>/<month>/<year>:<hour>:<minute>:<second><difference from GMT>`] です。例：  
[27/Jun/2000:11:27:29 -0500]
- `nCategoryId` にはカテゴリ ID が含まれ、`lpszCat` には対応するカテゴリ名が含まれます。
- `nEventId` には EMS イベント ID が含まれ、`lpszEvent` には対応するイベント名が含まれます。

斜体で表示された残りのフィールドは構造 `SmLog_EMS_t` のメンバです。

カテゴリ ID ( <code>nCategoryId</code> )	カテゴリ ( <code>lpszCat</code> )
<code>SmLogEmsCat_DirectoryUser</code>	"User"
<code>SmLogEmsCat_DirectoryAdmin</code>	"Admin"
<code>SmLogEmsCat_DirectorySession</code>	"Session"

イベント ID ( <code>nEventId</code> )	イベント ( <code>lpszEvent</code> )
<code>SmLogEmsEvent_CreateUser</code>	"CreateUser"
<code>SmLogEmsEvent_DeleteUser</code>	"DeleteUser"
<code>SmLogEmsEvent_ModifyUser</code>	"ModifyUser"
<code>SmLogEmsEvent_AssignUserRole</code>	"AssignUserRole"(sic)
<code>SmLogEmsEvent_RemoveUserRole</code>	"RemoveUserRole"
<code>SmLogEmsEvent_EnableUser</code>	"EnableUser"
<code>SmLogEmsEvent_DisableUser</code>	"DisableUser"
<code>SmLogEmsEvent_CreateOrg</code>	"CreateOrganization"
<code>SmLogEmsEvent_DeleteOrg</code>	"DeleteOrganization"
<code>SmLogEmsEvent_ModifyOrg</code>	"ModifyOrganization"
<code>SmLogEmsEvent_CreateRole</code>	"CreateRole"
<code>SmLogEmsEvent_DeleteRole</code>	"DeleteRole"
<code>SmLogEmsEvent_ModifyRole</code>	"ModifyRole"

イベント ID (nEventId)	イベント (IpszEvent)
SmLogEmsEvent_CreateResource	"CreateResource"
SmLogEmsEvent_DeleteResource	"DeleteResource"
SmLogEmsEvent_ModifyResource	"ModifyResource"
SmLogEmsEvent_AssignResourceRole	"AssignResourceRole"
SmLogEmsEvent_RemoveResourceRole	"RemoveResourceRole"
SmLogEmsEvent_Login	"Login"
SmLogEmsEvent_Logout	"Logout"
SmLogEmsEvent_LoginFail	"LoginFail"
SmLogEmsEvent_SessionTimeout	"SessionTimeout"
SmLogEmsEvent_AuthFail	"AuthFail"
SmLogEmsEvent_PasswordModify	"ChangePassword"
SmLogEmsEvent_CreateAdminRole	"CreateAdminRole"
SmLogEmsEvent_DeleteAdminRole	"DeleteAdminRole"
SmLogEmsEvent_ModifyAdminRole	"ModifyAdminRole"
SmLogEmsEvent_AssignAdminRole	"AssignAdminRole"
SmLogEmsEvent_RemoveAdminRole	"RemoveAdminRole"
SmLogEmsEvent_AddManagedOrg	"AddManagedOrg"
SmLogEmsEvent_RemoveManagedOrg	"RemoveManagedOrg"
SmLogEmsEvent_CreateObject	"CreateObject"
SmLogEmsEvent_DeleteObject	"DeleteObject"
SmLogEmsEvent_ModifyObject	"ModifyObject"



# 付録 A: SAML 2.0 プロパティ参照

---

このセクションには、以下のトピックが含まれています。

[SAML 2.0 プロパティについて \(P. 885\)](#)

[SAML 2.0 プロパティ参照 \(P. 886\)](#)

## SAML 2.0 プロパティについて

このリファレンスには、`Sm_PolicyApi_SAMLProviderProp_t` 構造体と関連付けられた SAML 2.0 メタデータ プロパティが含まれます。プロパティはファイル `SmPolicyApi45.h` で定義されています。

プロパティは、以下の 1 つ以上の SAML 2.0 オブジェクトに適用されます。

- SAML 2.0 アフィリエーション。

SAML 2.0 アフィリエーションは、プリンシパル用の一意の名前 ID の単一のフェデレートされたネームスペースを共有する一連のエンティティです。

- SAML 2.0 認証方式およびその関連のアイデンティティ プロバイダ定義。

アイデンティティ プロバイダは、アイデンティティ プロバイダおよびその関連の SAML 2.0 認証方式を設定したサービス プロバイダ用の SAML アサーションを作成します。

- サービス プロバイダ。

サービス プロバイダは、フェデレーション内のプリンシパルにサービス (アプリケーションおよびその他のリソースへのアクセスなど) を提供します。

サービス プロバイダは、SAML 2.0 認証方式を使用して、SAML アサーションの情報に基づいてプリンシパルを検証します。アサーションは認証方式と関連付けられたアイデンティティ プロバイダによって提供されます。

## SAML 2.0 プロパティ参照

このリファレンスには、`Sm_PolicyApi_SAMLProviderProp_t` 構造体と関連付けられた SAML 2.0 メタデータ プロパティが含まれます。プロパティはファイル `SmPolicyApi45.h` で定義されています。

プロパティは、以下の 1 つ以上の SAML 2.0 オブジェクトに適用されます。

- SAML 2.0 アフィリエーション。

SAML 2.0 アフィリエーションは、プリンシパル用の一意の名前 ID の単一のフェデレートされたネームスペースを共有する一連のエンティティです。

- SAML 2.0 認証方式およびその関連のアイデンティティ プロバイダ定義。

アイデンティティ プロバイダは、アイデンティティ プロバイダおよびその関連の SAML 2.0 認証方式を設定したサービス プロバイダ用の SAML アサーションを作成します。

- サービス プロバイダ。

サービス プロバイダは、フェデレーション内のプリンシパルにサービス（アプリケーションおよびその他のリソースへのアクセスなど）を提供します。

サービス プロバイダは、SAML 2.0 認証方式を使用して、SAML アサーションの情報に基づいてプリンシパルを検証します。アサーションは認証方式と関連付けられたアイデンティティ プロバイダによって提供されます。

プロパティ参照を読むときは、以下の事項に注意してください。

- 別途指定がないかぎり、文字列データ型のプロパティの最大文字数は以下のとおりです。
  - URI と URL は 1,024 文字未満である必要があります
  - 他のすべての文字列は 255 文字以下にする必要があります
- SAML\_TRUE または SAML\_FALSE のいずれかとしてブール値を指定します。

### SAML\_AFFILIATION

---

タイプ

文字列

---

適用先	SAML 2.0 認証、サービス プロバイダ
必須	いいえ
デフォルト	なし

### 説明

このオブジェクトと関連付けられる SAML 2.0 アフィリエーション。

サービス プロバイダは、アフィリエーションにわたって名前 ID プロパティを共有します。アイデンティティ プロバイダは、アフィリエーションにわたってユーザの不明瞭解消を共有します。

サービス プロバイダまたはアイデンティティ プロバイダは、1つの SAML 2.0 アフィリエーションのみに属することができます。

SAML アフィリエーションが指定される場合、名前 ID プロパティ（たとえば SAML\_SP\_NAMEID\_FORMAT）は使用されません。SiteMinder は、指定されたアフィリエーション内の名前 ID 情報を使用します。

注: アイデンティティ プロバイダはその関連する SAML 2.0 認証方式を通してアフィリエーションに割り当てられます。

### SAML\_AUDIENCE

タイプ	文字列
適用先	SAML 2.0 認証、サービス プロバイダ
必須	はい
デフォルト	なし

### 説明

サービス プロバイダの予期されるオーディエンスの URI。サービス プロバイダによって予期されるオーディエンスは、アサーションで指定されたオーディエンスに一致する必要があります。

オーディエンスは認証要求でも送信される場合があります。

**SAML\_DESCRIPTION**

タイプ	文字列
適用先	SAML 2.0 アフィリエーション、SAML 2.0 認証、サービス プロバイダ
必須	いいえ
デフォルト	なし

**説明**

アフィリエーション、認証方式またはサービス プロバイダ オブジェクトの簡単な説明。

**SAML\_DISABLE\_SIGNATURE\_PROCESSING**

タイプ	ブール値
適用先	SAML 2.0 認証、サービス プロバイダ
必須	いいえ
デフォルト	SAML_FALSE

**説明**

署名を含めた、すべての署名検証を無効にするかどうかを指定します。

プロバイダの初期設定中、およびデバッグ中に署名検証を無効にすると役立つ場合があります。通常のランタイム中に、このプロパティは SAML\_FALSE (署名処理が有効にされています) に設定される必要があります。

**SAML\_DSIG\_ALGO**

タイプ	文字列
適用先	SAML 2.0 認証、サービス プロバイダ
必須	いいえ
デフォルト	1

**説明**

次のいずれかの値の XML フェデレーション署名アルゴリズムを指定します。

- 1 = RSAwithSHA1 (デフォルト)
- 2 = RSAwithSHA256

**SAML\_DSIG\_VERINFO\_ISSUER\_DN**

タイプ	文字列
適用先	SAML 2.0 認証、サービスプロバイダ
必須	SAML 2.0 認証方式の場合 SAML_DISABLE_SIGNATURE_PROCESSING が SAML_FALSE で、以下のいずれかまたは両方が SAML_TRUE である場合に必須です。  SAML_SLO_REDIRECT_BINDING  SAML_ENABLE_SSO_POST_BINDING <ul style="list-style-type: none"> <li>■ サービスプロバイダの場合 SAML_DISABLE_SIGNATURE_PROCESSING が SAML_FALSE で、以下のいずれかまたは両方が SAML_TRUE である場合に必須です。 <ul style="list-style-type: none"> <li>■ SAML_SLO_REDIRECT_BINDING</li> <li>■ SAML_SP_REQUIRE_SIGNED_AUTHNREQUESTS</li> </ul> </li> </ul>
デフォルト	なし

**説明**

サービスプロバイダの証明書がインラインで提供されない場合、この値はキーストアで証明書を検索するために、SAML\_DSIG\_VERINFO\_SERIAL\_NUMBER と共に使用されます。

**SAML\_DSIG\_VERINFO\_SERIAL\_NUMBER**

タイプ	文字列
適用先	SAML 2.0 認証、サービスプロバイダ

必須	<p>SAML 2.0 認証方式の場合</p> <p>SAML_DISABLE_SIGNATURE_PROCESSING が SAML_FALSE で、以下のいずれかまたは両方が SAML_TRUE である場合に必須です。</p> <ul style="list-style-type: none"> <li>■ SAML_SLO_REDIRECT_BINDING</li> <li>■ SAML_ENABLE_SSO_POST_BINDING</li> </ul> <p>サービス プロバイダの場合</p> <p>SAML_DISABLE_SIGNATURE_PROCESSING が SAML_FALSE で、以下のいずれかまたは両方が SAML_TRUE である場合に必須です。</p> <ul style="list-style-type: none"> <li>■ SAML_SLO_REDIRECT_BINDING</li> <li>■ SAML_SP_REQUIRE_SIGNED_AUTHNREQUESTS</li> </ul>
デフォルト	なし

### 説明

サービス プロバイダ の証明書がインラインで提供されない場合、この値はキーストアで証明書を検索するために、SAML\_DSIG\_VERINFO\_ISSUER\_DN と共に使用されます。

### SAML\_ENABLE\_SSO\_ARTIFACT\_BINDING

タイプ	ブール値
適用先	SAML 2.0 認証、サービス プロバイダ
必須	いいえ
デフォルト	SAML_FALSE

### 説明

アーチファクトバインディングがサービス プロバイダによってサポートされ、アイデンティティ プロバイダによって有効にされるかどうかを指定します。

### SAML\_ENABLE\_SSO\_POST\_BINDING

タイプ	ブール値
適用先	SAML 2.0 認証、サービス プロバイダ

必須	いいえ
デフォルト	SAML_FALSE

**説明**

HTTP POST バインディングがサービスプロバイダによってサポートされ、アイデンティティプロバイダによって有効にされるかどうかを指定します。

SAML\_DSIG\_VERINFO\_ISSUER\_DN および SAML\_DSIG\_VERINFO\_SERIAL\_NUMBER も参照してください。

**SAML\_ENABLED**

タイプ	ブール値
適用先	サービスプロバイダ
必須	いいえ
デフォルト	SAML_TRUE

**説明**

サービスプロバイダがアクティブにされるかどうかを指定します。

**SAML\_IDP\_AD\_SEARCH\_SPEC**

タイプ	文字列
適用先	SAML 2.0 アフィリエーション、SAML 2.0 認証
必須	いいえ
デフォルト	なし

## 説明

AD ディレクトリの検索指定情報。

ユーザの特定が AD ディレクトリ内のユーザ上で実行されているが、AD 検索指定情報がこのプロパティに対して提供されていない場合、  
[SiteMinder ユーザ ディレクトリ プロパティ] ダイアログ ボックスで定義されたデフォルトの検索指定情報が使用されます。

このプロパティに検索指定情報を割り当てることは、以下の理由によって推奨されています。

- デフォルト検索指定情報を使用するときに、ポリシー サーバはアサーションから抽出された ID 内にすでにあるログイン ID プレフィックスおよびサフィックスを複製する場合があります。
- カスタム メッセージ コンシューマ プラグインで SAML 2.0 認証方式の機能を拡張している場合、ポリシー サーバが [SiteMinder ユーザ ディレクトリ プロパティ] ダイアログ ボックスで定義されているデフォルト検索指定情報でユーザの特定を行うと、プラグインはユーザの特定段階でコールされません。詳細については、SAML\_IDP\_PLUGIN\_CLASS を参照してください。

アフィリエーションに対して定義されていると、検索指定情報はアフィリエーションにわたってすべてのアイデンティティ プロバイダによって共有されます。

## SAML\_IDP\_ARTIFACT\_RESOLUTION\_DEFAULT\_SERVICE

タイプ	文字列
適用先	SAML 2.0 認証
必須	SAML_ENABLE_SSO_ARTIFACT_BINDING が SAML_TRUE である場合は必須です。
デフォルト	なし

## 説明

アイデンティティ プロバイダ用のデフォルト アーチファクト 解決サービスを指定する URL。

**SAML\_IDP\_BACKCHANNEL\_AUTH\_TYPE**

タイプ	整数
適用先	SAML 2.0 認証
必須	いいえ
デフォルト	0

**説明**

バック チャンネル上で使用する認証のタイプを指定します。有効な値は以下のとおりです。

- 0. 基本 - 認証用の指定されたサービス プロバイダ名およびパスワードを使用します。
- 1. クライアント証明書 - キーストア内の証明書を調べるために指定されたサービス プロバイダ ID およびパスワードを使用します。
- 2. 認証なし - 認証は必要ありません。

**SAML\_IDP\_CUSTOM\_SEARCH\_SPEC**

タイプ	文字列
適用先	SAML 2.0 アフィリエーション、SAML 2.0 認証
必須	いいえ
デフォルト	なし

**説明**

カスタム ユーザ ディレクトリの検索指定情報。ユーザの特定がカスタム ディレクトリ内のユーザに対して実行されているが、検索指定情報が提供されていない場合は、[SiteMinder ユーザ ディレクトリ プロパティ] ダイアログ ボックスで定義されているデフォルト検索指定情報が使用されます。

アフィリエーションに対して定義されていると、検索指定情報はアフィリエーションにわたってすべてのアイデンティティプロバイダによって共有されます。

カスタムメッセージコンシューマプラグインで SAML 2.0 認証方式の機能を拡張している場合、ポリシーサーバが [SiteMinder ユーザディレクトリプロパティ] ダイアログボックスで定義されているデフォルト検索指定情報でユーザの特定を行うと、プラグインはユーザの特定段階でコールされません。詳細については、SAML\_IDP\_PLUGIN\_CLASS を参照してください。

### SAML\_IDP\_LDAP\_SEARCH\_SPEC

タイプ	文字列
適用先	SAML 2.0 アフィリエーション、SAML 2.0 認証
必須	いいえ
デフォルト	なし

#### 説明

LDAP ディレクトリの検索指定情報。

ユーザの特定が LDAP ディレクトリ内のユーザ上で実行されているが、検索指定情報がこのプロパティに対して提供されていない場合、

[SiteMinder ユーザディレクトリプロパティ] ダイアログボックスで定義されたデフォルトの検索指定情報が使用されます。

このプロパティに検索指定情報を割り当てることは、以下の理由によって推奨されています。

- デフォルト検索指定情報を使用するときに、ポリシー サーバはアサーションから抽出された ID 内にすでにあるログイン ID プレフィックスおよびサフィックスを複製する場合があります。
- カスタム メッセージ コンシューマ プラグインで SAML 2.0 認証方式の機能を拡張している場合、ポリシー サーバが [SiteMinder ユーザ ディレクトリ プロパティ] ダイアログ ボックスで定義されているデフォルト検索指定情報でユーザの特定を行うと、プラグインはユーザの特定段階でコールされません。詳細については、SAML\_IDP\_PLUGIN\_CLASS を参照してください。

アフィリエーションに対して定義されていると、検索指定情報はアフィリエーションにわたってすべてのアイデンティティ プロバイダによって共有されます。

#### SAML\_IDP\_ODBC\_SEARCH\_SPEC

タイプ	文字列
適用先	SAML 2.0 アフィリエーション、SAML 2.0 認証
必須	いいえ
デフォルト	なし

## 説明

ODBC ディレクトリの検索指定情報。

ユーザの特定が ODBC ディレクトリ内のユーザ上で実行されているが、ODBC 検索指定情報がこのプロパティに対して提供されていない場合、[SiteMinder ユーザ ディレクトリ プロパティ] ダイアログ ボックスで定義されたデフォルトの検索指定情報が使用されます。

このプロパティに検索指定情報を割り当てることは、以下の理由によって推奨されています。

- デフォルト検索指定情報を使用するときに、ポリシー サーバはアサーションから抽出された ID 内にすでにあるログイン ID プレフィックス およびサフィックスを複製する場合があります。
- カスタム メッセージ コンシューマ プラグインで SAML 2.0 認証方式の機能を拡張している場合、ポリシー サーバが [SiteMinder ユーザ ディレクトリ プロパティ] ダイアログ ボックスで定義されているデフォルト検索指定情報でユーザの特定を行うと、プラグインはユーザの特定段階でコールされません。詳細については、SAML\_IDP\_PLUGIN\_CLASS を参照してください。

アフィリエーションに対して定義されていると、検索指定情報はアフィリエーションにわたってすべてのアイデンティティ プロバイダによって共有されます。

## SAML\_IDP\_PASSWORD

タイプ	文字列
適用先	SAML 2.0 認証
必須	SAML_IDP_BACKCHANNEL_AUTH_TYPE が 0 または 1 に設定される場合は必須。
デフォルト	なし

## 説明

バックチャネル認証に使用するパスワード。パスワードは、バックチャネル認証タイプ [基本] および [クライアント証明書] でのみ使用されません。

### SAML\_IDP\_PLUGIN\_CLASS

タイプ	文字列
適用先	SAML 2.0 認証
必須	いいえ
デフォルト	なし

#### 説明

この SAML 2.0 認証方式の機能を拡張する Java クラスの完全修飾名。カスタム機能はインターフェース `MessageConsumerPlugin.java` の実装によって提供されます。

認証には、ユーザの特定およびユーザの認証（特定されたユーザの認証情報の検証）の 2 つの段階があります。

プラグインが認証方式に対して設定される場合、以下のようにコールされます。

- 認証方式でユーザの特定を行えない場合は、ユーザの特定時。  
**注:** 検索指定情報が、特定が発生するユーザ ディレクトリ（たとえば、LDAP ディレクトリに対しては `SAML_IDP_LDAP_SEARCH_SPEC`）に対して提供されない場合、このプラグインはこの段階では呼び出されません。この場合、認証方式ではなくポリシー サーバが特定を実行します。
- ユーザが正常に検証される場合でも、デフォルト認証段階の最後。

SAML 2.0 認証方式は 1 つのメッセージ コンシューマ プラグインのみで拡張できます。

### SAML\_IDP\_PLUGIN\_PARAMS

タイプ	文字列
適用先	SAML 2.0 認証
必須	いいえ
デフォルト	なし

### 説明

SAML\_IDP\_PLUGIN\_CLASS で指定される、カスタム認証方式拡張へ渡されるパラメータ。

パラメータ文字列の構文はカスタム オブジェクトによって決定されます。

### SAML\_IDP\_REDIRECT\_MODE\_FAILURE

タイプ	整数
適用先	SAML 2.0 認証
必須	いいえ
デフォルト	0

### 説明

SAML\_IDP\_REDIRECT\_URL\_FAILURE 用のリダイレクトモード。有効な値は以下のとおりです。

- 0. 302 データなし -- HTTP 302 リダイレクト。ターゲットリソースの URL および認証失敗の理由はリダイレクト URL に追加されます。認証方式に渡される SAML 2.0 レスポンス メッセージは含まれません。
- 1. Http Post。 -- HTTP POST リダイレクト。認証方式に渡された SAML 2.0 レスポンス メッセージおよびアイデンティティプロバイダの ID は HTTP フォームによって生成されます。

### SAML\_IDP\_REDIRECT\_MODE\_INVALID

タイプ	整数
適用先	SAML 2.0 認証
必須	いいえ
デフォルト	0

**説明**

SAML\_IDP\_REDIRECT\_URL\_INVALID 用のリダイレクトモード。有効な値は以下のとおりです。

- 0. 302 データなし -- HTTP 302 リダイレクト。ターゲットリソースの URL および認証失敗の理由はリダイレクト URL に追加されます。認証方式に渡される SAML 2.0 レスポンス メッセージは含まれません。
- 1. Http Post。 -- HTTP POST リダイレクト。認証方式に渡された SAML 2.0 レスポンス メッセージおよびアイデンティティプロバイダの ID は HTTP フォームによって生成されます。

**SAML\_IDP\_REDIRECT\_MODE\_USER\_NOT\_FOUND**

タイプ	整数
適用先	SAML 2.0 認証
必須	いいえ
デフォルト	0

**説明**

SAML\_IDP\_REDIRECT\_URL\_USER\_NOT\_FOUND 用のリダイレクトモード。有効な値は以下のとおりです。

- 0. 302 データなし -- HTTP 302 リダイレクト。ターゲットリソースの URL および認証失敗の理由はリダイレクト URL に追加されます。認証方式に渡される SAML 2.0 レスポンス メッセージは含まれません。
- 1. Http Post。 -- HTTP POST リダイレクト。認証方式に渡された SAML 2.0 レスポンス メッセージおよびアイデンティティプロバイダの ID は HTTP フォームによって生成されます。

**SAML\_IDP\_REDIRECT\_URL\_FAILURE**

タイプ	文字列
適用先	SAML 2.0 認証
必須	いいえ
デフォルト	なし

### 説明

認証方式に渡される認証情報がユーザの認証のために受け入れられない場合に使用するリダイレクト URL。

#### SAML\_IDP\_REDIRECT\_URL\_INVALID

タイプ	文字列
適用先	SAML 2.0 認証
必須	いいえ
デフォルト	なし

### 説明

認証方式に渡される認証情報が SAML 2.0 標準に基づいた形式ではない場合に使用される リダイレクト URL。

#### SAML\_IDP\_REDIRECT\_URL\_USER\_NOT\_FOUND

タイプ	文字列
適用先	SAML 2.0 認証
必須	いいえ
デフォルト	なし

### 説明

以下のいずれかの状況で使用するリダイレクト URL。

- 認証方式で、渡された SAML 2.0 レスポンス メッセージからログイン ID を取得できない。
- 認証方式でユーザ ディレクトリ内のユーザを検出できない。

カスタム メッセージ コンシューマ プラグインで SAML 2.0 認証方式の機能を拡張している場合、ポリシー サーバが [SiteMinder ユーザ ディレクトリ プロパティ] ダイアログ ボックスで定義されているデフォルト検索指定情報でユーザの特定を行うと、プラグインはユーザの特定段階でコールされません。詳細については、SAML\_IDP\_PLUGIN\_CLASS を参照してください。

**SAML\_IDP\_REQUIRE\_ENCRYPTED\_ASSERTION**

タイプ	ブール値
適用先	SAML 2.0 認証
必須	いいえ
デフォルト	SAML_FALSE

**説明**

認証用に選択されたアサーションを暗号化する必要があるかどうかを指定します。このプロパティが **SAML\_TRUE** で、認証方式に暗号化されていないアサーションを渡される場合、アサーションは認証できません。

**SAML\_IDP\_REQUIRE\_ENCRYPTED\_NAMEID**

タイプ	ブール値
適用先	SAML 2.0 認証
必須	いいえ
デフォルト	SAML_FALSE

**説明**

アサーションに含まれていたプリンシパルの名前 ID を暗号化する必要があるかどうかを指定します。このプロパティが **SAML\_TRUE** で、名前 ID が暗号化されない場合、アサーションは認証できません。

**SAML\_IDP\_SAMLREQ\_ATTRIBUTE\_SERVICE**

タイプ	文字列
適用先	SAML 2.0 認証
必須	いいえ
デフォルト	なし

**説明**

属性認証機関上の属性サービスの URL。

**SAML\_IDP\_SAMLREQ\_ENABLE**

---

タイプ	ブール値
適用先	SAML 2.0 認証
必須	はい
デフォルト	False (0)

---

**説明**

SAML リクエストが有効になっているかどうかを示します。

**SAML\_IDP\_SAMLREQ\_NAMEID\_ALLOWED\_NESTED**

---

タイプ	ブール値
適用先	SAML 2.0 認証
必須	いいえ
デフォルト	False (0)

---

**説明**

名前識別子に対して DN 属性を選択する場合にネストされたグループが許可されるかどうかを示します。

**SAML\_IDP\_SAMLREQ\_NAMEID\_ATTR\_NAME**

---

タイプ	文字列
適用先	SAML 2.0 認証
必須	いいえ
デフォルト	なし

---

**説明**

NameIdType が 1 または 2 に設定される場合に名前を保持する属性名 (User または DN) 。 NameIdType が 1 または 2 に設定される場合、NameIdAttrName は有効な値が必要です。

**SAML\_IDP\_SAMLREQ\_NAMEID\_DN\_SPEC**

タイプ	文字列
適用先	SAML 2.0 認証
必須	いいえ
デフォルト	なし

**説明**

NameIdType が 2 に設定される場合に使用される DN 指定。 この場合、NameIdDNSpec には有効な値が必要です。

**SAML\_IDP\_SAMLREQ\_NAMEID\_FORMAT**

タイプ	ブール値
適用先	SAML 2.0 認証
必須	いいえ
デフォルト	なし

**説明**

SAML 2.0 名前識別子の URI。

**SAML\_IDP\_SAMLREQ\_NAMEID\_STATIC**

タイプ	文字列
適用先	SAML 2.0 認証
必須	いいえ

---

デフォルト	なし
-------	----

---

**説明**

NameIdType が 0 に設定される場合に名前識別子として使用されるスタティックテキスト。この場合、有効な値が NameIdStatic に対して指定される必要があります。

**SAML\_IDP\_SAMLREQ\_NAMEID\_TYPE**

---

タイプ	整数
適用先	SAML 2.0 認証
必須	いいえ
デフォルト	1

---

**説明**

名前識別子のタイプを表します。

0 - スタティック テキスト

1 - ユーザ属性

2 - DN 属性

**SAML\_IDP\_SAMELREQ\_REQUIRE\_SIGNED\_ASSERTION**

---

タイプ	ブール値
適用先	SAML 2.0 認証
必須	いいえ
デフォルト	False (0)

---

**説明**

属性クエリに応じて返されたアサーションが署名される必要があるかどうかを示します。

**SAML\_IDP\_SAMELREQ\_SIGN\_ATTRIBUTE\_QUERY**

タイプ	ブール値
適用先	SAML 2.0 認証
必須	いいえ
デフォルト	なし

**説明**

属性クエリに署名が必要かどうかを示します。

**SAML\_IDP\_SIGN\_AUTHNREQUESTS**

タイプ	ブール値
適用先	SAML 2.0 認証
必須	いいえ
デフォルト	False (0)

**説明**

認証要求が署名されるかどうかを指定します。

**SAML\_IDP\_SPID**

タイプ	文字列
適用先	SAML 2.0 認証
必須	はい
デフォルト	なし

**説明**

この認証方式によって保護されるサービス プロバイダの一意のプロバイダ ID。

**SAML\_IDP\_SPNAME**

タイプ	文字列
適用先	SAML 2.0 認証
必須	SAML_IDP_BACKCHANNEL_AUTH_TYPE が 0 または 1 に設定される場合は必須。
デフォルト	なし

**説明**

バックチャネル認証に含まれるサービスプロバイダの名前。サービスプロバイダ名は、バックチャネル認証タイプ [基本] および [クライアント証明書] で使用されます。

**SAML\_IDP\_SSO\_DEFAULT\_SERVICE**

タイプ	文字列
適用先	SAML 2.0 認証
必須	はい
デフォルト	なし

**説明**

アイデンティティプロバイダのシングルサインオンサービスの URL。たとえば以下ようになります。

<http://mysite.netegrity.com/affwebservices/public/saml2sso>

**SAML\_IDP\_SSO\_ENFORCE\_SINGLE\_USE\_POLICY**

タイプ	ブール値
適用先	SAML 2.0 認証
必須	いいえ
デフォルト	SAML_TRUE

## 説明

HTTP POST バインディングに使い捨てのポリシーを適用するべきかどうかを指定します。

このプロパティを **SAML\_TRUE** (デフォルト) に設定すると、**SAML POST** 固有の処理ルールに基づいて、サービスプロバイダサイトに対して 2 番目のセッションを確立するためにアサーションを「再使用」できなくなります。

使い捨てのポリシー要件は、ロードバランスおよびフェールオーバーが有効になったクラスタ化されたポリシー サーバ環境内でも強制されます。

## SAML\_IDP\_SSO\_REDIRECT\_MODE

タイプ	整数
適用先	SAML 2.0 認証
必須	いいえ
デフォルト	0

## 説明

ユーザがターゲットリソースにリダイレクトされる時にレスポンス属性情報を渡すためのメソッドを指定します。有効な値は以下のとおりです。

- 0. 302 データなし - レスポンス属性は渡されません。
- 1. 302 Cookie データ - レスポンス属性は HTTP Cookie データとして設定されます。認証方式によって発行された属性 Cookie は暗号化されていません。
- 2. サーバリダイレクト - レスポンス属性は HashMap オブジェクトとして渡されます。
- 3. 属性の保持 - 属性は取得できます。

サーバ側のリダイレクトでは、情報をサーバアプリケーション自体内のアプリケーションに渡すことができます。レスポンス属性データはユーザのブラウザに送信されません。このリダイレクトメソッドは **Java Servlet** 指定情報の一部であり、すべての標準準拠サーブレットコンテナによってサポートされています。

**SAML\_IDP\_SSO\_TARGET**

タイプ	文字列
適用先	SAML 2.0 認証
必須	いいえ
デフォルト	なし

**説明**

サービスプロバイダサイトでのターゲットリソースの URL。たとえば、ターゲットには Web ページまたはアプリケーションなどがあります。

**SAML\_IDP\_WINNT\_SEARCH\_SPEC**

タイプ	文字列
適用先	SAML 2.0 アフィリエーション、SAML 2.0 認証
必須	いいえ
デフォルト	なし

**説明**

WinNT ディレクトリの検索指定情報。ユーザの特定が WinNT ディレクトリ内のユーザに対して実行されているが、検索指定情報が提供されていない場合は、[SiteMinder ユーザディレクトリプロパティ] ダイアログボックスで定義されているデフォルト検索指定情報が使用されます。

アフィリエーションに対して定義されていると、検索指定情報はアフィリエーションにわたってすべてのアイデンティティプロバイダによって共有されます。

カスタムメッセージコンシューマプラグインで SAML 2.0 認証方式の機能を拡張している場合、ポリシーサーバが [SiteMinder ユーザディレクトリプロパティ] ダイアログボックスで定義されているデフォルト検索指定情報でユーザの特定を行うと、プラグインはユーザの特定段階でコールされません。詳細については、SAML\_IDP\_PLUGIN\_CLASS を参照してください。

**SAML\_IDP\_XPATH**

タイプ	文字列
適用先	SAML 2.0 アフィリエーション、SAML 2.0 認証
必須	いいえ
デフォルト	なし

**説明**

アサーションからユーザのログイン ID を抽出する XPath クエリ。その後、ログイン ID はユーザの特定のために使用されます。

デフォルトで、XPath が提供されない場合、SAML 2.0 レスポンスメッセージのアサーション/件名/名前 ID エlement からログイン ID を抽出する試みがなされます。

ログイン ID は、一度正常に抽出されると、ユーザディレクトリに対して指定された検索文字列に挿入され、特定段階が開始します。

アフィリエーションに対して定義された場合、XPath は、アフィリエーション間ですべてのアイデンティティプロバイダによって共有されます。

**SAML\_KEY\_AFFILIATION\_ID**

タイプ	文字列
適用先	SAML 2.0 アフィリエーション
必須	はい
デフォルト	なし

## 説明

アフィリエイトの URI。ID はサービス プロバイダおよびアイデンティティ プロバイダが同じアフィリエイトのメンバであることを確認するために使用されます。たとえば以下ようになります。

- サービス プロバイダがアイデンティティ プロバイダに対して認証要求を発行する場合、要求にはアフィリエイト ID が含まれます。アイデンティティ プロバイダは、サービス プロバイダが指定されたアフィリエイトに属することを確認します。
- アイデンティティ プロバイダがアサーションを生成し、サービス プロバイダに送信する場合、アサーションにはアフィリエイト ID が含まれます。サービス プロバイダは、アイデンティティ プロバイダが指定されたアフィリエイトに属することを確認します。
- シングル ログアウト中、ログアウト要求にはアフィリエイト ID も含まれます。ログアウト要求の受信時に、サービス プロバイダとアイデンティティ プロバイダは、互いに指定されたアフィリエイトに属することを確認します。

アフィリエイト ID は要求とアサーションの SPNameQualifier 属性で指定されます。

## SAML\_KEY\_IDP\_SOURCEID

タイプ	文字列
適用先	SAML 2.0 認証
必須	いいえ
デフォルト	SAML_KEY_IDPID 値の 16 進数にエンコードされた SHA-1 ハッシュ

## 説明

アーチファクト発行者に対する、16 進数にエンコードされた 20 バイトのシーケンス識別子。この値は、アサーションアーチファクト内のアーチファクト発行者を一意に識別します。

認証方式では、アイデンティティ プロバイダのメタデータを検索するキーとしてソース ID を使用します。

文字列長はちょうど 40 文字である必要があります。小文字の 16 進文字列のみが保存されます。

**SAML\_KEY\_IDPID**

タイプ	文字列
適用先	SAML 2.0 認証
必須	はい
デフォルト	なし

**説明**

この認証方式のアイデンティティ プロバイダのプロバイダ ID。この ID には以下が当てはまります。

- アサーション発行者を一意に識別します。
- アイデンティティ プロバイダのプロパティを検索するためのキーとして機能します。

**SAML\_KEY\_SPID**

タイプ	文字列
適用先	サービス プロバイダ
必須	はい
デフォルト	なし

**説明**

このサービス プロバイダの一意のプロバイダ ID。

**SAML\_MAJOR\_VERSION**

タイプ	整数
適用先	SAML 2.0 アフィリエーション、SAML 2.0 認証、サービス プロバイダ
必須	いいえ
デフォルト	2

**説明**

サポートされている SAML プロトコルのメジャーバージョン。値を指定する場合は、2 である必要があります。

**SAML\_MINOR\_VERSION**

---

タイプ	整数
適用先	SAML 2.0 アフィリエーション、SAML 2.0 認証、サービス プロバイダ
必須	いいえ
デフォルト	0

---

**説明**

サポートされている SAML プロトコルのマイナーバージョン。値を指定する場合は、0 である必要があります。

**SAML\_NAME**

---

タイプ	文字列
適用先	SAML 2.0 アフィリエーション、SAML 2.0 認証、サービス プロバイダ
必須	はい
デフォルト	なし

---

**説明**

アフィリエーション、認証方式またはサービス プロバイダの名前。

名前はグローバルに一意である必要があります。SAML 2.0 アフィリエーションおよびサービス プロバイダでは、名前は小文字である必要があります。

**SAML\_OID**

---

タイプ	文字列
-----	-----

---

適用先	SAML 2.0 アフィリエーション
必須	アフィリエーションオブジェクトが作成されている場合（SiteMinder がオブジェクト作成中にオブジェクト識別子を提供する）は必要ありません。カスタムコードが既存のオブジェクトを参照する場合は必要です。
デフォルト	なし

### 説明

アフィリエーションオブジェクトの一意のオブジェクト識別子。

[SAML アフィリエーションプロパティ] ダイアログボックスにはこのプロパティに対応するフィールドがありません。

### SAML\_SKEWTIME

タイプ	整数
適用先	SAML 2.0 認証、サービスプロバイダ
必須	いいえ
デフォルト	30

### 説明

アイデンティティプロバイダのシステムクロック時間と、サービスプロバイダのシステムクロック時間の間の秒単位の違い。

- サービスプロバイダでは、システムクロックがアイデンティティプロバイダとして機能するポリシーサーバと同期されていない場合に現在時間から差し引かれる秒数です。
- アイデンティティプロバイダでは、システムクロックがサービスプロバイダとして機能するポリシーサーバと同期されていない場合に現在時間から差し引かれる秒数です。

スキュー時間はアサーションおよびシングルログアウト要求の有効期間を計算するために使用されます。提供される値は正の整数である必要があります。

**SAML\_SLO\_REDIRECT\_BINDING**

タイプ	ブール値
適用先	SAML 2.0 認証、サービス プロバイダ
必須	いいえ
デフォルト	SAML_FALSE

**説明**

HTTP リダイレクト バインディングがシングル ログアウトに対してサポートされているかどうかを指定します。

SAML\_DSIG\_VERINFO\_ISSUER\_DN および SAML\_DSIG\_VERINFO\_SERIAL\_NUMBER も参照してください。

**SAML\_SLO\_SERVICE\_CONFIRM\_URL**

タイプ	文字列
適用先	SAML 2.0 認証、サービス プロバイダ
必須	いいえ
デフォルト	なし

**説明**

シングル ログアウトが完了した後にユーザがリダイレクトされる URL。

**SAML\_SLO\_SERVICE\_RESPONSE\_URL**

タイプ	文字列
適用先	SAML 2.0 認証、サービス プロバイダ
必須	いいえ
デフォルト	なし

**説明**

シングルログアウトサービスのレスポンスの場所。このプロパティでは、SLO レスポンスメッセージを要求メッセージが送信される場所とは異なる場所に送信できます。

**SAML\_SLO\_SERVICE\_URL**

タイプ	文字列
適用先	SAML 2.0 認証、サービス プロバイダ
必須	SAML_SLO_REDIRECT_BINDING が SAML_TRUE である場合は必須です。
デフォルト	なし

**説明**

HTTP-Redirect バインディングで、シングルログアウト要求が送られるアイデンティティプロバイダ URL。

**SAML\_SLO\_SERVICE\_VALIDITY\_DURATION**

タイプ	整数
適用先	SAML 2.0 認証、サービス プロバイダ
必須	いいえ
デフォルト	60 (値が提供されず、SAML_SLO_REDIRECT_BINDING が SAML_TRUE である場合に適用されます)

**説明**

シングルログアウト要求が有効な秒数。

提供される値は正の整数である必要があります。

SAML\_SKEWTIME も参照してください。

**SAML\_SP\_ARTIFACT\_ENCODING**

タイプ	文字列
-----	-----

---

適用先	サービス プロバイダ
必須	いいえ
デフォルト	FORM (値が提供されず、SAML_ENABLE_SSO_ARTIFACT_BINDING が SAML_TRUE である場合に適用されます)

---

### 説明

Artifact バインディングに使用するエンコードを指定します。有効な値は以下のとおりです。

- **FORM**。アーチファクトは SAMLart という名前の非表示コントロールでフォーム エンコードされます。
- **URL**。アーチファクトは SAMLart という名前の URL パラメータで URL エンコードされます。

FORM と URL のエンコードは SAML 2.0 仕様に従って実行されます。

### SAML\_SP\_ASSERTION\_CONSUMER\_DEFAULT\_URL

---

タイプ	文字列
適用先	サービス プロバイダ
必須	はい
デフォルト	なし

---

### 説明

生成されたアサーションが送信されるサービス プロバイダ URL。たとえば以下ようになります。

`http://mysite.netegrity.com/affwebservices/public/saml2assertionconsumer`

### SAML\_SP\_ATTRSVC\_AD\_SEARCH\_SPEC

---

タイプ	文字列
適用先	サービス プロバイダ
必須	いいえ
デフォルト	なし

---

**説明**

1 個の AD ディレクトリの検索指定情報。

**SAML\_SP\_ATTRSVC\_CUSTOM\_SEARCH\_SPEC**

タイプ	文字列
適用先	サービス プロバイダ
必須	いいえ
デフォルト	なし

**説明**

カスタム ディレクトリの検索指定情報です。

**SAML\_SP\_ATTRSVC\_ENABLE**

タイプ	ブール値
適用先	サービス プロバイダ
必須	いいえ
デフォルト	SAML_FALSE

**説明**

属性認証機関が有効かどうかを示します。

**SAML\_SP\_ATTRSVC\_LDAP\_SEARCH\_SPEC**

タイプ	文字列
適用先	サービス プロバイダ
必須	いいえ
デフォルト	なし

**説明**

1 個の LDAP ディレクトリの検索指定情報。

**SAML\_SP\_ATTRSVC\_ODBC\_SEARCH\_SPEC**

タイプ	文字列
適用先	サービス プロバイダ
必須	いいえ
デフォルト	なし

**説明**

1 個の ODBC ディレクトリの検索指定情報。

**SAML\_SP\_ATTRSVC\_REQUIRE\_SIGNED\_QUERY**

タイプ	ブール値
適用先	サービス プロバイダ
必須	いいえ
デフォルト	なし

**説明**

属性クエリに署名が必要かどうかを示します。

**SAML\_SP\_ATTRSVC\_SIGN\_ASSERTION**

タイプ	ブール値
適用先	サービス プロバイダ
必須	いいえ
デフォルト	なし

**説明**

SAML アサーションに署名が必要かどうかを示します。

**SAML\_SP\_ATTRSVC\_SIGN\_RESPONSE**

タイプ	ブール値
適用先	サービス プロバイダ
必須	いいえ
デフォルト	なし

**説明**

SAML レスポンスに署名が必要かどうかを示します。

**SAML\_SP\_ATTRSVC\_VALIDITY\_DURATION**

タイプ	整数
適用先	サービス プロバイダ
必須	いいえ
デフォルト	60

**説明**

アサーションが有効な秒数を指定します。

**SAML\_SP\_ATTRSVC\_WINNT\_SEARCH\_SPEC**

タイプ	文字列
適用先	サービス プロバイダ
必須	いいえ
デフォルト	なし

**説明**

1 個の WinNT ディレクトリの検索指定情報。

**SAML\_SP\_AUTHENTICATION\_LEVEL**

タイプ	整数
適用先	サービス プロバイダ
必須	いいえ
デフォルト	5

**説明**

このプロパティは、現在のアサーションと関連付けられたプリンシパルを認証する認証方式に必要な最小保護レベルを指定します。

**SAML\_SP\_AUTHENTICATION\_URL**

タイプ	文字列
適用先	サービス プロバイダ
必須	はい
デフォルト	なし

**説明**

このサービス プロバイダのユーザを認証するための保護された URL。

**SAML\_SP\_AUTHN\_CONTEXT\_CLASS\_REF**

タイプ	文字列
適用先	サービス プロバイダ
必須	いいえ
デフォルト	urn:oasis:names:tc:SAML:2.0:ac:classes:Password

## 説明

アサーションに対する信頼性を評価するためにサービスプロバイダが必要とする可能性がある情報のクラス。クラスはアサーションの `AuthnContextClassRef` エレメントで指定されます。

たとえば、デフォルト認証コンテキストクラスは `Password` です。プリンシパルが非保護の HTTP セッションに対するパスワードの表示を通じて認証するときに、このクラスが適用されます。

認証コンテキストクラスの他の例には、`InternetProtocol` (プロバイダ IP アドレスを通して認証)、`X509` (X.509 デジタル署名を通して認証)、および `Telephony` (`telephony` プロトコルを通して渡される固定回線電話番号の提供を通して認証) などがあります。

認証コンテキストクラスは以下の初期ステムを持つ URI です。

`urn:oasis:names:tc:SAML:2.0:ac:classes:`

SAML 2.0 認証コンテキスト指定情報は、認証コンテキストクラスとして提供できる URI を定義します。また、クラスは、サービスプロバイダに対して定義された認証レベルに適合する必要があります。

### SAML\_SP\_COMMON\_DOMAIN

タイプ	文字列
適用先	サービスプロバイダ
必須	<code>SAML_SP_ENABLE_IPD</code> が <code>SAML_TRUE</code> である場合は必須です。
デフォルト	なし

## 説明

アイデンティティプロバイダディスカバリプロファイル用の共通の Cookie ドメイン。ドメインは `SAML_SP_IPD_SERVICE_URL` で指定されたホストのサブセットである必要があります。

### SAML\_SP\_CUSTOM\_TIME\_OUT

タイプ	文字列
適用先	サービスプロバイダ

必須	いいえ
デフォルト	なし

**説明**

アサーションに設定される `SessionNotOnOrAfter` パラメータの値を指定します。このプロパティは、`SAML_SP_SESSION_NOTORAFETER_TYPE` が `Custom` に設定される場合にのみ有効です。

**SAML\_SP\_DOMAIN**

タイプ	文字列
適用先	サービスプロバイダ
必須	はい
デフォルト	なし

**説明**

サービスプロバイダが定義されるアフィリエイトドメインの一意の ID。

[SAML サービスプロバイダ プロパティ] ダイアログボックスにはこのプロパティに対応するフィールドがありません。

**SAML\_SP\_ENABLE\_IPD**

タイプ	ブール値
適用先	サービスプロバイダ
必須	いいえ
デフォルト	<code>SAML_FALSE</code>

**説明**

アイデンティティプロバイダディスカバリプロファイルが有効かどうかを指定します。

**SAML\_SP\_ENCRYPT\_ASSERTION**

タイプ	ブール値
適用先	サービス プロバイダ
必須	いいえ
デフォルト	SAML_FALSE

**説明**

サービス プロバイダ サイトで生成されたアサーションを暗号化するかどうかを指定します。デフォルトでは、アサーションは暗号化されません。

**SAML\_SP\_ENCRYPT\_BLOCK\_ALGO**

タイプ	文字列
適用先	サービス プロバイダ
必須	いいえ
デフォルト	tripleDES

**説明**

使用するブロック暗号化アルゴリズムのタイプ。有効な値は以下のとおりです。

- tripleDES。3つの個別の56ビットキーを使用するデータ暗号化標準。
- aes-128。Advanced Encryption Standard、キー長は128ビットです。
- aes-256。Advanced Encryption Standard、キー長は256ビットです。

**SAML\_SP\_ENCRYPT\_CERT\_ISSUER\_DN**

タイプ	文字列
適用先	サービス プロバイダ

必須	<p>以下のいずれかの状況では、はい。</p> <ul style="list-style-type: none"> <li>■ 以下のいずれかが <code>SAML_TRUE</code> である場合。 <ul style="list-style-type: none"> <li><code>SAML_SP_ENCRYPT_ID</code></li> <li><code>SAML_SP_ENCRYPT_ASSERTION</code></li> </ul> </li> <li>■ いずれかのアサーション属性ステートメントが暗号化を必要とする場合。これらの属性は、[SAML サービス プロバイダ プロパティ] ダイアログボックスの [属性] タブで定義されます。</li> </ul>
----	--

デフォルト	なし
-------	----

### 説明

暗号化に使用される公開鍵証明書の発行者 DN 部分。このプロパティは、キーストア内のサービス プロパティの証明書がインラインで提供されない場合にそれを見つけるために `SAML_SP_ENCRYPT_CERT_SERIAL_NUMBER` と共に使用されます。

### [SAML\\_SP\\_ENCRYPT\\_CERT\\_SERIAL\\_NUMBER](#)

タイプ	文字列
適用先	サービス プロバイダ
必須	<p>以下のいずれかの状況では、はい。</p> <ul style="list-style-type: none"> <li>■ 以下のいずれかが <code>SAML_TRUE</code> である場合。 <ul style="list-style-type: none"> <li><code>SAML_SP_ENCRYPT_ID</code></li> <li><code>SAML_SP_ENCRYPT_ASSERTION</code></li> </ul> </li> <li>■ いずれかのアサーション属性ステートメントが暗号化を必要とする場合。これらの属性は、[SAML サービス プロバイダ プロパティ] ダイアログボックスの [属性] タブで定義されます。</li> </ul>

デフォルト	なし
-------	----

### 説明

暗号化に使用される公開鍵証明書のシリアル番号部分。このプロパティは、キーストア内のサービス プロパティの証明書がインラインで提供されない場合にそれを見つけるために `SAML_SP_ENCRYPT_CERT_ISSUER_DN` と共に使用されます。

**SAML\_SP\_ENCRYPT\_ID**

タイプ	ブール値
適用先	サービス プロバイダ
必須	いいえ
デフォルト	SAML_FALSE

**説明**

生成されたアサーション内の名前 ID をサービス プロバイダ サイトで暗号化する必要があるかどうかを指定します。デフォルトでは、名前 ID は暗号化されません。

**SAML\_SP\_ENCRYPT\_KEY\_ALGO**

タイプ	文字列
適用先	サービス プロバイダ
必須	いいえ
デフォルト	rsa-v15

**説明**

使用する暗号化キー アルゴリズムのタイプ。有効な値は以下のとおりです。

- **rsa-v15**。RSA 暗号化、バージョン 1.5。
- **rsa-oaep**。最適な Asymmetric Encryption Padding エンコードおよび RSA 暗号化。

**SAML\_SP\_ENDTIME**

タイプ	長整数（10 進数の文字列として保存）
適用先	サービス プロバイダ
必須	いいえ
デフォルト	なし

### 説明

アサーションを生成する必要がある場合の期限（時間）。

このプロパティは、アサーション生成の時間制限を定義するために、**SAML\_SP\_STARTTIME**、および **Sm\_PolicyApi\_SAMLSP\_t** 構造の **[pszTimeGrid]** フィールドと共に使用されます。

この値は標準的な **time\_t** 値から作成されます。ただし、これは **10** 進の文字列として保存されます。長整数のデータ型として **SAML\_SP\_ENDTIME** を使用する必要がある場合は、必ず値を変換してください。

時間制限をすぐに終了するには、**SAML\_SP\_ENDTIME** を **0** に設定します。

### SAML\_SP\_IDP\_SOURCEID

タイプ	文字列
適用先	サービス プロバイダ
必須	いいえ
デフォルト	SAML_SP_IDPID 値の 16 進数にエンコードされた SHA-1 ハッシュ

### 説明

アーチファクト発行者に対する、16 進数にエンコードされた 20 バイトのシーケンス識別子。この値は、アサーションアーチファクト内のアーチファクト発行者を一意に識別します。

文字列長はちょうど 40 文字である必要があります。小文字の 16 進文字列のみが保存されます。

### SAML\_SP\_IDPID

タイプ	文字列
適用先	サービス プロバイダ
必須	はい
デフォルト	なし

**説明**

アサーションを生成するアイデンティティプロバイダのプロバイダ ID。

**SAML\_SP\_IGNORE\_REQ\_AUTHNCONTEXT**

タイプ	文字列
適用先	サービスプロバイダ
必須	いいえ
デフォルト	0

**説明**

アイデンティティプロバイダが AuthnRequest メッセージの「RequestedAuthnContext」を無視するか（値が 1 の場合）または無視しないか（0 の場合）を指定します。

**SAML\_SP\_IPD\_SERVICE\_URL**

タイプ	文字列
適用先	サービスプロバイダ
必須	SAML_SP_ENABLE_IPD が SAML_TRUE である場合は必須です。
デフォルト	なし

**説明**

アイデンティティプロバイダディスカバリプロファイルのホスト URL。

**SAML\_SP\_NAMEID\_ATTRNAME**

タイプ	文字列
適用先	SAML 2.0 アフィリエーション、サービスプロバイダ
必須	SAML_SP_NAMEID_TYPE が 1（ユーザ属性）または 2（DN 属性）に設定されているときは、はい
デフォルト	なし

### 説明

以下の値のいずれかです。

- SAML\_SP\_NAMEID\_TYPE が 1 に設定されている場合、このプロパティは名前識別子が含まれているユーザ属性の名前を指定します。
- SAML\_SP\_NAMEID\_TYPE が 2 に設定されている場合、このプロパティはグループまたは組織単位 DN と関連付けられた属性を指定します。

### SAML\_SP\_NAMEID\_DNSPEC

タイプ	文字列
適用先	SAML 2.0 アフィリエーション、サービス プロバイダ
必須	SAML_SP_NAMEID_TYPE が 2 (DN 属性) に設定されている場合は、はい
デフォルト	なし

### 説明

関連付けられた名前 ID 属性を取得するために使用されるグループまたは組織単位 DN。

SiteMinder がネストされたグループ内の属性を検索することを許可できます。詳細については、構造 Sm\_PolicyApi\_SAMLSPAttr\_t の [pszValue] フィールドの説明を参照してください。

### SAML\_SP\_NAMEID\_FORMAT

タイプ	文字列
適用先	SAML 2.0 アフィリエーション、サービス プロバイダ
必須	いいえ
デフォルト	未指定

## 説明

以下のいずれかの名前 ID 形式の値の完全 URI。

- 未指定
- [Email Address]
- X509 サブジェクト名
- Windows ドメイン修飾名
- Kerberos プリンシパル名
- エンティティ識別子
- 永続的な識別子
- 一時的な識別子

たとえば、デフォルト形式である未指定の完全 URI は以下のとおりです。

`urn:oasis:names:tc:SAML:2.0:nameid-format:unspecified`

これらの形式の説明については、以下の SAML 2.0 指定情報を参照してください。

OASIS セキュリティ アサーション マークアップ 言語 (SAML) V2.0 用のアサーションおよびプロトコル

**注:** SAML アサーションが SAML\_AFFILIATION で指定される場合、このプロパティと他の SAML\_SP\_NAMEID... プロパティは使用されません。SiteMinder は、指定されたアフィリエーション内の名前 ID 情報を使用しません。

## SAML\_SP\_NAMEID\_STATIC

タイプ	文字列
適用先	SAML 2.0 アフィリエーション、サービス プロバイダ
必須	SAML_SP_NAMEID_TYPE が 0 (スタティック) に設定されている場合は、はい
デフォルト	なし

## 説明

すべての名前識別子に使用されるスタティック テキスト。

**SAML\_SP\_NAMEID\_TYPE**

タイプ	整数
適用先	SAML 2.0 アフィリエーション、サービス プロバイダ
必須	いいえ
デフォルト	1

**説明**

名前識別子のタイプ。有効な値は以下のとおりです。

- 0. スタティック テキスト。
- 1. ユーザ属性。
- 2. DN 属性。

**SAML\_SP\_ONE\_TIME\_USE**

タイプ	ブール値
適用先	サービス プロバイダ
必須	いいえ
デフォルト	SAML_FALSE

**説明**

アサーションを 1 回だけ使用することを示す要素をアサーションに追加するかどうかを指定します。

**SAML\_SP\_PASSWORD**

タイプ	文字列
適用先	サービス プロバイダ
必須	SAML_ENABLE_SSO_ARTIFACT_BINDING が SAML_TRUE である場合は必須です。
デフォルト	なし

**説明**

バックチャネルによるサービスプロバイダアクセスに使用されるパスワード。

**SAML\_SP\_PERSISTENT\_COOKIE**

タイプ	ブール値
適用先	サービスプロバイダ
必須	いいえ
デフォルト	SAML_FALSE

**説明**

アイデンティティプロバイダディスカバリプロファイル Cookie を永続的にするかどうかを指定します。SAML\_SP\_ENABLE\_IPD が SAML\_TRUE である場合にのみ適用されます。

**SAML\_SP\_PLUGIN\_CLASS**

タイプ	文字列
適用先	サービスプロバイダ
必須	いいえ
デフォルト	なし

**説明**

アサーション生成プログラムプラグインの完全修飾 Java クラス名。

アサーション生成プログラムプラグインでは、アサーションのコンテンツをカスタマイズできます。詳細については、オンラインの「SiteMinder Java API ドキュメント」（Javadoc およびガイド）を参照してください。

**SAML\_SP\_PLUGIN\_PARAMS**

タイプ	文字列
適用先	サービスプロバイダ

---

必須	いいえ
デフォルト	なし

---

**説明**

SAML\_SP\_PLUGIN\_CLASS で指定されたアサーション生成プログラム プラグインに渡されるパラメータ。

**SAML\_SP\_REQUIRE\_SIGNED\_AUTHNREQUESTS**

---

タイプ	ブール値
適用先	サービス プロバイダ
必須	いいえ
デフォルト	SAML_FALSE

---

**説明**

認証要求が署名される必要があるかどうかを指定します。

**SAML\_SP\_REUSE\_SESSION\_INDEX**

---

タイプ	ブール値
適用先	サービス プロバイダ
必須	いいえ
デフォルト	0

---

## 説明

SiteMinder が単一のブラウザセッション内の同じパートナーに対してアサーション内の同じセッションインデックスを送るかどうかを示します。ユーザが同じブラウザウィンドウを使用して同じパートナーと複数回フェデレーションを行う場合、このプロパティを設定すると、各セッションで同じセッションインデックスを送信するように IdP に指示されます。プロパティのデフォルト値 (0) は、シングルサインオンが発生するたびに、新しいセッションインデックスを生成するように SiteMinder に指示します。

有効な値は以下のとおりです。

0

同じセッションインデックスを再利用しません。

1

同じセッションインデックスを再利用します。

## SAML\_SP\_STARTTIME

タイプ	長整数 (10 進数の文字列として保存)
適用先	サービスプロバイダ
必須	いいえ
デフォルト	なし

## 説明

アサーションを生成するための時間制限が有効になる時間。この値は、標準の `time_t` 形式で保存されます。

このプロパティは、アサーション生成の時間制限を定義するために、`SAML_SP_ENDTIME`、および `Sm_PolicyApi_SAMLSP_t` 構造の `[pszTimeGrid]` フィールドと共に使用されます。

この値は標準的な `time_t` 値から作成されます。ただし、これは 10 進の文字列として保存されます。長整数のデータ型として `SAML_SP_STARTTIME` を使用する必要がある場合は、必ず値を変換してください。

時間制限をすぐに開始するには `SAML_SP_STARTTIME` を 0 に設定します。

**SAML\_SP\_VALIDITY\_DURATION**

タイプ	整数
適用先	サービス プロバイダ
必須	いいえ
デフォルト	60

**説明**

生成されたアサーションが有効な秒数。

提供される値は正の整数である必要があります。

SAML\_SKEWTIME も参照してください。

**SAML\_SP\_SESSION\_NOTORAFTER\_TYPE**

タイプ	文字列
適用先	サービス プロバイダ
必須	いいえ
デフォルト	アサーション有効期間の使用

## 説明

このプロパティにより、アサーションの `SessionNotOnOrAfter` パラメータに対して設定される値が決まります。サードパーティ SP は、`SessionNotOnOrAfter` の値を使用して、独自のセッションタイムアウトを設定できます。

`SiteMinder` は、SP として機能している場合、`SessionNotOnOrAfter` 値を無視します。代わりに、`SiteMinder` SP は、ターゲットリソースを保護する設定済みの SAML 認証方式に相当するレルムタイムアウトに基づき、セッションタイムアウトを設定します。

## アサーション有効期間の使用

アサーション有効期間に基づく `SessionNotOnOrAfter` 値を計算します。

## 省略

アサーションに `SessionNotOnOrAfter` パラメータを含めないように IdP に命じます。

## IDP セッション

IdP セッションタイムアウトに基づく `SessionNotOnOrAfter` 値を計算します。タイムアウトは認証 URL 用の IdP レルムで設定されます。このオプションを使用すると、IdP および SP セッションタイムアウト値を同期することができます。

## カスタム

アサーションに `SessionNotOnOrAfter` パラメータの独自の値を指定できます。このオプションを選択する場合は、`SAML_SP_CUSTOM_TIME_OUT` プロパティに時間を入力します。

## SAML\_SSOECPPROFILE

タイプ	ブール値
適用先	SAML 2.0 認証、サービスプロバイダ
必須	いいえ
デフォルト	SAML_FALSE

**説明**

アイデンティティプロバイダまたはサービスプロバイダが SAML 2.0 拡張クライアントおよびプロキシプロファイル要求をサポートするかどうかを指定します。

**SAML2\_CUSTOM\_ENABLE\_INVALID\_REQUEST\_URL**

タイプ	ブール値
適用先	カスタムエラーページ
必須	いいえ
デフォルト	なし

**説明**

カスタムエラーリダイレクトプロセスが無効な要求に有効かどうかを指定します。

**SAML2\_CUSTOM\_ENABLE\_SERVER\_ERROR\_URL**

タイプ	ブール値
適用先	カスタムエラーページ
必須	いいえ
デフォルト	なし

**説明**

カスタムエラーリダイレクトプロセスをサーバエラーに対して有効にするかどうかを指定します。

**SAML2\_CUSTOM\_ENABLE\_INVALID\_REQUEST\_URL**

タイプ	ブール値
適用先	カスタムエラーページ
必須	いいえ

デフォルト	なし
-------	----

**説明**

カスタムエラーリダイレクトプロセスが無効な要求に有効かどうかを指定します。

**SAML2\_CUSTOM\_INVALID\_REQUEST\_REDIRECT\_MODE**

タイプ	ブール値
適用先	カスタムエラーページ
必須	いいえ
デフォルト	なし

**説明**

無効な要求に対するリダイレクトモードを指定します。有効な値は以下のとおりです。

- **0. 302 データなし -- HTTP 302 リダイレクト。** ターゲットリソースの URL および認証失敗の理由はリダイレクト URL に追加されます。認証方式に渡される SAML 2.0 レスポンスメッセージは含まれません。
- **1. Http Post。 -- HTTP POST リダイレクト。** 認証方式に渡された SAML 2.0 レスポンスメッセージおよびアイデンティティプロバイダの ID は HTTP フォームによって生成されます。

**SAML2\_CUSTOM\_INVALID\_REQUEST\_REDIRECT\_URL**

タイプ	文字列
適用先	カスタムエラーページ
必須	いいえ
デフォルト	なし

**説明**

- 無効な要求に対するリダイレクト URL を指定します。

**SAML2\_CUSTOM\_SERVER\_ERROR\_REDIRECT\_MODE**

タイプ	ブール値
適用先	カスタム エラー ページ
必須	いいえ
デフォルト	なし

**説明**

内部サーバエラーに対するリダイレクト モードを指定します。有効な値は以下のとおりです。

- **0. 302 データなし** -- HTTP 302 リダイレクト。ターゲットリソースの URL および認証失敗の理由はリダイレクト URL に追加されます。認証方式に渡される SAML 2.0 レスポンス メッセージは含まれません。
- **1. Http Post**。 -- HTTP POST リダイレクト。認証方式に渡された SAML 2.0 レスポンス メッセージおよびアイデンティティプロバイダの ID は HTTP フォームによって生成されます。

**SAML2\_CUSTOM\_SERVER\_ERROR\_REDIRECT\_URL**

タイプ	文字列
適用先	カスタム エラー ページ
必須	いいえ
デフォルト	なし

**説明**

内部サーバエラーに対するリダイレクト URL を指定します。

**SAML2\_CUSTOM\_UNAUTHORIZED\_ACCESS\_REDIRECT\_MODE**

タイプ	ブール値
適用先	カスタム エラー ページ
必須	いいえ

---

デフォルト なし

---

### 説明

禁止されたアクセスに対するリダイレクトモードを指定します。有効な値は以下のとおりです。

- 0. 302 データなし -- HTTP 302 リダイレクト。ターゲットリソースの URL および認証失敗の理由はリダイレクト URL に追加されます。認証方式に渡される SAML 2.0 レスポンスメッセージは含まれません。
- 1. Http Post。 -- HTTP POST リダイレクト。認証方式に渡された SAML 2.0 レスポンスメッセージおよびアイデンティティプロバイダの ID は HTTP フォームによって生成されます。

### SAML2\_CUSTOM\_UNAUTHORIZED\_ACCESS\_REDIRECT\_URL

---

タイプ 文字列

---

適用先 カスタムエラーページ

---

必須 いいえ

---

デフォルト なし

---

### 説明

禁止されたアクセスエラーに対するリダイレクト URL を指定します。