

SiteMinder

Perl 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写、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す。

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。

CA Technologies 製品リファレンス

このマニュアルでは、以下の CA Technologies 製品に言及しています。

SiteMinder

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。

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。

- [AddAssociationMultiValue メソッド](#) (P. 219) - 説明を更新しました。 CQ 176626 を解決します。
- (ユーザ メソッドに) **SetPath** メソッドを追加しました。 ドキュメント CQ 180610; Eng 179050 を解決します。

目次

第 1 章: Perl スクリプト作成の概要 25

SiteMinder コマンド ライン インターフェースについて	25
インストール パス	26
Perl の場所	26
スクリプトを実行する場所	27
CLI の例: ポリシー ストア オブジェクトの作成	27
CLI の例: 個々のプロパティの表示と設定	29
サンプル スクリプトの場所	30
関連マニュアル	30
オブジェクト依存性ポスター	31

第 2 章: CLI エージェント API について 33

エージェントおよびエージェント API	33
エージェント API に対するスクリプトの作成	34
シングル サインオンとエージェント API	35
カスタム エージェントのシングル サインオン サポート	36
標準エージェントのシングル サインオン サポート	37
セッション情報	38
セッション変数の利点	38
SiteMinder のサポート	39
セッション変数を使用するための要件	39
セッション終了時のクリーンアップ	39
オブジェクトとオブジェクト階層	40

第 3 章: CLI エージェント API メソッド 41

CLI エージェント管理メソッド	41
AddServerConfig メソッド -- エージェント API オブジェクトにポリシー サーバ設定を追加	41
Connect メソッド -- エージェント API とポリシー サーバ間の接続を確立	43
CreateBootstrapFile メソッド -- エージェントに接続するためのブートストラップ ファイルを生成	44
CreateUser メソッド -- ユーザ オブジェクトを作成	45
Disconnect メソッド -- エージェントとポリシー サーバ間の接続を解除	46
DoManagement メソッド -- ポリシー サーバからのエージェント コマンドを要求	47

GetResource メソッド -- 指定されたリソースを取得	48
IncrementRefCount メソッド -- 参照数をインクリメント	48
New メソッド -- エージェント API を構築	49
PrintDebugTrace メソッド -- コンソールにトレース情報を出力	50
ProcessAuthResponse メソッド -- パスワードの変更ログインから情報を抽出	51
SetErrorCallback メソッド -- エラー メッセージを処理するサブルーチンを登録	52
SetTraceCallback メソッド -- トレース メッセージを処理するサブルーチンを登録	53
リソース メソッド	54
GetAuthType メソッド -- 必要な認証情報のタイプを取得	54
IsProtected メソッド -- SiteMinder がリソースを保護するかどうかを確認	56
CLI エージェント レスポンス メソッド	57
GetAttributes メソッド -- 利用可能なレスポンス属性のリストを取得	57
GetSession メソッド -- レスポンスからセッションを取得	58
レスポンス属性メソッド	59
GetFlags メソッド -- レスポンス属性のフラグを取得	59
GetID メソッド -- レスポンス属性の ID またはエージェント コマンドの ID を取得	59
GetName メソッド -- レスポンス属性の名前を取得	62
GetTTL メソッド -- レスポンス属性の TTL 値を取得	63
GetValue メソッド -- レスポンス属性の値を取得	63
サーバ設定メソッド	64
IPAddress メソッド -- ポリシー サーバの IP アドレスを設定または取得	64
セッション メソッド	65
AddParameter メソッド -- パラメータ リストにセッション変数名と値のペアを追加	65
DelVariables メソッド -- セッションストアからセッション変数を削除	66
GetID メソッド -- セッション ID を取得	67
GetReason メソッド -- セッションの理由 ID を取得	67
GetSpec メソッド -- 暗号化セッション指定情報を取得	69
GetVariables メソッド -- セッションストアからセッション変数を取得	70
IdleTimeout メソッド -- セッションのアイドルタイムアウト値を取得	71
MaxTimeout メソッド -- セッションの最大タイムアウト値を取得	72
SetVariables メソッド -- セッションストアにセッション変数を書き込み	72
シングルサインオン トークン メソッド	73
Decode メソッド -- シングルサインオン トークンをデコード	73
GetString メソッド -- SSO トークンオブジェクトの文字列表現を取得	75
GetVersion メソッド -- SSO トークンの SiteMinder バージョンを取得	76
IsThirdParty メソッド -- トークンがカスタムかどうかを判定	76
ユーザ メソッド	77
Audit メソッド -- エージェント キャッシュから実行された許可を監査	77
Certificate メソッド -- ユーザの X.509 証明書を設定または取得	78
CertificateFile メソッド -- ファイルを使用してユーザの X.509 証明書を設定または取得	79

CreateSSOToken メソッド -- シングル サインオン トークン オブジェクトを作成.....	80
CustomData メソッド -- カスタム認証データを設定または取得.....	81
FormData メソッド -- HTML フォームベースの認証データを設定または取得.....	81
GetResponse メソッド --IsAuthorized または Login 後のレスポンスを返す.....	82
Impersonate メソッド -- 1 人のユーザが別のユーザに偽装することを許可.....	83
IsAuthorized メソッド -- ユーザが許可されるかどうかを判定.....	84
IsAuthorizedEx メソッド -- ユーザが許可されるかどうかを判定.....	86
Login メソッド -- セッション ログインおよび検証を実行	87
Logout メソッド -- ユーザをセッションからログアウト	88
Name メソッド -- ユーザのユーザ名を設定または取得	89
Password メソッド -- ユーザのパスワードを設定または取得.....	90
Validate メソッド -- セッション指定情報を検証.....	90

第 4 章: CLI エージェント操作 93

リソース保護.....	93
レスポンスとレスポンス属性	95
レスポンス属性の取得.....	96
セッション管理.....	97
ポリシー サーバコマンド	99

第 5 章: ポリシー管理 API について 101

ポリシー サーバおよびポリシー管理 API.....	101
ポリシー管理 API の場所.....	101
ポリシー管理 API に対するスクリプトの作成	102
スクリプト実行パフォーマンスの強化.....	102
フェデレーションプログラミング	103
SAML アサーション.....	103
SAML 1.x.....	104
SAML 2.0	106
WS-フェデレーション	111
サンプル スクリプト.....	112
アフィリエイト ドメイン	112
CLI 認証方式の設定	113
設定情報.....	113
設定テーブル.....	115

第 6 章: CLI ポリシー管理メソッド 159

第 7 章: CLI 管理者メソッド 161

AuthScheme メソッド -- 認証方式を設定または取得	161
Description メソッド -- 管理者の説明を設定または取得	162
ManageAllDomains メソッド -- ポリシー サーバオブジェクトを管理する権限を追加または取り消し	163
ManageDomainObjects メソッド -- ドメインオブジェクトを管理する権限を付与または取り消し	164
ManageKeysAndPwdPolicy メソッド -- キーおよびパスワードポリシーを管理する権限を追加または取り消し	165
ManageUsers メソッド -- ユーザを管理する権限を付与または取り消し	166
Name メソッド -- 管理者の名前を設定または取得	166
Password メソッド -- 管理者のパスワードを設定または取得	167
UserDirectory メソッド -- 外部ユーザディレクトリを設定または取得	167
CLI アフィリエイト属性メソッド	168
GetAttrType メソッド -- アフィリエイト属性タイプを取得	168
GetValue メソッド -- アフィリエイト属性の値を取得	169
CLI アフィリエイト ドメイン メソッド	169
AddAdmin メソッド -- アフィリエイト ドメインと管理者を関連付け	169
AddUserDir メソッド -- アフィリエイト ドメインとユーザディレクトリを関連付け	170
CreateAffiliate メソッド -- アフィリエイトオブジェクトを作成します	171
CreateSAMLServiceProvider メソッド -- SAML サービスプロバイダを作成	174
CreateWSFEDResourcePartner メソッド -- WS-Federation リソース パートナーを作成	177
DeleteAffiliate メソッド -- ドメインからアフィリエイトを削除	179
DeleteSAMLServiceProvider メソッド -- SAML サービスプロバイダを削除	180
DeleteWSFEDResourcePartner メソッド -- リソース パートナーを削除	181
Description メソッド -- 説明を取得または設定	181
GetAffiliate メソッド -- アフィリエイトオブジェクトを取得	182
GetAllAdmins メソッド -- 管理者をすべて取得	183
GetAllAffiliates メソッド -- ドメイン内のすべてのアフィリエイトを取得	183
GetAllSAMLServiceProviders メソッド -- アフィリエイト ドメインと関連付けられたすべてのサービスプロバイダを取得	184
GetAllWSFEDResourcePartners メソッド -- すべての WSFED リソース パートナーを取得	184
GetSAMLServiceProvider メソッド -- 指定されたサービスプロバイダを取得	185
GetSAMLServiceProviderByID メソッド -- 指定されたサービスプロバイダを取得	186
GetUserDirSearchOrder メソッド -- ユーザディレクトリの検索順序を取得	186
GetWSFEDResourcePartner メソッド -- リソース パートナーを取得	187
GetWSFEDResourcePartnerById メソッド -- ID によってリソース パートナーを取得	188

Name	メソッド -- アフィリエイト ドメイン名を設定または取得.....	188
RemoveAdmin	メソッド -- アフィリエイト ドメインから管理者の関連付けを解除.....	189
RemoveUserDir	メソッド -- アフィリエイト ドメインからユーザ ディレクトリの関連付けを解 除	190
SetUserDirSearchOrder	メソッド -- ディレクトリ オブジェクトの検索順序を設定.....	190
CLI	アフィリエイト オブジェクト メソッド.....	191
AddAttribute	メソッド -- 新しいアフィリエイト属性を追加.....	191
AddUser	メソッド -- アフィリエイト オブジェクトに新しいユーザを追加	194
AllowNotification	メソッド -- イベント通知プロパティを設定または取得	194
AssertionPluginClass	メソッド -- アサーション生成プログラム プラグインの名前を設定または 取得	195
AssertionPluginParameters	メソッド -- パラメータ文字列を設定または取得	196
Audience	メソッド -- URI を設定または取得	197
AuthURL	メソッド -- URL を設定または取得	198
ConsumerURL	メソッド -- URL を設定または取得.....	198
CreateIPHostConfigName	メソッド -- 指定されたホスト名から IP 設定オブジェクトを作成	199
CreateIPConfigRange	メソッド -- IP 設定 オブジェクトを作成.....	200
CreateIPConfigSingleHost	メソッド -- 指定された IP アドレスから IP 設定オブジェクトを作成	201
CreateIPConfigSubnetMask	メソッド -- IP 設定 オブジェクトを作成.....	201
DeleteIPConfig	メソッド -- IP 設定 オブジェクトを削除.....	202
Description	メソッド -- アフィリエイト オブジェクトの説明を設定または取得.....	203
GetAllAttributes	メソッド -- アフィリエイト オブジェクトの属性を取得	203
GetAllIPConfigs	メソッド -- アフィリエイトのすべての IP 設定オブジェクトを取得	204
GetAllUsers	メソッド -- アフィリエイトと関連付けられたすべてのユーザを取得	205
IsEnabled	メソッド -- アフィリエイトの有効化フラグを設定または取得.....	205
Name	メソッド -- アフィリエイト名を設定または取得	206
Password	メソッド -- アフィリエイトのパスワードを設定または取得.....	207
RemoveAttribute	メソッド -- アフィリエイトから属性を削除	207
RemoveUser	メソッド -- アフィリエイトからユーザを削除	208
SAMLProfile	メソッド -- SAML プロファイルのタイプを設定または取得	208
SAMLVersion	メソッド -- アフィリエイトの SAML バージョンを設定または取得.....	209
Save	メソッド -- ポリシー ストアにアフィリエイトを保存.....	210
SessionSyncInterval	メソッド -- セッション同期プロパティを設定または取得	211
SharedSessioning	メソッド -- 共有セッションプロパティを設定または取得	212
SkewTime	メソッド -- スキュー時間プロパティを設定または取得.....	213
ValidityDuration	メソッド -- SAML アサーションが有効な期間を設定または取得.....	213
CLI	エージェント メソッド.....	214
ConvertFromLegacy	メソッド -- v4.x エージェントを v5.x エージェントに変換	214
ConvertToLegacy	メソッド -- v5.x エージェントを v4.x エージェントに変換.....	215
Description	メソッド -- エージェントの説明を設定または取得.....	215

IPAddress メソッド -- エージェントの IP アドレスを設定または取得	216
Name メソッド -- エージェントの名前を設定または取得	216
RealmHintAttrID メソッド -- ヒント属性を設定または取得	217
SharedSecret メソッド -- v4.x エージェント用の共有秘密キーを設定または取得	218
CLI エージェントの設定方法	218
AddAssociation メソッド -- この設定用の名前と値を追加	218
AddAssociationMultiValue メソッド -- 複数值設定パラメータを追加	219
Description メソッド -- エージェント設定オブジェクトの説明を設定または取得	220
GetAssociations メソッド -- すべての設定パラメータのリストを取得	221
Name メソッド -- エージェント設定オブジェクト名を設定または取得	221
RemoveAssociation メソッド -- 設定パラメータを削除	222
CLI エージェント設定パラメータ メソッド	223
Name メソッド -- エージェント設定パラメータの名前部分を設定または取得	223
Flags メソッド -- 暗号化フラグ属性を設定または取得	223
Value メソッド -- エージェント設定パラメータの値を設定または取得	224
CLI エージェント タイプ メソッド	225
GetDescription メソッド -- エージェント タイプの説明を取得	225
GetName メソッド -- エージェント タイプの名前を取得	225
CLI 認証および許可マップ メソッド	226
AuthDir メソッド -- 認証ディレクトリを設定または取得	226
AzDir メソッド -- 許可ディレクトリを設定または取得	227
MapType メソッド -- 認証および許可マップのタイプを設定または取得	227
CLI 認証方式メソッド	228
AddMessageConsumerPluginToSAML1xScheme メソッド -- メッセージ コンシューマ プラグイン クラス名を追加	228
AddRedirectURLToSAML1xScheme メソッド -- 認証方式にリダイレクト値を追加	229
AddTargetConfigToSAML1xScheme メソッド -- デフォルト ターゲット設定を設定	230
CustomLib メソッド -- 共有ライブラリの名前を設定または取得	231
CustomParam メソッド -- 認証方式に渡される情報を設定または取得	232
CustomSecret メソッド -- カスタム認証方式用の共有秘密キーを設定または取得	233
Description メソッド -- 認証方式の説明を設定または取得	233
GetMessageConsumerPluginFromSAML1xScheme メソッド -- メッセージ コンシューマ プラグイ ンクラス名を取得	234
GetRedirectURLFromSAML1xScheme メソッド -- リダイレクト URL を取得	235
GetTargetConfigFromSAML1xScheme メソッド -- デフォルト ターゲット設定を取得	236
IgnorePwd メソッド -- パスワード ポリシーを確認するかどうかを指定	236
IsRadius メソッド -- 認証方式で RADIUS エージェントをサポートするかどうかを判定	237
IsTemplate メソッド -- 認証方式がテンプレートかどうかを判定	238
IsUsedByAdmin メソッド -- 方式で管理者を認証するかどうかを判定	239
Name メソッド -- 認証方式の名前を設定または取得	240

ProtectionLevel メソッド -- 認証方式の保護レベルを設定または取得	240
Save メソッド -- ポリシーストアに認証方式を保存	241
SaveCredentials メソッド -- ユーザ認証情報を保存できるかどうかを判定	242
Type メソッド -- 認証方式タイプを設定または取得	242
CLI 証明書マッピング メソッド	243
AttrMap メソッド -- 証明書マッピングの属性マップを設定または取得	243
CacheCRL メソッド -- キャッシュ証明書失効リスト (CRL) エントリをキャッシュに入れるかどうかを判定	244
CertRequired メソッド -- 証明書の検証が必要かどうかを判定	245
CRLUserDirectory メソッド -- 証明書失効リスト (CRL) が存在する LDAP ディレクトリを設定または取得	245
Description メソッド -- 証明書マップの説明を設定または取得	246
DirectoryType メソッド -- ユーザディレクトリのタイプを設定または取得	247
EnableCRL メソッド -- 取り消された証明書に対し証明書失効リスト (CRL) を確認するかどうかを判定	247
IssuerDN メソッド -- 証明書発行者の DN を設定または取得	248
UseDistributionPoints メソッド -- 証明書失効リスト (CRL) 検索で配布ポイントを使用するかどうかを判定	249
VerifySignature メソッド -- SiteMinder で認証機関の署名を検証するかどうかを判定	250
CLI クラスタ メソッド	251
AddServer メソッド -- クラスタにサーバを追加	251
GetAllServers メソッド -- クラスタ内のすべてのサーバの配列を取得	252
CLI ドメイン メソッド	252
AddAdmin メソッド -- ドメインに管理者を追加	252
AddUserDir メソッド -- ドメインとユーザディレクトリを関連付け	253
CreatePolicy メソッド -- ドメイン内のポリシーを作成および設定	254
CreateRealm メソッド -- ドメイン内にトップレベル レalmを作成および設定	255
CreateResponse メソッド -- レスポンスを作成	258
CreateResponseGroup メソッド -- ドメイン用のレスポンス グループを作成	259
CreateRuleGroup メソッド -- ドメインのルール グループを作成	260
DeleteGroup メソッド -- ドメインからグループを削除	261
DeletePolicy メソッド -- ポリシーを削除	261
DeleteRealm メソッド -- ドメイン内のレalmを削除	262
DeleteResponse メソッド -- レスポンスを削除	262
Description メソッド -- ドメインの説明を設定または取得	263
GetAllPolicies メソッド -- ドメインと関連付けられたすべてのポリシーを取得	264
GetAllRealms メソッド -- ドメイン内のすべてのトップレベル レalmを取得	264
GetAllResponseGroups メソッド -- ドメインと関連付けられたすべてのレスポンス グループを取得	265
GetAllResponses メソッド -- ドメインと関連付けられたすべてのレスポンスを取得	265

GetAllRuleGroups メソッド-- ドメインと関連付けられたすべてのルール グループを取得.....	266
GetPolicy メソッド-- ドメイン内のポリシーを取得.....	266
GetRealm メソッド-- ドメイン内のトップレベル レalmを取得.....	267
GetResponse メソッド-- ドメインと関連付けられたレスポンスを取得.....	268
GetResponseGroup メソッド-- 指定されたレスポンス グループを取得.....	268
GetRuleGroup メソッド-- 指定されたルール グループを取得.....	269
GetUserDirSearchOrder メソッド-- 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 クトを取得.....	269
GlobalPoliciesApply メソッド-- グローバル ポリシーに対してドメインが有効かどうかを判定.....	270
Name メソッド-- ドメイン名を設定または取得.....	271
RemoveAdmin メソッド-- ドメインから管理者の関連付けを解除.....	271
RemoveUserDir メソッド-- ドメインからユーザ ディレクトリの関連付けを解除.....	272
SetUserDirSearchOrder-- ユーザ ディレクトリ オブジェクトの検索順序を再調整.....	273
CLI グループ メソッド.....	274
Add メソッド-- エージェント、レスポンス、ルールまたはネストされたグループ オブジェク トをグループに追加.....	274
Contains メソッド-- グループに指定されたエージェント、レスポンス、ルールまたはネストさ れたグループ オブジェクトが含まれるかどうかを判定.....	275
Description メソッド-- グループ オブジェクトの説明を設定または取得.....	276
GetAgent メソッド-- グループから指定されたエージェント オブジェクトを取得.....	276
GetAgentGroup メソッド-- グループ内のネストされたエージェント グループ オブジェクトを 取得.....	277
GetAgentType メソッド-- グループに含まれているエージェント オブジェクトのタイプを取得.....	278
GetAllAgentGroups メソッド-- グループ内のネストされたすべてのエージェント グループ オ ブジェクトを取得.....	278
GetAllAgents メソッド-- グループ内のすべてのエージェント オブジェクトを取得.....	279
GetAllResponseGroups メソッド-- グループ内のネストされたすべてのレスポンス グループ オ ブジェクトを取得.....	279
GetAllResponses メソッド-- グループ内のすべてのレスポンス オブジェクトを取得.....	280
GetAllRuleGroups メソッド-- グループ内のネストされたすべてのルール グループ オブジェク トを取得.....	281
GetAllRules メソッド-- グループ内のすべてのルール オブジェクトを取得.....	281
GetResponse メソッド-- グループから指定されたレスポンス オブジェクトを取得.....	282
GetResponseGroup メソッド-- グループ内のネストされたレスポンス グループ オブジェク トを取得.....	283
GetRule メソッド-- グループから指定されたルール オブジェクトを取得.....	283
GetRuleGroup メソッド-- グループ内のネストされたルール グループ オブジェクトを取得.....	284
Name メソッド-- グループ オブジェクトの名前を設定または取得.....	285
Remove メソッド-- グループから指定されたグループ メンバを削除.....	285
CLI ホスト設定メソッド.....	286
AddCluster メソッド-- ホスト設定に空のクラスタを追加.....	286

AddServer メソッド -- ホスト設定に非クラスタ化サーバを追加	287
Description メソッド -- ホスト設定オブジェクトの説明を設定または取得	288
EnableFailover メソッド -- フェールオーバー有効化フラグを設定または取得	289
FailoverThreshold メソッド -- フェールオーバーしきい値パーセンテージを設定または取得	290
GetAllClusters メソッド -- ポリシー管理クラスタ オブジェクトの配列を取得	291
GetAllServers メソッド -- 非クラスタ化サーバオブジェクトの配列を取得	291
MaxSocketsPerPort メソッド -- TCP/IP ソケットの最大数を設定または取得	292
MinSocketsPerPort メソッド -- TCP/IP ソケットの最小数を設定または取得	293
Name メソッド -- ホスト設定オブジェクトの名前を設定または取得	293
NewSocketStep メソッド -- ホスト設定の新しいソケット ステップ値を設定または取得	294
RemoveAllClusters メソッド -- このホスト設定と関連付けられたすべてのクラスタ オブジェ クトを削除	295
RemoveAllServers メソッド -- ホスト設定からすべての非クラスタ化ポリシー サーバ オブジェ クトを削除	295
RequestTimeout メソッド -- 要求タイムアウト値を設定または取得	296
CLI 初期化メソッド	297
CreateSession メソッド -- ポリシー サーバセッションを作成	297
DisableAudit メソッド -- 監査を有効または無効にするフラグを設定	297
DisableCacheUpdates メソッド -- 非推奨	298
DisableManagementWatchDog メソッド -- SiteMinder 管理ウォッチドッグの有効化状態を読み 込みまたは設定	299
DisableValidation メソッド -- ポリシー サーバオブジェクトの検証の有効化状態を読み込みま たは設定	300
EnableCache メソッド -- 非推奨	300
LoadAgentTypeDictionary メソッド -- エージェント タイプ ディクショナリの有効化状態を読み 込みまたは設定	301
New メソッド -- ポリシー管理 API のコンストラクタ	302
PreLoadCache メソッド -- キャッシュの事前ロードの有効化状態を読み込みまたは設定	302
PrintDebugTrace メソッド -- デバッグ (トレース) 情報のサンプルの出力を有効化または無効 化	303
CLI IP 設定メソッド	304
GetEndIPAddress メソッド -- 終了 IP アドレスを取得	304
GetHostName メソッド -- ホスト名 IP アドレス制限と関連付けられたホスト名を取得	305
GetIPAddress メソッド -- IP アドレス制限の IP アドレスを取得	306
GetSubnetMask メソッド -- サブネットアドレス用のサブネット マスクを取得	307
GetType メソッド -- IP アドレス制限のタイプを取得	307
CLI ODBC クエリ方式メソッド	308
Description メソッド -- ODBC クエリ方式の説明を設定または取得	308
Name メソッド -- ODBC クエリ方式名を設定または取得	309
QueryAuthenticateUser メソッド -- ユーザのパスワードをフェッチするクエリを設定または取 得	309

QueryEnumerate メソッド -- ユーザ オブジェクトの名前をリストするクエリを設定または取得	310
QueryGetGroupProp メソッド -- グループ プロパティの値をフェッチするクエリを設定または取得	311
QueryGetGroupProps メソッド -- グループ プロパティのリストを設定または取得	312
QueryGetGroups メソッド -- ユーザがメンバであるグループの名前をフェッチするクエリを設定または取得	313
QueryGetObjInfo メソッド -- オブジェクトのクラスをフェッチするクエリを設定または取得	314
QueryGetUserProp メソッド -- ユーザ プロパティの値をフェッチするクエリを設定または取得	315
QueryGetUserProps メソッド -- ユーザ プロパティのリストを設定または取得	316
QueryInitUser メソッド -- データベースにユーザが存在するかどうかを判定するクエリを設定または取得	316
QueryIsGroupMember メソッド -- 特定のユーザのグループ メンバシップをリストするクエリを設定または取得	317
QueryLookup メソッド -- オブジェクトをフェッチするクエリを設定または取得	318
QueryLookupGroup メソッド -- グループ名をフェッチするクエリを設定または取得	319
QueryLookupUser メソッド -- ユーザ名をフェッチするクエリを設定または取得	320
QuerySetGroupProp メソッド -- グループ プロパティの値を設定するクエリを設定または取得	321
QuerySetPassword メソッド -- ユーザ パスワードを変更するクエリを設定または取得	322
QuerySetUserProp メソッド -- ユーザ プロパティの値を設定するクエリを設定または取得	323
CLI パスワード ポリシー メソッド	324
AllowNestedGroups メソッド -- パスワード ポリシーのネストされたグループに対する設定を許可	324
AllowLowerPriorityPolicies メソッド -- 優先度の低いパスワード ポリシーを評価するかどうかを判定するフラグを設定	325
AuthLoginTrackFailure メソッド -- ログイン追跡データが失敗した場合にユーザのログインを許可	326
BadLoginDisablementPeriod メソッド -- ユーザ アカウントが無効になるまでの時間数 (分) を設定または取得	327
Description メソッド -- パスワード ポリシーの説明を設定または取得	327
DictionaryMatch メソッド -- ディクショナリ確認用のパスワードに資格を与えるのに必要な文字の最小数を設定	328
DictionaryPath メソッド -- ディクショナリ ファイルの場所を設定または取得	329
DisableAfterInactivityExpiration メソッド -- 非アクティブ ユーザのアカウントを無効化	329
DisableAfterPwdExpiration メソッド -- ユーザのパスワードが期限切れになった後にユーザのアカウントを無効化	330
EntireDir メソッド -- パスワード ポリシーをディレクトリ全体に適用するかどうかを判定	331
ExpirationDelay メソッド -- パスワードを使用できる日数を指定	332
IsEnabled メソッド -- パスワード ポリシーを有効化または無効化	333
MaxLoginFailures メソッド -- ログイン試行に失敗できる最大回数を設定または取得	334
MaxLoginInactive メソッド -- 非アクティブ状態が許可される日数を設定または取得	334

Name メソッド -- パスワード ポリシー名を設定または取得	335
PwdAddRegExpMatch メソッド -- 新しいパスワードが一致する必要がある表現のリストに正規表現を追加.....	335
PwdAddRegExpNoMatch メソッド -- 新しいパスワードが一致してはならない表現のリストに正規表現を追加.....	336
PwdAllowDigits メソッド -- パスワードに数字を使用できるかどうかを指定	337
PwdAllowLowercase メソッド -- パスワードに小文字を使用できるかどうかを指定	338
PwdAllowNonAlphaNum メソッド -- パスワードに英数字以外の文字を使用できるかどうかを指定	338
PwdAllowNonPrintable メソッド -- パスワードに印刷不能文字を使用できるかどうかを指定	339
PwdAllowPunctuation メソッド -- パスワードに句読点文字を使用できるかどうかを指定.....	340
PwdAllowUpperCase メソッド -- パスワードに大文字を使用できるかどうかを指定.....	341
PwdExpiryWarning メソッド -- パスワードが期限切れになることをユーザに事前に通知する場合の日数を設定または取得.....	341
PwdForceLowerCase Method -- 新しいパスワードの大文字を小文字に変換するかどうかを判定	342
PwdForceUpperCase Method -- 新しいパスワードの小文字を大文字に変換するかどうかを判定	343
PwdGetAllRegExpMatch メソッド -- 新しいパスワードが一致する必要がある正規表現の名前タグを取得.....	343
PwdGetAllRegExpNoMatch メソッド -- 新しいパスワードが一致してはならない正規表現の名前タグを取得.....	344
PwdGetRegExp メソッド -- 指定された名前タグ用の正規表現を取得	345
PwdIgnoreSequence メソッド -- 新しいパスワードを計算するときにシーケンスを無視するかどうかを判定.....	345
PwdMaxLength メソッド -- ユーザ パスワードの最大長を設定または取得	346
PwdMaxRepeatingChar メソッド -- 同一文字の最大数を設定または取得	347
PwdMinAlpha メソッド -- パスワードに使用する必要があるアルファベット文字の最小数を設定または取得.....	348
PwdMinAlphaNum メソッド -- パスワードに使用する必要がある英数字の最小数を設定または取得	348
PwdMinLength メソッド -- ユーザ パスワードの最小長を設定または取得.....	349
PwdMinLowercase メソッド -- パスワードに使用する必要がある小文字の最小数を設定または取得	349
PwdMinNonAlpha メソッド -- パスワードに使用する必要がある英数字以外の文字の最小数を設定または取得.....	350
PwdMinNonPrintable メソッド -- パスワードに使用する必要がある印刷不能文字の最小数を設定または取得.....	351
PwdMinNumbers メソッド -- パスワードに使用する必要がある数字の最小数を設定または取得	351
PwdMinProfileMatch メソッド -- ユーザの個人情報と照合する最小文字シーケンスを指定	352
PwdMinPunctuation メソッド -- パスワードに使用する必要がある句読点の最小数を設定または取得.....	353

PwdMinUppercase メソッド -- パスワードに使用する必要がある大文字の最小数を設定または取得	353
PwdPercentDiff メソッド -- 新しいパスワードに使用する必要がある異なる文字のパーセンテージを設定または取得	354
PwdPolicyPriority メソッド -- パスワードの評価優先度設定を設定または取得	355
PwdRedirectionURL メソッド -- ユーザがリダイレクトされる場合の URL を設定または取得	355
PwdRemoveRegExp メソッド -- 指定された名前タグに関連付けられた正規表現を削除	356
PwdReuseCount メソッド -- 使用する必要がある新しいパスワードの数を指定	357
PwdReuseDelay メソッド -- パスワードを再利用する前にユーザが待機する必要がある日数を指定	357
ReEnableAfterIncorrectPwd メソッド -- 不正なパスワードの入力後にユーザ アカウントを再有効化するかどうかを判定	358
Save メソッド -- ポリシー ストアにパスワード ポリシーを保存	359
StripEmbeddedWhitespace メソッド -- 新しいパスワードから埋め込まれた空白を取り去るかどうかを判定	359
StripLeadingWhitespace メソッド -- 新しいパスワードから先頭の空白を取り去るかどうかを判定	360
StripTrailingWhitespace メソッド -- 新しいパスワードから後続の空白を取り去るかどうかを判定	361
TrackLoginDetails メソッド -- 認証試行および成功したログインを追跡するかどうかを判定	361
UserDirClass メソッド -- パスワード ポリシーがディレクトリの一部に適用される場合のディレクトリ クラスを設定または取得	362
UserDirectory メソッド -- パスワード ポリシーのユーザ ディレクトリを設定または取得	363
UserDirPath メソッド -- パスワード ポリシーがディレクトリの一部に適用される場合のディレクトリ パスを設定または取得	363
CLI ポリシー メソッド	364
ActiveExpr メソッド -- ポリシーに関連付けられたアクティブな式を設定または取得	364
AddRule メソッド -- ポリシーにルールを追加	364
AddUser メソッド -- ポリシーにユーザを追加	365
AllowNested メソッド -- AllowNested フラグを設定または取得	366
CreateIPHostConfigName メソッド -- IP アドレス設定を作成	367
CreateIPConfigRange メソッド -- IP アドレス設定を作成	368
CreateIPConfigSingleHost メソッド -- IP アドレス設定を作成	369
CreateIPConfigSubnetMask メソッド -- IP アドレスおよびサブネット マスクに基づいて IP アドレス設定を作成	369
DeleteIPConfig メソッド -- 指定された IP 設定 オブジェクトを削除	370
Description メソッド -- ポリシーの説明を設定または取得	371
EnforceANDEvaluation メソッド -- ANDUser/Group フラグを設定または取得	372
ExcludeUser メソッド -- ポリシーからユーザを除外または含む	373
GetAllIPConfigs メソッド -- ポリシー内のすべての IP アドレス制限オブジェクトを取得	374
GetAllRules メソッド -- ポリシーと関連付けられたすべてのルールを取得	374

GetAllUsers メソッド -- ポリシーと関連付けられたすべてのユーザを取得	375
IsEnabled メソッド -- ポリシーを有効化または無効化	376
Name メソッド -- ポリシー名を設定または取得	376
RemoveResponse メソッド -- ポリシー内の設定されたルール用のレスポンスを削除	377
RemoveRule メソッド -- ポリシーから指定されたルールを削除	378
RemoveUser メソッド -- ポリシーからユーザを削除	378
SetResponse メソッド -- ポリシー内の設定されたルール用のレスポンスを設定	379
CLI ポリシー サーバ接続メソッド	380
GetPorts メソッド -- 非推奨	380
GetServerAddress メソッド -- ポリシー サーバのホスト名または IP アドレスを取得	380
GetServerPort メソッド -- ポリシー サーバまたはサーバクラスタの TCP ポートを取得	381
CLI レルム メソッド	382
Agent メソッド -- レルムのエージェントを設定または取得	382
AuthScheme メソッド -- レルムの認証方式を設定または取得	382
AzUserDir メソッド -- レルムの許可ユーザディレクトリを設定または取得	383
CreateChildRealm メソッド -- 子レルムを作成および設定	384
CreateRule メソッド -- レルム下のルールを作成および設定	386
DeleteChildRealm メソッド -- レルム内のトップレベルレルムを削除	388
DeleteRule メソッド -- レルム内の既存のルールを削除	389
Description メソッド -- レルムの説明を設定または取得	389
Flush メソッド -- リソース キャッシュからレルムをフラッシュ	390
GetAllChildRealms メソッド -- レルム内のすべてのトップレベルレルムを取得	391
GetAllRules メソッド -- レルムと関連付けられたルールを取得	391
GetChildRealm メソッド -- レルム下のトップレベルの子レルムを取得	392
GetDomain メソッド -- レルムと関連付けられたドメインを取得	392
GetRule メソッド -- レルム内の既存のルールを取得	393
IdleTimeout メソッド -- ユーザがレルムで非アクティブ状態を維持できる最大時間を設定または取得	394
MaxTimeout メソッド -- ユーザがレルムにアクセスできる最大時間を設定または取得	394
Name メソッド -- レルム名を設定または取得	395
ProcessAuEvents メソッド -- レルム内の認証イベント フラグを設定または取得	396
ProcessAzEvents メソッド -- レルム内の許可イベント フラグを設定または取得	396
ProtectResource メソッド -- 現在のリソース保護フラグの例を設定または取得	397
RegScheme メソッド -- レルムの登録方式を設定または取得	398
SessionDrift メソッド -- セッションドリフトを設定または取得	399
ResourceFilter メソッド -- レルム リソース フィルタを設定または取得	399
SyncAudit メソッド -- 同期監査フラグを設定または取得	400
CLI 登録方式メソッド	401
Description メソッド -- 登録方式の説明を設定または取得	401
EnableLogging メソッド -- 登録方式のログ記録を有効化または無効化	402

Name メソッド -- 登録方式名を設定または取得	402
TemplatePath メソッド -- 登録方式のパスを設定または取得	403
UserDirectory メソッド -- 登録方式のユーザディレクトリを設定または取得	404
WelcomePageURL メソッド -- 登録方式のようこそページ URL を設定または取得	404
CLI ポリシー管理 レスポンス メソッド	405
CreateAttribute メソッド -- レスポンスのスタティック レスポンス属性を作成	405
DeleteAttribute メソッド -- レスポンス内のレスポンス属性を削除	407
Description メソッド -- レスポンスの説明を設定または取得	408
GetAllAttributes メソッド -- 設定されたレスポンス属性のリストを取得	408
Name メソッド -- レスポンス名を設定または取得	409
CLI レスポンス属性メソッド	410
GetAgentTypeAttrName メソッド -- エージェント タイプ属性の名前を取得	410
GetTTL メソッド -- Time To Live (TTL) 設定を取得	410
GetValue メソッド -- レスポンス属性値を取得	411
CLI ルール メソッド	411
AccessType メソッド -- 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を設定または取得	411
Action メソッド -- ルールのアクションを設定または取得	412
ActiveExpr メソッド -- ルールのアクティブな式を設定または取得	414
Agent メソッド -- グローバルルールと関連付けられたエージェント オブジェクトまたはエージェント グループ オブジェクトを設定または取得	414
Description メソッド -- ルールの説明を設定または取得	415
IsEnabled メソッド -- ルールを有効化または無効化	416
Name メソッド -- ルール名を設定または取得	417
RegexMatch メソッド -- 正規表現パターン一致が有効かどうかを判定	417
Resource メソッド -- ルールによって保護されたリソースを設定または取得	418
CLI SAML 2.0 アフィリエーション メソッド	419
GetAffiliatedSAMLAuthSchemes メソッド -- この SAML アフィリエーションと関連付けられた SAML 2.0 認証方式を取得	419
GetAffiliatedSAMLServiceProviders メソッド -- この SAML アフィリエーションと関連付けられた SAML 2.0 サービス プロバイダを取得	419
Property メソッド -- 指定された SAML 2.0 メタデータ プロパティを設定または取得	420
Save メソッド -- この SAML 2.0 アフィリエーションの SAML 2.0 メタデータ プロパティの変更を保存	421
CLI SAML 2.0 インデックス化エンドポイント メソッド	422
GetACSIndex メソッド -- アサーション コンシューマ サービス オブジェクトのインデックス値を取得	422
GetACSBinding メソッド -- アサーション コンシューマ サービス オブジェクトのプロトコルバインディングを取得	422
GetACSURL メソッド -- アサーション コンシューマ サービス オブジェクトの URL 値を取得	423

GetIsDefault メソッド -- アサーション コンシューマ サービス オブジェクトの IsDefault 値を取 得	423
CLI SAML 2.0 リクエスト属性メソッド	424
GetAttrNameFormat メソッド -- SAML リクエスト属性の名前形式を取得	424
GetLocalName メソッド -- SAML リクエスト属性のローカル名を取得	425
GetName メソッド -- SAML リクエスト属性の名前を取得	425
CLI SAML 2.0 サービス プロバイダ メソッド	426
AddAssertionConsumerService メソッド -- アサーション コンシューマ サービスを SAML サービ ス プロバイダ オブジェクトに追加	427
AddAttribute メソッド -- 属性を SAML 2.0 サービス プロバイダに追加	428
AddUser メソッド -- ユーザを SAML 2.0 サービス プロバイダに追加	430
CreateIPConfigHostName メソッド -- サービス プロバイダの IP 設定オブジェクトを作成	430
CreateIPConfigRange メソッド -- サービス プロバイダの IP 設定オブジェクトを作成	431
CreateIPConfigSingleHost メソッド -- サービス プロバイダの IP 設定オブジェクトを作成	432
CreateIPConfigSubnetMask メソッド -- サービス プロバイダの IP 設定オブジェクトを作成	433
DeleteIPConfig メソッド -- 指定された IP 設定 オブジェクトを削除	434
GetAllAttributes メソッド -- SAML 2.0 サービス プロバイダのすべての属性を取得	435
GetAllIPConfigs メソッド -- すべての IP 設定オブジェクトを取得	435
GetAllAssertionConsumerServices メソッド -- すべてのアサーション コンシューマ サービスを 取得	436
GetAllUsers メソッド -- すべてのユーザを取得	436
Property メソッド -- メタデータ プロパティを設定または取得	437
RemoveAssertionConsumerService メソッド -- アサーション コンシューマ サービスを削除	438
RemoveAttribute Method -- 指定された属性を削除	439
RemoveUser Method -- 指定されたユーザを削除	439
Save メソッド -- メタデータ プロパティに対して行った変更を保存	440
CLI SAML 2.0 サービス プロバイダ属性メソッド	441
GetAttrNameFormat メソッド -- 属性名の形式を取得	441
GetValue メソッド -- サービス プロバイダ属性値を取得	442
CLI セッション メソッド	443
AddAttributeToSAMLScheme メソッド -- 新しい属性を認証方式に追加	443
AddTrustedHost メソッド -- トラステッド ホスト オブジェクトを作成または変更	444
CreateAdmin メソッド -- システムレベル管理者を作成	445
CreateAffDomain メソッド -- アフィリエイト ドメインを作成	447
CreateAgent メソッド -- SiteMinder エージェントを作成	447
CreateAgentConfig メソッド -- エージェント設定オブジェクトを作成	449
CreateAgentGroup メソッド -- エージェント グループを作成	449
CreateAuthAzMap メソッド -- ディレクトリ マッピング オブジェクトを作成	450
CreateAuthScheme メソッド -- 認証方式を作成	452
CreateCustomCertMap メソッド -- カスタム証明書マップを作成	454

CreateDomain メソッド--ポリシー ドメインオブジェクトを作成.....	455
CreateExactCertMap メソッド--ユーザディレクトリ属性に一致する証明書マップを作成.....	457
CreateGlobalPolicy メソッド--グローバルポリシーを作成.....	458
CreateGlobalResponse メソッド--グローバルレスポンスを作成.....	459
CreateGlobalResponseGroup メソッド--ドメイン固有のルールグループを作成.....	460
CreateGlobalRule メソッド--グローバルルールを作成.....	461
CreateGlobalRuleGroup メソッド--ドメイン固有のルールグループを作成.....	463
CreateHostConfig メソッド--ホスト設定オブジェクトを作成.....	463
CreateODBCQueryScheme メソッド--ODBC クエリ方式を作成.....	465
CreatePwdPolicy メソッド--パスワードポリシーを作成.....	469
CreateRegScheme メソッド--登録方式を作成.....	473
CreateSAMLAffiliation メソッド--SAML 2.0 アフィリエーションオブジェクトを作成.....	475
CreateSAMLAuthScheme メソッド--SAML 認証方式オブジェクトを作成.....	477
CreateSingleCertMap メソッド--単一属性証明書マップを作成.....	481
CreateTrustedHost メソッド--トラステッドホストオブジェクトを作成.....	483
CreateUserDir メソッド--ユーザディレクトリオブジェクトを作成.....	484
CreateWSFEDAuthScheme メソッド--WS-Federation 認証方式を作成.....	489
DeleteAdmin メソッド--管理者を削除.....	491
DeleteAffDomain メソッド--アフィリエイトドメインを削除.....	492
DeleteAgent メソッド--エージェントを削除.....	493
DeleteAgentConfig メソッド--エージェント設定オブジェクトを削除.....	493
DeleteAuthAzMap メソッド--認証および許可マップを削除.....	494
DeleteAuthScheme メソッド--認証方式を削除.....	495
DeleteCertMap メソッド--証明書マップを削除.....	496
DeleteDomain メソッド--ポリシードメインを削除.....	496
DeleteGlobalPolicy メソッド--グローバルポリシーを削除.....	497
DeleteGlobalResponse メソッド--グローバルレスポンスを削除.....	498
DeleteGlobalRule メソッド--グローバルルールを削除.....	499
DeleteGroup メソッド--エージェントグループを削除.....	499
DeleteHostConfig メソッド--ホスト設定オブジェクトを削除.....	500
DeleteODBCQueryScheme メソッド--ODBC クエリ方式を削除.....	501
DeletePwdPolicy メソッド--パスワードポリシーを削除.....	501
DeleteRegScheme メソッド--登録方式を削除.....	502
DeleteSAMLAffiliation メソッド--SAML アフィリエーションを削除.....	503
DeleteTrustedHost メソッド--トラステッドホストを削除.....	504
DeleteUserDir メソッド--ユーザディレクトリを削除.....	504
GetAdmin メソッド--管理者を取得.....	505
GetAffDomain メソッド--アフィリエイトドメインを取得.....	506
GetAgent メソッド--エージェントを取得.....	506
GetAgentConfig メソッド--エージェント設定オブジェクトを取得.....	507

GetAgentGroup メソッド-- エージェント グループを取得	508
GetAgentType メソッド-- エージェント タイプを取得	508
GetAllAdmins メソッド-- すべての管理者のリストを取得	510
GetAllAffDomains メソッド-- すべてのアフィリエイト ドメインのリストを取得	510
GetAllAgentConfigs メソッド-- すべてのエージェント設定オブジェクトのリストを取得	511
GetAllAgentGroups メソッド-- すべてのエージェント グループ オブジェクトのリストを取得	512
GetAllAgents メソッド-- すべてのエージェントのリストを取得	512
GetAllAuthAzMaps メソッド-- すべての AuthAz マップのリストを取得	513
GetAllAuthSchemes メソッド-- 認証方式のリストを取得	513
GetAllCertMaps メソッド-- 証明書マッピング オブジェクトのリストを取得	514
GetAllDomains メソッド-- すべてのドメインのリストを取得	515
GetAllGlobalPolicies メソッド-- グローバル ポリシー オブジェクトのリストを取得	515
GetAllGlobalResponses メソッド-- すべてのグローバル レスポンス オブジェクトのリストを取 得	516
GetAllGlobalRules メソッド-- すべてのグローバル ルール オブジェクトのリストを取得	517
GetAllHostConfigs メソッド-- すべてのホスト設定オブジェクトのリストを取得	517
GetAllODBCQuerySchemes メソッド-- すべての ODBC クエリ方式のリストを取得	518
GetAllPwdPolicies メソッド-- すべてのパスワード ポリシーのリストを取得	518
GetAllRegSchemes メソッド-- すべての登録方式のリストを取得	519
GetAllSAMLAffiliations メソッド-- すべての SAML 2.0 アフィリエーションのリストを取得	520
GetAllSAMLSchemeAttributes メソッド-- すべてのリクエスト属性のリストを取得	520
GetAllTrustedHosts メソッド-- すべてのトラステッドホスト オブジェクトのリストを取得	521
GetAllUserDirs メソッド-- すべてのユーザディレクトリのリストを取得	521
GetAllVariableTypes メソッド-- すべての変数タイプ オブジェクトのリストを取得	522
GetAuthScheme メソッド-- 認証方式オブジェクトを取得	523
GetCertMap メソッド-- 証明書マッピング オブジェクトを取得	524
GetDomain メソッド-- ドメイン オブジェクトを取得	525
GetGlobalPolicy メソッド-- グローバル ポリシー オブジェクトを取得	526
GetGlobalResponse メソッド-- グローバル レスポンス オブジェクトを取得	526
GetGlobalRule メソッド-- グローバル ルール オブジェクトを取得	527
GetHostConfig メソッド-- ホスト設定オブジェクトを取得	528
GetODBCQueryScheme メソッド-- ODBC クエリ方式オブジェクトを取得	528
GetPwdPolicy メソッド-- パスワード ポリシー オブジェクトを取得	529
GetRegScheme メソッド-- 登録方式オブジェクトを取得	530
GetSAMLAffiliation メソッド-- SAML 2.0 アフィリエーション オブジェクトを取得	530
GetSAMLAffiliationById メソッド-- ID により SAML 2.0 アフィリエーション オブジェクトを取 得	531
GetSharedSecretPolicy メソッド-- 共有秘密キー ポリシー オブジェクトを取得	532
GetTrustedHost メソッド-- トラステッドホスト オブジェクトを取得	532
GetUserDir メソッド-- ユーザディレクトリ オブジェクトを取得	533

GetVariableType メソッド -- 変数タイプ オブジェクトを取得	534
RemoveAttributeFromSAMLScheme メソッド -- SAML 方式から属性を削除	535
SAMLAuthSchemeProperties メソッド -- SAML メタデータ プロパティを設定または取得	536
WSFEDAuthSchemeProperties メソッド -- WS-Federation プロパティを設定または取得	537
CLI 共有秘密キーのロールオーバー メソッド	539
Enabled メソッド -- ポリシーのロールオーバー有効化フラグを設定または取得	539
RolloverFrequency メソッド -- ポリシーのロールオーバー頻度を設定または取得	540
RolloverPeriod メソッド -- ポリシーのロールオーバー期間を設定または取得	541
Save メソッド -- 共有秘密キー ポリシー オブジェクトを保存	542
CLI トラストドホスト メソッド	543
GetDescription メソッド -- トラストドホストの説明を取得	543
GetIPAddress メソッド -- トラストドホストの IP アドレスを取得	543
GetName メソッド -- トラストドホストの名前を取得	544
GetSecret メソッド -- トラストドホストの共有秘密キーを取得	544
RolloverEnabled メソッド -- 共有秘密キーのロールオーバー フラグを設定するまたは取得	545
SetSecret メソッド -- トラストドホストの共有秘密キーを設定	546
CLI ユーザ メソッド	547
DisableByAdmin メソッド -- Disabled-by-Administrator フラグを設定または取得	547
DisableInactive メソッド -- Disabled-by-Inactivity フラグを設定または取得	549
DisableMaxLoginFail メソッド -- Disabled-by-Max-Login-Failure フラグを設定または取得	551
DisablePwdExpired メソッド -- Disabled-by-Password-Expired フラグを設定または取得	552
ForcePwdChange メソッド -- Force-Password-Change フラグを設定または取得	554
GetClass メソッド -- ユーザ クラスを取得	555
SetPassword メソッド -- 新しいパスワードを設定	556
UserPasswordState メソッド -- パスワード状態オブジェクトを設定または取得	557
ValidatePassword メソッド -- パスワードを検証	558
CLI ユーザディレクトリ メソッド	559
AnonymousIDAttr メソッド -- 匿名の DN 名を設定または取得	559
ChalRespAttr メソッド -- チャレンジ/レスポンス名を設定または取得	560
Description メソッド -- ユーザディレクトリの説明を設定または取得	560
DisabledAttr メソッド -- 無効化属性の名前を設定または取得	561
EmailAttr メソッド -- 電子メール属性名を設定または取得	562
EnableSecurityContext メソッド -- セキュリティ コンテキスト フラグを設定または取得	563
GetContents メソッド -- ユーザディレクトリのすべてのユーザを取得	564
GetNamespace メソッド -- ユーザディレクトリ ネームスペースを取得	564
IsSecure メソッド -- セキュア認証フラグを設定または取得	565
LookupEntry メソッド -- 指定されたパターンに一致するユーザを取得	566
MaxResults メソッド -- 最大検索結果を設定または取得	566
Name メソッド -- ユーザディレクトリ名を設定または取得	567
ODBCQueryScheme メソッド -- ODBC クエリ方式を設定または取得	568

Password メソッド -- ユーザのパスワードを設定または取得.....	569
PwdAttr メソッド -- パスワード属性名を設定または取得.....	569
PwdDataAttr メソッド -- パスワードデータ属性名を設定または取得.....	570
RequireCredentials メソッド -- 認証情報が必要かどうかを設定または取得.....	571
SearchRoot メソッド -- ディレクトリ検索ルートを設定または取得.....	572
SearchScope メソッド -- LDAP ディレクトリ検索範囲を設定または取得.....	573
SearchTimeout メソッド -- 最大ディレクトリ検索時間を設定または取得.....	574
Server メソッド -- ディレクトリ従属値を設定または取得.....	575
SetPath メソッド -- ユーザパスの設定.....	576
UIDAttr メソッド -- ユニバーサル ID 属性名を設定または取得.....	577
UserLookupEnd メソッド -- ユーザ DN 検索エンドポイントを設定または取得.....	578
UserLookupStart メソッド -- ユーザ DN 検索開始ポイントを設定または取得.....	579
Username メソッド -- ユーザ名を設定または取得.....	580
ValidateEntry メソッド -- ユーザディレクトリ エントリを検証.....	580
CLI ユーザパスワード状態メソッド.....	581
DisabledTime メソッド -- オブジェクトが無効化された時間を設定または取得.....	581
LastPWChangeTime メソッド -- パスワードが最後に変更された時間を設定または取得.....	582
LastLoginTime メソッド -- 最後のログイン時間を設定または取得.....	583
LoginFailures メソッド -- ログインの失敗数を設定または取得.....	583
CLI 変数タイプメソッド.....	584
GetName メソッド -- 変数タイプ オブジェクトの名前を取得.....	584
WS-フェデレーション リソース パートナー メソッド.....	585
AddAttribute メソッド -- リソース パートナーに属性を追加.....	585
AddUser メソッド -- リソース パートナーにユーザを追加.....	586
CreateIPConfigHostName メソッド -- 指定されたホストに基づいてオブジェクトを作成.....	587
CreateIPConfigSingleHost メソッド -- 1 つのアドレスに基づいてオブジェクトを作成.....	587
CreateIPConfigSubnetMask メソッド -- サブネット アドレスに基づいてオブジェクトを作成.....	588
DeleteIPConfig メソッド -- 指定された IP 設定 オブジェクトを削除.....	589
GetAllAttributes メソッド -- リソース パートナーのすべての属性を取得.....	590
GetAllIPConfigs メソッド -- サービス プロバイダのすべての IP 設定オブジェクトを取得.....	590
GetAllUsers メソッド -- リソース パートナーと関連付けられたすべてのユーザを取得.....	591
Property メソッド -- リソース パートナー プロパティを設定または取得.....	592
RemoveAttribute メソッド -- WSFED リソース パートナーから属性を削除.....	593
RemoveUser メソッド -- リソース パートナーから指定されたユーザを削除.....	593
Save メソッド -- リソース パートナーのメタデータを保存.....	594
CLI WS-フェデレーション リソース パートナー属性メソッド.....	595
GetAttrNameFormat メソッド -- 属性名の形式を取得.....	595
GetValue メソッド -- 属性値を取得.....	596

第 8 章: CLI ポリシー管理操作 597

セッションの初期化.....	597
システム オブジェクトの作成および管理	599
エージェント オブジェクトの作成	600
オブジェクト プロパティの表示および変更	600
ドメイン範囲を持つオブジェクト	601
1つのオブジェクトを取得して別のオブジェクトを作成	602
オブジェクトのプロパティの管理	602
ドメイン範囲またはグローバル範囲を持つオブジェクト	604
許可変数.....	606
特定の変数タイプに応じた変数の設定	606
オブジェクトへの変更の保存	612
パスワード ポリシーの変更	613
パスワード状態の管理	613
レスポンスとレスポンス属性の作成	616
新しい認証方式によるレルムの更新	617
認証方式テンプレートのデフォルト値の表示	618
認証方式の作成.....	619
共有秘密キー ロールオーバー ポリシーの変更	620
ファイルへのドメインおよびレルム レポートの書き込み	621
認証および許可イベント処理の無効化	622
ポリシー サーバ負荷分散の管理	623
クラスタの設定.....	624
クラスタがすべて失敗するとき	626

第 9 章: コマンドライン インターフェース制限 627

第 10 章: CLI プロパティリファレンス 628

SAML 2.0 プロパティ	629
WSFED プロパティ	679

第 1 章: Perl スクリプト作成の概要

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 コマンドライン インターフェースについて](#) (P. 25)

[インストールパス](#) (P. 26)

[スクリプトを実行する場所](#) (P. 27)

[CLI の例: ポリシー ストア オブジェクトの作成](#) (P. 27)

[CLI の例: 個々のプロパティの表示と設定](#) (P. 29)

[サンプルスクリプトの場所](#) (P. 30)

[関連マニュアル](#) (P. 30)

SiteMinder コマンドライン インターフェースについて

SiteMinder コマンドライン インターフェース (CLI) では、コマンドラインからカスタム Perl スクリプトを実行して、タスクを実行できます。

スクリプト作成インターフェースには以下の API が含まれます。

- エージェント API-- 保護されているリソースのアクセス制御がポリシー定義に従って動作しているかどうかをテストできます。
- ポリシー管理 API-- ポリシー ストアを設計して管理できます。

コマンドライン インターフェースには以下の機能があります。

- 管理 UI の迅速かつ軽快な代替機能
- 複数のポリシー ストアに対してポリシー サーバ設計と管理タスクを実行する効率的な方法。
- ポリシー ストア間で個別のオブジェクトを移行する機能。

コマンドライン インターフェースでは、管理 UI を通じて実行できるポリシー ストア操作のほとんどを実行できますが、すべてではありません。

詳細情報:

[コマンドライン インターフェース制限](#) (P. 627)

インストール パス

デフォルトでは、**SiteMinder** コマンド ライン インターフェースは以下の場所にインストールされます。

`<sm-ps-root>/CLI`

`<sm-ps-root>` は、ポリシー サーバ ソフトウェアをインストールしたルートディレクトリです。

Perl の場所

Perl の完全なバージョンはポリシー サーバと共にインストールされます。コマンド ライン インターフェースに対してスクリプトを実行するときは、システム上に存在する場合がある他の Perl インタープリタではなく、ポリシー サーバと共にインストールされる Perl インタープリタを使用する必要があります。

インストールプログラムは以下のデフォルトの場所に Perl をインストールします。

`<sm-ps-root>/CLI/bin`

システムに Perl の別のバージョンをインストールしている場合は、システムの PATH 環境変数内の他の Perl の場所の前に上に示す Perl の場所が来ることを確認してください。

スクリプトを実行する場所

Perl エージェントおよびポリシー管理 API は以下のマシン上で使用できます。

- エージェント API--ポリシー サーバがインストールされているマシン上。 エージェント API はローカル ポリシー サーバまたはリモート ポリシー サーバにアクセスできます。
- ポリシー管理 API--ポリシー サーバがインストールされているマシン上。

これらの API に対してスクリプトを実行するには、以下のコマンドライン構文を使用します。

```
perl scriptname
```

注: ポリシー管理 API で構築したスクリプトは、ポリシー サーバをインストールした同じユーザとして実行する必要があります（たとえば、UNIX プラットフォーム上の `smuser`）。

CLI の例: ポリシー ストア オブジェクトの作成

ドメイン エンジニアリングの管理者であると仮定します。 そのドメイン内でレルム ドキュメントを作成する必要があります。 管理 UI を使用して、以下の手順を実行できます。

1. SiteMinder 管理ソフトウェアにログインします。
2. レルムを追加するドメイン エンジニアリングを右クリックします。

3. [レルムの作成] をクリックし、[リソース] タブ上のフィールド用に以下の設定情報を指定します。

- 名前: **documentation**
- 説明: **Source files for manuals**
- エージェント: **agent1**
- リソース フィルタ: **/mysite/docs/***
- 認証方式: **Basic**

レルム用の他のすべてのデフォルトを受け入れます（リソース保護を含めて、デフォルトで有効にされるもの）。

4. 新しいレルムの作成を確認するために [OK] をクリックします。

同じ操作を実行するためにスクリプトを作成する場合は、以下のようになります。

```
#Initialize the Policy Management API
use Netegrity::PolicyMgtAPI;

$policyapi = Netegrity::PolicyMgtAPI->New();

print "Step 1. Log in the admin and create an API session.\n";
$session = $policyapi->CreateSession("adminid", "adminpwd");

print "Step 2. Select the domain for the new realm.\n";
$domain=$session->GetDomain("engineering");

#Get the realm's agent and authorization scheme info.\n";
$agent=$session->GetAgent("agent1");
$authscheme=$session->GetAuthScheme("Basic");
```

```
print "Step 3. Create and configure the realm.\n";
$realm=$domain->CreateRealm("documentation",
                             $agent,
                             $authscheme,
                             "Source files for manuals",
                             "/mysite/docs/*" );

print "Step 4. Confirm the creation of the realm.\n";
if ($realm == undef) {
    print "Realm creation failed.\n";
}
else {
    print "Realm creation succeeded.\n";
}
```

注: 通常は、ポリシー ストア オブジェクト名は大文字と小文字を区別します。上記の例で、基本認証方式およびエンジニアリング ドメインは大文字と小文字を区別します。さらに、エージェント名は、常に小文字でポリシー ストアに書き込まれます。既存のエージェントはユーザのスク립ト内で小文字で参照される必要があります。

CLI の例: 個々のプロパティの表示と設定

ポリシー管理 API オブジェクト (PolicyMgtRealm など) は、ポリシー ストア内のオブジェクトの個々のプロパティを表示して変更できる多くの取得および設定メソッドを提供します。これらの取得および設定メソッドを使用して、管理 UI のプロパティ フィールドを使用するときと同じ要領でオブジェクトのプロパティを表示および編集します。

以下のスク립トはリソース フィルタ プロパティを変更します。

```
use Netegrity::PolicyMgtAPI;

$policyapi = Netegrity::PolicyMgtAPI->New();
$session = $policyapi->CreateSession("adminid", "adminpwd");
$domain=$session->GetDomain("engineering");

$realm=$domain->GetRealm("documentation");
if($realm->ResourceFilter() eq "/mysite/docs/*") {
    $filter=$realm->ResourceFilter("/mysite/docs/*.doc");
}
```

```
if ($filter eq undef) {  
    print "Error changing resource filter.¥n";  
}  
else {  
    print "Resource filter changed to: " . $filter . "¥n";  
}
```

以下の一般的なルールに注意してください。

- メソッドへ引数を渡すと、メソッドはセッター メソッドとして動作します。たとえば、次のようになります。

```
$realm->ResourceFilter("/mysite/docs/*.doc");
```

- 引数を指定しないと、メソッドはゲッター メソッドとして動作します --たとえば、以下のようになります。

```
$filter=$realm->ResourceFilter();
```

- 取得および設定メソッドでは、既存または新規のプロパティ値が返されます。

サンプル スクリプトの場所

以下のデフォルトの場所にサンプル スクリプトがインストールされます。

```
<sm-ps-root>/CA/siteminder/CLI/examples
```

サンプル スクリプトを使用する前に、必ずスクリプトの始めに定義される、サイト固有の変数（管理者の認証情報およびユーザ ストア場所など）の値を変更してください。

関連マニュアル

以下の **SiteMinder** ドキュメントでポリシー サーバとエージェントの操作に関する追加の情報を検索できます。

- *CA SiteMinder ポリシー サーバ設定ガイド*
- *CA SiteMinder ポリシー サーバ管理ガイド*
- *CA SiteMinder Web エージェント設定ガイド*

オブジェクト依存性ポスター

ポスター *Scripting Interface for Perl: Object Dependencies* が SiteMinder と共に含まれます。ポスターは、従属オブジェクトを操作する前に作成または取得する必要がある Perl オブジェクトを示します。それぞれのオブジェクトはそのすべてのメソッドと共に表示されます。

第 2 章: CLI エージェント API について

このセクションには、以下のトピックが含まれています。

[エージェントおよびエージェント API \(P. 33\)](#)

[エージェント API に対するスクリプトの作成 \(P. 34\)](#)

[シングル サインオンとエージェント API \(P. 35\)](#)

[セッション情報 \(P. 38\)](#)

[オブジェクトとオブジェクト階層 \(P. 40\)](#)

エージェントおよびエージェント API

SiteMinder 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。

エージェントは、不正な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。リソースの要求を発行すると、要求はエージェントを介して転送されます。エージェントは、ポリシー サーバから受信する情報に基づいて要求を受理すべきか拒否すべきかを決定します。

エージェント API (モジュール `Netegrity::AgentAPI`) では、エージェントの操作をテストするスクリプトを作成できます。エージェント API で実行できるタスクには、SiteMinder テストツールで実行できるタスクの多くが含まれます。エージェント API では、ユーザのセッションに関する情報を表示することもできます。

エージェント API に対するスクリプトの作成

すべてのスクリプトで、ある特定の共通基本手順が必要になります。

エージェント API に対してスクリプトを作成する方法

1. 以下のようにスクリプトの始めのエージェント API を参照します。
2. 以下のように `New()` メソッドを使用してエージェント API オブジェクトを作成します。

```
$agentapi=Netegrity::AgentAPI->New("agtName","shrdSqrt");
```

注: 2 番目の引数で共有秘密キーを指定すると、v4.x エージェントが作成されます。共有秘密キーを省略すると、v5.x または 6.x エージェントが作成されます。

3. ポリシー サーバに関する設定情報を指定します。IP アドレスは必要ですが、他の引数のデフォルトを受け入れることができます。

```
$serverconfig = $agentapi->AddServerConfig($ipAddr);
```

4. ポリシー サーバに接続します。

- v4.x エージェントの場合

```
$agentapi->Connect();
```

- v5.x 以降のエージェントの場合

```
$agentapi->Connect("../..../Program  
Files/Netegrity/webagent/config/smhost.conf");
```

例: リソース上の保護の確認

これらの基本手順を完了した後、さまざまなエージェント操作を実行できます。たとえば、指定されたリソースがエージェントによって保護されるかどうかを確認できます。また、その場合は、以下のようにリソースにアクセスするのに必要な認証情報を探すことができます。

```
$resource = $agentapi->GetResource("/companyXYZ/private/");  
if($resource->IsProtected() == SM_AGENTAPI_YES) {  
    print "Authentication Type: ".$resource->GetAuthType();  
}  
else {  
    print "The resource is not protected."  
}
```

シングル サインオンとエージェント API

シングル サインオン環境では、別のエージェントによって保護されたレールムにアクセスするときに、指定されたエージェントを介して正しく認証するユーザは再認証の必要がありません。

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** という名前のシングル サインオン **Cookie** によって可能になります。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。

Cookie のコンテンツは **Cookie** から取得され、暗号化された文字列の形式で **Cookie** に書き込まれます。暗号化された文字列はトークンと呼ばれます。

エージェント API には、カスタム エージェント スクリプトが標準の **SiteMinder Web** エージェントとトークン情報を共有できる以下のメソッドが含まれます。

- **CreateSSOToken()**。ユーザが正常にログインした後、カスタム エージェント スクリプトはユーザに関する情報をこのメソッドに渡します。メソッドは、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に基づいてシングル サインオン トークン オブジェクトを作成します。
- **GetString()**。トークン オブジェクトを作成した後に、カスタム エージェント スクリプトは、暗号化された文字列としてトークンを取得するために **GetString()** をコールします。その後、スクリプトは **SMSESSION** **Cookie** にトークン文字列を書き込みます。
- **Decode()**。このメソッドは、現在のトークン オブジェクト用のトークンを復号し、指定された情報を抽出します。このメソッドは、トークン内の最後のアクセス タイムスタンプを更新するために使用することもできます。
- **GetVersion()**。現在のトークン オブジェクトの **SiteMinder** バージョンを取得します。
- **IsThirdParty()**。カスタム（サードパーティ）エージェントまたは標準の **SiteMinder** エージェントによって現在のトークン オブジェクトが作成されたかどうかを指定します。

カスタム エージェントのシングル サインオン サポート

標準の **SiteMinder Web** エージェントによってログインするときではなく、カスタム エージェントを介してログインするときにシングル サインオン オブジェクトを作成できます。

ここに示すのは、初期ログインをカスタム エージェント スクリプトを介して行うときの、シングル サインオン環境でのイベントの代表的シーケンスです。

- ユーザはカスタム エージェント スクリプトを介してログインします。
- カスタム エージェント スクリプトは、ユーザを認証するために **Login()** をコールします。ユーザは認証情報を要求されます。
- カスタム エージェント スクリプトは **CreateSSOToken()** をコールし、ユーザに関する情報（要求するクライアントのユーザ名、ユーザ **DN**、**IP** アドレス）をそれに渡します。**SiteMinder** は、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 オブジェクトにこの情報を追加します。**SiteMinder** はトークン内の情報の暗号化も実行します。
- カスタム エージェント スクリプトは、**GetString()** をコールして暗号化された文字列形式でトークンを取得します。
- カスタム エージェント スクリプトはユーザのブラウザで **SMSESSION Cookie** を作成し、**Cookie** にトークン文字列を書き込みます。
- 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。
- 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。

標準エージェントのシングル サインオン サポート

Perl エージェント API で作成されたカスタム エージェント スクリプトは、標準の SiteMinder Web エージェントによって作成された **SMSESSION Cookie** を受け入れることができます。

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バージョンに必要な **QMR** バージョンの詳細については、「**SDK リリース ノート**」を参照してください。

カスタム エージェントによって作成された **SMSESSION Cookie** を受け入れるように、適切な **QMR** アップグレードと共に **SiteMinder** エージェントを有効にするには、エージェントのエージェント設定ファイル (IIS サーバでの **LocalConfig.conf** またはその他のサーバでの **WebAgent.conf**) または中央の設定ファイル (v5.x 以降の場合) に、以下のエントリが含まれている必要があります。

```
AcceptTPCookie="yes"
```

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よって定義されたセッション情報は、**セッション変数**と呼ばれる名前と値のペアによって構成されます。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。

パッケージ **AgentSession** は、セッション変数の設定、取得、削除およびパラメータ リストの定義のための以下のメソッドを提供します。

- **SetVariables()**
- **GetVariables()**
- **DelVariables()**
- **AddParameter()**

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、

- 最大 **4KB** のデータを各セッション変数値に対して保存できます。
- 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。サーバに関するセッション情報の一元化によって、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めて、クロスドメインセッション管理などの機能を有効にできます。

SiteMinder のサポート

コマンドラインインターフェースは以下をサポートします。

- SiteMinder v5.x ポリシー サーバに対して実行される SiteMinder v4.x および v5.x エージェント
- SiteMinder v6.x ポリシー サーバに対して実行される SiteMinder v4.x 以降のエージェント

エージェント API は、SiteMinder ポリシー サーバがインストールされているマシン上でアクセスできます。 エージェント API はローカル ポリシー サーバまたはリモート ポリシー サーバにアクセスできます。

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。
- ポリシー サーバ UI でのレルム設定時に、[永続セッション] は、セッション中にアクセスするレルムの少なくとも 1 つに対して選択する必要があります。永続セッションに対して設定されたレルム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。

オブジェクトとオブジェクト階層

エージェント API には、エージェント操作を実行できる一連のオブジェクトが含まれています。

エージェント API 内のオブジェクトは階層的です。オブジェクトを使用するには、その前に別のオブジェクト内のメソッドによって作成または取得する必要があります。

以下の階層は、エージェント API 内のオブジェクトを示しています。他のオブジェクトを作成または取得する、より高位レベルのオブジェクトでメソッドを検索するためにそれを使用します。たとえば、**AgentResource** オブジェクトを必要とする場合は、そのすぐ上 (**AgentAPI**) のオブジェクトを参照して、適切なメソッドを選択します。

AgentAPI

AgentResource

AgentResponseAttr

AgentSvrCfg

AgentUser

AgentResponse

AgentResponseAttr

AgentSession

AgentResponseAttr

SSOToken

第 3 章: CLI エージェント API メソッド

このセクションには、以下のトピックが含まれています。

[CLI エージェント管理メソッド \(P. 41\)](#)
[リソースメソッド \(P. 54\)](#)
[CLI エージェント レスポンス メソッド \(P. 57\)](#)
[レスポンス属性メソッド \(P. 59\)](#)
[サーバ設定メソッド \(P. 64\)](#)
[セッションメソッド \(P. 65\)](#)
[シングルサインオン トークン メソッド \(P. 73\)](#)
[ユーザメソッド \(P. 77\)](#)

CLI エージェント管理メソッド

AddServerConfig メソッド -- エージェント API オブジェクトにポリシー サーバ設定を追加

AddServerConfig メソッドは、エージェント API オブジェクトに 1 つ以上のポリシー サーバ設定を追加します。ポリシー サーバとエージェント API 間の接続を確立するには、その前にこのメソッドを少なくとも 1 回コールする必要があります。

構文

AddServerConfig メソッドの形式は以下のとおりです。

```
Netegrity::AgentAPI->AddServerConfig(IPAddress[, minConn]  
[, maxConn][, stepConn][, timeout][, azPort][, auPort]  
[, acPort])
```

パラメータ

AddServerConfig メソッドは、以下のパラメータを受け入れます。

IPAddress (文字列)

ポリシー サーバの IP アドレスを指定します。

minConn (int)

(オプション) 許可されている接続の最小数を指定します。

maxConn (int)

(オプション) 許可されている接続の最大数を指定します。

stepConn (int)

(オプション) 既存の接続がすべて使用されているときに追加する接続の数を指定します。

timeout (int)

(オプション) エージェント API がポリシー サーバへの接続試行を停止するまでの秒単位の時間を指定します。

azPort (int)

(オプション) 許可サービス用のポート番号を指定します。

auPort (int)

(オプション) 認証サービス用のポート番号を指定します。

acPort (int)

(オプション) アカウンティング サービス用のポート番号を指定します。

戻り値

AddServerConfig メソッドは、以下の値を返します。

- AgentSvrCfg (オブジェクト)

注釈

SiteMinder v6.0 に導入された単一プロセス ポリシー サーバが、もともとは別個の認証、許可およびアカウンティングプロセスを 1 つの結合プロセスに組み合わせることにより、複数の要求が 1 つの TCP ポートを通ります。その結果、パラメータ **azPort**、**auPort** および **acPort** はすべて同じポート番号を参照します。3 つの引数は下位互換性のために維持されます。

Connect メソッド -- エージェント API とポリシー サーバ間の接続を確立

Connect メソッドは、エージェント API とポリシー サーバ間の接続を確立します。

構文

Connect メソッドの形式は以下のとおりです。

```
Netegrity::AgentAPI->Connect([bootFile])
```

パラメータ

Connect メソッドは以下のパラメータを受け入れます。

bootFile (文字列)

(オプション) v5.x エージェントに接続するために使用される設定ファイルの名前を指定します。

注: SmHost.conf などの既存のファイルを使用するか、新規ファイルを作成できます。詳細については、メソッド

[AgentAPI->CreateBootstrapFile](#) (P. 44) を参照してください。

戻り値

Connect メソッドは以下のいずれかの値を返します。

- SM_AGENTAPI_YES (値 = 1)
接続が成功したことを指定します。
- SM_AGENTAPI_FAILURE (値 = -1)
エージェント API がポリシー サーバに到達できなかったことを指定します。
- SM_AGENTAPI_TIMEOUT (値 = -2)
メソッドがタイムアウトしたことを指定します。
- SM_AGENTAPI_NOCONNECTION (値 = -3)
初期化が失敗したことを指定します。

CreateBootstrapFile メソッド -- エージェントに接続するためのブートストラップ ファイルを生成

CreateBootstrapFile メソッドは、v5.x エージェントに接続するためにポリシー サーバが使用できるブートストラップ ファイルを生成します。ブートストラップ ファイルには、メソッドのパラメータによって指定された設定が含まれています。いったん接続されると、ポリシー サーバはその他の設定を提供できます。SmHost.conf はブートストラップ ファイルの例です。

構文

CreateBootstrapFile メソッドの形式は以下のとおりです。

```
Netegrity::AgentAPI->CreateBootstrapFile(trustedHostName, ipAddress, hostConfigName, sharedSecret, registrationDataFileName)
```

パラメータ

CreateBootstrapFile メソッドは、以下のパラメータを受け入れます。

trustedHostName (文字列)

トラステッドホストの名前を指定します。

ipAddress (文字列)

ポリシー サーバの IP アドレスを指定します。

hostConfigName (文字列)

ホスト設定オブジェクトの名前を指定します。

sharedSecret (文字列)

ホストによって使用される共有秘密キーの値を指定します。

registrationDataFileName (文字列)

メソッドによって生成されるブートストラップ ファイルの名前を指定します。

戻り値

CreateBootstrapFile メソッドは以下のいずれかの値を返します。

- file_name (文字列)
新規に作成したファイルの名前を指定します。
- empty_string (文字列)
メソッドが失敗したことを指定します。

例

以下のコードフラグメントは、CreateBootstrapFile メソッドの使用方法の例です。

```
# A shared secret is not specified in the context of a v5.x agent.  
# A v5.x agent by the specified name needs to exist in the policy  
# store. Use Netegrity::AgentAPI.  
  
$agentapi = Netegrity::AgentAPI->New($agentname);  
  
# Create the bootstrap file.  
  
$agentapi->CreateBootstrapFile($hostname, $ipaddr, $hconfname, $secret, $fname);  
  
# Notice that in the v5.x context, the connect() method takes a filename.  
  
$agentapi->Connect($fname);
```

CreateUser メソッド -- ユーザオブジェクトを作成

CreateUser メソッドは、パラメータ内のメソッドに渡される情報が含まれたユーザオブジェクトを作成します。Perl は、ログインなどのユーザ操作を実行するためにこのオブジェクトを使用します。

注: このメソッドはユーザディレクトリに新規ユーザを作成しません。

構文

CreateUser メソッドの形式は以下のとおりです。

```
Netegrity::AgentAPI->CreateUser(Username, Password[, cert][, certLength])
```

パラメータ

CreateUser メソッドは、以下のパラメータを受け入れます。

Username (文字列)

ユーザの ID を指定します。

Password (文字列)

ユーザのパスワードを指定します。

cert (文字列)

(オプション) ユーザ認証用の X.509 証明書を指定します。

certLength (int)

(オプション) 証明書の長さを指定します。

戻り値

CreateUser メソッドは以下のいずれかの値を返します。

- AgentUser (オブジェクト)
- undef

メソッドが失敗したことを指定します。

Disconnect メソッド -- エージェントとポリシー サーバ間の接続を解除

Disconnect メソッドは、エージェントとポリシー サーバ間の接続を閉じます。

構文

Disconnect メソッドの形式は以下のとおりです。

```
Netegrity::AgentAPI->Disconnect()
```

パラメータ

Disconnect メソッドは、パラメータを受け入れません。

戻り値

Disconnect メソッドは以下のいずれかの値を返します。

- SM_AGENT_SUCCESS (値 = 0)
接続が正しく閉じたことを指定します。
- SM_AGENT_FAILURE (値 = 1)
接続が正しく閉じなかったことを指定します。

DoManagement メソッド -- ポリシー サーバからのエージェント コマンドを要求

DoManagement メソッドは、ポリシー サーバからの保留中のエージェント コマンドのリストを要求し、配列でそれらを返します。配列内のコマンドにアクセスするには、AgentResponseAttr->GetID メソッドを呼び出します。

構文

DoManagement メソッドの形式は以下のとおりです。

```
Netegrity::AgentAPI->DoManagement()
```

パラメータ

DoManagement メソッドは、パラメータを受け入れません。

戻り値

DoManagement メソッドは以下のいずれかの値を返します。

- AgentResponseAttr (配列)
- undef
保留中のエージェント コマンドがないことを指定します。

GetResource メソッド -- 指定されたリソースを取得

GetResource メソッドは、指定されたリソースを取得します。

構文

GetResource メソッドの形式は以下のとおりです。

```
Netegrity::AgentAPI->GetResource(resName[, action][, clientIP])
```

パラメータ

GetResource メソッドは、以下のパラメータを受け入れます。

resName (文字列)

取得するリソースの名前を指定します。

action (文字列)

(オプション) 実行する HTTP アクションを指定します (たとえば GET、POST または PUT)。

clientIP (文字列)

(オプション) クライアントの IP アドレスを指定します。

戻り値

GetResource メソッドは、以下の値を返します。

- AgentResource (オブジェクト)

IncrementRefCount メソッド -- 参照数をインクリメント

エージェント API がマルチスレッドモードで動作しているときに、IncrementRefCount メソッドをコールできます。Perl 関数フォークをコールした後に、たとえば、子プロセス内のこのメソッドをコールできます。

構文

IncrementRefCount メソッドの形式は以下のとおりです。

```
Netegrity::AgentAPI->IncrementRefCount()
```

パラメータ

IncrementRefCount メソッドは、パラメータを受け入れません。

戻り値

IncrementRefCount メソッドは、値を返しません。

New メソッド -- エージェント API を構築

New メソッドは、エージェント API オブジェクトを構築します。エージェント API オブジェクトを使用する前にコールする必要があります。

構文

New メソッドの形式は以下のとおりです。

```
Netegrity::AgentAPI->New(agentName[, sharedSecret][, failover])
```

パラメータ

New メソッドは、以下のパラメータを受け入れます。

agentName (文字列)

ポリシーストアに表示されたとおりのエージェントの名前を指定します。

注: エージェント名は大文字/小文字を区別しません。

psharedSecret (文字列)

(オプション) ポリシーストアに表示されたとおりの共有秘密キーを指定します。

注: 共有秘密キーの値を指定すると、**v4.x** エージェントオブジェクトが作成されます。共有秘密キーの値を指定しないと、**v5.x** エージェントオブジェクトが作成されます。共有秘密キーは大文字と小文字を区別します。

failover (int)

(オプション) 以下のようにフェールオーバー操作を有効にすべきか無効にすべきかを指定します。

- 値 = -1
フェールオーバー操作の有効化を指定します。
- 値 = 0
フェールオーバー操作の無効化を指定します。

戻り値

New メソッドは、以下の値を返します。

- AgentAPI (オブジェクト)

PrintDebugTrace メソッド -- コンソールにトレース情報を出力

PrintDebugTrace メソッドは、コンソールへのエラーまたはトレース情報の出力を有効または無効にし、それぞれ 1 または 0 の値を返します。

構文

PrintDebugTrace メソッドの形式は以下のとおりです。

```
Netegrity::AgentAPI->PrintDebugTrace([debugFlag])
```

パラメータ

PrintDebugTrace メソッドは以下のパラメータを受け入れます。

debugFlag (int)

(オプション) 以下のように、コンソールへのエラーまたはトレース情報の出力の有効化または無効化を指定します。

- 値 = 1
出力の有効化を指定します。
- 値 = 0
出力の無効化を指定します。

戻り値

PrintDebugTrace メソッドは以下のいずれかの整数値を返します。

- 値 = 1
出力が有効であることを指定します。
- 値 = 0
出力が無効であることを指定します。

ProcessAuthResponse メソッド -- パスワードの変更ログインから情報を抽出

ProcessAuthResponse メソッドは、ログインまたはパスワードの変更から返される適切な情報を抽出します。

構文

ProcessAuthResponse メソッドの形式は以下のとおりです。

```
Netegrity::AgentAPI->GetResource(UserCreds)
```

パラメータ

ProcessAuthResponse メソッドは以下のパラメータを受け入れます。

UserCreds (UserCredentials)

ログインまたはパスワードの変更コールからのユーザ認証情報を指定します。

戻り値

ProcessAuthResponse メソッドは、適切な情報を文字列として返します。

SetErrorCallback メソッド -- エラー メッセージを処理するサブルーチンを登録

SetErrorCallback メソッドは、SiteMinder エラー メッセージを処理するユーザ定義の Perl サブルーチンを登録します。複数のサブルーチンが登録される場合、SiteMinder は最新の登録済みサブルーチンをコールします。

構文

SetErrorCallback メソッドの形式は以下のとおりです。

```
Netegrity::AgentAPI->SetErrorCallback(subref)
```

パラメータ

SetErrorCallback メソッドは以下のパラメータを受け入れます。

subref (文字列)

以下のように、Perl サブルーチンの名前またはサブルーチンの参照を指定します。

- 名前

例：

```
$agentapi->SetErrorCallback("CustomErrorHandler");
```

- リファレンス

例：

```
$agentapi->SetErrorCallback(%&CustomErrorHandler);
```

戻り値

SetErrorCallback メソッドは以下のいずれかの整数値を返します。

- 値 = 1

登録が成功したことを指定します。

- 値 = 0

登録が失敗したことを指定します。

SetTraceCallback メソッド -- トレース メッセージを処理するサブルーチンを登録

SetTraceCallback メソッドは、SiteMinder トレース メッセージを処理するユーザ定義の Perl サブルーチンを登録します。複数のサブルーチンが登録される場合、SiteMinder は最新の登録済みサブルーチンをコールします。

構文

SetTraceCallback メソッドの形式は以下のとおりです。

```
Netegrity::AgentAPI->SetTraceCallback(subref[, mode][, delim]  
[, configFileName])
```

パラメータ

SetTraceCallback メソッドは、以下のパラメータを受け入れます。

subref (文字列)

以下のように、Perl サブルーチンの名前またはサブルーチンの参照を指定します。

- 名前

例：

```
$agentapi->SetErrorCallback("CustomTraceHandler");
```

- リファレンス

例：

```
$agentapi->SetErrorCallback(%&CustomTraceHandler);
```

mode (文字列)

(オプション) 以下のようにトレース メッセージの出力形式を指定します。

- default

角かっこによって囲まれたフィールドを指定します。

- fixed

固定幅フィールドを指定します。

- **delim**

フィールドを区切るための文字の使用を指定します。

- **xml**

XML のようなタグで囲まれたフィールドを指定します。

delim (文字列)

(オプション) モードが **delim** に設定されるときに区切り文字として使用する文字を指定します。

configFileName (文字列)

(オプション) トレースに含めるデータを指定する設定ファイルの名前を指定します。

デフォルト: ファイル名が指定されない場合は、デフォルト設定が使用されます。

戻り値

SetTraceCallback メソッドは以下のいずれかの整数値を返します。

- 値 = 1

登録が成功したことを指定します。

- 値 = 0

登録が失敗したことを指定します。

リソース メソッド

GetAuthType メソッド -- 必要な認証情報のタイプを取得

GetAuthType メソッドは、リソースまたはレルムへの正しいログインに必要な認証情報のタイプを取得します。

構文

GetAuthType メソッドの形式は以下のとおりです。

```
Netegrity::AgentResource->GetAuthType()
```

パラメータ

GetAuthType メソッドは、パラメータを受け入れません。

戻り値

GetAuthType メソッドは以下のいずれかの値を返します。

- Sm_AuthApi_Cred_None (値 = 0x00)
認証情報が必要ではないことを指定します。
注: この許可タイプは匿名のレルムに使用されます。
- Sm_AuthApi_Cred_Basic (値 = 0x01)
ユーザ名とパスワードが必要であることを指定します。
- Sm_AuthApi_Cred_Digest (値 = 0x02)
要約プロトコルを使用してユーザ名とパスワードを交換する必要があることを指定します。
- Sm_AuthApi_Cred_X509Cert (値 = 0x04)
完全な X.509 クライアント証明書が必要であることを指定します。
- Sm_AuthApi_Cred_X509CertUserDN (値 = 0x08)
X.509 クライアント証明書に基づくユーザ DN が必要であることを指定します。
- Sm_AuthApi_Cred_X509CertIssuerDN (値 = 0x10)
X.509 クライアント証明書に基づく発行者 DN が必要であることを指定します。
- Sm_AuthApi_Cred_CertOrBasic (値 = 0x20)
利用可能な場合は、証明書が必要であることを指定します。
注: 証明書が利用可能でない場合は、ユーザ名とパスワードが必要です。
- Sm_AuthApi_Cred_NTChalResp (値 = 0x40)
NT チャレンジ/レスポンス プロトコルを使用してユーザ名とパスワードを交換する必要があることを指定します。
- Sm_AuthApi_Cred_CertOrForm (値 = 0x80)
X.509 クライアント証明書またはフォームベースの認証方式が必要であることを指定します。

- `Sm_AuthApi_Cred_SSLRequired` (値 = 0x01000000)
SSL 接続が必要であることを指定します。
- `Sm_AuthApi_Cred_FormRequired` (値 = 0x02000000)
ユーザを HTML フォームにリダイレクトする必要があることを指定します。
- `Sm_AuthApi_Cred_AllowSaveCreds` (値 = 0x04000000)
30 日間認証情報を保存できることを指定します
注: 認証情報が保存される場合、ユーザは保護されているリソースにアクセスするごとに、それらを再入力する必要はありません。
- `Sm_AuthApi_Cred_PreserveSessionID` (値 = 0x08000000)
現在のセッションが有効な限り、セッション ID が保存されることを指定します。
- `Sm_AuthApi_Cred_DoNotChallenge` (値 = 0x10000000)
ユーザが認証情報のチャレンジを受けないことを指定します。
- `undef`
リソースが保護されないことを指定します。

IsProtected メソッド -- SiteMinder がリソースを保護するかどうかを確認

IsProtected メソッドは、SiteMinder が定義済みのリソースを保護するかどうかを確認します。

構文

IsProtected メソッドの形式は以下のとおりです。

```
Netegrity::AgentResource->IsProtected()
```

パラメータ

IsProtected メソッドは、パラメータを受け入れません。

戻り値

IsProtected メソッドは以下のいずれかの値を返します。

- SM_AGENTAPI_YES (値 = 1)
リソースが保護されることを指定します。
- SM_AGENTAPI_NO (値 = 2)
リソースが保護されないことを指定します。
- SM_AGENTAPI_FAILURE (値 = -1)
ポリシー サーバに到達できなかったことを指定します。
- SM_AGENTAPI_TIMEOUT (値 = -2)
メソッドがタイムアウトしたことを指定します。
- SM_AGENTAPI_NOCONNECTION (値 = -3)
初期化が失敗したことを指定します。

CLI エージェント レスポンス メソッド

GetAttributes メソッド -- 利用可能なレスポンス属性のリストを取得

GetAttributes メソッドは、利用可能なレスポンス属性のリストを取得して、配列にそれらを返します。 AgentUser->Login or AgentUser->GetResponse のコールに成功した後に、このメソッドを使用します。配列内のレスポンス属性のタイプを特定するには、AgentResponseAttr->GetID メソッドをコールします。

構文

GetAttributes メソッドの形式は以下のとおりです。

```
Netegrity::AgentResponse->GetAttributes()
```

パラメータ

GetAttributes メソッドは、パラメータを受け入れません。

戻り値

GetAttributes メソッドは以下のいずれかの値を返します。

- AgentResponseAttr (配列)
- undef

メソッドが失敗したことを指定します。

GetSession メソッド -- レスポンスからセッションを取得

GetSession メソッドは、レスポンスからセッション オブジェクトを取得します。

構文

GetSession メソッドの形式は以下のとおりです。

```
Netegrity::AgentResponse->GetSession()
```

パラメータ

GetSession メソッドは、パラメータを受け入れません。

戻り値

GetSession メソッドは以下のいずれかの値を返します。

- AgentSession (オブジェクト)
- undef

メソッドが失敗したことを指定します。

レスポンス属性メソッド

GetFlags メソッド -- レスポンス属性のフラグを取得

GetFlags メソッドは、レスポンス属性のフラグを取得します。

構文

GetFlags メソッドの形式は以下のとおりです。

```
Netegrity::AgentResponseAttr->GetFlags()
```

パラメータ

GetFlags メソッドは、パラメータを受け入れません。

戻り値

GetFlags メソッドは、以下の値を返します。

- existing_response_attribute_flags

GetID メソッド -- レスポンス属性の ID またはエージェント コマンドの ID を取得

GetID メソッドはレスポンス属性 ID またはエージェント コマンド ID を取得します。

構文

GetID メソッドの形式は以下のとおりです。

```
Netegrity::AgentResponseAttr->GetID()
```

パラメータ

GetID メソッドは、パラメータを受け入れません。

戻り値

GetID メソッドは、AgentResponse->GetAttributes のコール後にレスポンス属性 ID を返すか、AgentAPI->DoManagement のコール後にエージェント コマンド ID を返します。いずれの場合も、戻り値のタイプは long 型です。

- AgentResponse->GetAttributes は、レスポンス属性のリストを取得して、それらを配列に返します。AgentResponse->GetAttributes のコール後に GetID をコールすると、GetID は以下のいずれかのレスポンス属性 ID (long 型) を返します。
 - SM_AGENTAPI_ATTR_AUTH_DIR_OID (値 = 151)
ユーザが認証されたユーザ ディレクトリ用の内部 SiteMinder オブジェクト ID を指定します。
 - SM_AGENTAPI_ATTR_USERUNIVERSALID (値 = 152)
ユーザのユニバーサル ID を指定します。
 - SM_AGENTAPI_ATTR_IDENTITYSPEC (値 = 156)
ユーザの識別チケットを指定します。
注: ユーザ追跡が有効な場合は、この値が返されます。
 - SM_AGENTAPI_ATTR_SESSIONDRIFT (値 = 167)
実際のセッション データが Cookie に保存されたセッション データに対して検証される最大時間を分単位で指定します。
 - SM_AGENTAPI_ATTR_AUTH_DIR_NAME (値 = 213)
[SiteMinder ユーザ ディレクトリ] ダイアログ ボックスの [名前] フィールドに表示されるとおりのディレクトリ名を指定します。
 - SM_AGENTAPI_ATTR_AUTH_DIR_SERVER (値 = 214)
[SiteMinder ユーザ ディレクトリ] ダイアログ ボックスの [サーバ] フィールドに表示されるとおりのサーバ指定情報を指定します。
 - SM_AGENTAPI_ATTR_AUTH_DIR_NAMESPACE (値 = 215)
次のようにユーザ ディレクトリのネームスペースを指定します。
LDAP、AD、WinNT または ODBC。

- SM_AGENTAPI_ATTR_USERMSG (値 = 216)

認証試行の後にユーザに提示するテキストを指定します。

注: このテキストは認証のチャレンジまたは認証に失敗した理由として提示できます。

- SM_AGENTAPI_ATTR_USERDN (値 = 218)

ユーザの識別名を指定します。

- AgentAPI->DoManagement は、ポリシー サーバからの保留中のエージェント コマンドのリストを取得し、配列にそれらを返します。
AgentAPI->DoManagement のコール後に GetID をコールすると、GetID は以下のいずれかのエージェント コマンド ID (long 型) を返します。

- SM_AGENTAPI_AFFILIATE_KEY_UPDATE (値 = 189)

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 (値 = 190)

「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。

注: 暗号化された値には、24 バイトのバイナリ データが含まれています。

- SM_AGENTAPI_AGENT_KEY_UPDATE_LAST (値 = 191)

「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。

注: 暗号化された値には、24 バイトのバイナリ データが含まれています。

- SM_AGENTAPI_AGENT_KEY_UPDATE_CURRENT (値 = 192)

「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。

注: 暗号化された値には、24 バイトのバイナリ データが含まれています。

- SM_AGENTAPI_AGENT_KEY_UPDATE_PERSISTENT (値 = 193)

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。

注: 暗号化された値には、24 バイトのバイナリ データが含まれています。

- SM_AGENTAPI_CACHE_FLUSH_ALL (値 = 194)
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 (値 = 195)
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 (値 = 196)
特定のユーザのすべてのキャッシュ情報をフラッシュするようにエージェントに指示します。
- SM_AGENTAPI_CACHE_FLUSH_ALL_REALMS (値 = 197)
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 (値 = 198)
特定のユーザのすべてのリソース情報をフラッシュするようにエージェントに指示します。

GetName メソッド -- レスポンス属性の名前を取得

GetName メソッドは、レスポンス属性の名前を取得します。

構文

GetName メソッドの形式は以下のとおりです。

```
Netegrity::AgentResponseAttr->GetName()
```

パラメータ

GetName メソッドは、パラメータを受け入れません。

戻り値

GetName メソッドは、以下の値を返します。

- response_attribute_name (文字列)

GetTTL メソッド -- レスポンス属性の TTL 値を取得

GetTTL メソッドは、レスポンス属性の Time To Live (TTL) 値を取得します。

構文

GetTTL メソッドの形式は以下のとおりです。

```
Netegrity::AgentResponseAttr->GetTTL()
```

パラメータ

GetTTL メソッドは、パラメータを受け入れません。

戻り値

GetTTL メソッドは、以下の値を返します。

- Time_To_Live_value (long)

GetValue メソッド -- レスポンス属性の値を取得

GetValue メソッドは、レスポンス属性の値を取得します。

構文

GetValue メソッドの形式は以下のとおりです。

```
Netegrity::AgentResponseAttr->GetValue()
```

パラメータ

GetValue メソッドは、パラメータを受け入れません。

戻り値

GetValue メソッドは、以下の値を返します。

- response_attribute_value (文字列)

サーバ設定メソッド

IPAddress メソッド -- ポリシー サーバの IP アドレスを設定または取得

IPAddress メソッドは、ポリシー サーバの IP アドレスを設定または取得します。

構文

IPAddress メソッドの形式は以下のとおりです。

```
Netegrity::AgentSvrCfg->IPAddress([IPAddress])
```

パラメータ

IPAddress メソッドは以下のパラメータを受け入れます。

IPAddress (文字列)

(オプション) ポリシー サーバの IP アドレスを指定します。

戻り値

IPAddress メソッドは、以下の値を返します。

- *IP_address* (文字列)

ポリシー サーバの IP アドレスを指定します。

セッション メソッド

AddParameter メソッド -- パラメータ リストにセッション変数名と値のペアを追加

AddParameter メソッドは、パラメータ リストにセッション変数名と値のペアを追加します。セッション変数には、ビジネス ロジック、証明書情報およびアフィリエイト アプリケーション用の SAML アサーションなどのクライアント アプリケーションに関するデータが含まれます。このデータはユーザ セッションにバインドされます。

構文

AddParameter メソッドの形式は以下のとおりです。

```
Netegrity::AgentSession->AddParameter(varName [,varValue][, varFlag])
```

パラメータ

AddParameter メソッドは、以下のパラメータを受け入れます。

varName (文字列)

パラメータ リストに追加する変数の名前を指定します。

制限： 最大長は 255 文字です。

varValue (文字列)

(オプション) パラメータ リストに追加する変数の値を指定します。

制限： 最大長はターゲット データ ストアによって決定されます。

varFlag (int)

(オプション) 以下のように、GetVariables メソッドでセッション ストアから変数名と値のペアを削除するかどうかを指定します。

- 0 (ゼロ)

GetVariables でセッション ストアから変数名と値のペアを取得するがそれを削除しないことを指定します。

- 1 (one)

GetVariables でセッション ストアから変数名と値のペアを取得してそれを削除することを指定します。

戻り値

AddParameter メソッドは、値を返しません。

注釈

以下のセッション変数メソッドをコールして、パラメータ リスト内の名前と値のペアを管理できます。

- AgentSession->DelVariables
- AgentSession->GetVariables
- AgentSession->SetVariables

複数の変数を管理するには、変数ごとに AddParameter を 1 回コールしてから、AgentSession->DelVariables、AgentSession->GetVariables、および AgentSession->SetVariables メソッドをコールします。AddParameter は変数をパラメータ リストに追加しますが、AgentSession->DelVariables、AgentSession->GetVariables、および AgentSession->SetVariables は、パラメータ リストをクリアします。

セッション変数メソッドを使用するには、その前に以下の条件を満たしている必要があります。

- 少なくとも 1 つのレلمで永続セッションを作成する必要があります。
- SiteMinder セッションストアを有効にする必要があります。

注: 詳細については、「ポリシー サーバ管理ガイド」と「ポリシー サーバ設定ガイド」を参照してください。

DelVariables メソッド -- セッション ストアからセッション変数を削除

DelVariables メソッドは、セッション ストアからパラメータ リスト内のセッション変数を削除します。最初に AgentSession->AddParameter をコールして、パラメータ リスト内の変数を指定します。

構文

DelVariables メソッドの形式は以下のとおりです。

```
Netegrity::AgentSession->DelVariables()
```

パラメータ

DelVariables メソッドは、パラメータを受け入れません。

戻り値

DelVariables メソッドは、値を返しません。

注釈

DelVariables メソッドは、パラメータ リストをクリアします。

GetID メソッド -- セッション ID を取得

GetID メソッドは、セッション ID を取得します。

構文

GetID メソッドの形式は以下のとおりです。

```
Netegrity::AgentSession->GetID()
```

パラメータ

GetID メソッドは、パラメータを受け入れません。

戻り値

GetID メソッドは、以下の値を返します。

- session_ID (文字列)

GetReason メソッド -- セッションの理由 ID を取得

GetReason メソッドは、セッションの理由 ID を取得します。理由 ID は、失敗した認証または許可の理由を指定します。

構文

GetReason メソッドの形式は以下のとおりです。

```
Netegrity::AgentSession->GetReason()
```

パラメータ

GetReason メソッドは、パラメータを受け入れません。

戻り値

GetReason メソッドは以下のいずれかの値 (long 型) を返します。

- Sm_Api_Reason_None (値 = 0)
- Sm_Api_Reason_PwMustChange (値 = 1)
- Sm_Api_Reason_InvalidSession (値 = 2)
- Sm_Api_Reason_RevokedSession (値 = 3)
- Sm_Api_Reason_ExpiredSession (値 = 4)
- Sm_Api_Reason_AuthLevelTooLow (値 = 5)
- Sm_Api_Reason_UnknownUser (値 = 6)
- Sm_Api_Reason_UserDisabled (値 = 7)
- Sm_Api_Reason_InvalidSessionId (値 = 8)
- Sm_Api_Reason_InvalidSessionIp (値 = 9)
- Sm_Api_Reason_CertificateRevoked (値 = 10)
- Sm_Api_Reason_CRLOutOfDate (値 = 11)
- Sm_Api_Reason_CertRevokedKeyCompromised (値 = 12)
- Sm_Api_Reason_CertRevokedAffiliationChange (値 = 13)
- Sm_Api_Reason_CertOnHold (値 = 14)
- Sm_Api_Reason-TokenCardChallenge (値 = 15)
- Sm_Api_Reason_ImpersonatedUserNotInDir (値 = 16)
- Sm_Api_Reason_Anonymous (値 = 17)
- Sm_Api_Reason_PwWillExpire (値 = 18)
- Sm_Api_Reason_PwExpired (値 = 19)
- Sm_Api_Reason_ImmedPWChangeRequired (値 = 20)
- Sm_Api_Reason_PWChangeFailed (値 = 21)
- Sm_Api_Reason_BadPWChange (値 = 22)

- Sm_Api_Reason_PWChangeAccepted (値 = 23)
- Sm_Api_Reason_ExcessiveFailedLoginAttempts (値 = 24)
- Sm_Api_Reason_AccountInactivity (値 = 25)
- Sm_Api_Reason_NoRedirectConfigured (値 = 26)
- Sm_Api_Reason_ErrorMessageIsRedirect (値 = 27)
- Sm_Api_Reason_Next_Tokencode (値 = 28)
- Sm_Api_Reason_New_PIN_Select (値 = 29)
- Sm_Api_Reason_New_PIN_Sys_Tokencode (値 = 30)
- Sm_Api_Reason_New_User_PIN_Tokencode (値 = 31)
- Sm_Api_Reason_New_PIN_Accepted (値 = 32)
- Sm_Api_Reason_Guest (値 = 33)
- Sm_Api_Reason_PWSelfChange (値 = 34)
- Sm_Api_Reason_ServerException (値 = 35)
- Sm_Api_Reason_UnknownScheme (値 = 36)
- Sm_Api_Reason_UnsupportedScheme (値 = 37)
- Sm_Api_Reason_Misconfigured (値 = 38)
- Sm_Api_Reason_BufferOverflow (値 = 39)
- Sm_Api_Reason_SetPersistentSessionFailed (値 = 40)

GetSpec メソッド -- 暗号化セッション指定情報を取得

GetSpec メソッドは、暗号化セッション指定情報を取得します。セッション指定情報には、セッション開始時刻、タイムアウトパラメータおよびユーザの識別名などのユーザセッションに関する情報が含まれます。シングルサインオンアプリケーションの場合のように、複数の Web サイトにわたってセッションを識別するためにセッション指定情報を使用できます。

構文

GetSpec メソッドの形式は以下のとおりです。

```
Netegrity::AgentSession->GetSpec()
```

パラメータ

GetSpec メソッドは、パラメータを受け入れません。

戻り値

GetSpec メソッドは、以下の値を返します。

- encrypted_session_specification (文字列)

注釈

セッション指定情報が保存される場所は、以下のセッションのタイプによって異なります。

- 非永続セッション

セッション指定情報は **Cookie** に保存されます。

- 永続セッション

セッション指定情報は、セッションストア データベースに保存されます。また、オプションでクライアントの **Cookie** に保存されます。

注: セッションチケットは、**Web** エージェントのキャッシュに保存された実際のユーザセッションデータのインデックスとして使用されます。

GetVariables メソッド -- セッションストアからセッション変数を取得

GetVariables メソッドは、セッションストアからパラメータ リスト内のセッション変数を取得します。最初に **AgentSession->AddParameter** をコールして、パラメータ リスト内の変数を指定します。取得後に変数をセッションストアから削除するには、**varFlag** を **1 (one)** に設定した状態で **AgentSession->AddParameter** をコールします。

構文

GetVariables メソッドの形式は以下のとおりです。

```
Netegrity::AgentSession->GetVariables()
```

パラメータ

GetVariables メソッドは、パラメータを受け入れません。

戻り値

GetVariables メソッドは、以下の値を返します。

- AgentResponseAttr (配列)

注: 一部のセッション変数は取得できません。この確認を行うには、AgentResponseAttr->GetFlags を関連付けられた AgentResponseAttr オブジェクトに対してコールし、以下の戻り値についてテストします。

SM_AGENTAPI_RESPATTR_FLAGS_UNRESOLVED (値 = 2)

注釈

GetVariables メソッドは、パラメータ リストをクリアします。

IdleTimeout メソッド -- セッションのアイドル タイムアウト値を取得

IdleTimeout メソッドは、秒単位のセッションのアイドル タイムアウト値を取得します。アイドル タイムアウト値はユーザセッションを非アクティブな状態にできる時間の最大長です。この時間を超過すると、セッションは有効ではなくなり、ユーザは再認証する必要があります。

構文

IdleTimeout メソッドの形式は以下のとおりです。

```
Netegrity::AgentSession->IdleTimeout()
```

パラメータ

IdleTimeout メソッドは、パラメータを受け入れません。

戻り値

IdleTimeout メソッドは、以下の値を返します。

- idle_timeout_value (long)

MaxTimeout メソッド -- セッションの最大タイムアウト値を取得

MaxTimeout メソッドは、秒単位のセッションの最大タイムアウト値を取得します。最大タイムアウト値はユーザセッションをアクティブな状態にできる時間の長さです。この時間を超過すると、ユーザは再認証する必要があります。

構文

MaxTimeout メソッドの形式は以下のとおりです。

```
Netegrity::AgentSession->MaxTimeout()
```

パラメータ

MaxTimeout メソッドは、パラメータを受け入れません。

戻り値

MaxTimeout メソッドは、以下の値を返します。

- maximum_timeout_value (long)

SetVariables メソッド -- セッションストアにセッション変数を書き込み

SetVariables メソッドは、パラメータリスト内のセッション変数をセッションストアに書き込みます。最初に `AgentSession->AddParameter` をコールして、リスト内の変数を指定します。変数がセッションストアに存在する場合、その値は新しい値に更新されます。変数が存在しない場合は、新しい変数が作成されます。

構文

SetVariables メソッドの形式は以下のとおりです。

```
Netegrity::AgentSession->SetVariables()
```

パラメータ

SetVariables メソッドは、パラメータを受け入れません。

戻り値

SetVariables メソッドは、値を返しません。

注釈

SetVariables メソッドは、パラメータ リストをクリアします。

シングル サインオントークン メソッド

Decode メソッド -- シングル サインオントークンをデコード

Decode メソッドはシングル 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。さらに、ゼロ以外の値をこのメソッドに渡して、トークンの最後にアクセスされたタイムスタンプを更新するオプションがあります。更新されたトークンを文字列形式で取得するには、SSOToken->GetString をコールして、トークン文字列を SMSESSION Cookie に書き込みます。

構文

Decode メソッドの形式は以下のとおりです。

```
Netegrity::SSOToken->Decode([update])
```

パラメータ

Decode メソッドは以下のパラメータを受け入れます。

update (int)

(オプション) 以下のように更新されたトークンが要求されるかどうかを指定します。

- 値 = ゼロ以外

更新されたトークンが要求されることを指定します。

- 値 = 0 (デフォルト)

更新されたトークンが要求されないことを指定します。

戻り値

Decode メソッドは以下のいずれかの値を返します。

- 以下のサブセットが含まれる属性の配列
 - ATTR_CLIENTIP
ユーザが保護されているリソースの要求を開始したマシンの IP アドレスを指定します。
 - ATTR_DEVICENAME
トークンをデコードするエージェントの名前を指定します。
 - ATTR_IDLESESSIONTIMEOUT
セッションの最大アイドル時間を指定します。
 - ATTR_LASTSESSIONTIME
ポリシー サーバがセッション内で最後にアクセスされた時間を指定します。
 - ATTR_MAXSESSIONTIMEOUT
セッションをアクティブにできる最大時間を指定します。
 - ATTR_SESSIONID
ログイン コールから返されるセッション ID を指定します。
 - ATTR_SESSIONSPEC
ログイン コールから返されるセッション指定情報を指定します。
 - ATTR_STARTSESSIONTIME
ログインに成功した後にセッションがいつ開始されたかを指定します。
 - ATTR_USERDN
ユーザの識別名を指定します。
 - ATTR_USERNAME
ユーザの名前を指定します。

- undef

メソッドが失敗したことを指定します。

注釈

シングルサインオン オブジェクトを作成するには、`AgentUser->CreateSSOToken` をコールします。

GetString メソッド -- SSO トークン オブジェクトの文字列表現を取得

`GetString` メソッドは、シングルサインオン トークン オブジェクトの文字列表現を取得します。`GetString` のコール後に、`SMSESSION Cookie` にトークン文字列を書き込むことができます。

構文

`GetString` メソッドの形式は以下のとおりです。

```
Netegrity::SSOToken->GetString()
```

パラメータ

`GetString` メソッドは、パラメータを受け入れません。

戻り値

`GetString` メソッドは、以下の値を返します。

- `SSO_token` (文字列)

注釈

`GetString` は、`CreateSSOToken` でシングルサインオン トークン オブジェクトを作成した後にコールできます。`Decode` で最後にアクセスされたトークンのタイム スタンプを更新した後に `GetString` をコールすることもできます。

GetVersion メソッド -- SSO トークンの SiteMinder バージョンを取得

GetVersion メソッドは、シングル サインオン トークンの SiteMinder バージョンを取得します。

構文

GetVersion メソッドの形式は以下のとおりです。

```
Netegrity::SSOToken->GetVersion()
```

パラメータ

GetVersion メソッドは、パラメータを受け入れません。

戻り値

GetVersion メソッドは、以下の値を返します。

- version (int)

シングル サインオン トークンの SiteMinder バージョンを指定します。

IsThirdParty メソッド -- トークンがカスタムかどうかを判定

IsThirdParty メソッドは、トークンが元はカスタム (またはサードパーティ) エージェントによって作成され、標準の SiteMinder エージェントによってまだ更新されていないかどうかを判定します。

構文

IsThirdParty メソッドの形式は以下のとおりです。

```
Netegrity::SSOToken->IsThirdParty()
```

パラメータ

IsThirdParty メソッドは、パラメータを受け入れません。

戻り値

IsThirdParty メソッドは以下のいずれかの整数値を返します。

- 値 = ゼロ以外

トークンが元はカスタム エージェントによって作成され、標準の SiteMinder エージェントによってまだ更新されていないことを指定します。

- 値 = 0

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ないか、標準の SiteMinder エージェントによって更新されていることを指定します。

ユーザ メソッド

Audit メソッド -- エージェント キャッシュから実行された許可を監査

Audit メソッドは、エージェント キャッシュから実行された許可を監査します。

構文

Audit メソッドの形式は以下のとおりです。

```
Netegrity::AgentUser->Audit()
```

パラメータ

Audit メソッドは、パラメータを受け入れません。

戻り値

Audit メソッドは以下のいずれかの値を返します。

- SM_AGENTAPI_YES (値 = 1)

監査が成功したことを指定します。

- SM_AGENTAPI_NO (値 = 2)

監査が成功しなかったことを指定します。

- **SM_AGENTAPI_FAILURE** (値 = -1)
ポリシー サーバに到達できなかったことを指定します。
- **SM_AGENTAPI_TIMEOUT** (値 = -2)
メソッドがタイムアウトしたことを指定します。
- **SM_AGENTAPI_NOCONNECTION** (値 = -3)
初期化が失敗したことを指定します。

Certificate メソッド -- ユーザの X.509 証明書を設定または取得

Certificate メソッドは、ユーザの X.509 証明書を設定または取得します。このメソッドは、ユーザ オブジェクトの現在のインスタンスと関連付けられた証明書データにのみ影響します。

構文

Certificate メソッドの形式は以下のとおりです。

```
Netegrity::AgentUser->Certificate([cert, certBinaryLen])
```

パラメータ

Certificate メソッドは、以下のパラメータを受け入れます。

cert (文字列)

(オプション) 設定する証明書データを指定します。

certBinaryLen (int)

(オプション) 証明書の長さを指定します。

戻り値

Certificate メソッドは以下のいずれかの値を返します。

- (**\$String**, **\$Length**)
新規または既存の証明書のデータおよび長さを指定します。
- **undef**
メソッドが失敗したことを指定します。

CertificateFile メソッド -- ファイルを使用してユーザの X.509 証明書を設定または取得

CertificateFile メソッドは、指定された証明書ファイルを使用して、ユーザの X.509 証明書を設定または取得します。

構文

CertificateFile メソッドの形式は以下のとおりです。

```
Netegrity::AgentUser->CertificateFile([certFile[, format]])
```

パラメータ

CertificateFile メソッドは、以下のパラメータを受け入れます。

certFile (文字列)

(オプション) 証明書ファイルの完全パスおよびファイル名を指定します。

format (文字列)

(オプション) 証明書ファイルの形式を指定します。

デフォルト : base64 エンコード化 X.509 (値 = 1)

注: デフォルトはサポートされているファイル形式のみです。

戻り値

CertificateFile メソッドは、以下の値を返します。

- (\$String, \$Length)

新規または既存の証明書のデータおよび長さを指定します。

CreateSSOToken メソッド -- シングル サインオン トークン オブジェクトを作成

CreateSSOToken メソッドは、有効なユーザ セッションからシングル サインオン トークン オブジェクトを作成します。トークンには、カスタム エージェントが標準の **SiteMinder Web** エージェントと共有できる暗号化されたセッションおよびその他の情報が含まれます。標準とカスタムのエージェント間のシングル サインオンを作成するには、エージェントが同じドメインに存在する必要があります。シングル サインオン オブジェクトを作成するには、ユーザは **SiteMinder** エージェントではなくカスタム エージェントにログインする必要があります。

構文

CreateSSOToken メソッドの形式は以下のとおりです。

```
Netegrity::AgentUser->CreateSSOToken(szDn, szName, szIP)
```

パラメータ

CreateSSOToken メソッドは、以下のパラメータを受け入れます。

szDn (文字列)

ユーザの識別名を指定します。

szName (文字列)

ユーザの名前を指定します。

szIP (文字列)

保護されているリソースの要求をユーザが開始したマシンの IP アドレスを指定します。

戻り値

CreateSSOToken メソッドは、以下の値を返します。

- SSOToken (オブジェクト)

注釈

トークン オブジェクトを文字列形式で取得するには、**GetString** メソッドを使用して、トークン文字列を **SMSESSION Cookie** に書き込みます。トークンをデコードするには、その属性のサブセットを取得して **Decode** メソッドを使用します。

CustomData メソッド -- カスタム認証データを設定または取得

CustomData メソッドは、カスタム認証データを設定または取得します。このメソッドはカスタム テンプレートに基づく認証方式をテストするために使用されます。カスタム認証データの形式およびコンテンツは、各 Web サイトの要件に従ってカスタム定義されます。

構文

CustomData メソッドの形式は以下のとおりです。

```
Netegrity::AgentUser->CustomData([customData, length])
```

パラメータ

CustomData メソッドは、以下のパラメータを受け入れます。

customData (文字列)

(オプション) 設定するカスタム認証データを指定します。

length (int)

(オプション) カスタム認証データの長さを指定します。

戻り値

CustomData メソッドは以下のいずれかの値を返します。

- (\$String, \$Length)
新規または既存のカスタム データおよび長さを指定します。
- undef
メソッドが失敗したことを指定します。

FormData メソッド -- HTML フォームベースの認証データを設定または取得

FormData メソッドは、HTML フォームベースの認証データを設定または取得します。このメソッドは HTML フォーム テンプレートに基づく認証方式をテストするために使用されます。formData 文字列は、アンパサンド (&) 文字によって区切られた属性名と値のペアから構成されます。

例：

```
"PASSWORD=$password1&email=$username1@mycompany.com"
```

構文

FormData メソッドの形式は以下のとおりです。

```
Netegrity::AgentUser->FormData([formData])
```

パラメータ

FormData メソッドは以下のパラメータを受け入れます。

formData (文字列)

(オプション) 設定する HTML フォームベースの認証データを指定します。

戻り値

FormData メソッドは以下のいずれかの値を返します。

- authentication_data
新規または既存の HTML フォームベースの認証データを指定します。
- undef
メソッドが失敗したことを指定します。

GetResponse メソッド --IsAuthorized または Login 後のレスポンスを返す

GetResponse メソッドは、AgentUser->IsAuthorized または AgentUser->Login のコール後に、ユーザが許可されているかどうかに関係なく、レスポンスを返します。

構文

GetResponse メソッドの形式は以下のとおりです。

```
Netegrity::AgentUser->GetResponse()
```

パラメータ

GetResponse メソッドは、パラメータを受け入れません。

戻り値

GetResponse メソッドは以下のいずれかの値を返します。

- AgentResponse (オブジェクト)
- undef

GetResponse をコールする前に AgentUser->IsAuthorized または AgentUser->Login をコールしなかったために、メソッドが失敗したことを指定します。

Impersonate メソッド -- 1 人のユーザが別のユーザに偽装することを許可

Impersonate メソッドは、そのユーザとしてログインして、1 人のユーザが別のユーザに偽装できるようにします。たとえば、顧客サービス担当者は、顧客のソフトウェア問題をよりよく理解するために顧客に偽装することができます。

構文

Impersonate メソッドの形式は以下のとおりです。

```
Netegrity::AgentUser->Impersonate(username, resource)
```

パラメータ

Impersonate メソッドは、以下のパラメータを受け入れます。

username (文字列)

偽装するユーザの ID を指定します。

resource (AgentResource オブジェクト)

ログインするリソースを指定します。

戻り値

Impersonate メソッドは以下のいずれかの値を返します。

- SM_AGENTAPI_YES (値 = 1)
偽装に成功したことを指定します。
- SM_AGENTAPI_NO (値 = 2)
偽装に失敗したことを指定します。

- **SM_AGENTAPI_FAILURE** (値 = -1)
操作が失敗したことを指定します。
- **SM_AGENTAPI_TIMEOUT** (値 = -2)
メソッドがタイムアウトしたことを指定します。
- **SM_AGENTAPI_NOCONNECTION** (値 = -3)
初期化が失敗したことを指定します。

注釈

Impersonate メソッドは、偽装する人の元のセッションを破壊せずに、新しいセッションを作成します。インパーソネーションセッションを終了し、偽装する人の元のセッションをリストアするには、**AgentUser->Logout** をコールします。

偽装する対象にできるユーザは 1 回に 1 人のみです。インパーソネーションセッションを連鎖させることはできません。

インパーソネーションは、**Impersonation Authorization Scheme** によって保護されるレルムで開始します。偽装する人は、レルムのユーザに偽装することを許可されている必要があります。また、ユーザはレルム内で偽装することの対象として許可されている必要があります。

ユーザ インパーソネーションの詳細については、「**ポリシー サーバ設定ガイド**」を参照してください。

IsAuthorized メソッド -- ユーザが許可されるかどうかを判定

IsAuthorized メソッドは、指定されたリソース上で指定されたアクションを実行する権限がユーザに与えられているかどうかを判定します。このメソッドが **AgentUser->Login** をコールするのは、**AgentUser->Login** がコールされていない場合です。このメソッドをコールした後に、**AgentUser->GetResponse** をコールします。

構文

IsAuthorized メソッドの形式は以下のとおりです。

```
Netegrity::AgentUser->IsAuthorized(resource[, clientIP][, transID])
```

パラメータ

IsAuthorized メソッドは、以下のパラメータを受け入れます。

resource (AgentResource オブジェクト)

確認するリソースを指定します。

clientIP (文字列)

(オプション) クライアントの IP アドレスを指定します。

transID (文字列)

(オプション) セキュリティ アクティビティとアプリケーション アクティビティを関連付けるためにエージェントが使用するユーザ定義のトランザクション ID を指定します。

戻り値

IsAuthorized メソッドは以下のいずれかの値を返します。

- **SM_AGENTAPI_YES** (値 = 1)
ユーザが許可されることを指定します。
- **SM_AGENTAPI_NO** (値 = 2)
ユーザが許可されないことを指定します。
- **SM_AGENTAPI_FAILURE** (値 = -1)
ポリシー サーバに到達できなかったことを指定します。
- **SM_AGENTAPI_TIMEOUT** (値 = -2)
メソッドがタイムアウトしたことを指定します。
- **SM_AGENTAPI_NOCONNECTION** (値 = -3)
初期化が行われなかったことを指定します。

IsAuthorizedEx メソッド -- ユーザが許可されるかどうかを判定

IsAuthorizedEx メソッドは、指定されたリソース上で指定されたアクションを実行する権限がユーザに与えられているかどうかを判定します。このメソッドが **AgentUser->Login** をコールするのは **AgentUser->Login** がコールされていない場合です。このメソッドをコールした後に、**AgentUser->GetResponse** をコールします。

構文

IsAuthorizedEx メソッドの形式は以下のとおりです。

```
Netegrity::AgentUser->IsAuthorizedEx(resource[, clientIP][, transID])
```

パラメータ

IsAuthorizedEx メソッドは、以下のパラメータを受け入れます。

resource (AgentResource オブジェクト)

確認するリソースを指定します。

clientIP (文字列)

(オプション) クライアントの IP アドレスを指定します。

transID (文字列)

(オプション) セキュリティ アクティビティとアプリケーション アクティビティを関連付けるためにエージェントが使用するユーザ定義のトランザクション ID を指定します。

戻り値

IsAuthorizedEx メソッドは以下のいずれかの値を返します。

- **SM_AGENTAPI_YES** (値 = 1)
ユーザが許可されることを指定します。
- **SM_AGENTAPI_NO** (値 = 2)
ユーザが許可されないことを指定します。
- **SM_AGENTAPI_FAILURE** (値 = -1)
ポリシー サーバに到達できなかったことを指定します。

- `SM_AGENTAPI_TIMEOUT` (値 = -2)
メソッドがタイムアウトしたことを指定します。
- `SM_AGENTAPI_NOCONNECTION` (値 = -3)
初期化が行われなかったことを指定します。

Login メソッド -- セッション ログインおよび検証を実行

Login メソッドは、セッション ログインおよび検証を実行します。このメソッドをコールする前に、ターゲット リソース に対し `AgentResource->IsProtected` をコールします。

構文

Login メソッドの形式は以下のとおりです。
`Netegrity::AgentUser->Login(resource[, clientIP])`

パラメータ

Login メソッドは、以下のパラメータを受け入れます。

resource (AgentResource オブジェクト)

ログインするリソースを指定します。

clientIP (文字列)

(オプション) クライアントの IP アドレスを指定します。

戻り値

Login メソッドは以下のいずれかの値を返します。

- `SM_AGENTAPI_YES` (値 = 1)
ユーザ ログインに成功したことを指定します。
- `SM_AGENTAPI_NO` (値 = 2)
ユーザ ログインに失敗したことを指定します。
- `SM_AGENTAPI_CHALLENGE` (値 = 3)
認証にチャレンジが必要であることを指定します。
- `SM_AGENTAPI_FAILURE` (値 = -1)
操作が失敗したことを指定します。

- **SM_AGENTAPI_TIMEOUT** (値 = -2)
メソッドがタイムアウトしたことを指定します。
- **SM_AGENTAPI_NOCONNECTION** (値 = -3)
オブジェクトが接続されなかったことを指定します。

注釈

すでにログインしている 1 人のユーザが別のユーザとして再度ログインできるようにするには、**AgentUser->Impersonate** をコールします。

Logout メソッド -- ユーザをセッションからログアウト

Logout メソッドは、ユーザをセッションからログアウトします。このメソッドはオプションとしてコールします。ユーザ オブジェクトが Perl スクリプトでその範囲を超えると、ユーザは自動的にログアウトされます。

構文

Logout メソッドの形式は以下のとおりです。

```
Netegrity::AgentUser->Logout()
```

パラメータ

Logout メソッドは、パラメータを受け入れません。

戻り値

Logout メソッドは以下のいずれかの値を返します。

- **SM_AGENTAPI_YES** (値 = 1)
ユーザが正しくログアウトされたことを指定します。
- **SM_AGENTAPI_NO** (値 = 2)
ユーザ ログアウトに失敗したことを指定します。
- **SM_AGENTAPI_CHALLENGE** (値 = 3)
認証にチャレンジが必要であることを指定します。
- **SM_AGENTAPI_FAILURE** (値 = -1)
操作が失敗したことを指定します。

- **SM_AGENTAPI_TIMEOUT** (値 = -2)
メソッドがタイムアウトしたことを指定します。
- **SM_AGENTAPI_NOCONNECTION** (値 = -3)
オブジェクトが接続されなかったことを指定します。

注釈

1 人のユーザが別のユーザに偽装している間に **Logout** をコールすると、インパーソネーションセッションが終了し、偽装する人の元のセッションがリストアされます。**AgentUser->Impersonate** をコールすると、1 人のユーザが別のユーザに偽装するか、別のユーザとしてログインできます。

Name メソッド -- ユーザのユーザ名を設定または取得

Name メソッドは、ユーザのユーザ名を設定または取得します。

構文

Name メソッドの形式は以下のとおりです。

```
Netegrity::AgentUser->Name([username])
```

パラメータ

Name メソッドは以下のパラメータを受け入れます。

username (文字列)

(オプション) 設定するユーザ名を指定します。

戻り値

Name メソッドは、以下の値を返します。

- **username** (文字列)
新規または既存のユーザ名を指定します。

注釈

ユーザ名の設定は、ユーザ オブジェクトの現在のインスタンスにのみ影響します。ディレクトリ内のユーザのエントリには影響しません。

Password メソッド -- ユーザのパスワードを設定または取得

Password メソッドは、ユーザのパスワードを設定または取得します。

構文

Password メソッドの形式は以下のとおりです。

```
Netegrity::AgentUser->Password([password])
```

パラメータ

Password メソッドは以下のパラメータを受け入れます。

password (文字列)

(オプション) 設定するパスワードを指定します。

戻り値

Password メソッドは、以下の値を返します。

- *password* (文字列)

新規または既存のパスワードを指定します。

注釈

パスワードの設定は、ユーザ オブジェクトの現在のインスタンスにのみ影響します。ディレクトリ内のユーザのエントリには影響しません。

Validate メソッド -- セッション指定情報を検証

Validate メソッドはユーザ セッションが期限切れになっていないか、終了または無効になっていないかを確認して、セッション指定情報を検証します。この確認は、セッションの存続中に随時に行われる場合があります。

構文

Validate メソッドの形式は以下のとおりです。

```
Netegrity::AgentUser->Validate(resource[, clientIP][, transID])
```

パラメータ

Validate メソッドは、以下のパラメータを受け入れます。

resource (AgentResource オブジェクト)

ログインするリソースを指定します。

clientIP (文字列)

(オプション) クライアントの IP アドレスを指定します。

transID (文字列)

(オプション) ユーザ定義のトランザクション ID を指定します。

戻り値

Validate メソッドは以下のいずれかの値を返します。

- SM_AGENTAPI_YES (値 = 1)
操作に成功したことを指定します。
- SM_AGENTAPI_NO (値 = 2)
ユーザがログインされなかったことを指定します。
- SM_AGENTAPI_FAILURE (値 = -1)
操作が失敗したことを指定します。
- SM_AGENTAPI_TIMEOUT (値 = -2)
メソッドがタイムアウトしたことを指定します。
- SM_AGENTAPI_NOCONNECTION (値 = -3)
オブジェクトが接続されなかったことを指定します。

注釈

ポリシー サーバはセッション指定情報またはセッション ID を以下のように検証します。

- セッション ID がセッション指定情報なしで指定される場合、ポリシーサーバは検証用のセッション ID を使用します。
- セッション ID がセッション指定情報と共に指定される場合、ポリシーサーバはセッション指定情報に対してセッション ID を検証します。
- クライアントの IP アドレスがセッション指定情報と共に指定される場合、ポリシーサーバはセッション指定情報に対して IP アドレスを検証します。

第 4 章: CLI エージェント操作

このセクションには、以下のトピックが含まれています。

[リソース保護](#) (P. 93)

[レスポンスとレスポンス属性](#) (P. 95)

[セッション管理](#) (P. 97)

[ポリシー サーバ コマンド](#) (P. 99)

リソース保護

ユーザがサイトにログインして、保護されているリソースにアクセスを試みるときに、通常の場合エージェントは以下の質問に答える必要があります。

- 要求されたリソースは保護されていますか?
- ユーザはログインに対して認証できますか?
- リソースにアクセスする権限はユーザに与えられていますか?

以下のスクリプトは、どのようにエージェント API を使用して、これらの基本的なエージェント質問に対処し応答できるかを示しています。

```
use Netegrity::AgentAPI;

#Define script variables
$agent = "agent1";
$secret = "oursecret";
$ip = "127.0.0.1";
$respath = "/mysite/hr/payroll.htm";
$username = "userid";
$password = "userpwd";

print "\nStep 1. Connecting to Policy Server...\n";
$agentapi = Netegrity::AgentAPI->New($agent, $secret);
$serverconfig = $agentapi->AddServerConfig($ip);
$status=$agentapi->Connect();
```

```
die "FATAL: Connect() failed with error code " .
    $status unless($status==SM_AGENTAPI_YES);

$resource = $agentapi->GetResource($respath);
print "Step 2. Is the resource protected?";
if ($resource->IsProtected == SM_AGENTAPI_YES) {
    print "Resource ".$respath." is protected.";

    print "Step 3. User login...";
    $user = $agentapi->CreateUser($username, $pwd);
    print "Logging in user ".$user->Name().";
    $status = $user->Login($resource);
    if($status==SM_AGENTAPI_YES) {
        print $user->Name() . " logged in successfully!";

        print "Step 4. User authorized for the resource?";
        $status = $user->IsAuthorized($resource);
        if($status==SM_AGENTAPI_YES) {
            print $user->Name()." is authorized for " .
                $respath . "
        }
    }
    else {
        print $user->Name()." is not authorized for " .
            $respath . "
    }
}
else {
    print "Couldn't log in user " . $username . "
}
}
else {
    print "Resource ".$respath." is not protected."
}
}
```

レスポンスとレスポンス属性

エージェントが `Login()` または `IsAuthorized()` へのコールを通じて要求を発行した後、要求に対するレスポンスが返されます。ユーザーが許可されるかどうかに関係なくレスポンスが返されます。

返されたレスポンス オブジェクトと共に、以下のエージェント API オブジェクトを取得できます。

- セッション オブジェクト。セッション オブジェクトでは、セッション ID およびタイムアウト値などのセッション情報を取得できます。
- レスポンス属性オブジェクト。ポリシー サーバは、エージェントに情報を送信するためにレスポンス属性を使用します。エージェントはその後クライアント アプリケーションにその情報を渡すことができます。

レスポンス属性の取得

コールのこのシーケンスは、ユーザの識別名、ユーザと関連付けられたディレクトリの名前、およびユーザの認証と関連付けられた任意のテキストなどの既知の属性を取得します。「オンライン エージェント API リファレンス」には、`GetAttributes()` が取得できる既知の属性のリストがあります。

レスポンス属性を取得する方法

1. `AgentUser` メソッドの `Login()` および `GetResponse()` とコールします。
2. `GetResponse()` から返される `AgentResponse` オブジェクトと共に、`GetAttributes()` をコールします。

例：レスポンス用の属性の取得

以下のスクリプトは、ログイン要求に対するレスポンス用の属性を取得します。スクリプトはその後 `AgentResponseAttr` オブジェクトでメソッドをコールし、属性の ID、任意の属性値などの属性情報を表示します。

```
use Netegrity::AgentAPI;

$agentname="agent1";
$ip="127.0.0.1";
$sharedsecret="oursecret";
$username="userid";
$userpwd="userpwd";
```

```
$agentapi=Netegrity::AgentAPI->New($agentname,$sharedsecret);
#Add Policy Server configuration info...
$serverconfig = $agentapi->AddServerConfig($ip);
$agentapi->Connect();

$resource=$agentapi->GetResource("/mysite/hr/payroll.htm");
# Test whether the resource is protected. If it is,
# log in the user and get the attributes of the response.
if($resource->IsProtected() == SM_AGENTAPI_YES) {
    $user = $agentapi->CreateUser($username,$userpwd);
    print "Logging in user ".$user->Name()."...\n";
    $status=$user->Login($resource);
    if($status==SM_AGENTAPI_YES) {
        $response=$user->GetResponse();
        @attr = $response->GetAttributes();
        foreach $attr(@attr) {
            print "Attribute ID = " . $attr->GetID()."\n";
            print "TTL = " . $attr->GetTTL()."\n";
            print "Value = " . $attr->GetValue()."\n";
            print "Name = " . $attr->GetName()."\n";
        }
    }
} else {
    print "The resource is not protected.\n\n";
}
```

セッション管理

AgentSession オブジェクトを取得した後、セッション管理操作を実行できます。セッション タイムアウト値、セッション ID およびセッション指定情報などのセッションに関する情報を取得できます。

シングルサインオン操作の場合のように、複数のサイトにわたってセッションを識別するために**セッション指定情報**を使用できます。また、**GetReason()** をコールして、失敗した認証または許可試行の理由コードを取得できます。

例：ユーザのログイン

以下の例ではユーザをログインして、ログイン試行に対するレスポンスを取得し、ユーザのセッションのセッション オブジェクトを取得して、セッションに関するさまざまな詳細を出力します。

```
use Netegrity::AgentAPI;

#Define script variables
$agent = "agent1";
$secret = "oursecret";
$ip = "127.0.0.1";
$respath = "/mysite/hr/payroll.htm";
$username = "userid";
$pwd = "userpwd";

#Establish the connection and create needed objects
$agentapi = Netegrity::AgentAPI->New($agent, $secret);
$serverconfig = $agentapi->AddServerConfig($ip);
$agentapi->Connect();
$user = $agentapi->CreateUser($username, $pwd);
print "Logging in user ".$user->Name().".\n";
$resource = $agentapi->GetResource($respath);
#Log in the user
$status = $user->Login($resource);
if($status==SM_AGENTAPI_YES) {
    print $user->Name() . " logged in successfully!\n\n";
    #Get the login response
    $response=$user->GetResponse();
    #Get the session object
    $session=$response->GetSession();
    if ($session != undef) {
        print "Printing session details:\n";
        print "Session reason=".$session->GetReason()."\n";
        print "Session IdleTimeout=".$session->IdleTimeout()."\n";
        print "Session Maxtimeout=".$session->MaxTimeout()."\n";
        print "Session ID=".$session->GetID()."\n";
        print "Session specification=".$session->GetSpec()."\n\n";
    }
}
else {
    print "Couldn't log in user " . $username . ".\n\n";
}
```

ポリシー サーバコマンド

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。エージェントは、ポリシー サーバ上のポリシーおよび管理上の変更と一致するようにそのキャッシュと暗号化キーを管理するためにこのプロトコルを使用します。

対応するエージェントの変更を必要とするポリシー サーバの変更が行われると、ポリシー サーバは、エージェントで変更を行うためのコマンドを発行します。 **DoManagement()** をコールして、これらのコマンドを取得できます。 **DoManagement()** は、 **AgentResponseAttr** オブジェクトの配列を返します。ポリシー サーバから保留中の場合がある管理コマンドを取得するためにこのオブジェクトで **GetID()** をコールします。

以下のスクリプトは保留中の管理コマンドについて確認します。 コマンドが見つかり、スクリプトは以下のように各コマンド ID、および必要に応じてコマンドと関連付けられた値を出力します。

```
use Netegrity::AgentAPI;

$agentname="agent1";
$ip="127.0.0.1";
$secret="oursecret";

$agentapi = Netegrity::AgentAPI->New($agentname,$secret);
$serverconfig = $agentapi->AddServerConfig($ip);
$agentapi->Connect();

@attr = $agentapi->DoManagement();

if (@attr == undef) {
    print "No commands are pending from the Policy Server.";
}
else {
    print "IDs of commands pending from the Policy Server:";
    foreach $attr(@attr) {
        if ($attr->GetID()==SM_AGENTAPI_AFFILIATE_KEY_UPDATE ||
            $attr->GetID()==SM_AGENTAPI_CACHE_FLUSH_THIS_USER ||
            $attr->GetID()==SM_AGENTAPI_CACHE_FLUSH_THIS_REALM) {
            print "\n Command ID: " . $attr->GetID();
            print "\n \tValue: " . $attr->GetValue();
        }
    }
    else {
        #No need to print values for the IDs below. 値
        #will contain either binary data or no data.
        print "\nCommand ID: " . $attr->GetID();
    }
}
}
```

第 5 章: ポリシー管理 API について

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 サーバおよびポリシー管理 API \(P. 101\)](#)

[ポリシー管理 API に対するスクリプトの作成 \(P. 102\)](#)

[フェデレーションプログラミング \(P. 103\)](#)

[アフィリエイトドメイン \(P. 112\)](#)

[CLI 認証方式の設定 \(P. 113\)](#)

ポリシー サーバおよびポリシー管理 API

ポリシー サーバは、保護されているリソースおよびそれらのリソースにアクセスするための要件を説明するデータを管理します。また、ポリシー サーバは、保護されているドメインの管理者に関する情報を管理します。このセキュリティ データは LDAP などのポリシー ストアにあります。

ポリシー管理 API（モジュール `Netegrity::PolicyMgtAPI`）では、管理 UI で実行できる、ほとんどのポリシー サーバ設計と管理操作を実行できます。たとえば、以下の操作を実行できます。

- ドメイン、レルム、ルールおよびポリシーなどのポリシー ストア オブジェクトの作成、変更、削除
- 管理者およびユーザ ディレクトリの管理

さらに、ポリシー管理 API データ管理オブジェクト（`PolicyMgtDataMgr`）では、`SiteMinder smobjexport` および `smobjimport` ツールで可能となるようなポリシー ストアまたはドメイン全体ではなく、あるポリシー ストアから別のストアに特定のオブジェクトをコピーできます。

ポリシー管理 API の場所

ポリシー管理 API は、ターゲット ポリシー サーバが存在するマシン上にインストールする必要があります。ポリシー管理 API はリモート ポリシー サーバにアクセスできません。ただし、ポリシー サーバがリモートポリシー ストアをポイントするように設定されている限り、ポリシー ストアはリモート マシン上に存在できます。

ポリシー管理 API に対するスクリプトの作成

ポリシー管理 API に対してスクリプトを作成するときは、以下の基本手順を実行します。

1. 以下のようにスクリプトの始めのポリシー管理 API を参照します。

```
use Netegrity::PolicyMgtAPI;
```

2. 以下のように **New()** メソッドを使用してポリシー管理 API オブジェクトを作成します。

```
$policymgtapi = Netegrity::PolicyMgtAPI->New();
```

3. オプションで、**DisableValidation()** などの **PolicyMgtAPI** メソッドによって 1 つ以上のポリシー サーバ初期化フラグを設定します。デフォルトでは、初期化フラグはすべて 0 に設定されます。

4. 以下のように、ポリシー サーバでセッションを作成します。

```
$session = $policymgtapi->CreateSession("userid",  
                                         "password",  
                                         "127.0.0.1" );
```

ポリシー サーバ オブジェクトに対して操作を実行できるようになります。たとえば、ポリシー サーバ内の設定されたエージェントのリストを取得して印刷できます。

```
@agents = $session->GetAllAgents();  
foreach $agent (@agents) {  
    print "Agent Name = " . $agent->Name() . "\n";  
}
```

スクリプト実行パフォーマンスの強化

ポリシー管理スクリプトを使用してポリシー ストアで変更を加えるのにかかる時間を短縮できます。そのためには、初期化時に **PreLoadCache()** で 0 を渡します。デフォルトでは、キャッシュの事前ロードは無効になっています。

フェデレーション プログラミング

SiteMinder フェデレーション プログラミング インターフェースでは、インターネット上でフェデレートされた関係にあるビジネスが、いずれかのフェデレートされたサイトにアクセスするユーザに関して情報を安全に共有できます。ユーザは認証情報を 2 回以上入力することなく、連携サイトにアクセスできます。

アフィリエイトされたサイトでは、アサーションと呼ばれる業界標準の Security Assertion Markup Language (SAML) ドキュメントを通じてユーザ情報を共有します。

SiteMinder のフェデレーション製品では、SAML 1.x および SAML 2.0 プロトコルをサポートしています。

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
$aaffDom=$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
$sp=$aaffDom->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 属性認証機関

SiteMinder は、リモート サイトに事前決定されたユーザ属性の値を要求し、許可決定のベースとしてそれらの値を使用して、リソースへのユーザ アクセスの許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インして、料金について問い合わせるとします。 顧客は取り扱い店によって認証されますが、競争力のある料金を提供するため、取り扱い店は顧客の好みの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。 航空会社は、品質コードの値（たとえば 1A）を返します。その後、取り扱い店は顧客に適切な料金シートを表示します。

この例では、レンタカー取扱店が SAML リクエストとして機能し、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。

ユーザ属性に基づいて許可をサポートするために、Perl ポリシー管理 API には、`PolicyMgtSession` オブジェクトに以下の 3 つのメソッドが含まれます。

- `AddAttributeToSAMLScheme()`
- `GetAllSAMLSchemeAttributes()`
- `RemoveAttributeFromSAMLScheme()`

`PolicyMgtSAMLServiceProvider->AddAttribute` メソッドは、SAML リクエストが要求できるサービス プロバイダ（属性認証機関）への属性の追加をサポートします。

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ときに、アサーション コンシューマ サービスの複数のエンドポイントを指定できます。これは、サービス プロバイダで SAML アサーションを使用できるようにするサービスです。それぞれのエンドポイントには、アサーション コンシューマ サービス URL への単一の明示的参照ではなく、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、アイデンティティ プロバイダに送信するアサーションのサービス プロバイダによる要求の一部として使用できます。これにより、異なるプロトコル バインディング用にサービス プロバイダで別のアサーション コンシューマ サービスを有効にできます。

Perl スクリプトでは、以下のように `PolicyMgtSAMLServiceProvider` オブジェクトで `AddAssertionConsumerServiceToSAMLSP()` メソッドを使用して、インデックス値、プロトコルバインディングおよびインデックス付けエンドポイントの URL を指定します。

```
$res = $sp->AddAssertionConsumerServiceToSAMLSP(index, protocolBinding, URL);
```

このメソッドは `PolicyMgtSAMLSPACS` オブジェクトを返します。すべてのアサーション コンシューマ サービス オブジェクトを取得、およびアサーション コンシューマ サービスを削除するための `PolicyMgtSAMLServiceProvider` のメソッドもあります。

`PolicyMgtSAMLSPACS` オブジェクトには、インデックス、プロトコルバインディングおよびアサーション コンシューマ URL の値を取得するメソッドが含まれます。

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。

この環境内の 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-Federation セキュリティ トークンを提供し、それらをリソース パートナーへ渡します。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。

Perl スクリプトでは、以下のように

`PolicyMgtAffDomain->CreateWSFEDResourcePartner` メソッドをコールして、リソース パートナーを定義します。

```
$aff = $affDomain->CreateWSFEDResourcePartner(propsHash_ref);
```

`propsHash_ref` は、リソース パートナーに対して定義されたメタデータ プロパティのハッシュ テーブルへの参照です。

このメソッドは `PolicyMgtWSFEDResourcePartner` オブジェクトを返します。`PolicyMgtWSFEDResourcePartner` オブジェクトには、リソース パートナーでユーザを管理するためのメソッドが含まれます（`AddUser`、`GetAllUsers` および `RemoveUser`）。`PolicyMgtWSFEDResourcePartner->Property()` メソッドは、データ ストアに変更をサブミットしないことに注意してください。`PolicyMgtWSFEDResourcePartner->Save()` メソッドをコールする必要があります。

Perl スクリプトでアカウント パートナーを定義するには、`PolicyMgtSession->CreateWSFEDAuthScheme()` をコールして、WS-Federation 認証方式のインスタンスを作成します。この認証方式用にメタデータ プロパティを設定または取得するには、`PolicyMgtSession->WSFEDAuthSchemeProperties()` をコールします。

WS-Federation 認証方式を特に削除または取得するメソッドはありません。他のタイプの認証方式の場合と同じように、`DeleteAuthScheme`、`GetAuthScheme` および `GetAllAuthSchemes` を使用します。.

詳細情報:

[WS-フェデレーション リソース パートナー メソッド \(P. 585\)](#)

[CLI WS-フェデレーション リソース パートナー属性メソッド \(P. 595\)](#)

サンプル スクリプト

サンプル スクリプトの `AffiliateDemo.pl`、`SAMLServiceProvider.pl`、`WSFEDAccountPartner.pl`、および `WSFEDResourcePartner.pl` が、レガシー フェデレーションと共に提供されます。

まだインストールしていない場合は、`ampolicy.smdif` をポリシー ストアへインポートしてからサンプル スクリプトを使用してください。`ampolicy.smdif` のデフォルトの場所は、`<siteminder_install_dir>%db%SMdif` です。

アフィリエイトドメイン

フェデレートされたビジネス ネットワーク内のサイトを表すアフィリエイト オブジェクトは、ポリシー サーバアフィリエイト ドメイン内に含まれます。アフィリエイト ドメインには、**SAML 1.x** アフィリエイトおよび **SAML 2.0** サービス プロバイダを含めることができます。

また、アフィリエイト ドメインには、ドメイン内のアフィリエイトとサービス プロバイダと関連付けられた 1 つ以上のユーザディレクトリ、およびドメインを管理できる管理者アカウントへの参照が含まれます。

アフィリエイト オブジェクトおよびその存在場所であるアフィリエイト ドメインを設定するには、コマンドライン インターフェースを使用できます。

CLI 認証方式の設定

Perl スクリプトを介して認証方式を設定するときに、通常は 管理 UI の [認証方式プロパティ] ダイアログ ボックスを通じて指定する情報を指定します。このセクションでは、ポリシー管理 API を使用して、指定された認証方式を設定するために必要な 情報について説明します。

注: 認証方式を変更するときは、必ずすべての設定メソッドをコールした後に `Save()` をコールしてください。

設定情報

通常は、`CreateAuthScheme()` で方式を作成するとき、または `PolicyMgtAuthScheme` オブジェクト内のメソッドで方式を変更するときに、認証方式を設定します。

注: このルールの例外は **SAML 2.0** テンプレートに基づく認証方式です。メソッド `CreateSAMLAuthScheme()` で **SAML 2.0** 認証方式を作成し設定します。

認証方式用の以下の種類の設定情報を提供できます。すべての認証方式 テンプレートですべてのカテゴリの設定情報を使用するとは限りません。

- 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: Perl ポリシー管理 API でテンプレートとして認証方式を設定することは、SiteMinder v6.0 SP3 では非推奨となっています。

- 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?

ユーザの認証情報が保存されているかどうかを示すフラグ。

- 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。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。

設定テーブル

以下のテーブルは、認証方式の設定に役立ちます。それぞれのテーブルは特定の認証方式タイプに適用され、以下の情報が含まれています。

- 特定の種類の設定値を設定または取得する **PolicyMgtAuthScheme** メソッド。
- デフォルト設定値（該当する場合）、または指定する設定値を表す斜体表示の変数名。
- 認証方式を作成しているときに、設定値を指定するために使用する **CreateAuthScheme()** 引数の名前。

情報タイプ列の値は異なる認証方式で別の目的に使用できます。たとえば、TeleID 認証方式では、共有秘密キーを使用して暗号化シードを提供します。

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にこの表を使用します。

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	Type(<i>templateObject</i>) CreateAuthScheme() param: <i>schemeTemplate</i> 匿名方式タイプです。

情報タイプ	値の割り当てと意味
説明	Description(<i>schemeDesc</i>) CreateAuthScheme() param: <i>schemeDesc</i> 認証方式の説明。
保護レベル	ProtectionLevel(0) CreateAuthScheme() param: <i>protLevel</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	CustomLib("smauthanon") CreateAuthScheme() param: <i>schemeLib</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	CustomParam(<i>param</i>) CreateAuthScheme() param: <i>schemeParam</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	CustomSecret("") CreateAuthScheme() param: <i>secret</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?	IsTemplate(<i>templateFlag</i>) CreateAuthScheme() param: <i>isTemplate</i> 方式がテンプレートではないことを示すには 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?	IsUsedByAdmin(0) CreateAuthScheme() param: <i>isUsedByAdmin</i> 0 に設定 -- 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?	SaveCredentials(0) CreateAuthScheme() param: <i>saveCreds</i> ユーザ認証情報が保存されないことを示すには 0 に設定します。
RADIUS ですか?	IsRadius(0) CreateAuthScheme() param: <i>isRadius</i> 0 に設定 -- 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?	IgnorePwd(1) CreateAuthScheme() param: <i>ignorePwd</i> 1 に設定 -- 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

SSL を介した基本方式タイプに基づいて認証方式を設定するときはこのテーブルを使用します。

情報タイプ	値の割り当てと意味
方式タイプ	Type(<i>templateObject</i>) CreateAuthScheme() param: <i>schemeTemplate</i> SSL を介した基本方式タイプです。
説明	Description(<i>schemeDesc</i>) CreateAuthScheme() param: <i>schemeDesc</i> 認証方式の説明。
保護レベル	ProtectionLevel(<i>nLevel</i>) CreateAuthScheme() param: <i>protLevel</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 10 です。
ライブラリ	CustomLib("smauthcert") CreateAuthScheme() param: <i>schemeLib</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	CustomParam(<i>param</i>) CreateAuthScheme() param: <i>schemeParam</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) の名前が含まれる文字列。形式 https://server/SCC?basic 以下の例ではデフォルトの SCC を使用します。 https://my.server.com/siteminderagent/ nocert/smgetcred.scc?basic
共有秘密キー	CustomSecret("") CreateAuthScheme() param: <i>secret</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?	IsTemplate(<i>templateFlag</i>) CreateAuthScheme() param: <i>isTemplate</i> 方式がテンプレートではないことを示すには 0 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?	IsUsedByAdmin(0) CreateAuthScheme() param: <i>isUsedByAdmin</i> この方式には 0 を設定します。
認証情報を保存しますか?	SaveCredentials(0) CreateAuthScheme() param: <i>saveCreds</i> ユーザ認証情報が保存されないことを示すには 0 に設定します。
RADIUS ですか?	IsRadius(0) CreateAuthScheme() param: <i>isRadius</i> 0 に設定 -- 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?	IgnorePwd(flag) CreateAuthScheme() param: <i>ignorePwd</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。

基本テンプレート

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。

情報タイプ	値の割り当てと意味
方式タイプ	Type(<i>templateObject</i>) CreateAuthScheme() param: <i>schemeTemplate</i> 基本方式タイプです。
説明	Description(<i>schemeDesc</i>) CreateAuthScheme() param: <i>schemeDesc</i> 認証方式の説明。
保護レベル	ProtectionLevel(<i>nLevel</i>) CreateAuthScheme() param: <i>protLevel</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	CustomLib("smauthdir") CreateAuthScheme() param: <i>schemeLib</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	CustomParam("") CreateAuthScheme() param: <i>schemeParam</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	CustomSecret("") CreateAuthScheme() param: <i>secret</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?	IsTemplate(<i>templateFlag</i>) CreateAuthScheme() param: <i>isTemplate</i> 方式がテンプレートではないことを示すには 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?	IsUsedByAdmin(1) CreateAuthScheme() param: <i>isUsedByAdmin</i> 1 に設定 -- 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?	SaveCredentials(0) CreateAuthScheme() param: <i>saveCreds</i> ユーザ認証情報が保存されないことを示すには 0 に設定します。
RADIUS ですか?	IsRadius(1) CreateAuthScheme() param: <i>isRadius</i> 1 に設定 -- 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?	IgnorePwd(<i>flag</i>) CreateAuthScheme() param: <i>ignorePwd</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。

カスタム テンプレート

カスタ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表を使用します。SiteMinder SDK で利用可能な C 言語認証 API を使用して、カスタム方式を作成します。

情報タイプ	値の割り当てと意味
方式タイプ	Type(<i>templateObject</i>) CreateAuthScheme() param: <i>schemeTemplate</i> カスタム方式タイプです。

情報タイプ	値の割り当てと意味
説明	Description(<i>schemeDesc</i>) CreateAuthScheme() param: <i>schemeDesc</i> 認証方式の説明。
保護レベル	ProtectionLevel(<i>nLevel</i>) CreateAuthScheme() param: <i>protLevel</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	CustomLib(<i>customLibName</i>) CreateAuthScheme() param: <i>schemeLib</i> C 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	CustomParam(<i>param</i>) CreateAuthScheme() param: <i>schemeParam</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	CustomSecret(<i>secret</i>) CreateAuthScheme() param: <i>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?	IsTemplate(<i>templateFlag</i>) CreateAuthScheme() param: <i>isTemplate</i> 方式がテンプレートではないことを示すには 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?	IsUsedByAdmin(<i>flag</i>) CreateAuthScheme() param: <i>isUsedByAdmin</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 true (1)、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 false (0) を指定します。デフォルトは 0 です。
認証情報を保存しますか?	SaveCredentials(0) CreateAuthScheme() param: <i>saveCreds</i> ユーザ認証情報が保存されないことを示すには 0 に設定します。

情報タイプ	値の割り当てと意味
RADIUS ですか?	IsRadius(0) CreateAuthScheme() param: <i>isRadius</i>
パスワードチェックを無視しますか?	IgnorePwd(flag) CreateAuthScheme() param: <i>ignorePwd</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。

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にこのテーブルを使用します。

情報タイプ	値の割り当てと意味
方式タイプ	Type(<i>templateObject</i>) CreateAuthScheme() param: <i>schemeTemplate</i> HTML フォーム方式タイプです。
説明	Description(<i>schemeDesc</i>) CreateAuthScheme() param: <i>schemeDesc</i> 認証方式の説明。
保護レベル	ProtectionLevel(<i>nLevel</i>) CreateAuthScheme() param: <i>protLevel</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	CustomLib("smauthhtml") CreateAuthScheme() param: <i>schemeLib</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	<code>CustomParam(<i>param</i>)</code> <code>CreateAuthScheme()</code> param: <i>schemeParam</i> 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 AL= 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。たとえば、以下のようになります。 AL=Password,SSN,age,zipcode; 完全なパラメータ形式は次のとおりです。 attr-list;https://server/fcc 以下の例ではデフォルトの FCC を使用します。 AL=PASSWORD,SSN,age,zipcode; <code>http://my.server.com/siteminderagent/forms/login.fcc</code>
共有秘密キー	<code>CustomSecret("")</code> <code>CreateAuthScheme()</code> param: <i>secret</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?	<code>IsTemplate(<i>templateFlag</i>)</code> <code>CreateAuthScheme()</code> param: <i>isTemplate</i> 方式がテンプレートではないことを示すには 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?	<code>IsUsedByAdmin(0)</code> <code>CreateAuthScheme()</code> param: <i>isUsedByAdmin</i> 0 に設定 -- 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>SaveCredentials(<i>credFlag</i>)</code> <code>CreateAuthScheme()</code> param: <i>saveCreds</i> ユーザ認証情報を保存することを示すには 1、ユーザ認証情報を保存しないことを示すには 0 に設定します。デフォルトは 0 です。
RADIUS ですか?	<code>IsRadius(0)</code> <code>CreateAuthScheme()</code> param: <i>isRadius</i> 0 に設定 -- 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>IgnorePwd(<i>flag</i>)</code> <code>CreateAuthScheme()</code> param: <i>ignorePwd</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。

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にこのテーブルを使用します。

情報タイプ	値の割り当てと意味
方式タイプ	Type(<i>templateObject</i>) CreateAuthScheme() param: <i>schemeTemplate</i> RADIUS CHAP/PAP 方式タイプです。
説明	Description(<i>schemeDesc</i>) CreateAuthScheme() param: <i>schemeDesc</i> 認証方式の説明。
保護レベル	ProtectionLevel(<i>nLevel</i>) CreateAuthScheme() param: <i>protLevel</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	CustomLib("smauthchap") CreateAuthScheme() param: <i>schemeLib</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	CustomParam(<i>param</i>) CreateAuthScheme() param: <i>schemeParam</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。
共有秘密キー	CustomSecret("") CreateAuthScheme() param: <i>secret</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?	IsTemplate(<i>templateFlag</i>) CreateAuthScheme() param: <i>isTemplate</i> 方式がテンプレートではないことを示すには 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?	IsUsedByAdmin(0) CreateAuthScheme() param: <i>isUsedByAdmin</i> 0 に設定 -- 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?	SaveCredentials(0) CreateAuthScheme() param: <i>saveCreds</i> ユーザ認証情報が保存されないことを示すには 0 に設定します。
RADIUS ですか?	IsRadius(1) CreateAuthScheme() param: <i>isRadius</i> 1 に設定 -- 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?	setIgnorePwCheck(flag) 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。

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。

情報タイプ	値の割り当てと意味
方式タイプ	Type(<i>templateObject</i>) CreateAuthScheme() param: <i>schemeTemplate</i> RADIUS サーバ方式タイプです。
説明	Description(<i>schemeDesc</i>) CreateAuthScheme() param: <i>schemeDesc</i> 認証方式の説明。
保護レベル	ProtectionLevel(<i>nLevel</i>) CreateAuthScheme() param: <i>protLevel</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	CustomLib("smauthradius") CreateAuthScheme() param: <i>schemeLib</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	<code>CustomParam(param)</code> <code>CreateAuthScheme()</code> param: <i>schemeParam</i> RADIUS サーバの IP アドレスおよびポートが含まれる文字列。たとえば、 <code>123.123.12.12:1645</code> デフォルト UDP ポートは 1645 です。
共有秘密キー	<code>CustomSecret(secret)</code> <code>CreateAuthScheme()</code> param: <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?	<code>IsTemplate(templateFlag)</code> <code>CreateAuthScheme()</code> param: <i>isTemplate</i> 方式がテンプレートではないことを示すには 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?	<code>IsUsedByAdmin(1)</code> <code>CreateAuthScheme()</code> param: <i>isUsedByAdmin</i> 1 に設定 -- 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>SaveCredentials(0)</code> <code>CreateAuthScheme()</code> param: <i>saveCreds</i> ユーザ認証情報が保存されないことを示すには 0 に設定します。
RADIUS ですか?	<code>IsRadius(1)</code> <code>CreateAuthScheme()</code> param: <i>isRadius</i> 1 に設定 -- 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>IgnorePwd(flag)</code> <code>CreateAuthScheme()</code> param: <i>ignorePwd</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。

SafeWord HTML フォーム 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。

情報タイプ	値の割り当てと意味
方式タイプ	Type(<i>templateObject</i>) CreateAuthScheme() param: <i>schemeTemplate</i> SafeWord HTML フォーム方式タイプ。
説明	Description(<i>schemeDesc</i>) CreateAuthScheme() param: <i>schemeDesc</i> 認証方式の説明。
保護レベル	ProtectionLevel(<i>nLevel</i>) CreateAuthScheme() param: <i>protLevel</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 10 です。
ライブラリ	CustomLib("smauthenigmahtml") CreateAuthScheme() param: <i>schemeLib</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	CustomParam(<i>param</i>) CreateAuthScheme() param: <i>schemeParam</i> 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。 http://my.server.com/ siteminderagent/forms/safeword.fcc
共有秘密キー	CustomSecret("") CreateAuthScheme() param: <i>secret</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?	IsTemplate(<i>templateFlag</i>) CreateAuthScheme() param: <i>isTemplate</i> 方式がテンプレートではないことを示すには 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?	IsUsedByAdmin(1) CreateAuthScheme() param: <i>isUsedByAdmin</i> 1 に設定 -- 方式は管理者を認証するために使用できます。

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?	SaveCredentials(0) CreateAuthScheme() param: <i>saveCreds</i> ユーザ認証情報が保存されないことを示すには 0 に設定します。
RADIUS ですか?	IsRadius(1) CreateAuthScheme() param: <i>isRadius</i> 1 に設定 -- 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?	IgnorePwd(1) CreateAuthScheme() param: <i>ignorePwd</i> 1 に設定 -- 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。

情報タイプ	値の割り当てと意味
方式タイプ	Type(<i>templateObject</i>) CreateAuthScheme() param: <i>schemeTemplate</i> SafeWord 方式タイプ。
説明	Description(<i>schemeDesc</i>) CreateAuthScheme() param: <i>schemeDesc</i> 認証方式の説明。
保護レベル	ProtectionLevel(<i>nLevel</i>) CreateAuthScheme() param: <i>protLevel</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 10 です。
ライブラリ	CustomLib("smauthenigma") CreateAuthScheme() param: <i>schemeLib</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	CustomParam("") CreateAuthScheme() param: <i>schemeParam</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	CustomSecret("") CreateAuthScheme() param: <i>secret</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?	IsTemplate(<i>templateFlag</i>) CreateAuthScheme() param: <i>isTemplate</i> 方式がテンプレートではないことを示すには 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?	IsUsedByAdmin(1) CreateAuthScheme() param: <i>isUsedByAdmin</i> 1 に設定 -- 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?	SaveCredentials(0) CreateAuthScheme() param: <i>saveCreds</i> ユーザ認証情報が保存されないことを示すには 0 に設定します。
RADIUS ですか?	IsRadius(1) CreateAuthScheme() param: <i>isRadius</i> 1 に設定 -- 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?	IgnorePwd(1) CreateAuthScheme() param: <i>ignorePwd</i> 1 に設定 -- 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に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。

情報タイプ	値の割り当てと意味
方式タイプ	Type(<i>templateObject</i>) CreateAuthScheme() param: <i>schemeTemplate</i> SAML アーチファクト方式タイプ。

情報タイプ	値の割り当てと意味
説明	Description(<i>schemeDesc</i>) CreateAuthScheme() param: <i>schemeDesc</i> 認証方式の説明。
保護レベル	ProtectionLevel(<i>nLevel</i>) CreateAuthScheme() param: <i>protLevel</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	CustomLib("smauthsaml") CreateAuthScheme() param: <i>schemeLib</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	CustomParam(<i>param</i>) CreateAuthScheme() param: <i>schemeParam</i> 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 SAML アサーションプロデューサのサイト内のプロパティ ファイル <code>AMAssertionGenerator.properties</code> にあります。 ■ AssertionRetrievalURL。SAML アサーションプロデューサのサイトからアサーションを取得するための URL。 ■ Audience。アサーションプロデューサ サイトとアフィリエイト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。クエリは、ユーザを探す検索指定情報で代用されるデータを取得します。たとえば、 <pre>//saml:AttributeValue/SM:/SMContent/SM:Smlogin/SM:Username.text()</pre> このクエリは、Username エレメント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。名前と値のペアはセミコロン（;）で区切ります。この形式の例には LDAP および ODBC 属性が含まれます。 <pre>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</pre>
共有秘密キー	CustomSecret(<i>secret</i>) CreateAuthScheme() param: <i>secret</i> アフィリエイトサイトのパスワード。パスワードは、SAML アサーションが生成されるサイトでアフィリエイト用に入力されるパスワードに一致する必要があります。
テンプレートですか?	IsTemplate(0) CreateAuthScheme() param: <i>isTemplate</i> 方式がテンプレートではないことを示すには 0 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?	IsUsedByAdmin(0) CreateAuthScheme() param: <i>isUsedByAdmin</i> 0 に設定 -- 方式は管理者を認証するためには使用できません。
認証情報を保存しますか?	SaveCredentials(0) CreateAuthScheme() param: <i>saveCreds</i> ユーザ認証情報が保存されないことを示すには 0 に設定します。
RADIUS ですか?	IsRadius(0) CreateAuthScheme() param: <i>isRadius</i> 0 に設定 -- 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?	IgnorePwd(1) CreateAuthScheme() param: <i>ignorePwd</i> 1 に設定 -- 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。

情報タイプ	値の割り当てと意味
方式タイプ	Type(<i>templateObject</i>) CreateAuthScheme() param: <i>schemeTemplate</i> SAML POST 方式タイプ。
説明	Description(<i>schemeDesc</i>) CreateAuthScheme() param: <i>schemeDesc</i> 認証方式の説明。
保護レベル	ProtectionLevel(<i>nLevel</i>) CreateAuthScheme() param: <i>protLevel</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。

情報タイプ	値の割り当てと意味
ライブラリ	<code>CustomLib("smauthsaml")</code> <code>CreateAuthScheme()</code> param: <i>schemeLib</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	<code>CustomParam(param)</code> <code>CreateAuthScheme()</code> param: <i>schemeParam</i> 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。 アサーションプロデューサ サイトとアフィリエイト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。クエリは、ユーザを探す検索指定情報で代用されるデータを取得します。たとえば、 <pre>//saml:AttributeValue/SM:/SMContent /SM:Smlogin/SM:Username.text()</pre> 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。名前と値のペアはセミコロン (;) で区切ります。この形式の例には LDAP および ODBC 属性が含まれます。 <pre>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</pre>
共有秘密キー	CustomSecret("") CreateAuthScheme() param: <i>secret</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?	IsTemplate(0) CreateAuthScheme() param: <i>isTemplate</i> 方式がテンプレートではないことを示すには 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?	IsUsedByAdmin(0) CreateAuthScheme() param: <i>isUsedByAdmin</i> 0 に設定 -- 方式は管理者を認証するためには使用できません。
認証情報を保存しますか?	SaveCredentials(0) CreateAuthScheme() param: <i>saveCreds</i> ユーザ認証情報が保存されないことを示すには 0 に設定します。

情報タイプ	値の割り当てと意味
RADIUS ですか?	<code>IsRadius(0)</code> <code>CreateAuthScheme()</code> param: <i>isRadius</i> 0 に設定 -- 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>IgnorePwd(1)</code> <code>CreateAuthScheme()</code> param: <i>ignorePwd</i> 1 に設定 -- 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 方式タイプに基づく認証方式。SAML 2.0 サービスプロバイダによって設定されます。

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。

SAML 2.0 テンプレートに基づく認証方式は、以下の 2 つの点で他のタイプの認証方式と異なります。

- SAML 2.0 認証方式は、メソッド `CreateSAMLAuthScheme()` で作成されます。このメソッドは、`CreateAuthScheme()` の場合とまったく同じように `PolicyMgtAuthScheme` オブジェクトを作成します。
- SAML 2.0 認証方式には、以下の 2 つのセットのプロパティがあります。

- 以下のテーブルにリストされているプロパティ。これらのプロパティは `PolicyMgtAuthScheme` オブジェクトに保存されます。

通常、既存の SAML 2.0 認証方式で変更するように選択できるこのセット内のプロパティは、名前、説明および保護レベルのみです。`PolicyMgtAuthScheme` オブジェクト内の適切なメソッドでこれらのプロパティを変更します。

- 関連付けられたアイデンティティプロバイダのメタデータプロパティ。関連付けられたアイデンティティプロバイダは、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です。

これらのプロパティは `PolicyMgtAuthScheme` オブジェクトと共にハッシュテーブルとして保存されます。

SAML 2.0 認証方式に割り当てることができるメタデータ プロパティの詳細については、このガイドの「SAML 2.0 プロパティ リファレンス」を参照してください。

必要に応じて、メソッド `CreateSAMLAuthScheme()` は `CreateAuthScheme()` の代わりに参照されます。

情報タイプ	値の割り当てと意味
メタデータ プロパティ	<code>CreateSAMLAuthScheme()</code> param: <code>propsHash_ref</code> 認証方式オブジェクトと関連付けられた SAML 2.0 メタデータ プロパティのハッシュテーブル。 既存の SAML 2.0 認証方式と関連付けられたメタデータ プロパティを変更するために <code>SAMLAuthSchemeProperties()</code> をコールします。
方式タイプ	<code>Type(templateObject)</code> SAML 2.0 方式タイプ。
説明	<code>Description(schemeDesc)</code> <code>CreateSAMLAuthScheme()</code> param: <code>schemeDesc</code> 認証方式の説明。
保護レベル	<code>ProtectionLevel(nLevel)</code> <code>CreateSAMLAuthScheme()</code> param: <code>protLevel</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>CustomLib("smauthsaml")</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>CustomParam(param)</code> 空の文字列として設定します。
共有秘密キー	<code>CustomSecret("")</code> 空の文字列として設定します。この方式には適用できません。
テンプレートですか?	<code>IsTemplate(0)</code> 方式がテンプレートではないことを示すには 0 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?	IsUsedByAdmin(0) 0 に設定 -- 方式は管理者を認証するためには使用できません。
認証情報を保存しますか?	SaveCredentials(0) ユーザ認証情報が保存されないことを示すには 0 に設定します。
RADIUS ですか?	IsRadius(0) 0 に設定 -- 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?	IgnorePwd(1) 1 に設定 -- パスワードチェックを無視します。

SecurID HTML フォーム テンプレート

SecureID 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。

情報タイプ	値の割り当てと意味
方式タイプ	Type(<i>templateObject</i>) CreateAuthScheme() param: <i>schemeTemplate</i> SecureID HTML フォーム方式タイプ。
説明	Description(<i>schemeDesc</i>) CreateAuthScheme() param: <i>schemeDesc</i> 認証方式の説明。
保護レベル	ProtectionLevel(<i>nLevel</i>) CreateAuthScheme() param: <i>protLevel</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	CustomLib("smauthacehtml") CreateAuthScheme() param: <i>schemeLib</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	CustomParam(<i>param</i>) CreateAuthScheme() param: <i>schemeParam</i> 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 SecureID 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 <i>attr</i>;https://server/target 「https」内の「s」はオプションです。SSL 接続が必要かどうかによって異なります。 以下の例では、フォーム サポートで SecureID 認証を処理するためのデフォルトを使用します。 ace_id;https://my.server.com/siteminderagent/pwcgi/smpwservicescgi.exe
共有秘密キー	CustomSecret("") CreateAuthScheme() param: <i>secret</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?	IsTemplate(<i>templateFlag</i>) CreateAuthScheme() param: <i>isTemplate</i> 方式がテンプレートではないことを示すには 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?	IsUsedByAdmin(0) CreateAuthScheme() param: <i>isUsedByAdmin</i> 0 に設定 -- 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?	SaveCredentials(0) CreateAuthScheme() param: <i>saveCreds</i> ユーザ認証情報が保存されないことを示すには 0 に設定します。
RADIUS ですか?	IsRadius(0) CreateAuthScheme() param: <i>isRadius</i> 0 に設定 -- 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?	IgnorePwd(1) CreateAuthScheme() param: <i>ignorePwd</i> 1 に設定 -- パスワードチェックを無視します。

SecurID テンプレート

SecureID 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。

情報タイプ	値の割り当てと意味
方式タイプ	Type(<i>templateObject</i>) CreateAuthScheme() param: <i>schemeTemplate</i> SecureID 方式タイプ。
説明	Description(<i>schemeDesc</i>) CreateAuthScheme() param: <i>schemeDesc</i> 認証方式の説明。
保護レベル	ProtectionLevel(<i>nLevel</i>) CreateAuthScheme() param: <i>protLevel</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	CustomLib("smauthace") CreateAuthScheme() param: <i>schemeLib</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	CustomParam(<i>param</i>) CreateAuthScheme() param: <i>schemeParam</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。
共有秘密キー	CustomSecret("") CreateAuthScheme() param: <i>secret</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?	IsTemplate(<i>templateFlag</i>) CreateAuthScheme() param: <i>isTemplate</i> 方式がテンプレートではないことを示すには 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?	IsUsedByAdmin(1) CreateAuthScheme() param: <i>isUsedByAdmin</i> 1 に設定 -- 方式は管理者を認証するために使用できます。

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?	SaveCredentials(0) CreateAuthScheme() param: <i>saveCreds</i> ユーザ認証情報が保存されないことを示すには 0 に設定します。
RADIUS ですか?	IsRadius(1) CreateAuthScheme() param: <i>isRadius</i> 1 に設定 -- 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?	IgnorePwd(1) CreateAuthScheme() param: <i>ignorePwd</i> 1 に設定 -- 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 smaauthetsso 認証方式を設定するときにこのテーブルを使用します。SiteMinder SDK で利用可能な C 言語認証 API を使用して、カスタム方式を作成します。

情報タイプ	値の割り当てと意味
方式タイプ	Type(<i>templateObject</i>) CreateAuthScheme() param: <i>schemeTemplate</i> カスタム方式タイプです。
説明	Description(<i>schemeDesc</i>) CreateAuthScheme() param: <i>schemeDesc</i> 認証方式の説明。

情報タイプ	値の割り当てと意味
保護レベル	ProtectionLevel(<i>nLevel</i>) CreateAuthScheme() param: <i>protLevel</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	CustomLib("smauthetsso") CreateAuthScheme() param: <i>schemeLib</i> この認証方式に使用するライブラリの名前。

情報タイプ	値の割り当てと意味
パラメータ	CustomParam(<i>param</i>) CreateAuthScheme() param: <i>schemeParam</i> 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。 <code><Mode>; <Target>; <Admin>; <eTPS_Host></code> 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。 <code><Mode></code> 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。 ■ <code>cookie</code> -- eTrust SSO Cookie のみが受け入れ可能です ■ <code>cookieorbasic</code> -- 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。 ■ <code>cookieorforms</code> -- 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。 <code><Target></code> は <code>cookieorforms</code> 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 <code>Target</code> フィールドと同一です。 <code><Admin></code> は、ポリシー サーバの管理者用ログイン 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。 <code><eTPO_Host></code> 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。 SiteMinder は、eTrust SSO Cookie の検証を要求できるように、 <code><eTPS_Host></code> 上のポリシー サーバに対し <code><Admin></code> としてそれ自体を認証します。 例： <pre>"cookie; SMPS_sso; myserver.myco.com" "cookieorforms; /siteminderagent/forms/login.fcc; SMPS_sso; myserver.myco.com"</pre>
共有秘密キー	CustomSecret(<i>secret</i>) CreateAuthScheme() param: <i>secret</i> [パラメータ] フィールドで指定されたポリシー サーバ管理者のパスワード。

情報タイプ	値の割り当てと意味
テンプレートですか?	IsTemplate(<i>templateFlag</i>) CreateAuthScheme() param: <i>isTemplate</i> 方式がテンプレートではないことを示すには 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?	IsUsedByAdmin(<i>flag</i>) CreateAuthScheme() param: <i>isUsedByAdmin</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 true (1)、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 false (0) を指定します。デフォルトは 0 です。
認証情報を保存しますか?	SaveCredentials(0) CreateAuthScheme() param: <i>saveCreds</i> ユーザ認証情報が保存されないことを示すには 0 に設定します。
RADIUS ですか?	IsRadius(0) CreateAuthScheme() param: <i>isRadius</i> 0 に設定 -- 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?	IgnorePwd(<i>flag</i>) CreateAuthScheme() param: <i>ignorePwd</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。

TeleID テンプレート

TeleID 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。

情報タイプ	値の割り当てと意味
方式タイプ	Type(<i>templateObject</i>) CreateAuthScheme() param: <i>schemeTemplate</i> TeleID 方式タイプ。
説明	Description(<i>schemeDesc</i>) CreateAuthScheme() param: <i>schemeDesc</i> 認証方式の説明。

情報タイプ	値の割り当てと意味
保護レベル	ProtectionLevel(<i>nLevel</i>) CreateAuthScheme() param: <i>protLevel</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	CustomLib("smauthencotone") CreateAuthScheme() param: <i>schemeLib</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	CustomParam("") CreateAuthScheme() param: <i>schemeParam</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	CustomSecret(<i>seed</i>) CreateAuthScheme() param: <i>secret</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?	IsTemplate(<i>templateFlag</i>) CreateAuthScheme() param: <i>isTemplate</i> 方式がテンプレートではないことを示すには 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?	IsUsedByAdmin(1) CreateAuthScheme() param: <i>isUsedByAdmin</i> 1 に設定 -- 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?	SaveCredentials(0) CreateAuthScheme() param: <i>saveCreds</i> ユーザ認証情報が保存されないことを示すには 0 に設定します。
RADIUS ですか?	IsRadius(1) CreateAuthScheme() param: <i>isRadius</i> 1 に設定 -- 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?	IgnorePwd(1) CreateAuthScheme() param: <i>ignorePwd</i> 1 に設定 -- 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。

情報タイプ	値の割り当てと意味
方式タイプ	Type(<i>templateObject</i>) CreateAuthScheme() param: <i>schemeTemplate</i> Windows 認証（NTLM）方式タイプ。
説明	Description(<i>schemeDesc</i>) CreateAuthScheme() param: <i>schemeDesc</i> 認証方式の説明。
保護レベル	ProtectionLevel(<i>nLevel</i>) CreateAuthScheme() param: <i>protLevel</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	CustomLib("smauthntlm") CreateAuthScheme() param: <i>schemeLib</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	<code>CustomParam(<i>param</i>)</code> <code>CreateAuthScheme()</code> param: <i>schemeParam</i> <i>param</i> の値は、以下のようにこの方式で実行する認証のスタイルを決定します。 NTLM 認証（混合モードで実行される WinNT または Active Directory 用） 形式 <i>iis-web-server-url/path-to-ntc-file</i> 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。 例： <code>http://myiiswebserver.mycompany.com/siteminderagent/ntlm/creds.ntc</code> 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。 Windows 認証（ネイティブモードで実行される Active Directory 用） この認証スタイルの <i>param</i> では、リダイレクト URL の先頭に LDAP フィルタが追加されます。フィルタと URL はセミコロン (;) によって区切ります。例： <code>cn=%{UID},ou=Users,ou=USA,dc=%{DOMAIN},dc=mycompany,dc=com;http://myiiswebserver.mycompany.com/siteminderagent/ntlm/creds.ntc</code> 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。
共有秘密キー	<code>CustomSecret("")</code> <code>CreateAuthScheme()</code> param: <i>secret</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?	<code>IsTemplate(<i>templateFlag</i>)</code> <code>CreateAuthScheme()</code> param: <i>isTemplate</i> 方式がテンプレートではないことを示すには 0 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?	<code>IsUsedByAdmin(0)</code> <code>CreateAuthScheme()</code> param: <i>isUsedByAdmin</i> 0 に設定 -- 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>SaveCredentials(0)</code> <code>CreateAuthScheme()</code> param: <i>saveCreds</i> ユーザ認証情報が保存されないことを示すには 0 に設定します。
RADIUS ですか?	<code>IsRadius(0)</code> <code>CreateAuthScheme()</code> param: <i>isRadius</i> 0 に設定 -- 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>IgnorePwd(flag)</code> <code>CreateAuthScheme()</code> param: <i>ignorePwd</i> WinNT、および混在モードで実行される Active Directory の場合は、このプロパティを true (1) にする必要があります -- パスワード確認を無視します。 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

WS-Federation 方式タイプに基づく認証方式。WS-Federation リソース パートナーによって設定されます。

リソース パートナーでは、SAML 1.1 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。

WS-Federation テンプレートに基づく認証方式は、以下の 2 つの点で他のタイプの認証方式と異なります。

- WS-Federation 認証方式は、メソッド `CreateWSFEDAuthScheme()` で作成されます。このメソッドは、`CreateAuthScheme()` の場合とまったく同じように `PolicyMgtAuthScheme` オブジェクトを作成します。
- WS-Federation 認証方式には、以下の 2 つのセットのプロパティがあります。
 - 以下のテーブルにリストされているプロパティ。これらのプロパティは `PolicyMgtAuthScheme` オブジェクトに保存されます。

通常、既存の WS-Federation 認証方式で変更するように選択できるこのセット内のプロパティは、名前、説明および保護レベルのみです。`PolicyMgtAuthScheme` オブジェクト内の適切なメソッドでこれらのプロパティを変更します。
 - 関連付けられたアカウント パートナーのメタデータ プロパティ。関連付けられたアカウント パートナーは、リソース パートナーにアサーションを提供するパートナーです。

これらのプロパティは `PolicyMgtAuthScheme` オブジェクトと共にハッシュテーブルとして保存されます。

注: WS-Federation 認証方式に割り当てることができるメタデータ プロパティの詳細については、オンラインの「ポリシー管理 API リファレンス」の「WS-Federation プロパティ リファレンス」を参照してください。

この認証方式は CA SiteMinder® Federation レガシー フェデレーションを必要とします。レガシー フェデレーションには別途ライセンスが必要です。

情報タイプ	値の割り当てと意味

情報タイプ	値の割り当てと意味
メタデータ プロパティ	CreateWSFEDAuthScheme() param: propsHash_ref 認証方式オブジェクトと関連付けられた WS-Federation メタデータ プロパティのハッシュテーブル。 既存の WS-Federation 認証方式と関連付けられたメタデータ プロパティを変更するために WSFEDAuthSchemeProperties() をコールします。
方式タイプ	Type(templateObject) WSFED 方式タイプ。
説明	Description(schemeDesc) CreateWSFEDAuthScheme() param: schemeDesc 認証方式の説明。
保護レベル	ProtectionLevel(nLevel) CreateWSFEDAuthScheme() param: protLevel 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	CustomLib("smauthsaml") この方式タイプ用のデフォルト ライブラリ。
パラメータ	CustomParam(param) 空の文字列として設定します。
共有秘密キー	CustomSecret("") 空の文字列として設定します。この方式には適用できません。
テンプレートですか?	IsTemplate(0) 方式がテンプレートではないことを示すには 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?	IsUsedByAdmin(0) 0 に設定 -- 方式は管理者を認証するためには使用できません。
認証情報を保存しますか?	SaveCredentials(0) ユーザ認証情報が保存されないことを示すには 0 に設定します。
RADIUS ですか?	IsRadius(0) 0 に設定 -- 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?	IgnorePwd(1) 1 に設定 -- パスワードチェックを無視します。

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。

情報タイプ	値の割り当てと意味
方式タイプ	Type(<i>templateObject</i>) CreateAuthScheme() param: <i>schemeTemplate</i> X.509 クライアント証明書および基本方式タイプ。
説明	Description(<i>schemeDesc</i>) CreateAuthScheme() param: <i>schemeDesc</i> 認証方式の説明。
保護レベル	ProtectionLevel(<i>nLevel</i>) CreateAuthScheme() param: <i>protLevel</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	CustomLib("smauthcert") CreateAuthScheme() param: <i>schemeLib</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	CustomParam(<i>param</i>) CreateAuthScheme() param: <i>schemeParam</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、SSL 接続を介してユーザの X.509 証明書をリダイレクトします。形式 https://server:port/SCC?cert+basic 以下の例ではデフォルトの SCC を使用します。 https://my.server.com:80/siteminderagent/ cert/smgetcred.scc?cert+basic

情報タイプ	値の割り当てと意味
共有秘密キー	CustomSecret("") CreateAuthScheme() param: <i>secret</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?	IsTemplate(<i>templateFlag</i>) CreateAuthScheme() param: <i>isTemplate</i> 方式がテンプレートではないことを示すには 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?	IsUsedByAdmin(0) CreateAuthScheme() param: <i>isUsedByAdmin</i> 0 に設定 -- 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?	SaveCredentials(0) CreateAuthScheme() param: <i>saveCreds</i> ユーザ認証情報が保存されないことを示すには 0 に設定します。
RADIUS ですか?	IsRadius(0) CreateAuthScheme() param: <i>isRadius</i> 0 に設定 -- 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?	IgnorePwd(<i>flag</i>) CreateAuthScheme() param: <i>ignorePwd</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はこのテーブルを使用します。

情報タイプ	値の割り当てと意味
方式タイプ	Type(<i>templateObject</i>) CreateAuthScheme() param: <i>schemeTemplate</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	Description(<i>schemeDesc</i>) CreateAuthScheme() param: <i>schemeDesc</i> 認証方式の説明。

情報タイプ	値の割り当てと意味
保護レベル	ProtectionLevel(<i>nLevel</i>) CreateAuthScheme() param: <i>protLevel</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	CustomLib("smauthcert") CreateAuthScheme() param: <i>schemeLib</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	CustomParam(<i>param</i>) CreateAuthScheme() param: <i>schemeParam</i> 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、SSL 接続を介してユーザの X.509 証明書をリダイレクトします。形式 https://server:port/FCC?cert+forms 以下の例ではデフォルトの FCC を使用します。 https://my.server.com:80/siteminderagent/certoptional/forms/login.fcc?cert+forms
共有秘密キー	CustomSecret("") CreateAuthScheme() param: <i>secret</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?	IsTemplate(<i>templateFlag</i>) CreateAuthScheme() param: <i>isTemplate</i> 方式がテンプレートではないことを示すには 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?	IsUsedByAdmin(0) CreateAuthScheme() param: <i>isUsedByAdmin</i> 0 に設定 -- 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?	SaveCredentials(0) CreateAuthScheme() param: <i>saveCreds</i> ユーザ認証情報が保存されないことを示すには 0 に設定します。
RADIUS ですか?	IsRadius(0) CreateAuthScheme() param: <i>isRadius</i> 0 に設定 -- 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>IgnorePwd(flag)</code> <code>CreateAuthScheme()</code> param: <i>ignorePwd</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。

情報タイプ	値の割り当てと意味
方式タイプ	<code>Type(templateObject)</code> <code>CreateAuthScheme()</code> param: <i>schemeTemplate</i> X.509 クライアント証明書または基本方式タイプ。
説明	<code>Description(schemeDesc)</code> <code>CreateAuthScheme()</code> param: <i>schemeDesc</i> 認証方式の説明。
保護レベル	<code>ProtectionLevel(nLevel)</code> <code>CreateAuthScheme()</code> param: <i>protLevel</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	<code>CustomLib("smauthcert")</code> <code>CreateAuthScheme()</code> param: <i>schemeLib</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	CustomParam(<i>param</i>) CreateAuthScheme() param: <i>schemeParam</i> 以下の情報が含まれる文字列。 ■ SSL 接続を確立するためのサーバ。このサーバは、SSL 接続を介してユーザの X.509 証明書をリダイレクトします。 ■ 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。 ■ 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。 ■ 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] 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
共有秘密キー	CustomSecret("") CreateAuthScheme() param: <i>secret</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?	IsTemplate(<i>templateFlag</i>) CreateAuthScheme() param: <i>isTemplate</i> 方式がテンプレートではないことを示すには 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?	IsUsedByAdmin(0) CreateAuthScheme() param: <i>isUsedByAdmin</i> 0 に設定 -- 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?	SaveCredentials(0) CreateAuthScheme() param: <i>saveCreds</i> ユーザ認証情報が保存されないことを示すには 0 に設定します。

情報タイプ	値の割り当てと意味
RADIUS ですか?	<code>IsRadius(0)</code> <code>CreateAuthScheme()</code> param: <i>isRadius</i> 0 に設定 -- 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>IgnorePwd(flag)</code> <code>CreateAuthScheme()</code> param: <i>ignorePwd</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にこのテーブルを使用します。

情報タイプ	値の割り当てと意味
方式タイプ	<code>Type(templateObject)</code> <code>CreateAuthScheme()</code> param: <i>schemeTemplate</i> 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>Description(schemeDesc)</code> <code>CreateAuthScheme()</code> param: <i>schemeDesc</i> 認証方式の説明。
保護レベル	<code>ProtectionLevel(nLevel)</code> <code>CreateAuthScheme()</code> param: <i>protLevel</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	<code>CustomLib("smauthcertorform")</code> <code>CreateAuthScheme()</code> param: <i>schemeLib</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	CustomParam(<i>param</i>) CreateAuthScheme() param: <i>schemeParam</i> 以下の情報が含まれる文字列。 ■ SSL 接続を確立するためのサーバ。このサーバは、SSL 接続を介してユーザの X.509 証明書をリダイレクトします。 ■ 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。 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。 ■ 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。 ■ フォーム認証情報コレクタ (FCC) の名前およびパスです。 https://SSLserver:port/SFCC?certorform; [https://BasicServer/FCC] 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/forms/login.sfcc?certorform; https://my.BasicServer.com/ siteminderagent/forms/login.fcc
共有秘密キー	CustomSecret("") CreateAuthScheme() param: <i>secret</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?	IsTemplate(<i>templateFlag</i>) CreateAuthScheme() param: <i>isTemplate</i> 方式がテンプレートではないことを示すには 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?	IsUsedByAdmin(0) CreateAuthScheme() param: <i>isUsedByAdmin</i> 0 に設定 -- 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?	SaveCredentials(0) CreateAuthScheme() param: <i>saveCreds</i> ユーザ認証情報が保存されないことを示すには 0 に設定します。

情報タイプ	値の割り当てと意味
RADIUS ですか?	<code>IsRadius(0)</code> <code>CreateAuthScheme()</code> param: <i>isRadius</i> 0 に設定 -- 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>IgnorePwd(flag)</code> <code>CreateAuthScheme()</code> param: <i>ignorePwd</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。

情報タイプ	値の割り当てと意味
方式タイプ	<code>Type(templateObject)</code> <code>CreateAuthScheme()</code> param: <i>schemeTemplate</i> X.509 クライアント証明書方式タイプ。
説明	<code>Description(schemeDesc)</code> <code>CreateAuthScheme()</code> param: <i>schemeDesc</i> 認証方式の説明。
保護レベル	<code>ProtectionLevel(nLevel)</code> <code>CreateAuthScheme()</code> param: <i>protLevel</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	<code>CustomLib("smauthcert")</code> <code>CreateAuthScheme()</code> param: <i>schemeLib</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	CustomParam(<i>param</i>) CreateAuthScheme() param: <i>schemeParam</i> SSL 接続の確立に関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、SSL 接続を介してユーザの X.509 証明書を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	CustomSecret("") CreateAuthScheme() param: <i>secret</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?	IsTemplate(<i>templateFlag</i>) CreateAuthScheme() param: <i>isTemplate</i> 方式がテンプレートではないことを示すには 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?	IsUsedByAdmin(0) CreateAuthScheme() param: <i>isUsedByAdmin</i> 0 に設定 -- 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?	setAllowSaveCreds(0) ユーザ認証情報が保存されないことを示すには 0 に設定します。
RADIUS ですか?	IsRadius(0) CreateAuthScheme() param: <i>isRadius</i> 0 に設定 -- 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?	IgnorePwd(1) CreateAuthScheme() param: <i>ignorePwd</i> 1 に設定 -- パスワードチェックを無視します。

第 6 章: CLI ポリシー管理メソッド

第 7 章: CLI 管理者メソッド

このセクションには、以下のトピックが含まれています。

[AuthScheme メソッド -- 認証方式を設定または取得 \(P. 161\)](#)

[Description メソッド -- 管理者の説明を設定または取得 \(P. 162\)](#)

[ManageAllDomains メソッド -- ポリシー サーバ オブジェクトを管理する権限を追加または取り消し \(P. 163\)](#)

[ManageDomainObjects メソッド -- ドメイン オブジェクトを管理する権限を付与または取り消し \(P. 164\)](#)

[ManageKeysAndPwdPolicy メソッド -- キーおよびパスワード ポリシーを管理する権限を追加または取り消し \(P. 165\)](#)

[ManageUsers メソッド -- ユーザを管理する権限を付与または取り消し \(P. 166\)](#)

[Name メソッド -- 管理者の名前を設定または取得 \(P. 166\)](#)

[Password メソッド -- 管理者のパスワードを設定または取得 \(P. 167\)](#)

[UserDirectory メソッド -- 外部ユーザディレクトリを設定または取得 \(P. 167\)](#)

AuthScheme メソッド -- 認証方式を設定または取得

AuthScheme メソッド -- 外部ディレクトリに保存された管理者用の認証方式を設定または取得します。

構文

AuthScheme メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAdmin->AuthScheme( [authScheme] )
```

パラメータ

AuthScheme メソッドは以下のパラメータを受け入れます。

authScheme (PolicyMgtAuthScheme)

(オプション) 設定する認証方式を指定します。

戻り値

AuthScheme メソッドは以下のいずれかの値を返します。

- PolicyMgtAuthScheme オブジェクト
- 認証方式が存在しない場合、またはコールに失敗した場合は、**undef**

Description メソッド -- 管理者の説明を設定または取得

Description メソッドは、管理者の説明を設定または取得します。

構文

Description メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAdmin->Description( [adminDesc] )
```

パラメータ

Description メソッドは以下のパラメータを受け入れます。

adminDesc (文字列)

(オプション) 管理者の説明を指定します。

戻り値

Description メソッドは以下のいずれかの値を返します。

- 新規または既存の管理者ディレクトリの説明を指定します。
- 失敗の場合は、空の文字列

ManageAllDomains メソッド -- ポリシー サーバオブジェクトを管理する権限を追加または取り消し

ManageAllDomains メソッドは、すべてのシステム レベルとドメイン レベルのポリシー サーバオブジェクトを管理する権限を付与するか取り消します。

構文

ManageAllDomains メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAdmin->ManageAllDomains( [allDomFlag] )
```

パラメータ

ManageAllDomains メソッドは以下のパラメータを受け入れます。

allDomFlag (int)

(オプション) システムレベルとドメインレベルの権限を有効にするか (1 の値に設定)、無効にするか (0 の値に設定) を指定します。

戻り値

ManageAllDomains メソッドは以下のいずれかの値を返します。

- 管理者がすべてのドメインを管理できる場合は、1
- 管理者がすべてのドメインを管理できない場合は、0

注釈

権限には以下の要素が含まれます。

- 管理者、エージェント、ディレクトリ、ポリシー ドメイン、認証方式、登録方式、ODBC クエリ方式およびパスワード ポリシーなどのシステム レベル ポリシーストア オブジェクトの管理
- エージェントグループ、ディレクトリ マッピングおよび証明書マッピングの管理

注: これらのオブジェクトは、スクリプト作成インターフェースによって管理できません。

- ManageDomainObjects メソッドによって付与されるすべてのドメインレベル権限

ManageDomainObjects メソッド -- ドメイン オブジェクトを管理する権限を付与または取り消し

ManageDomainObjects メソッドは、ドメインレベルのポリシー サーバ オブジェクトを管理する権限を付与または取り消します。

構文

ManageDomainObjects メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAdmin->ManageDomainObjects( [domFlag] )
```

パラメータ

ManageDomainObjects メソッドは以下のパラメータを受け入れます。

domFlag (int)

(オプション) ドメイン オブジェクト管理権限を付与するか (1 の値に設定)、取り消すか (0 の値に設定) を指定します。

戻り値

ManageDomainObjects メソッドは以下のいずれかの値を返します。

- 管理者がドメイン オブジェクトを管理できる場合は、1
- 管理者がドメイン オブジェクトを管理できない場合は、0

注釈

権限には以下の要素が含まれます。

- ルール、レスポンス、ポリシーおよびレルムの管理
- ルールとレスポンスのグループの管理

注: これらのオブジェクトは、スクリプト作成インターフェースによって管理できません。

- リソース キャッシュからのレルムのフラッシング

ManageKeysAndPwdPolicy メソッド -- キーおよびパスワード ポリシーを管理する権限を追加または取り消し

ManageKeysAndPwdPolicy メソッドは、キーおよびパスワード ポリシーを管理する管理者権限を追加または取り消します。

構文

ManageKeysAndPwdPolicy メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAdmin->ManageKeysAndPwdPolicy( [pwdPolFlag] )
```

パラメータ

ManageKeysAndPwdPolicy メソッドは、以下のパラメータを受け入れます。

pwdPolFlag (int)

(オプション) 権限の付与または取り消しを指定します。このフラグを 1 に設定した場合は、管理者のタイプによって意味が異なります。

- システムレベル管理者はキーとパスワード ポリシーの両方を管理できます。
- ドメインレベル管理者はパスワード ポリシーのみを管理できます。

注: コマンドラインインターフェースでは、システム レベル管理者のみを作成できます。ドメインレベル管理者を作成するには、管理 UI を使用します。

このフラグを 0 に設定すると、これらの権限が取り消されます。

戻り値

ManageKeysAndPwdPolicy メソッドは以下のいずれかの値を返します。

- 権限が有効な場合は、1
- 権限が無効な場合は、0

ManageUsers メソッド -- ユーザを管理する権限を付与または取り消し

ManageUsers メソッドは、ユーザを管理する権限を付与または取り消します。

構文

ManageUsers メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAdmin->ManageUsers( [userFlag] )
```

パラメータ

ManageUsers メソッドは以下のパラメータを受け入れます。

userFlag (int)

(オプション) ユーザ管理権限を付与するか (値を 1 に設定)、取り消すか (値を 0 に設定) を指定します。

戻り値

ManageUsers メソッドは以下のいずれかの値を返します。

- 管理者がユーザを管理できる場合は、1
- 管理者がユーザを管理できない場合は、0

Name メソッド -- 管理者の名前を設定または取得

Name メソッドは、管理者の名前を設定または取得します。

構文

Name メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAdmin->Name( [adminName] )
```

パラメータ

Name メソッドは以下のパラメータを受け入れます。

adminName (文字列)

(オプション) 管理者の名前を指定します。

戻り値

Name メソッドは以下のいずれかの値を返します。

- 新規または既存の管理者名
- コールに失敗した場合は、**undef**

Password メソッド -- 管理者のパスワードを設定または取得

Password メソッドは、管理者のパスワードを設定または取得します。

構文

Password メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAdmin->Password([adminPwd])
```

パラメータ

Password メソッドは以下のパラメータを受け入れます。

adminPwd (文字列)

(オプション) 管理者パスワードを指定します。

戻り値

Password メソッドは以下のいずれかの値を返します。

- 新規または既存の管理者パスワード
- コールに失敗した場合は、**undef**

UserDirectory メソッド -- 外部ユーザ ディレクトリを設定または取得

UserDirectory メソッドは、管理者の外部ユーザ ディレクトリを設定または取得します。

構文

UserDirectory メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAdmin->UserDirectory([userDir])
```


パラメータ

UserDirectory メソッドは以下のパラメータを受け入れます。

userDir (PolicyMgtUserDir)

(オプション) 外部ユーザディレクトリを指定します。

戻り値

UserDirectory メソッドは以下のいずれかの値を返します。

- PolicyMgtUserDir オブジェクト
- ディレクトリが存在しない場合、またはコールに失敗した場合は、**undef**

CLI アフィリエイト属性メソッド

GetAttrType メソッド -- アフィリエイト属性タイプを取得

GetAttrType メソッドは、アフィリエイト属性のタイプを取得します。

構文

GetAttrType メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliateAttr->GetAttrType( )
```

パラメータ

GetAttrType メソッドは、パラメータを受け入れません。

戻り値

GetAttrType メソッドは以下のいずれかの値を返します。

- AFFILIATE_HTTP_HEADER_VARIABLE
- AFFILIATE_HTTP_COOKIE_VARIABLE

GetValue メソッド -- アフィリエイト属性の値を取得

GetValue メソッドは、アフィリエイト属性の値を取得します。

構文

GetValue メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliateAttr->GetValue( )
```

パラメータ

GetValue メソッドは、パラメータを受け入れません。

戻り値

GetValue メソッドは以下のいずれかの値を返します。

- アフィリエイト属性の値。
- コールに失敗した場合は、**undef**

CLI アフィリエイトドメイン メソッド

AddAdmin メソッド -- アフィリエイトドメインと管理者を関連付け

AddAdmin メソッドは、アフィリエイト ドメインと管理者を関連付けます。

構文

AddAdmin メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffDomain->AddAdmin(admin)
```

パラメータ

AddAdmin メソッドは以下のパラメータを受け入れます。

admin (PolicyMgtAdmin)

アフィリエイト ドメインと関連付ける管理者を指定します。

戻り値

AddAdmin メソッドは以下のいずれかの値を返します。

成功の場合は 0

失敗の場合は -1

AddUserDir メソッド -- アフィリエイトドメインとユーザ ディレクトリを関連付け

AddUserDir メソッドは、アフィリエイト ドメインとユーザ ディレクトリを関連付けます。

構文

メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffDomain->AddUserDir(userDir)
```

パラメータ

AddUserDir メソッドは以下のパラメータを受け入れます。

userDir (PolicyMgtUserDir)

アフィリエイト ドメインと関連付けるユーザ ディレクトリを指定します。

戻り値

AddUserDir メソッドは以下のいずれかの値を返します。

成功の場合は 0

失敗の場合は -1

CreateAffiliate メソッド -- アフィリエイト オブジェクトを作成します

CreateAffiliate メソッドは、アフィリエイト ドメイン内のアフィリエイト オブジェクトを作成し設定します。

構文

CreateAffiliate メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffDomain->CreateAffiliate( affName, password, authURL,  
validityDuration, skewTime [, affDesc] [, allowNotification] [, audience] [,  
enableFlag] [, shareSessioning] [, sessionSyncInterval] [, SAMLVersion] [,  
SAMLProfile] [,ConsumerURL] )
```

パラメータ

CreateAffiliate メソッドは、以下のパラメータを受け入れます。

affName (文字列)

アフィリエイト オブジェクトの名前を指定します。名前はすべてのアフィリエイト ドメインにわたって一意である必要があります。

password (文字列)

アフィリエイトで SiteMinder フェデレーション Web サービスにアクセスするために使用するパスワードを指定します。

authURL (文字列)

アフィリエイト ユーザの認証に使用する URL を指定します。

validityDuration (long)

SiteMinder によって生成された SAML アサーションが有効な秒数を指定します。アフィリエイトが指定された時間の後にアサーションを受信すると、アサーションが無効であると見なされます。

skewTime (long)

アサーションプロデューサ サイトのシステム クロック時間とアフィリエイト サイトのシステム クロック時間の間の秒単位の違いを指定します。スキュー時間は validityDuration に追加されます。時間は GMT が基準になります。

affDesc (文字列)

(オプション) アフィリエイトの説明を指定します。

allowNotification (int)

(オプション) イベント通知を許可するかどうかを指定します。アサーションプロデューサ サイト上でアフィリエイトから **SiteMinder** にイベント通知を送信できるようにするには、**1** に設定します。イベント通知サービスを無効にするには、**0** に設定します。デフォルトは **0** です (通知は無効化)。

audience (文字列)

(オプション) アサーションプロデューサとアフィリエイト間の許諾契約について説明するドキュメントの **URI**。この値はアフィリエイトに渡された **SAML** アサーションに含まれており、検証目的に使用できます。また、アフィリエイトは、関連情報を取得するためにオーディエンスドキュメントを解析できます。オーディエンス値は、**SAML** アフィリエイトエージェント用の **AffiliateConfig.xml** 設定ファイル内の **Assertion Audience** 設定に一致する必要があります。

enableFlag (int)

(オプション) アフィリエイトオブジェクトを有効にするかどうかを指定します。アフィリエイトオブジェクトを有効にするには **1** に設定し、無効にするには、**0** に設定します。デフォルトは **1** です (オブジェクトは有効化)。

shareSessioning (int)

(オプション) セッション情報を共有するかどうかを指定します。アサーションプロデューサおよびアフィリエイトでセッション情報を共有できるようにするには **1** に設定し、プロデューサおよびアフィリエイトで個別セッションを維持するには **0** に設定します。デフォルトは **0** です (セッションを共有せず)。共有セッションでは、一方のサイト上のセッションが終了すると、両方のサイト上のセッションが終了します。

sessionSyncInterval (long)

(オプション) アフィリエイトが共有セッションのステータスを検証するためにプロデューササイトにアクセスする頻度を秒単位で指定します。

SAMLVersion (long)

(オプション) SAML バージョンを指定します。以下の値のいずれかです。

- AFFILIATE_SAML_VER_1_0
- AFFILIATE_SAML_VER_1_1

SAML バージョンの指定は、ポリシー マネージャ API のセッションバージョンが少なくとも **v6.0 SP 1** である場合にのみ有効です。

SAMLProfile (long)

(オプション) SAML アサーションの送信および受信に使用されるプロファイルのタイプを指定します。有効なプロファイルは以下のとおりです。

AFFILIATE_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。

AFFILIATE_SAML_PROFILE_POST。生成された SAML アサーションは、ConsumerURL で指定された URL に対して POST 処理されます。

ポリシー管理 API のセッションバージョンが少なくとも **v6.0 SP 2** 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。以前のバージョンが含まれている場合、POST プロファイル要求は無視されます。また、アーチファクト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作成するように試行されます。

ConsumerURL (文字列)

(オプション) 要求するユーザのブラウザで、生成されたアサーションを POST 処理する必要がある 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。

戻り値

CreateAffiliate メソッドは以下のいずれかの値を返します。

成功した場合は、PolicyMgtAffiliate オブジェクト

失敗した場合は、**undef**

注釈

アフィリエイト オブジェクトは、フェデレートされたビジネス ネットワーク内のアフィリエイト サイトを表します。アフィリエイト オブジェクトおよびアフィリエイト ドメインは、SiteMinder レガシー フェデレーションを通じて利用可能です。

CreateSAMLServiceProvider メソッド -- SAML サービス プロバイダを作成

CreateSAMLServiceProvider メソッドは、SAML 2.0 サービス プロバイダ オブジェクトを作成します。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。SAML 2.0 サービス プロバイダ オブジェクト用に設定できるプロパティを以下にリストします。

既存のサービス プロバイダのプロパティを変更するには、PolicyMgtSAMLServiceProvider->Property メソッドをコールします。

構文

CreateSAMLServiceProvider メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffDomain->CreateSAMLServiceProvider(propsHash_ref)
```

パラメータ

CreateSAMLServiceProvider メソッドは以下のパラメータを受け入れます。

propsHash_ref (ハッシュ)

SAML 2.0 サービス プロバイダ用に定義されたメタデータ プロパティのハッシュテーブルの参照を指定します (たとえば、`myhash`)。

戻り値

CreateSAMLServiceProvider メソッドは以下のいずれかの値を返します。

- 成功したときの PolicyMgtSAMLServiceProvider オブジェクト
- 失敗の場合は `undef`

注釈

SAML 2.0 サービス プロバイダ オブジェクト用に以下のプロパティを定義できます。

一般的なプロパティ

- SAML_NAME
- SAML_DESCRIPTION
- SAML_SP_AUTHENTICATION_URL
- SAML_ENABLED
- SAML_SP_DOMAIN
- SAML_KEY_SPID
- SAML_SP_IDPID
- SAML_MAJOR_VERSION
- SAML_MINOR_VERSION
- SAML_SKEWTIME
- SAML_DISABLE_SIGNATURE_PROCESSING
- SAML_DSIG_VERINFO_ISSUER_DN
- SAML_DSIG_VERINFO_SERIAL_NUMBERSAML_KEY_SPID

名前 ID プロパティ

- SAML_SP_NAMEID_FORMAT
- SAML_SP_NAMEID_TYPE
- SAML_SP_NAMEID_STATIC
- SAML_SP_NAMEID_ATTRNAME
- SAML_SP_NAMEID_DNSPEC
- SAML_AFFILIATION

SSO プロパティ

- SAML_AUDIENCE
- SAML_SP_ASSERTION_CONSUMER_DEFAULT_URL
- SAML_ENABLE_SSO_ARTIFACT_BINDING
- SAML_SP_ARTIFACT_ENCODING
- SAML_SP_IDP_SOURCEID
- SAML_SP_PASSWORD
- SAML_ENABLE_SSO_POST_BINDING
- SAML_SSOECPPROFILE
- SAML_SP_REQUIRE_SIGNED_AUTHNREQUESTS
- SAML_SP_AUTHENTICATION_LEVEL
- SAML_SP_AUTHN_CONTEXT_CLASS_REF
- SAML_SP_VALIDITY_DURATION
- SAML_SP_STARTTIME
- SAML_SP_ENDTIME

SLO プロパティ

- SAML_SLO_REDIRECT_BINDING
- SAML_SLO_SERVICE_VALIDITY_DURATION
- SAML_SLO_SERVICE_URL
- SAML_SLO_SERVICE_RESPONSE_URL
- SAML_SLO_SERVICE_CONFIRM_URL

IPD プロパティ

- SAML_SP_ENABLE_IPD
- SAML_SP_IPD_SERVICE_URL
- SAML_SP_COMMON_DOMAIN
- SAML_SP_PERSISTENT_COOKIE

属性サービス プロパティ

- SAML_SP_ATTRSVC_ENABLE
- SAML_SP_ATTRSVC_VALIDITY_DURATION
- SAML_SP_ATTRSVC_SIGN_ASSERTION
- SAML_SP_ATTRSVC_LDAP_SEARCH_SPEC
- SAML_SP_ATTRSVC_ODBC_SEARCH_SPEC
- SAML_SP_ATTRSVC_WINNT_SEARCH_SPEC
- SAML_SP_ATTRSVC_CUSTOM_SEARCH_SPEC
- SAML_SP_ATTRSVC_AD_SEARCH_SPEC

暗号化プロパティ

- SAML_SP_ENCRYPT_ID
- SAML_SP_ENCRYPT_ASSERTION
- SAML_SP_ENCRYPT_BLOCK_ALGO
- SAML_SP_ENCRYPT_KEY_ALGO
- SAML_SP_ENCRYPT_CERT_ISSUER_DN
- SAML_SP_ENCRYPT_CERT_SERIAL_NUMBER

拡張プロパティ

- SAML_SP_PLUGIN_CLASS
- SAML_SP_PLUGIN_PARAMS

CreateWSFEDResourcePartner メソッド -- WS-Federation リソース パートナーを作成

CreateWSFEDResourcePartner メソッドは、アフィリエイト ドメイン用の WS-Federation リソース パートナーを作成します。

構文

CreateWSFEDResourcePartner メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffDomain->CreateWSFEDResourcePartner(propsHash_ref)
```

パラメータ

CreateWSFEDResourcePartner メソッドは、以下のパラメータを受け入れます。

propsHash_ref (ハッシュ)

次の定義を行う場合のメタデータ プロパティのハッシュテーブルの参照を指定します。

WS-Federation リソース パートナー、（たとえば、
C<¥%myhash>|"hashref"）

戻り値

CreateWSFEDResourcePartner メソッドは以下のいずれかの値を返します。

- 成功したときの PolicyMgtWSFEDResourcePartner オブジェクト
- 失敗の場合は **undef**

注釈

リソース パートナー用に以下のプロパティを定義できます。

一般的なプロパティ

- WSFED_NAME
- WSFED_DESCRIPTION
- WSFED_MAJOR_VERSION
- WSFED_MINOR_VERSION
- WSFED_SAML_MAJOR_VERSION
- WSFED_KEY_RPID
- WSFED_APID
- WSFED_SAML_MINOR_VERSION
- WSFED_RP_DOMAIN
- WSFED_ENABLED
- WSFED_RP_AUTHENTICATION_URL
- WSFED_KEY_RPID
- WSFED_APID

名前 ID プロパティ

- WSFED_RP_NAMEID_FORMAT
- WSFED_RP_NAMEID_TYPE
- WSFED_RP_NAMEID_STATIC
- WSFED_RP_NAMEID_ATTR_NAME
- WSFED_RP_NAMEID_DN_SPEC
- WSFED_RP_NAMEID_ALLOWED_NESTED

SSO プロパティ

- WSFED_RP_AUTHENTICATION_METHOD
- WSFED_RP_VALIDITY_DURATION
- WSFED_RP_ASSERTION_CONSUMER_DEFAULT_URL
- WSFED_RP_AUTHENTICATION_LEVEL
- WSFED_RP_STARTTIME
- WSFED_RP_ENDTIME

サインアウト プロパティ

- WSFED_RP_SLO_ENABLED
- WSFED_RP_SIGNOUT_CLEANUP_URL
- WSFED_RP_SIGNOUT_CONFIRM_URL

拡張プロパティ

- WSFED_RP_PLUGIN_CLASS
- WSFED_RP_PLUGIN_PARAMS

DeleteAffiliate メソッド -- ドメインからアフィリエイトを削除

DeleteAffiliate メソッドは、アフィリエイト ドメインから指定されたアフィリエイト オブジェクトを削除します。

構文

DeleteAffiliate メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffDomain->DeleteAffiliate(aff)
```

パラメータ

DeleteAffiliate メソッドは、以下のパラメータを受け入れます。

aff (PolicyMgtAffiliate)

削除するアフィリエイト オブジェクトを指定します。

戻り値

DeleteAffiliate メソッドは以下のいずれかの値を返します。

- 成功した場合またはアフィリエイト ドメインが見つからなかった場合は 0
- 失敗の場合は -1

DeleteSAMLServiceProvider メソッド -- SAML サービス プロバイダを削除

DeleteSAMLServiceProvider メソッドは、指定された SAML 2.0 サービス プロバイダを削除します。

構文

メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffDomain->DeleteSAMLServiceProvider(sp)
```

パラメータ

DeleteSAMLServiceProvider メソッドは、以下のパラメータを受け入れます。

sp (PolicyMgtSAMLServiceProvider)

削除するサービス プロバイダを指定します。

戻り値

DeleteSAMLServiceProvider メソッドは以下のいずれかの値を返します。

- 成功した場合またはサービス プロバイダが見つからなかった場合は 0
- 失敗の場合は -1

DeleteWSFEDResourcePartner メソッド -- リソース パートナーを削除

DeleteWSFEDResourcePartner メソッドは、リソース パートナーを削除します。

構文

DeleteWSFEDResourcePartner メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffDomain->DeleteWSFEDResourcePartner(rp)
```

パラメータ

DeleteWSFEDResourcePartner メソッドは以下のパラメータを受け入れます。

rp (PolicyMgtWSFEDResourcePartner オブジェクト)

削除するリソース パートナーを指定します。

戻り値

DeleteWSFEDResourcePartner メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことを指定します。
- 値 = -1
メソッドが失敗したことを指定します。

Description メソッド -- 説明を取得または設定

Description メソッドは、アフィリエイト ドメインの説明を設定または取得します。

構文

Description メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffDomain->Description([domainDesc])
```

パラメータ

Description メソッドは、以下のパラメータを受け入れます。

domainDesc (文字列)

(オプション) 設定する説明を指定します。

戻り値

Description メソッドは以下のいずれかの値を返します。

- 成功の場合は、アフィリエイト ドメインの新規または既存の説明
- 失敗の場合は **undef**

GetAffiliate メソッド -- アフィリエイト オブジェクトを取得

GetAffiliate メソッドは、指定されたアフィリエイト オブジェクトを取得します。

構文

GetAffiliate メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffDomain->GetAffiliate(affName)
```

パラメータ

GetAffiliate メソッドは、以下のパラメータを受け入れます。

affName (文字列)

取得するアフィリエイト オブジェクトの名前を指定します。

戻り値

GetAffiliate メソッドは以下のいずれかの値を返します。

- 成功したときの **PolicyMgtAffiliate** オブジェクト
- 指定されたアフィリエイト オブジェクトが存在しない場合またはコールに失敗する場合は、**undef**

GetAllAdmins メソッド -- 管理者をすべて取得

GetAllAdmins メソッドは、アフィリエイト ドメインと関連付けられたすべての管理者を取得します。

構文

GetAllAdmins メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffDomain->GetAllAdmins( )
```

パラメータ

GetAllAdmins メソッドは、パラメータを受け入れません。

戻り値

GetAllAdmins メソッドは以下の 1 個以上の値を返します。

- PolicyMgtAdmin オブジェクトの配列
- 管理者オブジェクトがアフィリエイト ドメインに関連付けられていない場合またはコールに失敗する場合は、**undef**

GetAllAffiliates メソッド -- ドメイン内のすべてのアフィリエイトを取得

GetAllAffiliates メソッドは、アフィリエイト ドメインと関連付けられたすべてのアフィリエイト オブジェクトを取得します。

構文

GetAllAffiliates メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffDomain->GetAllAffiliates( )
```

パラメータ

GetAllAffiliates メソッドは、パラメータを受け入れません。

戻り値

GetAllAffiliates メソッドは以下のいずれかの値を返します。

- 成功の場合は、PolicyMgtAffiliate オブジェクトの配列
- 失敗した場合は、**undef**

GetAllSAMLServiceProviders メソッド -- アフィリエイトドメインと関連付けられたすべてのサービス プロバイダを取得

GetAllSAMLServiceProviders メソッドは、アフィリエイト ドメインと関連付けられたすべての SAML 2.0 サービス プロバイダを取得します。

構文

GetAllSAMLServiceProviders メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffDomain->GetAllSAMLServiceProviders( )
```

パラメータ

GetAllSAMLServiceProviders メソッドは、パラメータを受け入れません。

戻り値

GetAllSAMLServiceProviders メソッドは以下のいずれかの値を返します。

- PolicyMgtSAMLServiceProvider オブジェクトの配列
- 失敗した場合は、**undef**

GetAllWSFEDResourcePartners メソッド -- すべての WSFED リソース パートナーを取得

GetAllWSFEDResourcePartners メソッドは、ドメインと関連付けられたすべてのリソース パートナーを取得します。

構文

GetAllWSFEDResourcePartners メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffDomain->GetAllWSFEDResourcePartners( )
```

パラメータ

GetAllWSFEDResourcePartners メソッドは、パラメータを受け入れません。

戻り値

GetAllWSFEDResourcePartners メソッドは以下のいずれかの値を返します。

- 成功の場合は、PolicyMgtWSFEDResourcePartner オブジェクトの配列
- 失敗の場合は **undef**

GetSAMLServiceProvider メソッド -- 指定されたサービス プロバイダを取得

GetSAMLServiceProvider メソッドは、指定された SAML 2.0 サービス プロバイダを取得します。

構文

GetSAMLServiceProvider メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffDomain->GetSAMLServiceProvider(spName)
```

パラメータ

GetSAMLServiceProvider メソッドは以下のパラメータを受け入れます。

spName (文字列)

取得するサービス プロバイダの名前を指定します。

戻り値

GetSAMLServiceProvider メソッドは以下のいずれかの値を返します。

- 成功したときの PolicyMgtSAMLServiceProvider オブジェクト
- 指定されたサービス プロバイダが存在しない場合またはコールに失敗する場合は、**undef**

GetSAMLServiceProviderById メソッド -- 指定されたサービス プロバイダを取得

GetSAMLServiceProviderById メソッドは、そのプロバイダ ID によって、指定された SAML 2.0 サービス プロバイダを取得します。

構文

GetSAMLServiceProviderById メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffDomain->GetSAMLServiceProviderById(spID)
```

パラメータ

GetSAMLServiceProviderById メソッドは以下のパラメータを受け入れます。

spID (文字列)

取得するサービス プロバイダのプロバイダ ID を指定します。

戻り値

GetSAMLServiceProviderById メソッドは以下のいずれかの値を返します。

- 成功したときの PolicyMgtSAMLServiceProvider オブジェクト
- 指定されたサービス プロバイダが存在しない場合またはコールに失敗する場合は、**undef**

GetUserDirSearchOrder メソッド -- ユーザ ディレクトリの検索順序を取得

GetUserDirSearchOrder メソッドは、アフィリエイト ドメインと関連付けられたユーザ ディレクトリ オブジェクトを取得します。返されるオブジェクトの順序は、ディレクトリのクエリ実行時に SiteMinder が使用する順序と同じです。

構文

GetUserDirSearchOrder メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffDomain->GetUserDirSearchOrder( )
```

パラメータ

GetUserDirSearchOrder メソッドは、パラメータを受け入れません。

戻り値

GetUserDirSearchOrder メソッドは以下のいずれかの値を返します。

- 成功の場合は、PolicyMgtUserDir オブジェクトの配列
- 失敗した場合は、**undef**

GetWSFEDResourcePartner メソッド -- リソース パートナーを取得

GetWSFEDResourcePartner メソッドは、ドメインと関連付けられた指定の WS-Federation リソース パートナーを取得します。

構文

GetWSFEDResourcePartner メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffDomain->GetWSFEDResourcePartner(rpName)
```

パラメータ

GetWSFEDResourcePartner メソッドは、以下のパラメータを受け入れます。

rpName (文字列)

取得するリソース パートナーの名前を指定します。

戻り値

GetWSFEDResourcePartner メソッドは、以下の値を返します。

- 成功したときの PolicyMgtWSFEDResourcePartner オブジェクト
- 指定されたリソース パートナーが存在しない場合またはコールに失敗する場合は、**undef**

GetWSFEDResourcePartnerById メソッド -- ID によってリソース パートナーを取得

GetWSFEDResourcePartnerById メソッドは、そのプロバイダ ID によって、指定されたリソース パートナーを取得します。

構文

GetWSFEDResourcePartnerById メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffDomain->GetWSFEDResourcePartnerById( rpID )
```

パラメータ

GetWSFEDResourcePartnerById メソッドは、以下のパラメータを受け入れます。

rpID (文字列)

取得するリソース パートナーの ID を指定します。

戻り値

GetWSFEDResourcePartnerById メソッドは、以下の値を返します。

- PolicyMgtWSFEDResourcePartner オブジェクト
- 指定されたリソース パートナーが存在しない場合またはコールに失敗する場合は、**undef**

Name メソッド -- アフィリエイトドメイン名を設定または取得

Name メソッドは、アフィリエイト ドメインの名前を設定または取得します。

構文

Name メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffDomain->Name( [domainName] )
```

パラメータ

Name メソッドは以下のパラメータを受け入れます。

domainName (文字列)

(オプション) 設定する名前を指定します。

戻り値

Name メソッドは以下のいずれかの値を返します。

- 新規または既存のアフィリエイト ドメイン名
- コールに失敗した場合は、**undef**

RemoveAdmin メソッド -- アフィリエイトドメインから管理者の関連付けを解除

RemoveAdmin メソッドは、アフィリエイト ドメインから指定された管理者の関連付けを解除します。

構文

RemoveAdmin メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffDomain->RemoveAdmin( admin )
```

パラメータ

RemoveAdmin メソッドは以下のパラメータを受け入れます。

admin (PolicyMgtAdmin)

アフィリエイト ドメインから削除する管理者を指定します。

戻り値

RemoveAdmin メソッドは以下のいずれかの値を返します。

- 成功の場合は 0
- 失敗の場合は -1

RemoveUserDir メソッド -- アフィリエイトドメインからユーザ ディレクトリの関連付けを解除

RemoveUserDir メソッドは、アフィリエイト 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。

構文

RemoveUserDir メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffDomain->RemoveUserDir( userDir )
```

パラメータ

RemoveUserDir メソッドは以下のパラメータを受け入れます。

userDir (PolicyMgtUserDir)

アフィリエイト ドメインから関連付けを解除するユーザ ディレクトリを指定します。

戻り値

RemoveUserDir メソッドは以下のいずれかの値を返します。

- 成功の場合は 0
- 失敗の場合は -1

SetUserDirSearchOrder メソッド -- ディレクトリ オブジェクトの検索順序を設定

SetUserDirSearchOrder メソッドは、アフィリエイト 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再調整します。

構文

SetUserDirSearchOrder メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffDomain->SetUserDirSearchOrder( dirArray )
```

パラメータ

SetUserDirSearchOrder メソッドは以下のパラメータを受け入れます。

dirArray (PolicyMgtUserDir)

ユーザ ディレクトリ オブジェクトの配列の参照を指定します (たとえば、`¥@myarray`)。

戻り値

SetUserDirSearchOrder メソッドは、以下の値を返します。

- 成功の場合は、(PolicyMgtUserDir) オブジェクトの配列
- 失敗した場合は、**undef**

CLI アフィリエイト オブジェクト メソッド

AddAttribute メソッド -- 新しいアフィリエイト属性を追加

AddAttribute メソッドは、アフィリエイト オブジェクトに新しいアフィリエイト属性を追加します。

構文

AddAttribute メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliate->AddAttribute(attrType, value)
```

パラメータ

AddAttribute メソッドは、以下のパラメータを受け入れます。

attrType (int)

以下のいずれかのアフィリエイト属性タイプを指定します。

- AFFILIATE_HTTP_HEADER_VARIABLE (値=1)。アフィリエイト属性は HTTP ヘッダ変数として利用可能になります。
- AFFILIATE_HTTP_COOKIE_VARIABLE (値=2)。アフィリエイト属性は HTTP Cookie 変数として利用可能になります。

value (int)

アフィリエイト属性の値を指定します。この値指定は、[SiteMinder アフィリエイト ダイアログ] の [名前と値のペア] 列に表示されます。値指定の形式は、以下のように追加するアフィリエイト属性のタイプによって異なります -- スタティック、ユーザ属性、または DN 属性。

スタティック。リテラル属性値。スタティック アフィリエイト属性は、アフィリエイト サイトでユーザに関する特定の情報をアプリケーションへ渡すのに役立ちます。たとえば、アフィリエイト サイトでのユーザの信用限度などです。

- 形式 : `VariableName=value`

`VariableName` は SAML アサーションでの属性を識別する名前です。また、値は属性値です。

- 例 : `climit=2000`

ユーザ属性。LDAP、WinNT または ODBC ユーザ ディレクトリ内のユーザのエントリに基づくユーザ プロファイル属性名。たとえば、ユーザの肩書きまたは電子メールアドレスの属性名などです。

- 形式 : `UserAttrVariableName=<%userattr="UserAttrName"%>`

`UserAttrVariableName` は SAML アサーションでの属性を識別する名前です。また、`UserAttrName` (引用符付き) は、ユーザ ディレクトリ内の属性の名前です。

`userattr=` は、形式に含める必要があるスタティック テキストです。名前/値ペアの `userattr=` 部分は、次のようにパーセント記号と角かっこで囲む必要があります。`<% .. %>`。

例 : `email_address=<%userattr="email"%>`

DN 属性。ユーザと関連付けられた LDAP または ODBC のディレクトリ オブジェクト内の属性の名前。ユーザが所属するグループ、ユーザ DN の一部をなす組織単位 (ou) などは、属性を DN 属性として参照できるディレクトリ オブジェクトの例です。たとえば、DN 属性では、ユーザのメンバーシップに基づいて、ユーザの部署を参照できます。

- 形式 : `DNVariableName=<#dn="DNSpec" attr="DNAttrName"#>`

`DNVariableName` は SAML アサーションでの属性を識別する名前です。`DNSpec` (引用符付き) は、ディレクトリ オブジェクトの DN です。また、`DNAttrName` (引用符付き) はディレクトリ オブジェクト属性の名前です。

`dn=` と `attr=` は、形式に含める必要があるスタティック テキスト文字列です。名前/値ペアの `dn=` および `attr=` 部分は、次のようにポンド記号と角かっこで囲む必要があります。`<#. ...#>`.

例 : `GroupName=<#dn="ou=home,o=security.com" attr="cn"#>`

`SiteMinder` で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、`DNSpec` の前に単一文字 (!) を付けます。たとえば、以下のように指定します。

`dn="!ou=home,o=security.com"`

戻り値

`AddAttribute` メソッドは、以下の値を返します。

- `PolicyMgtAffiliateAttr` オブジェクト
- 失敗した場合は、`undef`

注釈

アフィリエイト属性は、`SiteMinder` が `SAML` アサーションでアフィリエイトに提供する名前/値ペアです。属性には、ユーザの資格（ユーザのアフィリエイト サイトでの信用限度など）およびユーザのプロファイル（肩書きまたは電子メールアドレスなど）からの情報が含まれます。

アフィリエイト サイトでのアプリケーションは、アサーションからアフィリエイト属性を抽出すると、サイトでの他のアプリケーション用に `HTTP` ヘッダ変数または `HTTP Cookie` 変数として属性を利用可能にできます。

注: アフィリエイトに渡されるアサーションの合計サイズは、**4K** 以下でなければなりません。アフィリエイト オブジェクトに多くの属性が含まれる場合、この制限に違反する可能性があります。**3K** の最大のアサーション サイズが推奨されます。

AddUser メソッド -- アフィリエイト オブジェクトに新しいユーザを追加

AddUser メソッドは、アフィリエイト オブジェクトに新しいユーザを追加します。

構文

AddUser メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliate->AddUser( user )
```

パラメータ

AddUser メソッドは以下のパラメータを受け入れます。

user (PolicyMgtUser)

追加するユーザを指定します。

戻り値

AddUser メソッドは以下のいずれかの値を返します。

- 成功の場合は 0
- 失敗の場合は -1

AllowNotification メソッド -- イベント通知プロパティを設定または取得

AllowNotification メソッドは、イベント通知プロパティを設定または取得します。 イベント通知が有効な場合、アフィリエイトはアサーションプロデューサ サイト上の SiteMinder にユーザ アクティビティに関する通知を送信します。

構文

AllowNotification メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliate->AllowNotification( [notificationFlag] )
```

パラメータ

`AllowNotification` メソッドは以下のパラメータを受け入れます。

notificationFlag (int)

(オプション) イベント通知を有効にするかどうかを指定します。**1** はイベント通知を有効にすることを表し、**0** はイベント通知を無効にすることを表します。

戻り値

`AllowNotification` メソッドは以下のいずれかの値を返します。

- 新規または既存の通知フラグ設定
- 失敗した場合は、**undef**

AssertionPluginClass メソッド -- アサーション生成プログラム プラグインの名前を設定または取得

`AssertionPluginClass` メソッドは、アサーション生成プログラム プラグインの完全修飾クラス名を設定または取得します。

構文

`AssertionPluginClass` メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliate->AssertionPluginClass( [className] )
```

パラメータ

`AssertionPluginClass` メソッドは以下のパラメータを受け入れます。

className (文字列)

(オプション) たとえば、`com.samlproducer.assertionplugin.partner1` などのカスタム アサーション生成プログラム プラグインの完全修飾クラス名を指定します。

戻り値

AssertionPluginClass メソッドは以下のいずれかの値を返します。

- 新規または既存のクラス名
- プラグインが定義されていない場合またはコールに失敗する場合は、**undef**

注釈

プラグインは、SiteMinder によって生成されたデフォルト SAML アサーションの内容を変更できるようにするカスタム Java クラスです。SAML アサーションは、別途ライセンスされるレガシー フェデレーションで使用可能です。

アサーション生成プログラム プラグイン機能では、v6.0 SP 2 以上のポリシー管理 API セッションバージョンが必要です。アサーション生成プログラム プラグインにパラメータを渡すには、メソッド

PolicyMgtAffiliate->AssertionPluginParameters を使用できます。

アサーション生成プログラム プラグインを作成するには、Java SDK 内の AssertionGeneratorPlugin インターフェースを実装します。詳細については、「Java 用プログラミング ガイド」を参照してください。

AssertionPluginParameters メソッド -- パラメータ文字列を設定または取得

AssertionPluginParameters メソッドは、カスタム アサーション生成プログラム プラグインに渡すパラメータ文字列を設定または取得します。パラメータ文字列の構文はユーザ定義です。つまり、パラメータ文字列は、カスタム アサーション生成プログラムが必要とする規則に必ず従う必要があります。

構文

AssertionPluginParameters メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliate->AssertionPluginParameters( [parameter] )
```

パラメータ

AssertionPluginParameters メソッドは以下のパラメータを受け入れます。

parameters (文字列)

(オプション) プラグインに渡すパラメータ文字列を指定します。

戻り値

AssertionPluginParameters メソッドは以下のいずれかの値を返します。

- 新規または既存のパラメータ文字列
- コールに失敗した場合は、**undef**

Audience メソッド -- URI を設定または取得

Audience メソッドは、アサーションプロデューサ サイトとアフィリエイト間の許諾契約について説明するドキュメントの URI を設定または取得します。

この値はアフィリエイトに渡された SAML アサーションに含まれており、検証目的に使用できます。また、アフィリエイトは、関連情報を取得するためにオーディエンス ドキュメントを解析できます。オーディエンス値は、SAML アフィリエイトエージェント用の AffiliateConfig.xml 設定ファイル内の Assertion Audience 設定に一致する必要があります。

構文

Audience メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliate->Audience( [audience] )
```

パラメータ

Audience メソッドは以下のパラメータを受け入れます。

audience (文字列)

(オプション) 設定するオーディエンス URI を指定します。

戻り値

`AudienceObjects` メソッドは以下のいずれかの値を返します。

- 新規または既存の オーディエンス URI
- コールに失敗した場合は、**undef**

AuthURL メソッド -- URL を設定または取得

`AuthURL` メソッドは、アフィリエイト ユーザの認証に使用する URL を設定または取得します。

構文

`AuthURL` メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliate->AuthURL( [AuthURL] )
```

パラメータ

`AuthURL` メソッドは以下のパラメータを受け入れます。

AuthURL (文字列)

(オプション) 設定する認証 URL を指定します。

戻り値

`AuthURL` メソッドは以下のいずれかの値を返します。

- 新規または既存の URL
- コールに失敗した場合は、**undef**

ConsumerURL メソッド -- URL を設定または取得

`ConsumerURL` メソッドは、要求するユーザのブラウザで生成されたアサーションを POST 処理する必要がある URL を設定または取得します。

構文

`ConsumerURL` メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliate->ConsumerURL( [ConsumerURL] )
```

パラメータ

ConsumerURL メソッドは以下のパラメータを受け入れます。

ConsumerURL (文字列)

(オプション)生成されたアサーションを送信する URL を指定します。

戻り値

ConsumerURL メソッドは以下のいずれかの値を返します。

- 生成されたアサーションを送信する新規または既存の URL
- コールに失敗した場合は、**undef**

CreateIPHostConfigName メソッド -- 指定されたホスト名から IP 設定オブジェクトを作成

CreateIPHostConfigName メソッドは、指定されたホスト名から IP 設定オブジェクトを作成します。

構文

CreateIPHostConfigName メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliate->CreateIPConfigHostName( hostName )
```

パラメータ

CreateIPHostConfigName メソッドは以下のパラメータを受け入れます。

hostName (文字列)

IP 設定オブジェクトのベースにするホスト名を指定します。

戻り値

CreateIPHostConfigName メソッドは以下のいずれかの値を返します。

- PolicyMgtIPConfig オブジェクト
- コールに失敗した場合は、**undef**

注釈

指定されたホストからアフィリエイト サイトにアクセスするユーザのみがアフィリエイト サイトで受け入れられます。

CreateIPConfigRange メソッド -- IP 設定 オブジェクトを作成

CreateIPConfigRange メソッドは、指定された IP アドレスの範囲から IP 設定オブジェクトを作成します。

構文

CreateIPConfigRange メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliate->CreateIPConfigRange( ipAddr1, ipAddr2 )
```

パラメータ

CreateIPConfigRange メソッドは、以下のパラメータを受け入れます。

ipAddr1 (文字列)

アフィリエイト サイトにアクセスするための有効な IP アドレス範囲内の最初の IP アドレスを指定します。

ipAddr2 (int)

アフィリエイト サイトにアクセスするための有効な IP アドレス範囲内の最後の IP アドレスを指定します。

戻り値

CreateIPConfigRange メソッドは以下のいずれかの値を返します。

- PolicyMgtIPConfig オブジェクト
- コールに失敗した場合は、**undef**

注釈

指定された範囲内の IP アドレスからアフィリエイト サイトにアクセスするユーザのみがアフィリエイト サイトで受け入れられます。

CreateIPConfigSingleHost メソッド -- 指定された IP アドレスから IP 設定オブジェクトを作成

CreateIPConfigSingleHost メソッドは、指定された IP アドレスから IP 設定オブジェクトを作成します。

構文

CreateIPConfigSingleHost メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliate->CreateIPConfigSingleHost( ipAddr )
```

パラメータ

CreateIPConfigSingleHost メソッドは以下のパラメータを受け入れます。

ipAddr (文字列)

アフィリエイト サイトにアクセスする IP アドレスを指定します。

戻り値

CreateIPConfigSingleHost メソッドは以下のいずれかの値を返します。

- PolicyMgtIPConfig オブジェクト
- コールに失敗した場合は、**undef**

注釈

指定された IP アドレスからアフィリエイト サイトにアクセスするユーザーのみがアフィリエイト サイトで受け入れられます。

CreateIPConfigSubnetMask メソッド -- IP 設定 オブジェクトを作成

CreateIPConfigSubnetMask メソッドは、指定された IP アドレスおよびサブネット マスクから IP 設定オブジェクトを作成します。

構文

CreateIPConfigSubnetMask メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliate->CreateIPConfigSubnetMask( ipAddr, subnetMask )
```

パラメータ

CreateIPConfigSubnetMask メソッドは、以下のパラメータを受け入れます。

ipAddr (文字列)

サブネット アドレスを引き出すために使用する IP アドレスを指定します。

subnetMask (署名なし long)

サブネット アドレスを引き出すために使用するサブネット マスクを指定します。

戻り値

CreateIPConfigSubnetMask メソッドは以下のいずれかの値を返します。

- PolicyMgtIPConfig オブジェクト
- コールに失敗した場合は、**undef**

注釈

サブネット アドレスからアフィリエイト サイトにアクセスするユーザのみがアフィリエイト サイトで受け入れられます。サブネット アドレスは、渡された IP アドレスおよびサブネット マスクから引き出されます。

DeleteIPConfig メソッド -- IP 設定 オブジェクトを削除

DeleteIPConfig メソッドは、指定された IP 設定オブジェクトを削除します。

構文

DeleteIPConfig メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliate->DeleteIPConfig( IPConfig )
```

パラメータ

DeleteIPConfig メソッドは以下のパラメータを受け入れます。

IPConfig (PolicyMgtIPConfig)

削除する IP 設定オブジェクトを指定します。

戻り値

DeleteIPConfig メソッドは以下のいずれかの値を返します。

- 成功の場合は 0
- コールに失敗した場合は -1

Description メソッド -- アフィリエイト オブジェクトの説明を設定または取得

Description メソッドは、アフィリエイト オブジェクトの説明を設定または取得します。

構文

Description メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliate->Description( [affDesc] )
```

パラメータ

Description メソッドは以下のパラメータを受け入れます。

affDesc (文字列)

(オプション) 設定する説明を指定します。

戻り値

Description メソッドは以下のいずれかの値を返します。

- アフィリエイト オブジェクトの新規または既存の説明。
- コールに失敗した場合は、**undef**

GetAllAttributes メソッド -- アフィリエイト オブジェクトの属性を取得

GetAllAttributes メソッドは、アフィリエイト オブジェクトの既存のすべてのアフィリエイト属性を取得します。

構文

GetAllAttributes メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliate->GetAllAttributes( )
```

パラメータ

GetAllAttributes メソッドは、パラメータを受け入れません。

戻り値

GetAllAttributes メソッドは以下のいずれかの値を返します。

- PolicyMgtAffiliateAttr オブジェクトの配列
- コールに失敗した場合は、**undef**

GetAllIPConfigs メソッド -- アフィリエイトのすべての IP 設定オブジェクトを取得

GetAllIPConfigs メソッドは、アフィリエイト オブジェクトのすべての IP 設定オブジェクトを取得します。

構文

GetAllIPConfigs メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliate->GetAllIPConfigs( )
```

パラメータ

GetAllIPConfigs メソッドは、パラメータを受け入れません。

戻り値

GetAllIPConfigs メソッドは以下のいずれかの値を返します。

- PolicyMgtIPConfig オブジェクトの配列
- IP 設定オブジェクトが見つからなかった場合は、**undef**

GetAllUsers メソッド -- アフィリエイトと関連付けられたすべてのユーザを取得

GetAllUsers メソッドは、アフィリエイト オブジェクトと関連付けられたすべてのユーザを取得します。ユーザ ディレクトリが指定される場合、メソッドはアフィリエイトおよび特定のディレクトリと関連付けられたユーザのみを返します。

構文

GetAllUsers メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliate->GetAllUsers( [userDir] )
```

パラメータ

GetAllUsers メソッドは以下のパラメータを受け入れます。

userDir (PolicyMgtUserDir)

(オプション) アフィリエイト ユーザがメンバである必要があるユーザ ディレクトリを指定します。

戻り値

GetAllUsers メソッドは以下のいずれかの値を返します。

- PolicyMgtUser オブジェクトの配列
- ユーザが見つからなかった場合またはコールに失敗した場合は、**undef**

IsEnabled メソッド -- アフィリエイトの有効化フラグを設定または取得

IsEnabled メソッドは、アフィリエイト オブジェクトの有効化フラグを設定または取得します。

構文

IsEnabled メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliate->IsEnabled( [enableFlag] )
```

パラメータ

isEnabled メソッドは以下のパラメータを受け入れます。

enableFlag (int)

(オプション) 以下のようにアフィリエイト オブジェクトを有効にするかどうかを指定します。

- 値 1 はアフィリエイトを有効にします。
- 値 0 はアフィリエイトを無効にします。

戻り値

isEnabled メソッドは以下のいずれかの値を返します。

- アフィリエイト オブジェクトが有効な場合は、1
- アフィリエイト オブジェクトが無効な場合は、0
- コールに失敗した場合は、-1

Name メソッド -- アフィリエイト名を設定または取得

Name メソッドは、アフィリエイト オブジェクトの名前を設定または取得します。

構文

Name メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliate->Name( [affName] )
```

パラメータ

Name メソッドは以下のパラメータを受け入れます。

affName (文字列)

(オプション) 設定する名前を指定します。

戻り値

Name メソッドは以下のいずれかの値を返します。

- 新規または既存のアフィリエイト オブジェクト名
- コールに失敗した場合は、**undef**

Password メソッド -- アフィリエイトのパスワードを設定または取得

Password メソッドは、アフィリエイトで SiteMinder フェデレーション Web サービスにアクセスするために使用するパスワードを設定または取得します。

構文

Password メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliate->Password( [affPassword] )
```

パラメータ

Password メソッドは以下のパラメータを受け入れます。

affPassword (文字列)

(オプション) 設定するパスワードを指定します。

戻り値

Password メソッドは以下のいずれかの値を返します。

- 新規または既存のパスワード
- コールに失敗した場合は、**undef**

RemoveAttribute メソッド -- アフィリエイトから属性を削除

RemoveAttribute メソッドは、アフィリエイト オブジェクトから指定されたアフィリエイト属性を削除します。

構文

RemoveAttribute メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliate->RemoveAttribute( affiliateAttr )
```

パラメータ

RemoveAttribute メソッドは以下のパラメータを受け入れます。

affiliateAttr (PolicyMgtAffiliateAttr)

削除するアフィリエイト属性を指定します。

戻り値

RemoveAttribute メソッドは以下のいずれかの値を返します。

- 成功の場合は 0
- 失敗の場合は -1

RemoveUser メソッド -- アフィリエイトからユーザを削除

RemoveUser メソッドは、アフィリエイト オブジェクトから指定されたユーザを削除します。

構文

RemoveUser メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliate->RemoveUser( user )
```

パラメータ

RemoveUser メソッドは以下のパラメータを受け入れます。

user (タイプ)

削除するユーザを指定します。

戻り値

RemoveUser メソッドは以下のいずれかの値を返します。

- 成功の場合は 0
- 失敗の場合は -1

SAMLProfile メソッド -- SAML プロファイルのタイプを設定または取得

SAMLProfile メソッドは、アサーションの送信および受信に使用されるプロファイルのタイプを設定または取得します。

構文

SAMLProfile メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliate->SAMLProfile([SAMLProfile])
```

パラメータ

SAMLProfile メソッドは、以下のパラメータを受け入れます。

SAMLProfile (long)

(オプション) 以下のいずれかの有効な SAML プロファイルを指定します。

- **AFFILIATE_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。
- **AFFILIATE_SAML_PROFILE_POST**。生成された SAML アサーションは、PolicyMgtAffiliate->ConsumerURL メソッドで指定された URL に対して POST 処理されます。ポリシー管理 API のセッションバージョンが少なくとも v6.0 SP 2 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。

戻り値

SAMLProfile メソッドは以下のいずれかの値を返します。

- 新規または既存の SAML プロファイル タイプ
- コールに失敗した場合は、**undef**

SAMLVersion メソッド -- アフィリエイトの SAML バージョンを設定または取得

SAMLVersion メソッドは、アフィリエイトの SAML バージョンを設定または取得します。

構文

SAMLVersion メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliate->SAMLVersion( [SAMLVer] )
```

パラメータ

SAMLVersion メソッドは以下のパラメータを受け入れます。

SAMLVer (long)

(オプション) 設定する以下のいずれかの **SAML** バージョンを指定します。

- AFFILIATE_SAML_VER_1_0
- AFFILIATE_SAML_VER_1_1

戻り値

SAMLVersion メソッドは以下のいずれかの値を返します。

- 新規または既存の **SAML** バージョン
- コールに失敗した場合は、**undef**

注釈

SAML バージョンの指定は、ポリシー マネージャ API のセッションバージョンが少なくとも **v6.0 SP 1** である場合にのみ有効です。

Save メソッド -- ポリシー ストアにアフィリエイトを保存

Save メソッドは、ポリシー ストアにアフィリエイト オブジェクトを保存します。

構文

Save メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliate->Save( )
```

パラメータ

Save メソッドは、パラメータを受け入れません。

戻り値

Save メソッドは以下のいずれかの値を返します。

- 成功の場合は 0。
- 失敗の場合は -1。
- 変更を保存するためのユーザの権限が十分でない場合は、-4。
- パスとクラスが空の場合は、-10。

注釈

アフィリエイト オブジェクトに対して必要なすべての変更を行った後にこのメソッドを 1 回コールします。変更を有効にするには、このメソッドをコールする必要があります。

SessionSyncInterval メソッド -- セッション同期プロパティを設定または取得

SessionSyncInterval メソッドは、セッション同期間隔プロパティを設定または取得します。このプロパティは、アフィリエイトが共有セッションのステータスを検証するためにアサーションプロデューササイトにアクセスする頻度を秒単位で指定します。

構文

SessionSyncInterval メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliate->SessionSyncInterval( [SessionSyncInterval] )
```

パラメータ

SessionSyncInterval メソッドは以下のパラメータを受け入れます。

SessionSyncInterval (long)

(オプション) 設定するセッション同期間隔を指定します。

戻り値

SessionSyncInterval メソッドは以下のいずれかの値を返します。

- 新規または既存のセッション同期間隔
- コールに失敗した場合は、**undef**

SharedSessioning メソッド -- 共有セッション プロパティを設定または取得

SharedSessioning メソッドは、共有セッション プロパティを設定または取得します。共有セッションでは、一方のサイト上のセッションが終了すると、アサーションプロデューサ サイトおよびアフィリエイトの両方のセッションが終了します。

構文

SharedSessioning メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliate->ShareSessioning([shareFlag])
```

パラメータ

SharedSessioning メソッドは以下のパラメータを受け入れます。

shareFlag (int)

(オプション) 設定する共有セッション プロパティを指定します。

- アサーションプロデューサおよびアフィリエイトでセッション情報を共有できるようにするには、**1**
- プロデューサおよびアフィリエイトで個別のセッションを維持するには、**0**

戻り値

SharedSessioning メソッドは以下のいずれかの値を返します。

- 新規または既存の共有セッション プロパティ値
- コールに失敗した場合は、**undef**

SkewTime メソッド -- スキュー時間プロパティを設定または取得

SkewTime メソッドは、スキュー時間プロパティを設定または取得します。スキュー時間は、アサーションプロデューササイトのシステムクロック時間とアフィリエイトサイトのシステムクロック時間の間の秒単位の違いです。時間は GMT が基準になります。

構文

SkewTime メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliate->SkewTime( [SkewTime] )
```

パラメータ

SkewTime メソッドは以下のパラメータを受け入れます。

skewTime (long)

(オプション) 設定するスキュー時間を指定します。

戻り値

SkewTime メソッドは以下のいずれかの値を返します。

- 新規または既存のスキュー時間
- コールに失敗した場合は、**undef**

ValidityDuration メソッド -- SAML アサーションが有効な期間を設定または取得

ValidityDuration メソッドは、**SiteMinder** によって生成された **SAML** アサーションが有効な秒数を設定または取得します。アフィリエイトが指定された時間の後にアサーションを受信すると、アサーションが無効であると見なされます。

構文

ValidityDuration メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAffiliate->ValidityDuration( [ValidityDuration] )
```

パラメータ

ValidityDuration メソッドは以下のパラメータを受け入れます。

validityDuration (long)

(オプション) 設定する有効期間を指定します。

戻り値

ValidityDuration メソッドは以下のいずれかの値を返します。

- 新規または既存の有効期間
- コールに失敗した場合は、**undef**

CLI エージェント メソッド

ConvertFromLegacy メソッド -- v4.x エージェントを v5.x エージェントに変換

ConvertFromLegacy メソッドは、v4.x エージェントを v5.x エージェントに変換します。

構文

ConvertFromLegacy メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAgent->ConvertFromLegacy( )
```

パラメータ

ConvertFromLegacy メソッドは、パラメータを受け入れません。

戻り値

ConvertFromLegacy メソッドは以下のいずれかの値を返します。

- 成功の場合は 0
- 失敗の場合は -1

ConvertToLegacy メソッド -- v5.x エージェントを v4.x エージェントに変換

ConvertToLegacy メソッドは、v5.x エージェントを v4.x エージェントに変換します。

構文

ConvertToLegacy メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAgent->ConvertToLegacy( )
```

パラメータ

ConvertToLegacy メソッドは、パラメータを受け入れません。

戻り値

ConvertToLegacy メソッドは以下のいずれかの値を返します。

- 成功の場合は 0
- 失敗の場合は -1

Description メソッド -- エージェントの説明を設定または取得

Description メソッドは、エージェントの説明を設定または取得します。

構文

Description メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAgent->Description([agentDesc])
```

パラメータ

Description メソッドは以下のパラメータを受け入れます。

agentDesc (文字列)

(オプション) 設定する説明を指定します。

戻り値

Description メソッドは以下のいずれかの値を返します。

- 新規または既存のエージェントの説明
- 失敗の場合は、空の文字列

IPAddress メソッド -- エージェントの IP アドレスを設定または取得

IPAddress メソッドは、エージェントの IP アドレスを設定または取得します。

構文

IPAddress メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAgent->IPAddress([ipAddress])
```

パラメータ

IPAddress メソッドは以下のパラメータを受け入れます。

ipAddress (文字列)

(オプション) 設定する IP アドレスを指定します。

戻り値

IPAddress メソッドは以下のいずれかの値を返します。

- 新規または既存のエージェント IP アドレス
- コールに失敗した場合は、**undef**

Name メソッド -- エージェントの名前を設定または取得

Name メソッドは、エージェントの名前を設定または取得します。

構文

Name メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAgent->Name([agentName])
```

パラメータ

Name メソッドは以下のパラメータを受け入れます。

agentName (文字列)

(オプション) エージェントに割り当てる名前を指定します。

戻り値

Name メソッドは以下のいずれかの値を返します。

- 新規または既存のエージェントの名前
- コールに失敗した場合は、`undef`

RealmHintAttrID メソッド -- ヒント属性を設定または取得

RealmHintAttrID メソッドは、RADIUS エージェントのヒント属性を設定または取得します。

構文

RealmHintAttrID メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAgent->RealmHintAttrID([hintID])
```

パラメータ

RealmHintAttrID メソッドは以下のパラメータを受け入れます。

hintID (int)

(オプション) 設定するヒント属性 ID を指定します。

戻り値

RealmHintAttrID メソッドは以下のいずれかの値を返します。

- RADIUS エージェント用に設定する新規または既存のレルム ヒント属性
- 失敗した場合は、`-1`

SharedSecret メソッド -- v4.x エージェント用の共有秘密キーを設定または取得

SharedSecret メソッドは、v4.x エージェント用の共有秘密キーを設定または取得します。これは Web エージェント設定で使用する同じ共有秘密キーです。

構文

SharedSecret メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAgent->SharedSecret([sharedSecret])
```

パラメータ

SharedSecret メソッドは以下のパラメータを受け入れます。

sharedSecret (文字列)

(オプション) 設定する共有秘密キーを指定します。

戻り値

SharedSecret メソッドは以下のいずれかの値を返します。

- 新規または既存の共有秘密キー
- コールに失敗した場合は、`undef`

CLI エージェントの設定方法

AddAssociation メソッド -- この設定用の名前と値を追加

AddAssociation メソッドは、このエージェント設定用の設定パラメータ名と値を追加します。

構文

AddAssociation メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAgentConfig->AddAssociation(Name, Value, Flags)
```

パラメータ

AddAssociation メソッドは、以下のパラメータを受け入れます。

Name (文字列)

設定パラメータ名を指定します。

Value (文字列)

設定パラメータ値を指定します。

Flag (int)

以下のように暗号化フラグ値を指定します。

- 名前/値ペアが暗号化された形式で保存される場合は 1
- 名前/値ペアがプレーン テキストとして保存される場合は 0

戻り値

AddAssociation メソッドは以下のいずれかの値を返します。

- PolicyMgtAssociation オブジェクト
- コールに失敗した場合は、**undef**

AddAssociationMultiValue メソッド -- 複数值設定パラメータを追加

AddAssociationMultiValue メソッドは、このエージェント設定用の複数值設定パラメータを追加します。指定された設定パラメータが存在する場合、新しい値は古い値と連結されます。

注: このメソッドは 1 つのセッションで 2 回以上コールしないでください。そうしないと、新しい値は古い値に上書きします。

構文

AddAssociationMultiValue メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAgentConfig->AddAssociationMultiValue(Name, valueArray)
```

パラメータ

AddAssociationMultiValue メソッドは、以下のパラメータを受け入れます。

Name (文字列)

設定パラメータ名を指定します。

valueArray (文字列配列)

このパラメータ名に関連付けられた値の配列の参照を指定します (たとえば、`¥@myarray`)。

戻り値

AddAssociationMultiValue メソッドは以下のいずれかの値を返します。

- PolicyMgtAssociation オブジェクト
- コールに失敗した場合は、**undef**

注釈

複数値パラメータは、プレーンテキストとしてのみ保存できます。

Description メソッド -- エージェント設定オブジェクトの説明を設定または取得

Description メソッドは、エージェント設定オブジェクトの説明を設定または取得します。

構文

Description メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAgentConfig->Description([Description])
```

パラメータ

Description メソッドは、以下のパラメータを受け入れます。

Description (文字列)

(オプション) 設定する説明を指定します。

戻り値

Description メソッドは以下のいずれかの値を返します。

- 新規または既存のエージェント設定オブジェクトの説明
- コールに失敗した場合は、**undef**

GetAssociations メソッド -- すべての設定パラメータのリストを取得

GetAssociations メソッドは、このエージェント設定のすべての設定パラメータのリストを取得します。

構文

GetAssociations メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAgentConfig->GetAssociations( )
```

パラメータ

GetAssociations メソッドは、パラメータを受け入れません。

戻り値

GetAssociations メソッドは以下のいずれかの値を返します。

- **PolicyMgtAssociation** オブジェクトの配列。それぞれのオブジェクトには、設定パラメータ名およびその関連する値が含まれます。
- 設定パラメータ オブジェクトが存在しない場合またはコールに失敗した場合は、**undef**

Name メソッド -- エージェント設定オブジェクト名を設定または取得

Name メソッドは、エージェント設定オブジェクト名を設定または取得します。

構文

Name メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAgentConfig->Name([Name])
```

パラメータ

Name メソッドは、以下のパラメータを受け入れます。

Name (文字列)

(オプション) 設定する名前を指定します。

戻り値

Name メソッドは以下のいずれかの値を返します。

- 新規または既存のエージェント設定オブジェクト名
- コールに失敗した場合は、**undef**

RemoveAssociation メソッド -- 設定パラメータを削除

RemoveAssociation メソッドは、エージェント設定オブジェクトから設定パラメータ名と値のペアを削除します。

構文

RemoveAssociation メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAgentConfig->RemoveAssociation(assoc)
```

パラメータ

RemoveAssociation メソッドは以下のパラメータを受け入れます。

assoc (PolicyMgtAssociation)

削除する設定パラメータ名と値のペアを指定します。

戻り値

RemoveAssociation メソッドは以下のいずれかの値を返します。

- 成功の場合は 0
- コールに失敗した場合は、-1

CLI エージェント設定パラメータ メソッド

Name メソッド -- エージェント設定パラメータの名前部分を設定または取得

Name メソッドは、エージェント設定パラメータ名と値のペアの名前部分を設定または取得します。

構文

Name メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAssociation->Name([Name])
```

パラメータ

Name メソッドは、以下のパラメータを受け入れます。

Name (文字列)

(オプション) 設定する名前を指定します。

戻り値

Name メソッドは以下のいずれかの値を返します。

- エージェント設定パラメータの名前。
- 失敗した場合は、**undef**

Flags メソッド -- 暗号化フラグ属性を設定または取得

Flags メソッドは、エージェント設定名と値のペアの暗号化フラグ属性を設定または取得します。

構文

Flags メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAssociation->Flags([Flags])
```


パラメータ

Flags メソッドは以下のパラメータを受け入れます。

Flags (int)

(オプション) 設定するフラグ値を指定します。

戻り値

Flags メソッドは以下のいずれかの値を返します。

- 名前/値ペアが暗号化された形式の場合は **1**
- 名前/値ペアがプレーン テキストの場合は **0**
- コールに失敗した場合は、**undef**

Value メソッド -- エージェント設定パラメータの値を設定または取得

Value メソッドは、エージェント設定パラメータ名と値のペアの値部分を設定または取得します。

構文

Value メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAssociation->Value([Value])
```

パラメータ

Value メソッドは以下のパラメータを受け入れます。

Value (int)

(オプション) 設定する値を指定します。

戻り値

Value メソッドは以下のいずれかの値を返します。

- エージェント設定パラメータの値。
- 失敗した場合は、**undef**

CLI エージェント タイプ メソッド

GetDescription メソッド -- エージェント タイプの説明を取得

GetDescription メソッドは、エージェント タイプの説明を取得します。

構文

GetDescription メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAgentType->GetDescription( )
```

パラメータ

GetDescription メソッドは、パラメータを受け入れません。

戻り値

GetDescription メソッドは以下のいずれかの値を返します。

- 新規または既存のエージェント タイプの説明
- 失敗の場合は、NULL 文字列

GetName メソッド -- エージェント タイプの名前を取得

GetName メソッドは、たとえば、Web エージェントなどのエージェント タイプの名前を取得します。

構文

GetName メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAgentType->GetName( )
```

パラメータ

GetName メソッドは、パラメータを受け入れません。

戻り値

GetName メソッドは以下のいずれかの値を返します。

- エージェント タイプの名前
- 失敗の場合は、NULL 文字列

CLI 認証および許可マップ メソッド

AuthDir メソッド -- 認証ディレクトリを設定または取得

AuthDir メソッドは、認証および許可マップの認証ディレクトリを設定または取得します。

構文

AuthDir メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAuthAzMap->AuthDir([userDir])
```

パラメータ

AuthDir メソッドは以下のパラメータを受け入れます。

userDir (PolicyMgtUserDir)

(オプション) 設定する認証ディレクトリを指定します。

戻り値

AuthDir メソッドは以下のいずれかの値を返します。

- 新規または既存の PolicyMgtUserDir オブジェクト。
- コールに失敗した場合は、**undef**

AzDir メソッド -- 許可ディレクトリを設定または取得

AzDir メソッドは、認証および許可マップの許可ディレクトリを設定または取得します。

構文

AzDir メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAuthAzMap->AzDir([userDir])
```

パラメータ

AzDir メソッドは以下のパラメータを受け入れます。

userDir (PolicyMgtUserDir)

(オプション) 設定する許可ディレクトリを指定します。

戻り値

AzDir メソッドは以下のいずれかの値を返します。

- 新規または既存の PolicyMgtUserDir オブジェクト
- コールに失敗した場合は、**undef**

MapType メソッド -- 認証および許可マップのタイプを設定または取得

MapType メソッドは、認証および許可マップのタイプを設定または取得します。

構文

MapType メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAuthAzMap->MapType([mapType])
```

パラメータ

MapType メソッドは以下のパラメータを受け入れます。

mapType (int)

(オプション) マップタイプを指定します。以下の値が有効です。

AUTHAZMAPTYPE_DN (値=1)。マッピングは DN に基づいています。

AUTHAZMAPTYPE_UNIVERSALID (値=2)。マッピングはユニバーサル識別子に基づいています。

AUTHAZMAPTYPE_ATTR (値=3)。マッピングはディレクトリ内の属性に基づいています。

戻り値

MapType メソッドは以下のいずれかの値を返します。

- 新規または既存のマップタイプ
- コールに失敗した場合は、-1

CLI 認証方式メソッド

AddMessageConsumerPluginToSAML1xScheme メソッド -- メッセージコンシューマプラグイン クラス名を追加

AddMessageConsumerPluginToSAML1xScheme メソッドは、メッセージコンシューマプラグインクラス名およびパラメータを SAML1x 認証方式に追加します。

構文

AddMessageConsumerPluginToSAML1xScheme メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAuthScheme->AddMessageConsumerPluginToSAML1xScheme(pluginClass, pluginParam)
```

パラメータ

AddMessageConsumerPluginToSAML1xScheme メソッドは、以下のパラメータを受け入れます。

pluginClass (文字列)

メッセージ コンシューマ プラグイン クラス名を指定します。

pluginParam (文字列)

メッセージ コンシューマ プラグイン パラメータ名を指定します。

戻り値

AddMessageConsumerPluginToSAML1xScheme メソッドは以下のいずれかの値を返します。

- Sm_PolicyApi_Success
- Sm_PolicyApi_Failure

AddRedirectURLToSAML1xScheme メソッド -- 認証方式にリダイレクト値を追加

AddRedirectURLToSAML1xScheme メソッドは、リダイレクト URL、タイプおよびモードを SAML1x 認証方式に追加します。

構文

AddRedirectURLToSAML1xScheme メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAuthScheme->AddRedirectURLToSAML1xScheme(iTypeURL, URL, redirectMode)
```

パラメータ

AddRedirectURLToSAML1xScheme メソッドは、以下のパラメータを受け入れます。

iTypeURL (int)

以下のいずれかの値であるリダイレクト URL タイプを指定します。

- 0 -- ユーザが見つからない
- 1 -- 無効なメッセージ
- 2 -- 受け入れられない認証情報

URL (文字列)

リダイレクト URL サイトを指定します。

redirectMode (int)

以下のいずれかの値であるリダイレクト モードを指定します。

- 0 -- 302 データなし
- 1 -- POST

戻り値

AddRedirectURLToSAML1xScheme メソッドは以下のいずれかの値を返します。

- Sm_PolicyApi_Success
- Sm_PolicyApi_Failure

AddTargetConfigToSAML1xScheme メソッド -- デフォルト ターゲット設定を設定

AddTargetConfigToSAML1xScheme メソッドは、デフォルト ターゲットおよび QueryParameterOverridesTarget 設定を SAML1x 認証方式に設定します。

構文

AddTargetConfigToSAML1xScheme メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAuthScheme->AddTargetConfigToSAML1xScheme(pszTargetURL,  
iQPOverrideTarget)
```

パラメータ

AddTargetConfigToSAML1xScheme メソッドは、以下のパラメータを受け入れます。

pszTargetURL (文字列)

デフォルト ターゲット URL を指定します。

iQPOVERRIDE (int)

クエリ パラメータがデフォルト ターゲットの設定を無視するかどうかを指定します。

戻り値

AddTargetConfigToSAML1xScheme メソッドは以下のいずれかの値を返します。

- Sm_PolicyApi_Success
- Sm_PolicyApi_Failure

CustomLib メソッド -- 共有ライブラリの名前を設定または取得

CustomLib メソッドは、認証方式を実装する共有ライブラリの名前を設定または取得します。

構文

CustomLib メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAuthScheme->CustomLib([libName])
```

パラメータ

CustomLib メソッドは以下のパラメータを受け入れます。

libName (文字列)

(オプション) 共有ライブラリ名を指定します。

戻り値

CustomLib メソッドは以下のいずれかの値を返します。

- 新規または既存のライブラリ名
- コールに失敗した場合は、**undef**

注釈

事前定義済み認証方式タイプがそれぞれデフォルト ライブラリと共にリリースされますが、カスタム ライブラリを使用できます。 カスタム認証方式を使用する場合は、カスタム ライブラリを指定する必要があります。

CustomParam メソッド -- 認証方式に渡される情報を設定または取得

CustomParam メソッドは、認証方式に渡される情報を設定または取得します。認証方式用のデフォルト パラメータを受け入れるか、新しいパラメータを指定できます。

構文

CustomParam メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAuthScheme->CustomParam([param])
```

パラメータ

CustomParam メソッドは以下のパラメータを受け入れます。

param (文字列)

(オプション) 渡すパラメータ情報を指定します。

戻り値

CustomParam メソッドは以下のいずれかの値を返します。

- 新規または既存のパラメータ情報
- コールに失敗した場合は、**NULL** 文字列

CustomSecret メソッド -- カスタム認証方式用の共有秘密キーを設定または取得

CustomSecret メソッドは、カスタム認証方式用の共有秘密キーを設定または取得します。

構文

CustomSecret メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAuthScheme->CustomSecret([param])
```

パラメータ

CustomSecret メソッドは以下のパラメータを受け入れます。

param (文字列)

(オプション) 共有秘密キーを指定します。

戻り値

CustomSecret メソッドは以下のいずれかの値を返します。

- 新規または既存の共有秘密キー
- コールに失敗した場合は、NULL 文字列

Description メソッド -- 認証方式の説明を設定または取得

Description メソッドは、認証方式の説明を設定または取得します。

構文

Description メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAuthScheme->Description([schemeDesc])
```

パラメータ

Description メソッドは以下のパラメータを受け入れます。

schemeDesc (文字列)

(オプション) 説明を指定します。

戻り値

Description メソッドは以下のいずれかの値を返します。

- 新規または既存の認証方式の説明
- コールに失敗した場合は、空の文字列

GetMessageConsumerPluginFromSAML1xScheme メソッド -- メッセージ コンシューマ プラグイン クラス名を取得

GetMessageConsumerPluginFromSAML1xScheme メソッドは、メッセージ コンシューマ プラグイン クラス名およびパラメータを SAML1x 認証方式から取得します。

構文

GetMessageConsumerPluginFromSAML1xScheme メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAuthScheme->GetMessageConsumerPluginFromSAML1xScheme(pluginClass, pluginParam)
```

パラメータ

GetMessageConsumerPluginFromSAML1xScheme メソッドは、以下のパラメータを受け入れます。

pluginClass (文字列)

メッセージ コンシューマ プラグイン クラス名を指定します。

pluginParam (文字列)

メッセージ コンシューマ プラグイン パラメータ名を指定します。

戻り値

GetMessageConsumerPluginFromSAML1xScheme メソッドは以下のいずれかの値を返します。

- Sm_PolicyApi_Success
- Sm_PolicyApi_Failure

GetRedirectURLFromSAML1xScheme メソッド -- リダイレクト URL を取得

GetRedirectURLFromSAML1xScheme メソッドは、リダイレクト URL、タイプおよびモードを SAML1x 認証方式から取得します。

構文

GetRedirectURLFromSAML1xScheme メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAuthScheme->GetRedirectURLFromSAML1xScheme(iTypeURL, URL, redirectMode)
```

パラメータ

GetRedirectURLFromSAML1xScheme メソッドは、以下のパラメータを受け入れます。

iTypeURL (int)

以下のいずれかの値であるリダイレクト URL タイプを指定します。

- 0 -- ユーザが見つからない
- 1 -- 無効なメッセージ
- 2 -- 受け入れられない認証情報

URL (文字列)

リダイレクト URL サイトを指定します。

redirectMode (int)

以下のいずれかの値であるリダイレクト モードを指定します。

- 0 -- 302 データなし
- 1 -- POST

戻り値

GetRedirectURLFromSAML1xScheme メソッドは以下のいずれかの値を返します。

- Sm_PolicyApi_Success
- Sm_PolicyApi_Failure

GetTargetConfigFromSAML1xScheme メソッド -- デフォルト ターゲット設定を取得

GetTargetConfigFromSAML1xScheme メソッドは、デフォルト ターゲットおよび QueryParameterOverridesTarget 設定を SAML1x 認証方式から取得します。

構文

GetTargetConfigFromSAML1xScheme メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAuthScheme->GetTargetConfigFromSAML1xScheme(pszTargetURL,  
iQPOverrideTarget)
```

パラメータ

GetTargetConfigFromSAML1xScheme メソッドは、以下のパラメータを受け入れます。

pszTargetURL (文字列)

デフォルト ターゲット URL を指定します。

iQPOverrideTarget (int)

クエリ パラメータがデフォルト ターゲットの設定を無視するかどうかを指定します。

戻り値

GetTargetConfigFromSAML1xScheme メソッドは以下のいずれかの値を返します。

- Sm_PolicyApi_Success
- Sm_PolicyApi_Failure

IgnorePwd メソッド -- パスワード ポリシーを確認するかどうかを指定

IgnorePwd メソッドは、認証方式用のパスワード ポリシーを確認するかどうかを指定するフラグを設定または取得します。

構文

IgnorePwd メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAuthScheme->IgnorePwd([pwdFlag])
```

パラメータ

IgnorePwd メソッドは以下のパラメータを受け入れます。

pwdFlag (int)

(オプション) パスワード ポリシーを無視するかどうか (1 に設定)、または実施するかどうか (0 に設定) を指定します。

戻り値

IgnorePwd メソッドは以下のいずれかの値を返します。

- パスワード ポリシーを無視する場合は、1
- パスワード ポリシーの場合は、0
- コールに失敗した場合は、-1

IsRadius メソッド -- 認証方式で RADIUS エージェントをサポートするかどうかを判定

IsRadius メソッドは、認証方式で RADIUS エージェントをサポートするかどうかを指定するフラグを設定または取得します。

構文

IsRadius メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAuthScheme->IsRadius([radFlag])
```

パラメータ

IsRadius メソッドは以下のパラメータを受け入れます。

radFlag (int)

(オプション) 認証方式で RADIUS エージェントをサポートするかどうかを指定します (1=yes、0=no)。

戻り値

IsRadius メソッドは以下のいずれかの値を返します。

- 認証方式で RADIUS エージェントをサポートする場合は、1
- 認証方式で RADIUS エージェントをサポートしない場合は、0
- コールに失敗した場合は、-1

IsTemplate メソッド -- 認証方式がテンプレートかどうかを判定

IsTemplate メソッドは、認証方式がテンプレートかどうかを示すフラグ値を取得します。

構文

IsTemplate メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAuthScheme->IsTemplate( )
```

パラメータ

IsTemplate メソッドは、パラメータを受け入れません。

戻り値

IsTemplate メソッドは以下のいずれかの値を返します。

- 認証方式がテンプレートの場合は、1
- 認証方式がテンプレートでない場合は、0
- コールに失敗した場合は、-1

注釈

Perl ポリシー管理 API でテンプレートとして認証方式を設定することは、SiteMinder v6.0 SP3 では非推奨となっています。

IsUsedByAdmin メソッド -- 方式で管理者を認証するかどうかを判定

IsUsedByAdmin メソッドは、方式を使用して管理者を認証するかどうかを判定します。

構文

IsUsedByAdmin メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAuthScheme->IsUsedByAdmin([useAdminFlag])
```

パラメータ

IsUsedByAdmin メソッドは以下のパラメータを受け入れます。

useAdminFlag (int)

(オプション) 方式を使用して、管理者を認証するかどうかを指定します。

- 方式を管理者認証に使用できるようにする場合は、1
- 方式を管理者認証に使用できないようにする場合は、0

戻り値

IsUsedByAdmin メソッドは以下のいずれかの値を返します。

- 方式を使用して、管理者を認証できる場合は、1
- 方式を使用して、管理者を認証できない場合は、0
- コールに失敗した場合は、-1

Name メソッド -- 認証方式の名前を設定または取得

Name メソッドは、認証方式の名前を設定または取得します。

構文

Name メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAuthScheme->Name([authSchemeName])
```

パラメータ

Name メソッドは以下のパラメータを受け入れます。

authSchemeName (文字列)

(オプション) 認証方式に割り当てる名前を指定します。

戻り値

Name メソッドは以下のいずれかの値を返します。

- 新規または既存の認証方式の名前
- コールに失敗した場合は、`undef`

ProtectionLevel メソッド -- 認証方式の保護レベルを設定または取得

ProtectionLevel メソッドは、認証方式の保護レベルを設定または取得します。

構文

ProtectionLevel メソッドの形式は以下のとおりです。

```
codefirstNetegrity::PolicyMgtAuthScheme->ProtectionLevel([nlevel])
```

パラメータ

ProtectionLevel メソッドは以下のパラメータを受け入れます。

nlevel (int)

(オプション) 設定する保護レベルを指定します。

戻り値

ProtectionLevel メソッドは以下のいずれかの値を返します。

- 新規または既存の認証方式の保護レベル
- 失敗した場合は、-1

注釈

レベル値の有効範囲は 1 ～ 1000 です。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 0 に設定します。

Save メソッド -- ポリシー ストアに認証方式を保存

Save メソッドは、ポリシー ストアに認証方式を保存します。

構文

Save メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAuthScheme->Save( )
```

パラメータ

Save メソッドは、パラメータを受け入れません。

戻り値

Save メソッドは以下のいずれかの値を返します。

- 成功の場合は 0
- 失敗の場合は -1
- 変更を保存するためのユーザの権限が十分でない場合は、-4
- 方式オブジェクト識別子が見つからない場合は、-100

注釈

認証方式に対して必要なすべての変更を行った後にこのメソッドを 1 回 コールします。変更を有効にするには、このメソッドをコールする必要があります。

SaveCredentials メソッド -- ユーザ認証情報を保存できるかどうかを判定

SaveCredentials メソッドは、ユーザ認証情報を保存できるようにするフラグを設定または取得します。

構文

SaveCredentials メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAuthScheme->SaveCredentials([credFlag])
```

パラメータ

SaveCredentials メソッドは以下のパラメータを受け入れます。

credFlag (int)

(オプション) フラグ値を指定します。

- 認証情報を保存できる場合は、1
- 認証情報を保存できない場合は、0

戻り値

SaveCredentials メソッドは以下のいずれかの値を返します。

- ユーザ認証情報を保存できる場合は、1
- ユーザ認証情報を保存できない場合は、0

Type メソッド -- 認証方式タイプを設定または取得

Type メソッドは、認証方式タイプを設定または取得します。

構文

Type メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAuthScheme->Type([template])
```

パラメータ

Type メソッドは以下のパラメータを受け入れます。

template (PolicyMgtAuthScheme)

(オプション) 認証方式タイプを指定します。

戻り値

Type メソッドは以下のいずれかの値を返します。

- 新規または既存の認証方式タイプ
- コールに失敗した場合は、**undef**

CLI 証明書マッピング メソッド

AttrMap メソッド -- 証明書マッピングの属性マップを設定または取得

AttrMap メソッドは、証明書マッピングの属性マップを設定または取得します。

構文

AttrMap メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtCertMap->AttrMap ([attribute_map])
```

パラメータ

AttrMap メソッドは以下のパラメータを受け入れます。

attribute_map (文字列)

(オプション) 設定する属性マップを指定します。

戻り値

AttrMap メソッドは以下のいずれかの値を返します。

- 証明書マップの新規または既存の属性
- コールに失敗した場合は、空の文字列

CacheCRL メソッド -- キャッシュ証明書失効リスト(CRL)エントリをキャッシュに入れるかどうかを判定

CacheCRL メソッドは、Certificate Revocation List (CRL) エントリをキャッシュに入れるかどうかを判定するフラグを設定または取得します。このフラグを設定すると、SiteMinder は CRL の NextUpdate 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。

構文

CacheCRL メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtCertMap->CacheCRL([cacheFlag])
```

パラメータ

CacheCRL メソッドは以下のパラメータを受け入れます。

cacheFlag (int)

(オプション) CRL エントリをキャッシュに入れるかどうかを指定します。

- 1 は、キャッシュ エントリが使用されることを指定します
- 0 は、キャッシュ エントリが使用されないことを指定します

戻り値

CacheCRL メソッドは以下のいずれかの値を返します。

- 新規または既存のキャッシュ フラグ設定
- コールに失敗した場合は、-1

CertRequired メソッド -- 証明書の検証が必要かどうかを判定

CertRequired メソッドは、ユーザによって提供された証明書がユーザ ディレクトリ内のユーザのエントリに保存された証明書に一致することを検証するよう SiteMinder に要求するフラグを設定または取得します。ユーザ ディレクトリは、LDAP ユーザ ディレクトリである必要があります。

構文

CertRequired メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtCertMap->CertRequired([certFlag])
```

パラメータ

CertRequired メソッドは以下のパラメータを受け入れます。

certFlag (int)

(オプション) 以下のように証明書の検証が必要かどうかを指定します。

- 証明書の検証が必要な場合は、1
- 証明書の検証が必要でない場合は、0

戻り値

CertRequired メソッドは以下のいずれかの値を返します。

- 新規または既存のフラグ設定
- コールに失敗した場合は、-1

CRLUserDirectory メソッド -- 証明書失効リスト(CRL)が存在する LDAP ディレクトリを設定または取得

CRLUserDirectory メソッドは、証明書失効リスト (CRL) が存在する LDAP ユーザ ディレクトリを指定または取得します。

構文

CRLUserDirectory メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtCertMap->CRLUserDirectory([crlDir])
```

パラメータ

CRLUserDirectory メソッドは以下のパラメータを受け入れます。

crlDir (PolicyMgtUserDir)

(オプション) CRL が存在するユーザ ディレクトリを指定します。

戻り値

CRLUserDirectory メソッドは以下のいずれかの値を返します。

- PolicyMgtUserDir オブジェクト
- コールに失敗した場合は、**undef**

Description メソッド -- 証明書マップの説明を設定または取得

Description メソッドは、証明書マップの説明を設定または取得します。

構文

Description メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtCertMap->Description([certMapDesc])
```

パラメータ

Description メソッドは以下のパラメータを受け入れます。

certMapDesc (文字列)

(オプション) 設定する説明を指定します。

戻り値

Description メソッドは以下のいずれかの値を返します。

- 新規または既存の証明書マップの説明
- コールに失敗した場合は、空の文字列

DirectoryType メソッド -- ユーザ ディレクトリのタイプを設定または取得

DirectoryType メソッドは、ユーザ認証に含まれるユーザ ディレクトリのタイプを設定または取得します。

構文

DirectoryType メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtCertMap->DirectoryType([dirType])
```

パラメータ

DirectoryType メソッドは以下のパラメータを受け入れます。

dirType (int)

(オプション) 以下のいずれかのタイプのユーザ ディレクトリを指定します。

- Sm_PolicyApi_DirType_LDAP
- Sm_PolicyApi_DirType_WinNT
- Sm_PolicyApi_DirType_ODBC

戻り値

DirectoryType メソッドは以下のいずれかの値を返します。

- 新規または既存のディレクトリ タイプ
- コールに失敗した場合は、**undef**

EnableCRL メソッド -- 取り消された証明書に対し証明書失効リスト(CRL)を確認するかどうかを判定

EnableCRL メソッドは、取り消された証明書に対し証明書失効リスト (CRL) を確認するかどうかを判定するフラグを設定または取得します。

構文

EnableCRL メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtCertMap->EnableCRL([ckCRLflag])
```


パラメータ

EnableCRL メソッドは以下のパラメータを受け入れます。

ckCRLFlag (int)

(オプション) 以下のように **CRL** に対して証明書を確認するかどうかを指定します。

- 1 は、証明書を確認すべきであることを指定します
- 0 は、証明書を確認すべきでないことを指定します

戻り値

EnableCRL メソッドは以下のいずれかの値を返します。

- 新規または既存のフラグ設定
- コールに失敗した場合は、-1

注釈

CRL は、認証機関によって発行されて取り消された X.509 クライアント証明書のリストです。CRL に対する証明書の比較は、証明書が有効であることを確認するための 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、SiteMinder は、このユーザの証明書を CRL で検出し、認証を拒否します。

CRL 確認を有効にするには、メソッド `PolicyMgtCertMap->CRLUserDirectory` をコールして、CRL が存在するユーザディレクトリを指定します。

IssuerDN メソッド -- 証明書発行者の DN を設定または取得

IssuerDN メソッドは、証明書発行者の DN を設定または取得します。

構文

IssuerDN メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtCertMap->IssuerDN([issuerDN])
```

パラメータ

IssuerDN メソッドは以下のパラメータを受け入れます。

issuerDN (文字列)

(オプション) 設定する発行者 DN を指定します。

戻り値

IssuerDN メソッドは以下のいずれかの値を返します。

- 新規または既存の発行者 DN
- コールに失敗した場合は、空の文字列

UseDistributionPoints メソッド -- 証明書失効リスト(CRL)検索で配布ポイントを使用するかどうかを判定

UseDistributionPoints メソッドは、証明書失効リスト (CRL) 検索で配布ポイントを検索開始ポイントとして使用するかどうかを示すフラグを設定または取得します。

構文

UseDistributionPoints メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtCertMap->UseDistributionPoints([distPointsFlag])
```

パラメータ

UseDistributionPoints メソッドは、以下のパラメータを受け入れます。

distPointsFlag (int)

(オプション) 以下のように CRL 検索に配布ポイントを使用するかどうかを指定します。

- 1 は、配布ポイントを使用すべきであることを指定します
- 0 は、配布ポイントを使用すべきでないことを指定します

戻り値

UseDistributionPoints メソッドは以下のいずれかの値を返します。

- 新規または既存のフラグ設定
- コールに失敗した場合は、-1

注釈

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。CRL 確認の開始ポイントを指定して、配布ポイントにより CRL 全体の検索にかかる処理時間を節減します。

VerifySignature メソッド -- SiteMinder で認証機関の署名を検証するかどうかを判定

VerifySignature メソッドは、SiteMinder で証明書失効リスト (CRL) 内の認証機関の署名を検証するかどうかを示すフラグを設定または取得します。

構文

VerifySignature メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtCertMap->VerifySignature([verifyFlag])
```

パラメータ

VerifySignature メソッドは以下のパラメータを受け入れます。

verifyFlag (int)

(オプション) 以下のように CRL で CA の署名を検証するかどうかを指定します。

- 1 は、署名を検証すべきであることを指定します
- 0 は、署名を検証すべきでないことを指定します

戻り値

VerifySignature メソッドは以下のいずれかの値を返します。

- 新規または既存のフラグ設定
- コールに失敗した場合は、-1

CLI クラスタ メソッド

AddServer メソッド -- クラスタにサーバを追加

AddServer メソッドは、クラスタにサーバを追加します。

構文

AddServer メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtCluster->AddServer(Host, Port)
```

パラメータ

AddServer メソッドは、以下のパラメータを受け入れます。

Host (文字列)

(オプション) ホスト IP アドレスを指定します。

Port (int)

サーバのポートを指定します

戻り値

AddServer メソッドは以下のいずれかの値を返します。

- PolicyMgtServer オブジェクト
- コールに失敗した場合は、**undef**

注釈

クラスタ内のサーバは配列で参照されます。クラスタにサーバを追加すると、サーバ配列の終わりに追加されます。

要求がクラスタ内で利用できるキャパシティの最も高いサーバに送信される動的負荷分散により、サーバがクラスタに追加される順序は重要ではありません。

ホスト設定にクラスタ化されていないサーバを追加するには、PolicyMgtHostConfig->AddServer メソッドをコールします。

GetAllServers メソッド -- クラスタ内のすべてのサーバの配列を取得

GetAllServers メソッドは、クラスタ内のすべてのサーバの配列を取得します。

構文

GetAllServers メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtCluster->GetAllServers( )
```

パラメータ

GetAllServers メソッドは、パラメータを受け入れません。

戻り値

GetAllServers メソッドは以下のいずれかの値を返します。

- PolicyMgtServer オブジェクトの配列
- コールに失敗した場合は、**undef**

注釈

クラスタのメンバでないサーバを取得するには、
PolicyMgtHostConfig->GetAllServers メソッドをコールします。

CLI ドメイン メソッド

AddAdmin メソッド -- ドメインに管理者を追加

AddAdmin メソッドは、ドメインに管理者を追加します。

構文

AddAdmin メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtDomain->AddAdmin(admin)
```

パラメータ

AddAdmin メソッドは以下のパラメータを受け入れます。

admin (タイプ)

ドメインに追加する管理者を指定します。

戻り値

AddAdmin メソッドは以下のいずれかの値を返します。

- 成功の場合は 0
- コールに失敗した場合は、-1

注釈

管理者は、ドメイン内の **SiteMinder** オブジェクトを作成、編集、削除できます。

特定のドメインの管理者を作成するためにポリシー管理 API を使用することはできません。ただし、管理 UI を使用して、ドメインの管理者を作成する場合は、**PolicyMgtAffDomain->AddAdmin** メソッドをコールして、別のドメインにその管理者を追加できます。

AddUserDir メソッド -- ドメインとユーザ ディレクトリを関連付け

AddUserDir メソッドは、ドメインとユーザ ディレクトリを関連付けます。

構文

AddUserDir メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtDomain->AddUserDir(userDir)
```

パラメータ

AddUserDir メソッドは以下のパラメータを受け入れます。

userDir (PolicyMgtUserDir)

ドメインと関連付けるユーザ ディレクトリを指定します。

戻り値

AddUserDir メソッドは以下のいずれかの値を返します。

- 成功の場合は 0
- コールに失敗した場合は、-1

注釈

ユーザ認証時に、ユーザが提供した認証情報は、このユーザ ディレクトリに保存された認証情報と照合されます。

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。 検索順序を変更するには、PolicyMgtAffDomain -から SetUserDirSearchOrder メソッドを呼び出します。

CreatePolicy メソッド -- ドメイン内のポリシーを作成および設定

CreatePolicy メソッドは、ドメイン内のポリシーを作成および設定します。

構文

CreatePolicy メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtDomain->CreatePolicy(policyName [, policyDesc] [, enableFlag] [, activeExpr])
```

パラメータ

CreatePolicy メソッドは、以下のパラメータを受け入れます。

policyName (文字列)

ポリシーの名前を指定します。

policyDesc (文字列)

(オプション) ポリシーの説明を指定します。

enableFlag (int)

(オプション) 以下のようにポリシーを有効 (1) にするか無効 (0) にするかを指定します。 デフォルトは有効化です。

activeExpr (文字列)

(オプション) ポリシーのアクティブな式を指定します。

戻り値

CreatePolicy メソッドは以下のいずれかの値を返します。

- PolicyMgtPolicy オブジェクト
- コールに失敗した場合は、**undef**

CreateRealm メソッド -- ドメイン内にトップレベル レalmを作成および設定

CreateRealm メソッドは、ドメイン内にトップレベル レalmを作成および設定します。

構文

CreateRealm メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtDomain->CreateRealm(realName, agent, authScheme [, realmDesc]
[, resFilter] [, procAuthEvents] [, procAzEvents] [, protectAll] [, maxTimeout] [,
idleTimeout] [, syncAudit] [, azUserDir] [, regScheme])
```


パラメータ

CreateRealm メソッドは、以下のパラメータを受け入れます。

realmName (文字列)

レルムの名前を指定します。

agent (PolicyMgtAgent)

レルムを保護するエージェントまたはエージェントのグループを指定します。

authScheme (PolicyMgtAuthScheme)

レルムと関連付ける認証方式を指定します。

realmDesc (文字列)

(オプション) レルムの説明を指定します。

resFilter (文字列)

(オプション) レルムのリソース フィルタを指定します。

procAuthEvents (int)

(オプション) 認証イベントを処理するかどうかを指定します。有効にするには **1** に設定し、無効にするには **0** に設定します。デフォルトは有効化です。認証イベント処理はパフォーマンスに影響します。レルムのルールを認証イベントによってトリガしない場合は、このフラグを **0** に設定します。

procAzEvents (int)

(オプション) 許可イベントを処理するかどうかを指定します。有効にするには **1** に設定し、無効にするには **0** に設定します。デフォルトは有効化です。認可イベント処理はパフォーマンスに影響します。レルムのルールを許可イベントによってトリガしない場合は、このフラグを **0** に設定します。

protectAll (int)

(オプション) デフォルトのリソース保護をアクティブにするかどうかを指定します。有効にするには **1** に設定し、無効にするには **0** に設定します。デフォルトは有効化です。

***maxTimeout* (int)**

(オプション) 再認証が必要になる前に、ユーザがレルムにアクセスできる最大時間を秒単位で指定します。デフォルトは **7200** です (2 時間)。

***idleTimeout* (int)**

(オプション) 再認証が必要になる前に、ユーザがレルムで非アクティブ状態を維持できる最大時間を秒単位で指定します。デフォルトは **3600** です (1 時間)。

***syncAudit* (int)**

(オプション) 同期監査を有効にするフラグを指定します。有効にするには **1** に設定し、無効にするには **0** に設定します。このフラグが有効な場合は、リソースへのアクセスを許可する前に、**SiteMinder** はポリシー サーバおよびエージェント アクションをログ記録します。デフォルトは無効化です。

***azUserDir* (PolicyMgtUserDir)**

(オプション) レルム内のユーザが許可されるディレクトリを指定します。デフォルトはデフォルト ディレクトリです。

***regScheme* (タイプ)**

(オプション) レルム内のリソースにアクセスする新しいユーザを登録するために使用される登録方式を指定します。

戻り値

CreateRealm メソッドは以下のいずれかの値を返します。

- **PolicyMgtRealm** オブジェクト
- コールに失敗した場合は、**undef**

注釈

このメソッドは、非永続セッション用に設定されるレلمを作成します。
SiteMinder 5.0 永続セッション用のレلمを設定するには、管理 UI でレلمを編集します。

注: ポリシー管理 API は、以下のようにそのメソッドがコールされたオブジェクトの直接の子孫であるレلمのみを操作します。

- ドメイン下のレلمの場合は、ドメイン オブジェクト内のトップレベルのレلمのみを操作できます。
- レلم下のレلمの場合は、特定のレベルの直下のレلمのみを操作できます。

CreateResponse メソッド -- レスポンスを作成

CreateResponse メソッドは、レスポンスを作成します。

構文

CreateResponse メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtDomain->CreateResponse(resName, agentType [, resDesc])
```

パラメータ

CreateResponse メソッドは、以下のパラメータを受け入れます。

resName (文字列)

レスポンスの名前を指定します。

agentType (PolicyMgtAgentType)

レスポンスと関連付けられたエージェント タイプを指定します。エージェント タイプ オブジェクトを取得するには、
PolicyMgtSession->GetAgentType メソッドをコールします。

resDesc (文字列)

(オプション) レスポンスの説明を指定します。

戻り値

CreateResponse メソッドは以下のいずれかの値を返します。

- PolicyMgtResponse オブジェクト
- コールに失敗した場合は、**undef**

注釈

エージェントは、ある特定のイベントに基づいてレスポンスを返します。たとえば、不正なユーザが保護されている Web ページにアクセスしようとする、レスポンスによりユーザを適切なメッセージを表示する HTML ページにリダイレクトできます。

CreateResponseGroup メソッド -- ドメイン用のレスポンス グループを作成

CreateResponseGroup メソッドは、ドメイン用のレスポンス グループを作成します。

構文

CreateResponseGroup メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtDomain->CreateResponseGroup(groupName, agentType, [, groupDesc])
```

パラメータ

CreateResponseGroup メソッドは、以下のパラメータを受け入れます。

groupName (文字列)

グループの名前を指定します。

agentType (PolicyMgtAgentType)

レスポンス グループと関連付けられたエージェント タイプを指定します。 エージェント タイプ オブジェクトを取得するには、
PolicyMgtSession->GetAgentType メソッドをコールします。

groupDesc (文字列)

(オプション) グループの説明を指定します。

戻り値

CreateResponseGroup メソッドは以下のいずれかの値を返します。

- PolicyMgtGroup オブジェクト
- コールに失敗した場合は、**undef**

CreateRuleGroup メソッド -- ドメインのルール グループを作成

CreateRuleGroup メソッドは、ドメインのルール グループを作成します。

構文

CreateRuleGroup メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtDomain->CreateRuleGroup(groupName, agentType [, groupDesc])
```

パラメータ

CreateRuleGroup メソッドは、以下のパラメータを受け入れます。

groupName (文字列)

グループの名前を指定します。

agentType (PolicyMgtAgentType)

このルール グループと関連付けられたエージェント タイプを指定します。 エージェント タイプ オブジェクトを取得するには、
PolicyMgtSession->GetAgentType メソッドをコールします。

groupDesc (文字列)

(オプション) グループの説明を指定します。

戻り値

CreateRuleGroup メソッドは以下のいずれかの値を返します。

- PolicyMgtGroup オブジェクト
- コールに失敗した場合は、**undef**

DeleteGroup メソッド -- ドメインからグループを削除

DeleteGroup メソッドは、ドメイン内の指定されたグループを削除します。

構文

DeleteGroup メソッドの形式は以下のとおりです。

```
Netegrity::PolicyManagementDomain->DeleteGroup(group)
```

パラメータ

DeleteGroup メソッドは以下のパラメータを受け入れます。

group (PolicyMgrGroup)

削除するグループを指定します。

戻り値

DeleteGroup メソッドは以下のいずれかの値を返します。

- 成功した場合またはグループが見つからなかった場合は 0
- コールに失敗した場合は -1

DeletePolicy メソッド -- ポリシーを削除

DeletePolicy メソッドは、ポリシーを削除します。

構文

DeletePolicy メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtDomain->DeletePolicy(policy)
```

パラメータ

DeletePolicy メソッドは以下のパラメータを受け入れます。

policy (PolicyMgtPolicy)

削除するポリシーを指定します。

戻り値

DeletePolicy メソッドは以下のいずれかの値を返します。

- 成功の場合は 0
- コールに失敗した場合は -1

DeleteRealm メソッド -- ドメイン内のレルムを削除

DeleteRealm メソッドは、ドメイン内のトップレベル レルムを削除します。

構文

DeleteRealm メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtDomain->DeleteRealm(realm)
```

パラメータ

DeleteRealm メソッドは以下のパラメータを受け入れます。

realm (PolicyMgtRealm)

削除するレルムを指定します。

戻り値

DeleteRealm メソッドは以下のいずれかの値を返します。

- 成功した場合またはレルムが見つからなかった場合は 0
- コールに失敗した場合は -1

DeleteResponse メソッド -- レスポンスを削除

DeleteResponse メソッドは、レスポンスを削除します。

構文

DeleteResponse メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtDomain->DeleteResponse(response)
```

パラメータ

DeleteResponse メソッドは以下のパラメータを受け入れます。

response (PolicyMgtResponse)

削除するレスポンスを指定します。

戻り値

DeleteResponse メソッドは以下のいずれかの値を返します。

- 成功の場合は 0
- コールに失敗した場合は -1

Description メソッド --ドメインの説明を設定または取得

Description メソッドは、ドメインの説明を設定または取得します。

構文

Description メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtDomain->Description([domainDesc])
```

パラメータ

Description メソッドは以下のパラメータを受け入れます。

domainDesc (文字列)

(オプション) 設定する説明を指定します。

戻り値

Description メソッドは以下のいずれかの値を返します。

- 新規または既存のドメインの説明
- 失敗の場合は、空の文字列

GetAllPolicies メソッド -- ドメインと関連付けられたすべてのポリシーを取得

GetAllPolicies メソッドは、ドメインと関連付けられたすべてのポリシーを取得します。

構文

GetAllPolicies メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtDomain->GetAllPolicies( )
```

パラメータ

GetAllPolicies メソッドは、パラメータを受け入れません。

戻り値

GetAllPolicies メソッドは以下のいずれかの値を返します。

- PolicyMgtPolicy オブジェクトの配列
- 失敗した場合は、**undef**

GetAllRealms メソッド -- ドメイン内のすべてのトップレベル レalmを取得

GetAllRealms メソッドは、ドメイン内のすべてのトップレベル レalmを取得します。

構文

GetAllRealms メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtDomain->GetAllRealms( )
```

パラメータ

GetAllRealms メソッドは、パラメータを受け入れません。

戻り値

GetAllRealms メソッドは以下のいずれかの値を返します。

- PolicyMgtRealm オブジェクトの配列
- 失敗した場合は、**undef**

注釈

レルム下でトップレベル レルムをすべて取得するには、
`PolicyMgtRealm->GetAllChildRealms` メソッドをコールします。

GetAllResponseGroups メソッド -- ドメインと関連付けられたすべてのレスポンス グループを取得

`GetAllResponseGroups` メソッドは、ドメインと関連付けられたすべてのレスポンス グループを取得します。

構文

`GetAllResponseGroups` メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtDomain->GetAllResponseGroups( )
```

パラメータ

`GetAllResponseGroups` メソッドは、パラメータを受け入れません。

戻り値

`GetAllResponseGroups` メソッドが返すのは以下のいずれかの値

- `PolicyMgtGroup` オブジェクトの配列
- 失敗した場合は、**undef**

GetAllResponses メソッド -- ドメインと関連付けられたすべてのレスポンスを取得

`GetAllResponses` メソッドは、ドメインと関連付けられたすべてのレスポンスを取得します。

構文

`GetAllResponses` メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtDomain->GetAllResponses()
```

パラメータ

`GetAllResponses` メソッドは、パラメータを受け入れません。

戻り値

GetAllResponses メソッドが返すのは以下のいずれかの値

- PolicyMgtResponse オブジェクトの配列
- コールに失敗した場合は、**undef**

GetAllRuleGroups メソッド -- ドメインと関連付けられたすべてのルール グループを取得

GetAllRuleGroups メソッドは、ドメインと関連付けられたすべてのルールグループを取得します。

構文

GetAllRuleGroups メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtDomain->GetAllRuleGroups( )
```

パラメータ

GetAllRuleGroups メソッドは、パラメータを受け入れません。

戻り値

GetAllRuleGroups メソッドは以下のいずれかの値を返します。

- PolicyMgtGroup オブジェクトの配列
- コールに失敗した場合は、**undef**

GetPolicy メソッド -- ドメイン内のポリシーを取得

GetPolicy メソッドは、ドメイン内のポリシーを取得します。

構文

GetPolicy メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtDomain->GetPolicy(policyName)
```

パラメータ

GetPolicy メソッドは以下のパラメータを受け入れます。

policyName (文字列)

取得するポリシーを指定します。

戻り値

GetPolicy メソッドは以下のいずれかの値を返します。

- PolicyMgtPolicy オブジェクト
- コールに失敗する場合または指定されたポリシーが存在しない場合は、**undef**

GetRealm メソッド -- ドメイン内のトップレベル レalmを取得

GetRealm メソッドは、ドメイン内のトップレベル レalmを取得します。

構文

GetRealm メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtDomain->GetRealm(realmName)
```

パラメータ

GetRealm メソッドは以下のパラメータを受け入れます。

realmName (文字列)

取得するレalmを指定します。

戻り値

GetRealm メソッドは以下のいずれかの値を返します。

- PolicyMgtRealm オブジェクト
- コールに失敗する場合または指定されたレalmが存在しない場合は、**undef**

GetResponse メソッド -- ドメインと関連付けられたレスポンスを取得

GetResponse メソッドは、ドメインと関連付けられたレスポンスを取得します。

構文

GetResponse メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtDomain->GetResponse(resName)
```

パラメータ

GetResponse メソッドは以下のパラメータを受け入れます。

resName (文字列)

取得するレスポンスを指定します。

戻り値

GetResponse メソッドは以下のいずれかの値を返します。

- PolicyMgtResponse オブジェクト
- コールに失敗する場合または指定されたレスポンスが存在しない場合は、**undef**

GetResponseGroup メソッド -- 指定されたレスポンス グループを取得

GetResponseGroup メソッドは、指定されたレスポンス グループを取得します。

構文

GetResponseGroup メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtDomain->GetResponseGroup(groupName)
```

パラメータ

GetResponseGroup メソッドは以下のパラメータを受け入れます。

groupName (文字列)

取得するレスポンス グループの名前を指定します。

戻り値

GetResponseGroup メソッドは以下のいずれかの値を返します。

- PolicyMgtGroup オブジェクト
- コールに失敗した場合は、**undef**

GetRuleGroup メソッド -- 指定されたルール グループを取得

GetRuleGroup メソッドは、指定されたルール グループを取得します。

構文

GetRuleGroup メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtDomain->GetRuleGroup(groupName)
```

パラメータ

GetRuleGroup メソッドは以下のパラメータを受け入れます。

groupName (文字列)

取得するグループの名前を指定します。

戻り値

GetRuleGroup メソッドは以下のいずれかの値を返します。

- PolicyMgtGroup オブジェクト
- コールに失敗した場合は、**undef**

GetUserDirSearchOrder メソッド -- ドメインと関連付けられたユーザ ディレクトリ オブジェクトを取得

GetUserDirSearchOrder メソッドは、ドメインと関連付けられたユーザ ディレクトリ オブジェクトを取得します。

構文

GetUserDirSearchOrder メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtDomain->GetUserDirSearchOrder( )
```

パラメータ

GetUserDirSearchOrder メソッドは、パラメータを受け入れません。

戻り値

GetUserDirSearchOrder メソッドは以下のいずれかの値を返します。

- PolicyMgtUserDir オブジェクト
- コールに失敗した場合は、**undef**

注釈

返されるオブジェクトの順序は、ディレクトリのクエリ実行時に SiteMinder が使用する順序と同じです。検索順序を変更するには、PolicyMgtAffDomain から SetUserDirSearchOrder メソッドを呼び出します。

GlobalPoliciesApply メソッド -- グローバル ポリシーに対してドメインが有効かどうかを判定

GlobalPoliciesApply メソッドは、グローバル ポリシーに対してドメインが有効かどうかを示すフラグを設定または取得します。ドメインがグローバル ポリシーに対して有効な場合、グローバルおよびドメイン固有のポリシーをドメインに適用できます。

構文

GlobalPoliciesApply メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtDomain->GlobalPoliciesApply([globalFlag])
```

パラメータ

GlobalPoliciesApply メソッドは以下のパラメータを受け入れます。

globalFlag (int)

(オプション) グローバル ポリシーに対してドメインを有効にするかどうかを指定します。

- 1 は、グローバル ポリシーを有効にすることを指定します
- 0 は、グローバル ポリシーを無効にすることを指定します

戻り値

GlobalPoliciesApply メソッドは以下のいずれかの値を返します。

- 新規または既存のフラグ設定

Name メソッド -- ドメイン名を設定または取得

Name メソッドは、ドメイン名を設定または取得します。

構文

Name メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtDomain->Name([domainName])
```

パラメータ

Name メソッドは以下のパラメータを受け入れます。

domainName (文字列)

(オプション) ドメインに割り当てる名前を指定します。

戻り値

Name メソッドは以下のいずれかの値を返します。

- 新規または既存のドメイン名
- コールに失敗した場合は、**undef**

RemoveAdmin メソッド -- ドメインから管理者の関連付けを解除

RemoveAdmin メソッドは、ドメインから管理者の関連付けを解除します。

構文

RemoveAdmin メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtDomain->RemoveAdmin(admin)
```


パラメータ

RemoveAdmin メソッドは以下のパラメータを受け入れます。

admin (PolicyMgtAdmin)

ドメインから削除する管理者を指定します。

戻り値

RemoveAdmin メソッドは以下のいずれかの値を返します。

- 成功の場合は 0
- コールに失敗した場合は、-1

注釈

ポリシー ストアから管理者を削除するには、
PolicyMgtSession->DeleteAdmin メソッドも参照してください。

特定のドメインの管理者を作成するためにポリシー管理 API を使用することはできません。ただし、管理者が 管理 UI または PolicyMgtAffDomain->AddAdmin メソッドのいずれかによってドメインと関連付けられる場合は、RemoveAdmin メソッドをコールして、ドメインからその管理者を削除できます。

RemoveUserDir メソッド -- ドメインからユーザ ディレクトリの関連付けを解除

RemoveUserDir メソッドは、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。

構文

RemoveUserDir メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtDomain->RemoveUserDir(userDir)
```

パラメータ

RemoveUserDir メソッドは以下のパラメータを受け入れます。

userDir (PolicyMgtUserDir)

ドメインから関連付けを解除するユーザ ディレクトリを指定します。

戻り値

RemoveUserDir メソッドは以下のいずれかの値を返します。

- 成功の場合は 0
- コールに失敗した場合は、-1

SetUserDirSearchOrder-- ユーザ ディレクトリ オブジェクトの検索順序を再調整

SetUserDirSearchOrder メソッドは、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再調整します。

構文

SetUserDirSearchOrder メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtDomain->SetUserDirSearchOrder(dirArray)
```

パラメータ

SetUserDirSearchOrder メソッドは以下のパラメータを受け入れます。

dirArray ()

ユーザ ディレクトリ オブジェクトの配列の参照を指定します (たとえば、`¥@myarray`) 。

戻り値

SetUserDirSearchOrder メソッドは以下のいずれかの値を返します。

- PolicyMgtUserDir オブジェクトの配列
- コールに失敗した場合は、**undef**

CLI グループ メソッド

Add メソッド -- エージェント、レスポンス、ルールまたはネストされたグループ オブジェクトをグループに追加

Add メソッドは、エージェント、レスポンス、ルールまたはネストされたグループ オブジェクトをグループに追加します。

構文

Add メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtGroup->Add(newMember)
```

パラメータ

Add メソッドは以下のパラメータを受け入れます。

newMember (*objectType*)

グループに追加するメンバを指定します。 *objectType* は、以下のいずれかになります。

- PolicyMgtAgent
- PolicyMgtResponse
- PolicyMgtRule
- PolicyMgtGroup

戻り値

Add メソッドは以下のいずれかの値を返します。

- 成功の場合は 0
- コールに失敗した場合は、-1

Contains メソッド -- グループに指定されたエージェント、レスポンス、ルールまたはネストされたグループ オブジェクトが含まれるかどうかを判定

Contains メソッドは、グループに指定されたエージェント、レスポンス、ルールまたはネストされたグループ オブジェクトが含まれるかどうかを判定します。

構文

Contains メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtGroup->Contains(object)
```

パラメータ

Contains メソッドは以下のパラメータを受け入れます。

object (*objectType*)

確認するオブジェクトを指定します。 *objectType* は、以下のいずれかになります。

- PolicyMgtAgent
- PolicyMgtResponse
- PolicyMgtRule
- PolicyMgtGroup

戻り値

Contains メソッドは以下のいずれかの値を返します。

- グループに指定されたオブジェクトが含まれる場合は、1
- グループに指定されたオブジェクトが含まれない場合は、0
- コールに失敗した場合は、**undef**

Description メソッド -- グループ オブジェクトの説明を設定または取得

Description メソッドは、グループ オブジェクトの説明を設定または取得します。

構文

Description メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtGroup->Description([Description])
```

パラメータ

Description メソッドは以下のパラメータを受け入れます。

Description (文字列)

(オプション) 設定する説明を指定します。

戻り値

Description メソッドは以下のいずれかの値を返します。

- 新規または既存の説明
- コールに失敗した場合は、空の文字列

GetAgent メソッド -- グループから指定されたエージェント オブジェクトを取得

GetAgent メソッドは、グループから指定されたエージェント オブジェクトを取得します。

構文

GetAgent メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtGroup->GetAgent(agentName)
```

パラメータ

GetAgent メソッドは以下のパラメータを受け入れます。

agentName (文字列)

取得するエージェントの名前を指定します。

戻り値

GetAgent メソッドは以下のいずれかの値を返します。

- PolicyMgtAgent オブジェクト
- そのようなエージェントが見つからない場合、グループに別のタイプのオブジェクトが含まれる場合、またはコールが失敗した場合は、**undef**

GetAgentGroup メソッド -- グループ内のネストされたエージェント グループ オブジェクトを取得

GetAgentGroup メソッドは、グループ内のネストされたエージェント グループ オブジェクトを取得します。

構文

GetAgentGroup メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtGroup->GetAgentGroup(groupName)
```

パラメータ

GetAgentGroup メソッドは以下のパラメータを受け入れます。

groupName (文字列)

取得するエージェント グループの名前を指定します。

戻り値

GetAgentGroup メソッドは以下のいずれかの値を返します。

- PolicyMgtGroup オブジェクト
- コールに失敗した場合またはグループが存在しない場合は、**undef**

GetAgentType メソッド -- グループに含まれているエージェント オブジェクトのタイプを取得

GetAgentType メソッドは、グループに含まれているエージェント オブジェクトのタイプを取得します（たとえば、**Web** エージェント）。

構文

GetAgentType メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtGroup->GetAgentType( )
```

パラメータ

GetAgentType メソッドは、パラメータを受け入れません。

戻り値

GetAgentType メソッドは以下のいずれかの値を返します。

- PolicyMgtAgentType オブジェクト
- コールに失敗した場合は、**undef**

GetAllAgentGroups メソッド -- グループ内のネストされたすべてのエージェント グループ オブジェクトを取得

GetAllAgentGroups メソッドは、グループ内のネストされたすべてのエージェント グループ オブジェクトを取得します。

構文

GetAllAgentGroups メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtGroup->GetAllAgentGroups( )
```

パラメータ

GetAllAgentGroups メソッドは、パラメータを受け入れません。

戻り値

GetAllAgentGroups メソッドは以下のいずれかの値を返します。

- PolicyMgtGroup オブジェクトの配列
- コールに失敗した場合は、**undef**

GetAllAgents メソッド -- グループ内のすべてのエージェント オブジェクトを取得

GetAllAgents メソッドは、グループ内のすべてのエージェント オブジェクトを取得します。

構文

GetAllAgents メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtGroup->GetAllAgents( )
```

パラメータ

GetAllAgents メソッドは、パラメータを受け入れません。

戻り値

GetAllAgents メソッドは以下のいずれかの値を返します。

- PolicyMgtAgent オブジェクトの配列
- エージェントが見つからない場合、グループに別のタイプのオブジェクトが含まれる場合、またはコールが失敗した場合は、**undef**

GetAllResponseGroups メソッド -- グループ内のネストされたすべてのレスポンス グループ オブジェクトを取得

GetAllResponseGroups メソッドは、グループ内のネストされたすべてのレスポンス グループ オブジェクトを取得します。

構文

GetAllResponseGroups メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtGroup->GetAllResponseGroups( )
```


パラメータ

GetAllResponseGroups メソッドは、パラメータを受け入れません。

戻り値

GetAllResponseGroups メソッドは以下のいずれかの値を返します。

- PolicyMgtGroup オブジェクトの配列
- レスポンス グループが見つからない場合、グループに別のタイプのオブジェクトが含まれる場合、またはコールが失敗した場合は、**undef**

GetAllResponses メソッド -- グループ内のすべてのレスポンス オブジェクトを取得

GetAllResponses メソッドは、グループ内のすべてのレスポンス オブジェクトを取得します。

構文

GetAllResponses メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtGroup->GetAllResponses( )
```

パラメータ

GetAllResponses メソッドは、パラメータを受け入れません。

戻り値

GetAllResponses メソッドは以下のいずれかの値を返します。

- PolicyMgtResponse オブジェクトの配列
- レスポンス オブジェクトが見つからない場合、グループに別のタイプのオブジェクトが含まれる場合、またはコールが失敗した場合は、**undef**

GetAllRuleGroups メソッド -- グループ内のネストされたすべてのルール グループ オブジェクトを取得

GetAllRuleGroups メソッドは、グループ内のネストされたすべてのルール グループ オブジェクトを取得します。

構文

GetAllRuleGroups メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtGroup->GetAllRuleGroups( )
```

パラメータ

GetAllRuleGroups メソッドは、パラメータを受け入れません。

戻り値

GetAllRuleGroups メソッドは以下のいずれかの値を返します。

- PolicyMgtGroup オブジェクトの配列
- ルール グループが見つからない場合、グループに別のタイプのオブジェクトが含まれる場合、またはコールが失敗した場合は、**undef**

GetAllRules メソッド -- グループ内のすべてのルール オブジェクトを取得

GetAllRules メソッドは、グループ内のすべてのルール オブジェクトを取得します。

構文

GetAllRules メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtGroup->GetAllRules( )
```

パラメータ

GetAllRules メソッドは、パラメータを受け入れません。

戻り値

GetAllRules メソッドは以下のいずれかの値を返します。

- PolicyMgtRule オブジェクトの配列
- ルール オブジェクトが見つからない場合、グループに別のタイプのオブジェクトが含まれる場合、またはコールが失敗した場合は、**undef**

GetResponse メソッド -- グループから指定されたレスポンス オブジェクトを取得

GetResponse メソッドは、グループから指定されたレスポンス オブジェクトを取得します。

構文

GetResponse メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtGroup->GetResponse(responseName)
```

パラメータ

GetResponse メソッドは以下のパラメータを受け入れます。

responseName (タイプ)

取得するレスポンスの名前を指定します。

戻り値

GetResponse メソッドは以下のいずれかの値を返します。

- PolicyMgtResponse オブジェクト
- そのようなレスポンスが見つからない場合、グループに別のタイプのオブジェクトが含まれる場合、またはコールが失敗した場合は、**undef**

GetResponseGroup メソッド -- グループ内のネストされたレスポンス グループ オブジェクトを取得

GetResponseGroup メソッドは、グループ内のネストされたレスポンス グループ オブジェクトを取得します。

構文

GetResponseGroup メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtGroup->GetResponseGroup(groupName)
```

パラメータ

GetResponseGroup メソッドは以下のパラメータを受け入れます。

groupName (文字列)

取得するレスポンス グループの名前を指定します。

戻り値

GetResponseGroup メソッドは以下のいずれかの値を返します。

- PolicyMgtGroup オブジェクト
- グループが存在しない場合またはコールに失敗する場合は、**undef**

GetRule メソッド -- グループから指定されたルール オブジェクトを取得

GetRule メソッドは、グループから指定されたルール オブジェクトを取得します。

構文

GetRule メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtGroup->GetRule(ruleName)
```

パラメータ

GetRule メソッドは以下のパラメータを受け入れます。

ruleName (文字列)

取得するルールの名前を指定します。

戻り値

GetRule メソッドは以下のいずれかの値を返します。

- PolicyMgtRule オブジェクト
- そのようなルールが見つからない場合、グループに別のタイプのオブジェクトが含まれる場合、またはコールが失敗した場合は、**undef**

GetRuleGroup メソッド -- グループ内のネストされたルール グループ オブジェクトを取得

GetRuleGroup メソッドは、グループ内のネストされたルール グループ オブジェクトを取得します。

構文

GetRuleGroup メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtGroup->GetRuleGroup(groupName)
```

パラメータ

GetRuleGroup メソッドは以下のパラメータを受け入れます。

groupName (文字列)

取得するルール グループの名前を指定します。

戻り値

GetRuleGroup メソッドは以下のいずれかの値を返します。

- PolicyMgtGroup オブジェクト
- グループが存在しない場合またはコールに失敗する場合は、**undef**

Name メソッド -- グループ オブジェクトの名前を設定または取得

Name メソッドは、グループ オブジェクトの名前を設定または取得します。

構文

Name メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtGroup->Name([Name])
```

パラメータ

Name メソッドは以下のパラメータを受け入れます。

Name (文字列)

(オプション) 設定する名前を指定します。

戻り値

Name メソッドは以下のいずれかの値を返します。

- 新規または既存の名前
- コールに失敗した場合は、**undef**

Remove メソッド -- グループから指定されたグループ メンバを削除

Remove メソッドは、グループから指定されたグループ メンバを削除します。

構文

Remove メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtGroup->Remove(member)
```

パラメータ

Remove メソッドは、以下のパラメータを受け入れます。

member (*objectType*)

以下のいずれかのオブジェクト タイプである削除対象のグループメンバを指定します。

- PolicyMgtAgent
- PolicyMgtResponse
- PolicyMgtRule
- PolicyMgtGroup

戻り値

Remove メソッドは以下のいずれかの値を返します。

- 成功の場合は 0
- コールに失敗した場合は、**undef**

CLI ホスト設定メソッド

AddCluster メソッド -- ホスト設定に空のクラスタを追加

AddCluster メソッドは、ホスト設定に空のクラスタを追加します。
AddServer メソッドをコールして、クラスタにサーバを追加します。

構文

AddCluster メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtHostConfig->AddCluster( )
```

パラメータ

AddCluster メソッドは、パラメータを受け入れません。

戻り値

AddCluster メソッドは以下のいずれかの値を返します。

- 空の PolicyMgtCluster オブジェクト
- コールに失敗した場合は、**undef**

注釈

ホスト設定内のクラスタはクラスタ配列で参照されます。クラスタを追加すると、クラスタがクラスタ配列の終わりに追加されます。ホスト設定オブジェクトにクラスタを追加する順序によって、フェールオーバーシーケンスが決定されます。追加する最初のクラスタ（つまりクラスタ配列内の最初のクラスタ）は、プライマリ クラスタです。これは、SiteMinder が要求を送信するフェールオーバーシーケンスの最初のクラスタです。プライマリ クラスタに使用できる十分なサーバがないと（つまり、クラスタ内の利用可能なサーバの数がフェールオーバーしきい値より低い場合）、次のクラスタ（ホスト設定オブジェクトに追加された 2 番目のクラスタ）へのフェールオーバーが行われます。そのクラスタも失敗する場合、ホスト設定オブジェクトに追加された 3 番目のクラスタへのフェールオーバーが行われます（以後同じ要領）。

AddServer メソッド -- ホスト設定に非クラスタ化サーバを追加

AddServer メソッドは、ホスト設定に非クラスタ化サーバを追加します。

構文

AddServer メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtHostConfig->AddServer(Host [, AcctPort] [, AuthPort] [, AzPort])
```

パラメータ

AddServer メソッドは、以下のパラメータを受け入れます。

Host（文字列）

ポリシー サーバの IP アドレスを指定します。

AcctPort（文字列）

（オプション）アカウントिंग サーバの IP ポートを指定します。

AuthPort (文字列)

(オプション) 認証サーバの IP ポートを指定します。

AzPort (文字列)

(オプション) 許可サーバの IP ポートを指定します。

戻り値

AddServer メソッドは以下のいずれかの値を返します。

- 成功の場合は 0
- 失敗の場合は -1

注釈

SiteMinder v6.0 に導入された単一プロセス ポリシー サーバが、もともとは別個の認証、許可およびアカウンティングプロセスを 1 つの結合プロセスに組み合わせることにより、複数の要求が 1 つの TCP ポートを通ります。その結果、引数 AcctPort、AuthPort および AzPort はすべて同じポート番号を参照します。3 つの引数は下位互換性のために維持されます。

クラスタにサーバを追加するには、PolicyMgtCluster->AddServer メソッドをコールします。

Description メソッド -- ホスト設定オブジェクトの説明を設定または取得

Description メソッドは、ホスト設定オブジェクトの説明を設定または取得します。

構文

Description メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtHostConfig->Description([Description])
```

パラメータ

メソッドは以下のパラメータを受け入れます。

Description (文字列)

(オプション) 設定する説明を指定します。

戻り値

Description メソッドは以下のいずれかの値を返します。

- 新規または既存の説明
- コールに失敗した場合は、**undef**

EnableFailover メソッド -- フェールオーバー有効化フラグを設定または取得

EnableFailover メソッドは、フェールオーバー有効化フラグを設定または取得します。このフラグは、エージェントとポリシーサーバの通信をフェールオーバーまたはラウンドロビンによって行うかどうかを判定します。

構文

EnableFailover メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtHostConfig->EnableFailover([EnableFailover])
```

パラメータ

EnableFailover メソッドは以下のパラメータを受け入れます。

EnableFailover (int)

(オプション) 設定するフラグの値を指定します。

戻り値

EnableFailover メソッドは以下のいずれかの値を返します。

- 新規または既存のフラグ設定：
 - フェールオーバーの場合は 1
 - ラウンドロビンの場合は 0
- コールに失敗する場合は、-1

FailoverThreshold メソッド -- フェールオーバーしきい値パーセンテージを設定または取得

FailoverThreshold メソッドは、ホスト設定でのクラスタのフェールオーバーしきい値パーセンテージを設定または取得します。

構文

FailoverThreshold メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtHostConfig->FailoverThreshold([FailoverThreshold])
```

パラメータ

FailoverThreshold メソッドは以下のパラメータを受け入れます。

FailoverThreshold (int)

(オプション) 設定するフェールオーバーしきい値パーセンテージを指定します。

戻り値

FailoverThreshold メソッドは以下のいずれかの値を返します。

- 新規または既存のフェールオーバーしきい値パーセンテージ
- コールに失敗した場合は、**undef**

注釈

しきい値パーセンテージは、要求に対して利用可能である必要があるクラスタ内のサーバの最小数を表します。利用可能なサーバの数がしきい値以下になると、次のクラスタへのフェールオーバーが行われます。パーセンテージが表すサーバの数を判定するには、しきい値パーセンテージにクラスタ内のサーバ数を乗算して、次の最も高い整数まで丸めます。例：

- 5つのサーバのクラスタの 60 パーセント フェールオーバーしきい値では、クラスタ内の利用可能なサーバの数が 3 を下回ると、次のクラスタへのフェールオーバーが行われます。
- 同じクラスタの 61 パーセント フェールオーバーしきい値では、クラスタ内の利用可能なサーバの数が 4 を下回ると、フェールオーバーが行われます。

GetAllClusters メソッド -- ポリシー管理クラスタ オブジェクトの配列を取得

GetAllClusters メソッドは、ポリシー管理クラスタ オブジェクトの配列を取得します。

構文

GetAllClusters メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtHostConfig->GetAllClusters()
```

パラメータ

GetAllClusters メソッドは、パラメータを受け入れません。

戻り値

GetAllClusters メソッドは以下のいずれかの値を返します。

- PolicyMgtCluster オブジェクトの配列
- コールに失敗した場合は、**undef**

GetAllServers メソッド -- 非クラスタ化サーバ オブジェクトの配列を取得

GetAllServers メソッドは、ホスト設定の非クラスタ化サーバ オブジェクトの配列を取得します。

構文

GetAllServers メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtHostConfig->GetAllServers( )
```

パラメータ

GetAllServers メソッドは、パラメータを受け入れません。

戻り値

GetAllServers メソッドは以下のいずれかの値を返します。

- PolicyMgtServer オブジェクトの配列
- サーバオブジェクトが見つからない場合またはコールに失敗した場合は、**undef**

注釈

クラスタのメンバであるサーバを取得するには、
PolicyMgtCluster->GetAllServers メソッドをコールします。

MaxSocketsPerPort メソッド -- TCP/IP ソケットの最大数を設定または取得

MaxSocketsPerPort メソッドは、エージェントとポリシー サーバ間で開くことができる TCP/IP ソケットの最大数を設定または取得します。

構文

MaxSocketsPerPort メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtHostConfig->MaxSocketsPerPort([MaxSocketsPerPort])
```

パラメータ

MaxSocketsPerPort メソッドは以下のパラメータを受け入れます。

MaxSocketsPerPort (int)

(オプション) ポート当たりのソケットの新しい最大数を指定します。

戻り値

MaxSocketsPerPort メソッドは以下のいずれかの値を返します。

- ソケットの最大数を示す新規または既存の設定
- コールに失敗する場合は、-1

MinSocketsPerPort メソッド -- TCP/IP ソケットの最小数を設定または取得

MinSocketsPerPort メソッドは、エージェントとポリシー サーバ間で開く必要がある TCP/IP ソケットの最小数を設定または取得します。

構文

MinSocketsPerPort メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtHostConfig->MinSocketsPerPort([MinSocketsPerPort])
```

パラメータ

MinSocketsPerPort メソッドは以下のパラメータを受け入れます。

MinSocketsPerPort (int)

(オプション) 新しい最小ソケット値を指定します。

戻り値

MinSocketsPerPort メソッドは以下のいずれかの値を返します。

- ソケットの最小数を示す新規または既存の設定
- コールに失敗する場合は、-1

Name メソッド -- ホスト設定オブジェクトの名前を設定または取得

Description メソッドは、ホスト設定オブジェクトの名前を設定または取得します。

構文

Name メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtHostConfig->Name([Name])
```

パラメータ

Name メソッドは以下のパラメータを受け入れます。

Name (文字列)

(オプション) 設定する名前を指定します。

戻り値

Name メソッドは以下のいずれかの値を返します。

- 新規または既存の名前
- コールに失敗した場合は、**undef**

NewSocketStep メソッド -- ホスト設定の新しいソケット ステップ値を設定または取得

NewSocketStep メソッドは、ホスト設定の新しいソケット ステップ値を設定または取得します。この値は、需要が増大したときにエージェントとポリシー サーバ間で開く必要がある TCP/IP ソケットの増分数を表します。

構文

NewSocketStep メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtHostConfig->NewSocketStep([NewSocketStep])
```

パラメータ

NewSocketStep メソッドは以下のパラメータを受け入れます。

NewSocketStep (int)

(オプション) 設定する新しいソケット ステップ値を指定します。

戻り値

NewSocketStep メソッドは以下のいずれかの値を返します。

- 新規または既存のソケット ステップ値
- コールに失敗する場合は、**-1**

RemoveAllClusters メソッド -- このホスト設定と関連付けられたすべてのクラスタオブジェクトを削除

RemoveAllClusters メソッドは、このホスト設定と関連付けられたすべてのクラスタ オブジェクトを削除します。

構文

RemoveAllClusters メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtHostConfig->RemoveAllClusters()
```

パラメータ

RemoveAllClusters メソッドは、パラメータを受け入れません。

戻り値

RemoveAllClusters メソッドは以下のいずれかの値を返します。

- コールに成功する場合は、0
- コールに失敗する場合は、-1

RemoveAllServers メソッド -- ホスト設定からすべての非クラスタ化ポリシー サーバオブジェクトを削除

RemoveAllServers メソッドは、ホスト設定からすべての非クラスタ化 PolicyMgtServer オブジェクトを削除します。

構文

RemoveAllServers メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtHostConfig->RemoveAllServers()
```

パラメータ

RemoveAllServers メソッドは、パラメータを受け入れません。

戻り値

RemoveAllServers メソッドは以下のいずれかの値を返します。

- コールに成功する場合は、0
- コールに失敗する場合は、-1

RequestTimeout メソッド -- 要求タイムアウト値を設定または取得

RequestTimeout メソッドは、秒単位の要求タイムアウト値を設定または取得します。この値は、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の長さを表します。

構文

RequestTimeout メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtHostConfig->RequestTimeout([RequestTimeout])
```

パラメータ

RequestTimeout メソッドは以下のパラメータを受け入れます。

RequestTimeout (int)

(オプション) 設定する新しいタイムアウト値を指定します。

戻り値

RequestTimeout メソッドは以下のいずれかの値を返します。

- 新規または既存のタイムアウト値
- コールに失敗する場合は、-1

CLI 初期化メソッド

CreateSession メソッド -- ポリシー サーバ セッションを作成

CreateSession メソッドは、ポリシー サーバセッションを作成します。ポリシー サーバオブジェクトを操作するには、その前にセッションが必要です。必要な初期化およびログ記録はすべてこの段階で実行されます。

構文

CreateSession メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAPI->CreateSession(username, userpwd[, clientIP])
```

パラメータ

CreateSession メソッドは、以下のパラメータを受け入れます。

username (文字列)

管理者のログイン ID を指定します。

userpwd (文字列)

管理者のパスワードを指定します。

clientIP (文字列)

(オプション) ローカル マシンの IP アドレスを指定します。

CreateSession メソッドは以下のいずれかの値を返します。

- PolicyMgtSession オブジェクト
- コールに失敗した場合は、**undef**

DisableAudit メソッド -- 監査を有効または無効にするフラグを設定

DisableAudit メソッドは、監査を有効または無効にするフラグを設定します。

構文

DisableAudit メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAPI->DisableAudit([auditFlag])
```

パラメータ

DisableAudit メソッドは以下のパラメータを受け入れます。

auditFlag (int)

(オプション) フラグを設定する値を指定します。

- 監査を有効にするには 0
- 監査を無効にするには 1

戻り値

DisableAudit メソッドは以下のいずれかの値を返します。

- 引数が指定されない場合は、既存の有効化状態（0 または 1）。
- フラグ値がメソッドに渡される場合は、新しい有効化状態。

注釈

以下の操作の有効化状態を読み込みまたは設定します。

- 認証、許可および管理アクティビティなどの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。
- ユーザセッションの監視。

デフォルト状態は有効化です。有効化状態は新しいセッションの開始時にデフォルトに戻ります。

PolicyMgtAPI->CreateSession メソッドをコールした後に、有効化状態の設定を試行しても効果はありません。

DisableCacheUpdates メソッド -- 非推奨

DisbleCacheUpdates メソッドは SiteMinder v6.0 で非推奨です。このメソッドの影響を受けたキャッシュは自動的に有効にされます。

DisableManagementWatchDog メソッド -- SiteMinder 管理ウォッチドッグの有効化状態を読み込みまたは設定

DisableManagementWatchDog メソッド -- SiteMinder 管理ウォッチドッグの有効化状態を読み込みまたは設定します。

注: ウォッチドッグは内部的に使用され無効にはしません。

構文

DisableManagementWatchdog メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAPI->DisableManagementWatchDog([watchDogFlag])
```

パラメータ

DisableManagementWatchdog メソッドは以下のパラメータを受け入れます。

watchDogFlag (int)

(オプション) 設定するフラグの値を指定します。

- WatchDog を有効にするには 0
- WatchDog を無効にするには 1

戻り値

DisableManagementWatchdog メソッドは以下のいずれかの値を返します。

- 引数が指定されない場合は、既存の有効化状態（0 または 1）。
- フラグ値がメソッドに渡される場合は、新しい有効化状態。

注釈

デフォルト状態は有効化です。有効化状態は新しいセッションの開始時にデフォルトに戻ります。

PolicyMgtAPI->CreateSession をコールした後に、有効化状態の設定を試行しても効果はありません。

DisableValidation メソッド -- ポリシー サーバ オブジェクトの検証の有効化状態を読み込みまたは設定

DisableValidation メソッドは、ポリシー サーバ オブジェクトの検証に関して有効化状態を読み込みまたは設定します。

構文

DisableValidation メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAPI->DisableValidation([validationFlag])
```

パラメータ

DisableValidation メソッドは以下のパラメータを受け入れます。

validationFlag (int)

(オプション) フラグを設定する値を指定します。

- 検証を有効にするには 0
- 検証を無効にするには 1

戻り値

DisableValidation メソッドは以下のいずれかの値を返します。

- 引数が指定されない場合は、既存の有効化状態 (0 または 1) 。
- フラグ値がメソッドに渡される場合は、新しい有効化状態。

注釈

デフォルト状態は有効化です。有効化状態は新しいセッションの開始時にデフォルトに戻ります。

PolicyMgtAPI->CreateSession メソッドをコールした後に、有効化状態の設定を試行しても効果はありません。

EnableCache メソッド -- 非推奨

EnableCache メソッドは SiteMinder v6.0 で非推奨です。このリリースからは、このメソッドの影響を受けたキャッシュは自動的に有効にされます。

LoadAgentTypeDictionary メソッド -- エージェント タイプ ディクショナリの有効化状態を読み込みまたは設定

LoadAgentTypeDirectory メソッドは、ポリシー サーバによるエージェント タイプ ディクショナリのロードの有効化状態を読み込みまたは設定します。

構文

LoadAgentTypeDirectory メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAPI->LoadAgentTypeDictionary([loadFlag])
```

パラメータ

LoadAgentTypeDirectory メソッドは以下のパラメータを受け入れます。

loadFlag (int)

(オプション) フラグを設定する値を指定します。

エージェント タイプ ディクショナリのロードを無効にするには **0**

エージェント タイプ ディクショナリのロードを有効にするには **1**

戻り値

LoadAgentTypeDirectory メソッドは以下のいずれかの値を返します。

- 引数が指定されない場合は、既存の有効化状態 (0 または 1)。
- フラグ値がメソッドに渡される場合は、新しい有効化状態。

注釈

デフォルト状態は無効化です。有効化状態は新しいセッションの開始時にデフォルトに戻ります。

PolicyMgtAPI->CreateSession メソッドをコールした後に、有効化状態の設定を試行しても効果はありません。

New メソッド -- ポリシー管理 API のコンストラクタ

New メソッドは、ポリシー管理 API のコンストラクタです。このメソッドは、ポリシー管理 API を使用する前にコールする必要があります。

構文

New メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAPI->New( )
```

パラメータ

New メソッドは、パラメータを受け入れません。

戻り値

New メソッドは以下のいずれかの値を返します。

- PolicyMgtAPI オブジェクト
- コールに失敗した場合は、**undef**

PreLoadCache メソッド -- キャッシュの事前ロードの有効化状態を読み込みまたは設定

PreLoadCache メソッドは、ポリシー サーバによるキャッシュの事前ロードの有効化状態を読み込みまたは設定します。

構文

PreLoadCache メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAPI->PreLoadCache([cacheFlag])
```

パラメータ

PreLoadCache メソッドは以下のパラメータを受け入れます。

cacheFlag (int)

(オプション) フラグを設定する値を指定します。

- 0 はキャッシュ事前ロードを無効にします
- 1 はキャッシュ事前ロードを有効にします

戻り値

PreLoadCache メソッドは以下のいずれかの値を返します。

- 引数が指定されない場合は、既存の有効化状態（0 または 1）。
- フラグ値がメソッドに渡される場合は、新しい有効化状態。

注釈

デフォルト状態は無効化です。有効化状態は新しいセッションの開始時にデフォルトに戻ります。

PolicyMgtAPI->CreateSession メソッドをコールした後に、有効化状態の設定を試行しても効果はありません。

注: このフラグを無効にすると、ポリシー管理スクリプトを使用してポリシーストアで変更を加えるのにかかる時間を短縮できます。

PrintDebugTrace メソッド -- デバッグ(トレース)情報のサンプルの出力を有効化または無効化

PrintDebugTrace メソッドは、コンソールへのデバッグ（トレース）情報の出力を有効化または無効化します。

構文

PrintDebugTrace メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtAPI->PrintDebugTrace([debugFlag])
```

パラメータ

PrintDebugTrace メソッドは以下のパラメータを受け入れます。

debugFlag (int)

(オプション) フラグを設定する値を指定します。

- 0 はトレース出力を無効にします
- 1 はトレース出力を有効にします

戻り値

PrintDebugTrace メソッドは以下のいずれかの値を返します。

- トレース出力が無効な場合は、0
- トレース出力が有効な場合は、1

CLI IP 設定メソッド

GetEndIPAddress メソッド -- 終了 IP アドレスを取得

GetEndIPAddress メソッドは、IP アドレス範囲の終了 IP アドレスを取得します。

構文

GetEndIPAddress メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtIPConfig->GetEndIPAddress( )
```

パラメータ

GetEndIPAddress メソッドは、パラメータを受け入れません。

戻り値

GetEndIPAddress メソッドは以下のいずれかの値を返します。

- 受け入れられた IP アドレス範囲内の終了 IP アドレス。
- コールに失敗した場合は、**undef**

注釈

詳細については、メソッド [PolicyMgtAffiliate->CreateIPConfigRange](#) (P. 200) を参照してください。

GetHostName メソッド -- ホスト名 IP アドレス制限と関連付けられたホスト名を取得

GetHostName メソッドは、ホスト名 IP アドレス制限と関連付けられたホスト名を取得します。

構文

GetHostName メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtIPConfig->GetHostName()
```

パラメータ

GetHostName メソッドは、パラメータを受け入れません。

戻り値

GetHostName メソッドは以下のいずれかの値を返します。

- 要求が発行されるマシンのホスト名
- コールに失敗した場合は、**undef**

注釈

詳細については、メソッド [PolicyMgtAffiliate->CreateIPConfigHostName](#) (P. 199) を参照してください。

GetIPAddress メソッド -- IP アドレス制限の IP アドレスを取得

GetIPAddress メソッドは、以下のように IP アドレス制限の IP アドレスを取得します。

- IPCFG_TYPE_SINGLEHOST IP アドレスの場合は、ポリシーを起動させるために要求を発行する必要があるマシンの IP アドレスを取得します。詳細については、[PolicyMgtAffiliate->CreateIPConfigSingleHost](#) (P. 201) メソッドの説明を参照してください。
- IPCFG_TYPE_RANGE IP アドレス制限の場合は、受け入れられた IP アドレスの範囲内の開始 IP アドレスを取得します。詳細については、[PolicyMgtAffiliate->CreateIPConfigRange](#) (P. 200) メソッドの説明を参照してください。

IP アドレス制限のタイプを判定するには、GetType メソッドをコールします。

構文

GetIPAddress メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtIPConfig->GetIPAddress()
```

パラメータ

GetIPAddress メソッドは、パラメータを受け入れません。

戻り値

GetIPAddress メソッドは以下のいずれかの値を返します。

- 要求を発行する必要がある IP アドレス、または受け入れられた IP アドレスの範囲内の開始アドレス。
- コールに失敗した場合は、**undef**

GetSubnetMask メソッド -- サブネット アドレス用のサブネット マスクを取得

GetSubnetMask メソッドは、指定されたサブネット マスクおよび IP アドレスから引き出されるサブネット アドレス用のサブネット マスクを取得します。

構文

GetSubnetMask メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtIPConfig->GetSubnetMask( )
```

パラメータ

GetSubnetMask メソッドは、パラメータを受け入れません。

戻り値

GetSubnetMask メソッドは以下のいずれかの値を返します。

- サブネット マスク
- コールに失敗した場合は、**undef**

注釈

詳細については、[PolicyMgtPolicy->CreateIPConfigSubnetMask](#) (P. 369) メソッドの説明を参照してください。

GetType メソッド -- IP アドレス制限のタイプを取得

GetType メソッドは、IP アドレス制限のタイプを取得します。IP アドレス制限は、考慮できるように要求を発行する必要がある場所を指定します。

構文

GetType メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtIPConfig->GetType()
```

パラメータ

GetType メソッドは、パラメータを受け入れません。

戻り値

GetType メソッドは以下のいずれかの値を返します。

- IPCFG_TYPE_SINGLEHOST (値=1)。要求は指定された IP アドレスから行う必要があります。このタイプの IP アドレス制限は、PolicyMgtAffiliate->CreateIPConfigSingleHost メソッドで作成されます。
- IPCFG_TYPE_HOSTNAME (値=2)。要求は特定のホスト名を持つマシンから行う必要があります。このタイプの IP アドレス制限は、PolicyMgtAffiliate->CreateIPConfigHostName メソッドで作成されます。
- IPCFG_TYPE_SUBNETMASK (値=3)。要求は指定された IP アドレスから行われる必要があります。このタイプの IP アドレス制限は、PolicyMgtPolicy->CreateIPConfigSubnetMask メソッドで作成されます。
- IPCFG_TYPE_RANGE (値=4)。要求は IP アドレスの範囲から行う必要があります。このタイプの IP アドレス制限は、PolicyMgtAffiliate->CreateIPConfigRange メソッドで作成されます。
- コールに失敗した場合は、**undef**

CLI ODBC クエリ方式メソッド

Description メソッド -- ODBC クエリ方式の説明を設定または取得

Description メソッドは、ODBC クエリ方式の説明を設定または取得します。

構文

Description メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtODBCQueryScheme->Description([schemeDesc])
```

パラメータ

Description メソッドは以下のパラメータを受け入れます。

schemeDesc (文字列)

(オプション) ODBC クエリ方式の説明を指定します。

戻り値

Description メソッドは以下のいずれかの値を返します。

- 新規または既存の ODBC クエリ方式の説明
- コールに失敗した場合は、空

Name メソッド -- ODBC クエリ方式名を設定または取得

Name メソッドは、ODBC クエリ方式名を設定または取得します。

構文

Name メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtODBCQueryScheme->Name([schemeName])
```

パラメータ

Name メソッドは以下のパラメータを受け入れます。

schemeName (文字列)

ODBC クエリ方式名を指定します。

戻り値

Name メソッドは以下のいずれかの値を返します。

- 新規または既存の ODBC クエリ方式名
- コールに失敗した場合は、**undef**

QueryAuthenticateUser メソッド -- ユーザのパスワードをフェッチするクエリを設定または取得

QueryAuthenticateUser メソッドは、ユーザのパスワードをフェッチするクエリを設定または取得します。

構文

QueryAuthenticateUser メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtODBCQueryScheme->QueryAuthenticateUser([queryAuthUser])
```

パラメータ

QueryAuthenticateUser メソッドは以下のパラメータを受け入れます。

queryAuthUser (文字列)

(オプション) ユーザのパスワードをフェッチするクエリを指定します。

戻り値

QueryAuthenticateUser メソッドは以下のいずれかの値を返します。

- 新規または既存のクエリ
- コールに失敗した場合は、**undef**

注釈

サンプルクエリ (SiteMinder サンプル データベース スキーマ SmSampleUsers に基づく)。%s 表現は、以下のように、クエリが実行されるときに SiteMinder によって提供されるユーザの名前パラメータのプレースホルダです。

```
select Name from SmUser where Name = '%s' and Password = '%s'
```

Oracle データベース用のクエリ方式を設定して、Oracle の暗号化パスワード機能を使用している場合は、クエリ文字列全体を **connect** という語に置き換えます。このクエリに **connect** という語を使用すると、ユーザ名とパスワードは Oracle の暗号化パスワード機能によって評価されることが SiteMinder に通知されます。

QueryEnumerate メソッド -- ユーザ オブジェクトの名前をリストするクエリを設定または取得

QueryEnumerate メソッドは、ディレクトリ内のユーザ オブジェクトの名前をリストするクエリを設定または取得します。

構文

QueryEnumerate メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtODBCQueryScheme->QueryEnumerate([queryEnumerate])
```

パラメータ

QueryEnumerate メソッドは以下のパラメータを受け入れます。

queryEnumerate (文字列)

(オプション) ディレクトリ内のユーザ オブジェクトの名前をリストするクエリを指定します。

戻り値

QueryEnumerate メソッドは以下のいずれかの値を返します。

- 新規または既存のクエリ
- コールに失敗した場合は、**undef**

注釈

以下のようなサンプルクエリ (SiteMinder サンプル データベース スキーマ SmSampleUsers に基づく)。

```
select Name, 'Group' as Class from SmGroup order by Class
```

QueryGetGroupProp メソッド -- グループ プロパティの値をフェッチするクエリを設定または取得

QueryGetGroupProp メソッドは、グループ プロパティの値をフェッチするクエリを設定または取得します。プロパティは QueryGetGroupProps メソッドによって指定されたプロパティの 1 つである必要があります。

構文

QueryGetGroupProp メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtODBCQueryScheme->QueryGetGroupProp([queryGetGroupProp])
```

パラメータ

QueryGetGroupProp メソッドは以下のパラメータを受け入れます。

queryGetGroupProp (文字列)

(オプション) グループ プロパティをフェッチするクエリを指定します。

戻り値

QueryGetGroupProp メソッドは以下のいずれかの値を返します。

- 新規または既存のクエリ
- コールに失敗した場合は、**undef**

注釈

サンプル クエリ（SiteMinder サンプル データベース スキーマ SmSampleUsers に基づく）。%s 表現は、以下のように、クエリが実行されるときに SiteMinder によって提供されるプロパティ名およびグループ名パラメータのプレースホルダです。

```
select %s from SmGroup where Name = '%s'
```

QueryGetGroupProps メソッド -- グループ プロパティのリストを設定または取得

QueryGetGroupProps メソッドは、グループ プロパティのカンマ区切りリストを設定または取得します。これらの属性はグループの内容を検索する場合や、ポリシーをグループ属性に結び付ける場合に使用します。これらの属性をグループ名と同じテーブルに配置することをお勧めします。

構文

QueryGetGroupProps メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtODBCQueryScheme->QueryGetGroupProps([queryGetGroupProps])
```

パラメータ

QueryGetGroupProps メソッドは、以下のパラメータを受け入れます。

queryGetGroupProps（文字列）

（オプション）グループ プロパティのカンマ区切りリストを指定します。

戻り値

QueryGetGroupProps メソッドは以下のいずれかの値を返します。

- 新規または既存のグループ プロパティ リスト
- コールに失敗した場合は、**undef**

注釈

以下のサンプル リスト。

Name, GroupId

QueryGetGroups メソッド -- ユーザがメンバであるグループの名前をフェッチするクエリを設定または取得

QueryGetGroups メソッドは、ユーザがメンバであるグループの名前をフェッチするクエリを設定または取得します。

構文

QueryGetGroups メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtODBCQueryScheme->QueryGetGroups([queryGetGroups])
```

パラメータ

QueryGetGroups メソッドは、以下のパラメータを受け入れます。

queryGetGroups (文字列)

(オプション) ユーザ グループの名前をフェッチするクエリを指定します。

戻り値

QueryGetGroups メソッドは以下のいずれかの値を返します。

- 新規または既存のクエリ
- コールに失敗した場合は、**undef**

注釈

サンプル クエリ (SiteMinder サンプル データベース スキーマ SmSampleUsers に基づく)。 %s 表現は、以下のように、クエリが実行されるときに SiteMinder によって提供されるユーザ名パラメータのプレースホルダです。

```
select SmGroup.Name from SmGroup, SmUser, SmUserGroup where SmUser.Name = '%s' and  
SmUser.UserId = SmUserGroup.UserId and SmGroup.GroupId = SmUserGroup.GroupId
```

QueryGetObjInfo メソッド -- オブジェクトのクラスをフェッチするクエリを設定または取得

QueryGetObjInfo メソッドは、オブジェクトのクラスをフェッチするクエリを設定または取得します。

構文

QueryGetObjInfo メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtODBCQueryScheme->QueryGetObjInfo([queryGetObjInfo])
```

パラメータ

QueryGetObjInfo メソッドは以下のパラメータを受け入れます。

queryGetObjInfo (文字列)

(オプション) オブジェクトのクラスをフェッチするクエリを指定します。

戻り値

QueryGetObjInfo メソッドは以下のいずれかの値を返します。

- 新規または既存のクエリ
- コールに失敗した場合は、**undef**

注釈

サンプルクエリ (SiteMinder サンプルデータベース スキーマ SmSampleUsers に基づく)。%s 表現は、以下のように、クエリが実行されるときに SiteMinder によって提供されるユーザまたはグループ オブジェクト名のプレースホルダです。

```
select Name, 'User' from SmUser where Name = '%s' Union select Name, 'Group' from SmGroup where Name = '%s'
```

QueryGetUserProp メソッド -- ユーザ プロパティの値をフェッチするクエリを設定または取得

QueryGetUserProp メソッドは、ユーザ プロパティの値をフェッチするクエリを設定または取得します。プロパティは PolicyMgtODBCQueryScheme->QueryGetUserProps メソッドによって指定されたプロパティの 1 つである必要があります。

構文

QueryGetUserProp メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtODBCQueryScheme->QueryGetUserProp([queryGetUserProp])
```

パラメータ

QueryGetUserProp メソッドは以下のパラメータを受け入れます。

queryGetUserProp (文字列)

(オプション) ユーザ プロパティをフェッチするクエリを指定します。

戻り値

QueryGetUserProp メソッドは以下のいずれかの値を返します。

- 新規または既存のクエリ
- コールに失敗した場合は、**undef**

注釈

サンプル クエリ (SiteMinder サンプル データベース スキーマ SmSampleUsers に基づく)。%s 表現は、以下のように、クエリが実行されるときに SiteMinder によって提供されるプロパティ名およびユーザ名パラメータのプレースホルダです。

```
select %s from SmUser where Name = '%s'
```

QueryGetUserProps メソッド -- ユーザ プロパティのリストを設定または取得

QueryGetUserProps メソッドは、ユーザ プロパティのカンマ区切りリストを設定または取得します。プロパティはユーザ名と同じテーブルに存在します。

構文

QueryGetUserProps メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtODBCQueryScheme->QueryGetUserProps([queryGetUserProps])
```

パラメータ

QueryGetUserProps メソッドは以下のパラメータを受け入れます。

queryGetUserProps (文字列)

(オプション) ユーザ プロパティのカンマ区切りリストを指定します。

戻り値

QueryGetUserProps メソッドは以下のいずれかの値を返します。

- 新規または既存のユーザ プロパティ リスト
- コールに失敗した場合は、**undef**

注釈

以下のサンプル リスト。

```
Name, UserId, FirstName, LastName, TelephoneNumber, EmailAddress, PIN, Mileage, Disabled
```

QueryInitUser メソッド -- データベースにユーザが存在するかどうかを判定するクエリを設定または取得

QueryInitUser メソッドは、データベースに特定のユーザが存在するかどうかを判定するクエリを設定または取得します。

構文

QueryInitUser メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtODBCQueryScheme->QueryInitUser([queryGetInitUser])
```

パラメータ

QueryInitUser メソッドは以下のパラメータを受け入れます。

queryGetInitUser (文字列)

(オプション) データベースにユーザが存在するかどうかを判定するクエリを指定します。

戻り値

QueryInitUser メソッドは以下のいずれかの値を返します。

- 新規または既存のクエリ
- コールに失敗した場合は、**undef**

注釈

サンプルクエリ (SiteMinder サンプル データベース スキーマ SmSampleUsers に基づく)。%s 表現は、以下のように、クエリが実行されるときに SiteMinder によって提供されるユーザ名パラメータのプレースホルダです。

```
select Name from SmUser where Name = '%s'
```

QueryIsGroupMember メソッド -- 特定のユーザのグループ メンバシップをリストするクエリを設定または取得

QueryIsGroupMember メソッド -- 特定のユーザのグループ メンバシップをリストするクエリを設定または取得します。

構文

QueryIsGroupMember メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtODBCQueryScheme->QueryIsGroupMember([queryIsGroupMember])
```

パラメータ

QueryIsGroupMember メソッドは、以下のパラメータを受け入れます。

queryIsGroupMember (文字列)

(オプション) ユーザのグループ メンバシップを判定するクエリを指定します。

戻り値

QueryIsGroupMember メソッドは以下のいずれかの値を返します。

- 新規または既存のクエリ
- コールに失敗した場合は、**undef**

注釈

サンプル クエリ（SiteMinder サンプル データベース スキーマ SmSampleUsers に基づく）。%s 表現は、以下のように、クエリが実行されるときに SiteMinder によって提供されるユーザ名およびグループ名パラメータのプレースホルダです。

```
select Id from SmUserGroup where UserId = (select UserId from SmUser where Name = '%s')
and GroupId = (select GroupId from SmGroup where Name = '%s')
```

QueryLookup メソッド -- オブジェクトをフェッチするクエリを設定または取得

QueryLookup メソッドは、グループ テーブルで指定されたプロパティに基づいてオブジェクトをフェッチするクエリを設定または取得します。

構文

QueryLookup メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtODBCQueryScheme->QueryLookup([queryLookup])
```

パラメータ

QueryLookup メソッドは以下のパラメータを受け入れます。

queryLookup (文字列)

(オプション) オブジェクトをフェッチするクエリを指定します。

戻り値

QueryLookup メソッドは以下のいずれかの値を返します。

- 新規または既存のクエリ
- コールに失敗した場合は、**undef**

注釈

サンプルクエリ（SiteMinder サンプル データベース スキーマ SmSampleUsers に基づく）。%s 表現は、以下のように、クエリが実行されるときに SiteMinder によって提供されるパラメータのプレースホルダです。

```
select Name, 'User' as Class from SmUser where Name %s Union select Name, 'Group' as Class from SmGroup where Name %s order by Class
```

QueryLookupGroup メソッド -- グループ名をフェッチするクエリを設定または取得

QueryLookupGroup メソッドは、グループ テーブルで指定されたプロパティに基づいてグループ名をフェッチするクエリを設定または取得します。

構文

QueryLookupGroup メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtODBCQueryScheme->QueryLookupGroup([queryLookupGrp])
```

パラメータ

QueryLookupGroup メソッドは以下のパラメータを受け入れます。

queryLookupGrp（文字列）

（オプション）グループ名をフェッチするクエリを指定します。

戻り値

QueryLookupGroup メソッドは以下のいずれかの値を返します。

- 新規または既存のクエリ
- コールに失敗した場合は、**undef**

注釈

サンプルクエリ（SiteMinder サンプル データベース スキーマ SmSampleUsers に基づく）。%s 表現は、以下のように、クエリが実行されるときに SiteMinder によって提供されるパラメータのプレースホルダです。

```
select Name, 'Group' as Class from SmGroup where %s
```


QueryLookupUser メソッド -- ユーザ名をフェッチするクエリを設定または取得

QueryLookupUser メソッドは、ユーザテーブルで指定されたプロパティに基づいてユーザ名をフェッチするクエリを設定または取得します。

構文

QueryLookupUser メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtODBCQueryScheme->QueryLookupUser([queryLookupUsr])
```

パラメータ

QueryLookupUser メソッドは以下のパラメータを受け入れます。

queryLookupUsr (文字列)

(オプション) ユーザ名をフェッチするクエリを指定します。

戻り値

QueryLookupUser メソッドは以下のいずれかの値を返します。

- 新規または既存のクエリ
- コールに失敗した場合は、**undef**

注釈

サンプル クエリ (SiteMinder サンプル データベース スキーマ SmSampleUsers に基づく)。%s 表現は、以下のように、クエリが実行されるときに SiteMinder によって提供されるパラメータのプレースホルダです。

```
select Name, 'User' as Class from SmUser where %s
```

QuerySetGroupProp メソッド -- グループ プロパティの値を設定するクエリを設定または取得

QuerySetGroupProp メソッドは、グループ プロパティの値を設定するクエリを設定または取得します。プロパティは QueryGetGroupProps メソッドによって指定されたプロパティの 1 つである必要があります。

構文

QuerySetGroupProp メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtODBCQueryScheme->QuerySetGroupProp([querySetGroupProp])
```

パラメータ

QuerySetGroupProp メソッドは以下のパラメータを受け入れます。

querySetGroupProp (文字列)

(オプション) グループのプロパティ値を設定するクエリを指定します。

戻り値

QuerySetGroupProp メソッドは以下のいずれかの値を返します。

- 新規または既存のクエリ
- コールに失敗した場合は、**undef**

注釈

サンプル クエリ (SiteMinder サンプル データベース スキーマ SmSampleUsers に基づく)。%s 表現は、以下のように、クエリが実行されるときに SiteMinder によって提供されるプロパティ名、プロパティ値およびグループ名パラメータのプレースホルダです。

```
update SmGroup set %s = %s where Name = '%s'
```

QuerySetPassword メソッド -- ユーザ パスワードを変更するクエリを設定または取得

QuerySetPassword メソッドは、ユーザ パスワードを変更するクエリを設定または取得します。

構文

QuerySetPassword メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtODBCQueryScheme->QuerySetPassword([querySetPassword])
```

パラメータ

QuerySetPassword メソッドは以下のパラメータを受け入れます。

querySetPassword (文字列)

(オプション) ユーザ パスワードを変更するクエリを指定します。

戻り値

QuerySetPassword メソッドは以下のいずれかの値を返します。

- 新規または既存のクエリ
- コールに失敗した場合は、**undef**

注釈

サンプル クエリ (SiteMinder サンプル データベース スキーマ SmSampleUsers に基づく)。%s 表現は、以下のように、クエリが実行されるときに SiteMinder によって提供されるユーザ パスワードおよびユーザ名パラメータのプレースホルダです。

```
update SmUser set Password = '%s' where Name = '%s'
```

QuerySetUserProp メソッド -- ユーザ プロパティの値を設定するクエリを設定または取得

QuerySetUserProp メソッドは、ユーザ プロパティの値を設定するクエリを設定または取得します。プロパティは PolicyMgtODBCQueryScheme->QueryGetUserProps メソッドによって指定されたプロパティの 1 つである必要があります。

構文

QuerySetUserProp メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtODBCQueryScheme->QuerySetUserProp([querySetUserProp])
```

パラメータ

QuerySetUserProp メソッドは、以下のパラメータを受け入れます。

querySetUserProp (文字列)

(オプション) グループのプロパティ値を設定するクエリを指定します。

戻り値

QuerySetUserProp メソッドは以下のいずれかの値を返します。

- 新規または既存のクエリ
- コールに失敗した場合は、**undef**

注釈

サンプルクエリ (SiteMinder サンプル データベース スキーマ SmSampleUsers に基づく)。%s 表現は、以下のように、クエリが実行されるときに SiteMinder によって提供されるプロパティ名、プロパティ値およびユーザ名パラメータのプレースホルダです。

```
update SmUser set %s = %s where Name = '%s'
```

CLI パスワード ポリシー メソッド

AllowNestedGroups メソッド -- パスワード ポリシーのネストされたグループに対する設定を許可

AllowNestedGroups メソッドは、パスワード ポリシーのネストされたグループに対する設定を許可します。このメソッドは LDAP ディレクトリにのみ適用されます。

構文

AllowNestedGroups メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->AllowNestedGroups([groupFlag])
```

パラメータ

AllowNestedGroups メソッドは以下のパラメータを受け入れます。

groupFlag (int)

(オプション) 以下のように、ネストされたグループを許可するかどうかを指定します。

- ネストされたグループを許可する場合は、**1**
- ネストされたグループを許可しない場合は、**0**

戻り値

AllowNestedGroups メソッドは以下のいずれかの値を返します。

- ネストされたグループが許可されない場合は、**0**。
- ネストされたグループが許可される場合は、**1**。

AllowLowerPriorityPolicies メソッド -- 優先度の低いパスワード ポリシーを評価するかどうかを判定するフラグを設定

ApplyLowerPriorityPolicies メソッドは、現在のパスワード ポリシーを評価した後に優先度の低いパスワード ポリシーを評価するかどうかを判定するフラグを設定または取得します。

構文

ApplyLowerPriorityPolicies メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->ApplyLowerPriorityPolicies([lowerPriorityFlag])
```

パラメータ

ApplyLowerPriorityPolicies メソッドは、以下のパラメータを受け入れます。

lowerPriorityFlag (int)

(オプション) 優先度の低いパスワード ポリシーの評価を有効にするかどうかを指定します。

- 1 は優先度の低いパスワード ポリシーの評価を有効にします。
- 0 は優先度の低いパスワード ポリシーの評価を無効にします。

戻り値

ApplyLowerPriorityPolicies メソッドは以下のいずれかの値を返します。

- 新規または既存のフラグ設定
- コールに失敗した場合は、**undef**

AuthLoginTrackFailure メソッド -- ログイン追跡データが失敗した場合にユーザのログインを許可

AuthLoginTrackFailure メソッドは、ログイン追跡データのユーザ ディレクトリへの書き込みに失敗した場合にユーザがログインできるようにするフラグを設定または取得します。 ログイン追跡データにはログイン試行および成功したログインが含まれます。

構文

AuthLoginTrackFailure メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->AuthLoginTrackFailure([trackingFlag])
```

パラメータ

AuthLoginTrackFailure メソッドは以下のパラメータを受け入れます。

trackingFlag (int)

(オプション) 以下のように、ログイン追跡が失敗するときにユーザのログインを許可するかどうかを指定します。

- 1 はユーザのログインを許可します
- 0 はユーザのログインを許可しません

戻り値

AuthLoginTrackFailure メソッドは以下のいずれかの値を返します。

- 新規または既存のフラグ設定
- コールに失敗した場合は、**undef**

注釈

このフラグを有効にすると、ユーザはユーザ ディレクトリにログイン追跡データを書き込めない場合でもログインできます。 このフラグを無効にすると、ユーザはユーザ ディレクトリにログイン追跡データを書き込めない場合にログインできません。

BadLoginDisablementPeriod メソッド -- ユーザ アカウントが無効になるまでの時間数(分)を設定または取得

BadLoginDisablementPeriod メソッドは、ログイン試行に失敗した回数が多い場合にユーザ アカウントが無効化されるまでの時間 (分) を設定または取得します。

構文

BadLoginDisablementPeriod メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->BadLoginDisablementPeriod([disablementPeriod])
```

パラメータ

BadLoginDisablementPeriod メソッドは、以下のパラメータを受け入れます。

disablementPeriod (int)

(オプション) ユーザ アカウントが無効になるまでの許容時間数 (分) を指定します。

戻り値

BadLoginDisablementPeriod メソッドは以下のいずれかの値を返します。

- 新規または既存の無効化期間
- コールに失敗した場合は、**undef**

Description メソッド -- パスワード ポリシーの説明を設定または取得

Description メソッドは、パスワード ポリシーの説明を設定または取得します。

構文

Description メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->Description([policyDesc])
```


パラメータ

Description メソッドは以下のパラメータを受け入れます。

policyDesc (文字列)

(オプション) パスワード ポリシーの説明を指定します。

戻り値

Description メソッドは以下のいずれかの値を返します。

- 新規または既存のポリシーの説明
- コールに失敗した場合は、空の文字列。

DictionaryMatch メソッド -- ディクショナリ確認用のパスワードに資格を与えるのに必要な文字の最小数を設定

DictionaryMatch メソッドは、ディクショナリ確認用のパスワードに資格を与えるのに必要な文字の最小数を設定します。

構文

DictionaryMatch メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->DictionaryMatch([dicMatchLen])
```

パラメータ

DictionaryMatch メソッドは以下のパラメータを受け入れます。

dicMatchLen (int)

(オプション) 必要な文字の最小数を指定します。

戻り値

DictionaryMatch メソッドは以下のいずれかの値を返します。

- 新規または既存の最小設定
- コールに失敗した場合は、**undef**

DictionaryPath メソッド -- ディクショナリ ファイルの場所を設定または取得

DictionaryPath メソッドは、パスワードに使用できない語をリスト表示するディクショナリ ファイルの場所を設定または取得します。

構文

DictionaryPath メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->DictionaryPath([dicPath])
```

パラメータ

DictionaryPath メソッドは以下のパラメータを受け入れます。

dicPath (文字列)

(オプション) 新しいディクショナリ パスを指定します。

戻り値

DictionaryPath メソッドは以下のいずれかの値を返します。

- 新規または既存のディクショナリ パス。
- コールに失敗した場合は、**undef**

注釈

ディクショナリ ファイルは、すべてのポリシー サーバがアクセスできるディレクトリ内のテキスト ファイルである必要があります。

DisableAfterInactivityExpiration メソッド -- 非アクティブ ユーザのアカウントを無効化

DisableAfterInactivityExpiration メソッドは、指定された期間にわたって非アクティブだった場合にユーザ アカウントを無効にするためのフラグを設定または取得します。

構文

DisableAfterInactivityExpiration メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->DisableAfterInactivityExpiration([inactivityFlag])
```

パラメータ

`DisableAfterInactivityExpiration` メソッドは、以下のパラメータを受け入れます。

`inactivityFlag` (int)

(オプション) ユーザのアカウントを無効にするかどうかを指定します。

1 は非アクティブ状態の指定された期間の後にユーザのアカウントを無効にします

0 はアカウントを有効に維持し、パスワード変更を強制します

戻り値

`DisableAfterInactivityExpiration` メソッドは以下のいずれかの値を返します。

- 新規または既存のフラグ設定
- コールに失敗した場合は、**undef**

注釈

フラグが非アクティブ状態期間の経過後にユーザ アカウントを無効にしないように設定されている場合、ユーザは次のログインでパスワードを変更する必要があります。

DisableAfterPwdExpiration メソッド -- ユーザのパスワードが期限切れになった後にユーザのアカウントを無効化

`DisableAfterPwdExpiration` メソッドは、ユーザのパスワードが期限切れになった後にユーザのアカウントを無効にするためのフラグを設定または取得します。

構文

`DisableAfterPwdExpiration` メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->DisableAfterPwdExpiration([expireFlag])
```

パラメータ

DisableAfterPwdExpiration メソッドは以下のパラメータを受け入れます。

expireFlag (タイプ)

(オプション) 以下のように、ユーザのアカウントを無効にするかどうかを指定します。

1 は、ユーザのパスワードが期限切れになった後にユーザのアカウントを無効にします

0 はアカウントを有効に維持し、パスワード変更を強制します

戻り値

DisableAfterPwdExpiration メソッドは以下のいずれかの値を返します。

- 新規または既存のフラグ設定
- コールに失敗した場合は、**undef**

注釈

フラグがパスワードの期限切れ後にユーザ アカウントを無効にしないように設定されている場合、ユーザは次のログインでパスワードを変更する必要があります。

EntireDir メソッド -- パスワード ポリシーをディレクトリ全体に適用するかどうかを判定

EntireDir メソッドは、ディレクトリ全体またはその一部のみにパスワードポリシーを適用するかどうかを判定します。

構文

EntireDir メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->EntireDir([dirFlag])
```

パラメータ

EntireDir メソッドは、以下のパラメータを受け入れます。

dirFlag (int)

(オプション) パスワード ポリシーをディレクトリ全体に適用するかどうかを指定します。

1 はパスワード ポリシーをディレクトリ全体に適用します

0 はパスワード ポリシーをディレクトリの一部のみに適用します

戻り値

EntireDir メソッドは以下のいずれかの値を返します。

- ポリシーをディレクトリ全体に適用する場合は、1
- ポリシーをディレクトリの一部に適用する場合は、0

注釈

ディレクトリ全体の一部の指定については、

[PolicyMgtPwdPolicy->UserDirPath](#) (P. 363) メソッドおよび

[PolicyMgtPwdPolicy->UserDirClass](#) (P. 362) メソッドの説明を参照してください。

ExpirationDelay メソッド -- パスワードを使用できる日数を指定

ExpirationDelay メソッドは、変更が必要になるまでパスワードを使用できる日数を指定します。

構文

ExpirationDelay メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->ExpirationDelay([expDelay])
```

パラメータ

ExpirationDelay メソッドは以下のパラメータを受け入れます。

expDelay (int)

(オプション) パスワードを使用できる日数を指定します。

戻り値

ExpirationDelay メソッドは以下のいずれかの値を返します。

- 新規または既存の日数
- コールに失敗する場合は、-1

IsEnabled メソッド -- パスワード ポリシーを有効化または無効化

IsEnabled メソッドは、パスワード ポリシーを有効化または無効化します。

構文

IsEnabled メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->IsEnabled([enableFlag])
```

パラメータ

IsEnabled メソッドは以下のパラメータを受け入れます。

enableFlag (int)

(オプション) 以下のように、パスワード ポリシーを有効にするかどうかを指定します。

- 1 はパスワード ポリシーを有効にします
- 0 はパスワード ポリシーを無効にします

戻り値

IsEnabled メソッドは以下のいずれかの値を返します。

- ポリシーが有効な場合は、1
- ポリシーが無効な場合は、0

MaxLoginFailures メソッド -- ログイン試行に失敗できる最大回数を設定または取得

MaxLoginFailures メソッドは、ユーザアカウントが無効になるまでにユーザがログイン試行に失敗できる最大回数を設定または取得します。

構文

MaxLoginFailures メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->MaxLoginFailures([maxLogin])
```

パラメータ

MaxLoginFailures メソッドは以下のパラメータを受け入れます。

maxLogin (int)

(オプション) 失敗できるログイン試行の回数を指定します。

戻り値

MaxLoginFailures メソッドは以下のいずれかの値を返します。

- 新規または既存の失敗できるログイン試行設定
- コールに失敗した場合は、**undef**

MaxLoginInactive メソッド -- 非アクティブ状態が許可される日数を設定または取得

MaxLoginInactive メソッドは、ユーザのパスワードが期限切れになる前に許可される非アクティブ状態の日数を設定または取得します。

構文

MaxLoginInactive メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->MaxLoginInactive([maxLoginInactive])
```

パラメータ

MaxLoginInactive メソッドは、以下のパラメータを受け入れます。

maxLoginInactive (int)

(オプション) 非アクティブ状態の日数を指定します。

戻り値

MaxLoginInactive メソッドは以下のいずれかの値を返します。

- 新規または既存の最大非アクティブ期間設定
- コールに失敗した場合は、**undef**

Name メソッド -- パスワード ポリシー名を設定または取得

Name メソッドは、パスワード ポリシー名を設定または取得します。

構文

Name メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->Name([policyName])
```

パラメータ

Name メソッドは以下のパラメータを受け入れます。

policyName (文字列)

(オプション) パスワード ポリシー名を指定します。

戻り値

Name メソッドは以下のいずれかの値を返します。

- 新規または既存のポリシー名
- コールに失敗した場合は、**undef**

PwdAddRegExpMatch メソッド -- 新しいパスワードが一致する必要がある表現のリストに正規表現を追加

PwdAddRegExpMatch メソッドは、新しいパスワードが一致する必要がある表現のリストに正規表現を追加します。

構文

PwdAddRegExpMatch メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdAddRegExpMatch([tag] [, expression])
```


パラメータ

`PwdAddRegExpMatch` メソッドは、以下のパラメータを受け入れます。

tag (文字列)

(オプション) 正規表現の名前を指定します。

expression (文字列)

(オプション) 正規表現を指定します。

戻り値

`PwdAddRegExpMatch` メソッドは以下のいずれかの値を返します。

- 正規表現が正しく追加される場合は、0
- コールに失敗する場合は、-1

`PwdAddRegExpNoMatch` メソッド -- 新しいパスワードが一致してはならない表現のリストに正規表現を追加

`PwdAddRegExpNoMatch` メソッドは、新しいパスワードが一致してはならない表現のリストに正規表現を追加します。

構文

`PwdAddRegExpNoMatch` メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdAddRegExpNoMatch([tag] [, expression])
```

パラメータ

`PwdAddRegExpNoMatch` メソッドは、以下のパラメータを受け入れます。

tag (文字列)

(オプション) 正規表現の名前を指定します。

expression (文字列)

(オプション) 正規表現を指定します。

戻り値

PwdAddRegExpNoMatch メソッドは以下のいずれかの値を返します。

- 正規表現が正しく追加される場合は、0
- コールに失敗する場合は、-1

PwdAllowDigits メソッド -- パスワードに数字を使用できるかどうかを指定

PwdAllowDigits メソッドは、パスワードに数字を使用できるかどうかを指定するフラグを設定または取得します。

構文

PwdAllowDigits メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdAllowDigits([digitFlag])
```

パラメータ

PwdAllowDigits メソッドは以下のパラメータを受け入れます。

digitFlag (int)

(オプション) パスワードに数字を使用できるかどうかを指定します。

数字が許可される場合は、1。

数字が許可されない場合は、0。

戻り値

PwdAllowDigits メソッドは以下のいずれかの値を返します。

- 新規または既存のフラグ設定
- コールに失敗した場合は、**undef**

PwdAllowLowercase メソッド -- パスワードに小文字を使用できるかどうかを指定

PwdAllowLowercase メソッドは、パスワードに小文字を使用できるかどうかを指定するフラグを設定または取得します。

構文

PwdAllowLowercase メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdAllowLowercase([lcFlag])
```

パラメータ

PwdAllowLowercase メソッドは、以下のパラメータを受け入れます。

lcFlag (int)

(オプション) 以下のように、パスワードに小文字を使用できるかどうかを指定します。

- 1 は小文字を許可します
- 0 は小文字を許可しません

戻り値

PwdAllowLowercase メソッドは以下のいずれかの値を返します。

- 新規または既存のフラグ設定
- コールに失敗した場合は、**undef**

PwdAllowNonAlphaNum メソッド -- パスワードに英数字以外の文字を使用できるかどうかを指定

PwdAllowNonAlphaNum メソッドは、パスワードに英数字以外の文字を使用できるかどうかを指定するフラグを設定または取得します。

構文

PwdAllowNonAlphaNum メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdAllowNonAlphaNum([nonAlphaNumFlag])
```

パラメータ

PwdAllowNonAlphaNum メソッドは、以下のパラメータを受け入れます。

nonAlphaNumFlag (int)

(オプション) パスワードに英数字以外の文字を使用できるかどうかを指定します

- 1 は英数字以外の文字を許可します
- 0 は英数字以外の文字を許可しません

戻り値

PwdAllowNonAlphaNum メソッドは以下のいずれかの値を返します。

- 新規または既存のフラグ設定
- コールに失敗した場合は、**undef**

PwdAllowNonPrintable メソッド -- パスワードに印刷不能文字を使用できるかどうかを指定

PwdAllowNonPrintable メソッドは、パスワードに印刷不能文字を使用できるかどうかを指定するフラグを設定または取得します。印刷不能文字は、コンピュータ画面には表示されません。

構文

PwdAllowNonPrintable メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdAllowNonPrintable([nonPrintFlag])
```

パラメータ

PwdAllowNonPrintable メソッドは、以下のパラメータを受け入れます。

nonPrintFlag (int)

(オプション) 以下のように、パスワードに印刷不能文字を使用できるかどうかを指定します。

- 1 は印刷不能文字を許可します
- 0 は印刷不能文字を許可しません

戻り値

`PwdAllowNonPrintable` メソッドは以下のいずれかの値を返します。

- 新規または既存のフラグ設定
- コールに失敗した場合は、**undef**

`PwdAllowPunctuation` メソッド -- パスワードに句読点文字を使用できるかどうかを指定

`PwdAllowPunctuation` メソッドは、パスワードに句読点文字を使用できるかどうかを指定するフラグを設定または取得します。

構文

`PwdAllowPunctuation` メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdAllowPunctuation([punctuationMarkFlag])
```

パラメータ

`PwdAllowPunctuation` メソッドは、以下のパラメータを受け入れます。

punctuationMarkFlag (int)

(オプション) 以下のように、パスワードに句読点文字を使用できるかどうかを指定します。

- 1 は句読点文字を許可します
- 0 は句読点文字を許可しません

戻り値

`PwdAllowPunctuation` メソッドは以下のいずれかの値を返します。

- 新規または既存のフラグ設定
- コールに失敗した場合は、**undef**

PwdAllowUpperCase メソッド -- パスワードに大文字を使用できるかどうかを指定

PwdAllowUpperCase メソッドは、パスワードに大文字を使用できるかどうかを指定するフラグを設定または取得します。

構文

PwdAllowUpperCase メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdAllowUppercase([upperCaseFlag])
```

パラメータ

PwdAllowUpperCase メソッドは以下のパラメータを受け入れます。

upperCaseFlag (int)

(オプション) 以下のように、パスワードに大文字を使用できるかどうかを指定します。

- 1 は大文字を許可します
- 0 は大文字を許可しません。

戻り値

PwdAllowUpperCase メソッドは以下のいずれかの値を返します。

- 新規または既存のフラグ設定
- コールに失敗した場合は、**undef**

PwdExpiryWarning メソッド -- パスワードが期限切れになることをユーザに事前に通知する場合の日数を設定または取得

PwdExpiryWarning メソッドは、パスワードが期限切れになることをユーザに事前に通知する場合の日数を設定または取得します。

構文

PwdExpiryWarning メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdExpiryWarning([warningDays])
```

パラメータ

PwdExpiryWarning メソッドは、以下のパラメータを受け入れます。

warningDays (int)

(オプション) 事前通知の日数を指定します。

戻り値

PwdExpiryWarning メソッドは以下のいずれかの値を返します。

- 新規または既存の事前通知設定
- コールに失敗した場合は、**undef**

PwdForceLowerCase Method -- 新しいパスワードの大文字を小文字に変換するかどうかを判定

PwdForceLowerCase メソッドは、新しいパスワードの大文字を小文字に変換するかどうかを判定するフラグを設定または取得します。

構文

PwdForceLowerCase メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdForceLowerCase([forceLCFlag])
```

パラメータ

PwdForceLowerCase メソッドは、以下のパラメータを受け入れます。

forceLCFlag (int)

(オプション) 以下のように、強制的に新しいパスワードを小文字に変換することを指定します。

- 1 任意の大文字を小文字に変換します
- 0 は大文字を変換しません。

戻り値

PwdForceLowerCase メソッドは以下のいずれかの値を返します。

- 新規または既存のフラグ設定
- コールに失敗した場合は、**undef**

PwdForceUpperCase Method -- 新しいパスワードの小文字を大文字に変換するかどうかを判定

PwdForceUpperCase メソッドは、新しいパスワードの小文字を大文字に変換するかどうかを判定するフラグを設定または取得します。

構文

PwdForceUpperCase メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdForceUpperCase([forceUCFlag])
```

パラメータ

PwdForceUpperCase メソッドは、以下のパラメータを受け入れます。

forceUCFlag (int)

(オプション) 以下のように、強制的に新しいパスワードに大文字のみを使用することを指定します。

- 1 は大文字を強制します
- 0 は大文字を強制しません

戻り値

PwdForceUpperCase メソッドは以下のいずれかの値を返します。

- 新規または既存のフラグ設定
- コールに失敗した場合は、**undef**

PwdGetAllRegExpMatch メソッド -- 新しいパスワードが一致する必要がある正規表現の名前タグを取得

PwdGetAllRegExpMatch メソッドは、新しいパスワードが一致する必要があるすべての正規表現の名前タグを取得します。

構文

PwdGetAllRegExpMatch メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdGetAllRegExpMatch()
```


パラメータ

PwdGetAllRegExpMatch メソッドは、パラメータを受け入れません。

戻り値

PwdGetAllRegExpMatch メソッドは以下のいずれかの値を返します。

- 新しいパスワードが一致する必要がある正規表現の名前タグの配列。
- コールに失敗した場合は、**undef**

PwdGetAllRegExpNoMatch メソッド -- 新しいパスワードが一致してはならない正規表現の名前タグを取得

PwdGetAllRegExpNoMatch メソッドは、新しいパスワードが一致してはならないすべての正規表現の名前タグを取得します。

構文

PwdGetAllRegExpNoMatch メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdGetAllRegExpNoMatch()
```

パラメータ

PwdGetAllRegExpNoMatch メソッドは、パラメータを受け入れません。

戻り値

PwdGetAllRegExpNoMatch メソッドは以下のいずれかの値を返します。

- 新しいパスワードが一致してはならない正規表現の名前タグの配列。
- コールに失敗した場合は、**undef**

PwdGetRegExp メソッド -- 指定された名前タグ用の正規表現を取得

PwdGetRegExp メソッドは、指定された名前タグ用の正規表現を取得します。

構文

PwdGetRegExp メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdGetRegExp(tag)
```

パラメータ

PwdGetRegExp メソッドは以下のパラメータを受け入れます。

tag (文字列)

取得する正規表現の名前を指定します。

戻り値

PwdGetRegExp メソッドは以下のいずれかの値を返します。

- 指定された正規表現
- コールに失敗した場合は、**undef**

PwdIgnoreSequence メソッド -- 新しいパスワードを計算するときにシーケンスを無視するかどうかを判定

PwdIgnoreSequence メソッドは、前の文字と異なるパーセンテージが計算されるときにシーケンス（つまり文字位置）を無視するかどうかを指定します。

構文

PwdIgnoreSequence メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdIgnoreSequence([pwdPctSeq])
```

パラメータ

`PwdIgnoreSequence` メソッドは以下のパラメータを受け入れます。

`pwdPctSeq` (int)

(オプション) 以下のように、新しいパスワードを作成するときに文字のシーケンスを無視するかどうかを指定します。

- 1 は前のパスワードの差異パーセンテージを計算するときにシーケンスを無視します
- 0 はシーケンスを考慮します

戻り値

`PwdIgnoreSequence` メソッドは以下のいずれかの値を返します。

- 1 はシーケンスを無視
- 0 はシーケンスを考慮

注釈

たとえば、以下のようにユーザの前のパスワードが **BASEBALL12** であると想定します。

- このメソッドを 1 (シーケンスを無視) に設定すると、ユーザは新しいパスワードとして **12BASEBALL** を選択できません。それは、文字シーケンスにかかわらず、文字が前のパスワードの場合と同じためです。
- このメソッドを 0 (シーケンスを考慮) に設定すると、ユーザは新しいパスワードとして **12BASEBALL** を選択できます。これは文字が異なるシーケンスで 사용되는ためです。

セキュリティを高めるには、このメソッドに 1 を渡します。

PwdMaxLength メソッド -- ユーザ パスワードの最大長を設定または取得

`PwdMaxLength` メソッドは、ユーザ パスワードの最大長を設定または取得します。

構文

`PwdMaxLength` メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdMaxLength([maxPwdLength])
```

パラメータ

PwdMaxLength メソッドは以下のパラメータを受け入れます。

maxPwdLength (int)

(オプション) 最大パスワード長を指定します。

戻り値

PwdMaxLength メソッドは、新規または既存のパスワード長さ設定を返します。

PwdMaxRepeatingChar メソッド -- 同一文字の最大数を設定または取得

PwdMaxRepeatingChar メソッドは、パスワードで連続して使用できる同一文字の最大数を設定または取得します。

構文

PwdMaxRepeatingChar メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdMaxRepeatingChar([maxPwdRepeat])
```

パラメータ

PwdMaxRepeatingChar メソッドは以下のパラメータを受け入れます。

maxPwdRepeat (int)

(オプション) 繰り返し文字の最大数を指定します。

戻り値

PwdMaxRepeatingChar メソッドは、繰り返し文字の新規または既存の設定を返します。

PwdMinAlpha メソッド -- パスワードに使用する必要があるアルファベット文字の最小数を設定または取得

PwdMinAlpha メソッドは、パスワードに使用する必要があるアルファベット文字（A ～ Z、a ～ z）の最小数を設定または取得します。

構文

PwdMinAlpha メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdMinAlpha([pwdMinAlpha])
```

パラメータ

PwdMinAlpha メソッドは以下のパラメータを受け入れます。

pwdMinAlpha (int)

（オプション）必要なアルファベット文字の最小数を指定します。

戻り値

PwdMinAlpha メソッドは、アルファベット文字の新規または既存の最小数を返します。

PwdMinAlphaNum メソッド -- パスワードに使用する必要がある英数字の最小数を設定または取得

PwdMinAlphaNum メソッドは、パスワードに使用する必要がある英数字（A ～ Z、a ～ z）の最小数を設定または取得します。

構文

PwdMinAlphaNum メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdMinAlphaNum([pwdMinAlphaNum])
```

パラメータ

PwdMinAlphaNum メソッドは、以下のパラメータを受け入れます。

pwdMinAlphaNum (int)

（オプション）必要な英数字の最小数を指定します。

戻り値

PwdMinAlphaNum メソッドは、英数字の新規または既存の最小数を返します。

PwdMinLength メソッド -- ユーザ パスワードの最小長を設定または取得

PwdMinLength メソッドは、ユーザ パスワードの最小長を設定または取得します。

構文

PwdMinLength メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdMinLength([minPwdLength])
```

パラメータ

PwdMinLength メソッドは、以下のパラメータを受け入れます。

minPwdLength (int)

(オプション) ユーザ パスワードの最小長を指定します。

戻り値

PwdMinLength メソッドは、新規または既存の最小パスワード長を返します。

PwdMinLowercase メソッド -- パスワードに使用する必要がある小文字の最小数を設定または取得

PwdMinLowercase メソッドは、パスワードに使用する必要がある小文字の最小数を設定または取得します。

構文

PwdMinLowercase メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdMinLowercase([pwdMinLC])
```

パラメータ

PwdMinLowercase メソッドは以下のパラメータを受け入れます。

pwdMinLC (int)

(オプション) パスワードに使用する必要がある小文字の最小数を指定します。

戻り値

PwdMinLowercase メソッドは、新規または既存の小文字の最小数を返します。

PwdMinNonAlpha メソッド -- パスワードに使用する必要がある英数字以外の文字の最小数を設定または取得

PwdMinNonAlpha メソッドは、パスワードに使用する必要がある英数字以外の文字の最小数を設定または取得します。これらの文字には句読点、および @、\$、* などのキーボード上のその他の記号が含まれます。

構文

PwdMinNonAlpha メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdMinNonAlpha([pwdMinNonAlpha])
```

パラメータ

PwdMinNonAlpha メソッドは、以下のパラメータを受け入れます。

pwdMinNonAlpha (int)

(オプション) 必要な英数字以外の文字の最小数を指定します。

戻り値

PwdMinNonAlpha メソッドは、英数字以外の文字の新規または既存の最小数を返します。

PwdMinNonPrintable メソッド -- パスワードに使用する必要がある印刷不能文字の最小数を設定または取得

PwdMinNonPrintable メソッドは、パスワードに使用する必要がある印刷不能文字の最小数を設定または取得します。印刷不能文字は、コンピュータ画面には表示されません。

構文

PwdMinNonPrintable メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdMinNonPrintable([pwdMinNonPrint])
```

パラメータ

PwdMinNonPrintable メソッドは以下のパラメータを受け入れます。

pwdMinNonPrint (int)

(オプション) 必要な印刷不能文字の最小数を指定します。

戻り値

PwdMinNonPrintable メソッドは、印刷不能文字の新規または既存の最小数を返します。

PwdMinNumbers メソッド -- パスワードに使用する必要がある数字の最小数を設定または取得

PwdMinNumbers メソッドは、パスワードに使用する必要がある数字の最小数を設定または取得します。

構文

PwdMinNumbers メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdMinNumbers([pwdMinNum])
```

パラメータ

PwdMinNumbers メソッドは以下のパラメータを受け入れます。

pwdMinNum (int)

(オプション) 必要な数字の最小数を指定します。

戻り値

`PwdMinNumbers` メソッドは、数字の新規または既存の最小数を返します。

PwdMinProfileMatch メソッド -- ユーザの個人情報と照合する最小文字シーケンスを指定

`PwdMinProfileMatch` メソッドは、ユーザの個人情報と照合する最小文字シーケンスを指定します。

構文

`PwdMinProfileMatch` メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdMinProfileMatch([pwdMatchAttr])
```

パラメータ

`PwdMinProfileMatch` メソッドは以下のパラメータを受け入れます。

pwdMatchAttr (int)

(オプション) 照合するシーケンス文字の最小数を指定します。

戻り値

`PwdMinProfileMatch` メソッドは、新規または既存の最小設定を返します。

注釈

たとえば、この値が **4** に設定されている場合、**SiteMinder** 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。**SiteMinder** 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。

PwdMinPunctuation メソッド -- パスワードに使用する必要がある句読点の最小数を設定または取得

PwdMinPunctuation メソッドは、パスワードに使用する必要がある句読点の最小数を設定または取得します。これらの文字にはピリオド、カンマ、感嘆符、スラッシュ、ハイフン、ダッシュおよび他の句読点が含まれます。

構文

PwdMinPunctuation メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdMinPunctuation([pwdMinPunc])
```

パラメータ

PwdMinPunctuation メソッドは以下のパラメータを受け入れます。

pwdMinPunc (int)

(オプション) 必要な句読点の最小数を指定します。

戻り値

PwdMinPunctuation メソッドは、句読点の新規または既存の最小数を返します。

PwdMinUppercase メソッド -- パスワードに使用する必要がある大文字の最小数を設定または取得

PwdMinUppercase メソッドは、パスワードに使用する必要がある大文字の最小数を設定または取得します。

構文

PwdMinUppercase メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdMinUppercase([pwdMinUC])
```

パラメータ

`PwdMinUppercase` メソッドは以下のパラメータを受け入れます。

pwdMinUC (int)

(オプション) パスワードに使用する必要がある大文字の最小数を指定します。

戻り値

`PwdMinUppercase` メソッドは、新規または既存の大文字の最小数を返します。

PwdPercentDiff メソッド -- 新しいパスワードに使用する必要がある異なる文字のパーセンテージを設定または取得

`PwdPercentDiff` メソッド -- 以前のパスワードに使用した文字と異なる文字を新しいパスワードに含める必要があるパーセンテージを設定または取得します。値が **100** に設定されている場合、新しいパスワードには前のパスワードに含まれている文字を使用できません (パラメータ *PwdIgnoreSeq* が **0** に設定されていない場合)。

構文

`PwdPercentDiff` メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdPercentDiff([pwdPctDiff])
```

パラメータ

`PwdPercentDiff` メソッドは以下のパラメータを受け入れます。

pwdPctDiff (int)

(オプション) 最小パーセンテージ設定を指定します。

戻り値

`PwdPercentDiff` メソッドは、新規または既存の最小パーセンテージ設定を返します。

PwdPolicyPriority メソッド -- パスワードの評価優先度設定を設定または取得

PwdPolicyPriority メソッドは、パスワードの評価優先度設定を設定または取得します。ポリシーは降順（最初は 1000、最後は 1）で評価されます。

構文

PwdPolicyPriority メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdPolicyPriority([priority])
```

パラメータ

PwdPolicyPriority メソッドは、以下のパラメータを受け入れます。

priority (int)

(オプション) このパスワード ポリシーの評価優先度を指定します。

戻り値

PwdPolicyPriority メソッドは、新規または既存の評価優先度設定を返します。

PwdRedirectionURL メソッド -- ユーザがリダイレクトされる場合の URL を設定または取得

PwdRedirectionURL メソッドは、無効なパスワードが指定されたときにユーザがリダイレクトされる場合の URL を設定または取得します。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。

構文

PwdRedirectionURL メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdRedirectionURL([URL])
```

パラメータ

PwdRedirectionURL メソッドは以下のパラメータを受け入れます。

URL (文字列)

(オプション) リダイレクト URL を指定します。

戻り値

PwdRedirectionURL メソッドは以下のいずれかの値を返します。

- 新規または既存の URL
- コールに失敗した場合は、**undef**

PwdRemoveRegExp メソッド -- 指定された名前タグに関連付けられた正規表現を削除

PwdRemoveRegExp メソッドは、指定された名前タグに関連付けられた正規表現を削除します。

構文

PwdRemoveRegExp メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdRemoveRegExp(tag)
```

パラメータ

メソッドは以下のパラメータを受け入れます。

tag (文字列)

削除する正規表現の名前を指定します。

戻り値

PwdRemoveRegExp メソッドは以下のいずれかの値を返します。

- 成功の場合は 0
- コールに失敗する場合は、-1

PwdReuseCount メソッド -- 使用する必要がある新しいパスワードの数を指定

PwdReuseCount メソッドは、古いパスワードを再利用するまでに使用する必要がある新しいパスワードの数を指定します。

構文

PwdReuseCount メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdReuseCount([pwdReuseCount])
```

パラメータ

PwdReuseCount メソッドは、以下のパラメータを受け入れます。

pwdReuseCount (int)

(オプション) パスワード再利用設定を指定します。

戻り値

PwdReuseCount メソッドは、新規または既存のパスワード再利用設定を返します。

PwdReuseDelay メソッド -- パスワードを再利用する前にユーザが待機する必要がある日数を指定

PwdReuseDelay メソッドは、パスワードを再利用する前にユーザが待機する必要がある日数を指定します。

構文

PwdReuseDelay メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->PwdReuseDelay([pwdReuseDelay])
```

パラメータ

PwdReuseDelay メソッドは以下のパラメータを受け入れます。

pwdReuseDelay (タイプ)

(オプション) パスワード再利用遅延設定を指定します。

戻り値

`PwdReuseDelay` メソッドは、新規または既存のパスワード再利用遅延設定を返します。

ReEnableAfterIncorrectPwd メソッド -- 不正なパスワードの入力後にユーザ アカウントを再有効化するかどうかを判定

`ReEnableAfterIncorrectPwd` メソッドは、不正なパスワードの入力後にユーザ アカウントを再有効化するかどうかを判定します。

構文

`ReEnableAfterIncorrectPwd` メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->ReEnableAfterIncorrectPwd([groupFlag])
```

パラメータ

`ReEnableAfterIncorrectPwd` メソッドは以下のパラメータを受け入れます。

groupFlag (int)

(オプション) 不正なパスワードの入力後にユーザ アカウントを再有効化するかどうかを指定します。

- 0 はユーザ アカウントを無効にします
- 1 はユーザ アカウントを有効にします

戻り値

`ReEnableAfterIncorrectPwd` メソッドは以下のいずれかの値を返します。

- 不正なパスワードの入力後にユーザ アカウントを再有効化する場合は、1
- 不正なパスワードの入力後にユーザが 1 回のログイン試行を許可される場合は、0

Save メソッド -- ポリシー ストアにパスワード ポリシーを保存

Save メソッドは、ポリシー ストアにパスワード ポリシーを保存します。

構文

Save メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->Save( )
```

パラメータ

Save メソッドは、パラメータを受け入れません。

戻り値

Save メソッドは以下のいずれかの値を返します。

- コールに成功する場合は、0
- コールに失敗する場合は、-1
- 変更を保存するためのユーザの権限が十分でない場合は、-4。
- パスとクラスが空の場合は、10。

注釈

パスワード ポリシーに対して必要なすべての変更を行った後にこのメソッドを1回コールします。変更を有効にするには、このメソッドをコールする必要があります。

StripEmbeddedWhitespace メソッド -- 新しいパスワードから埋め込まれた空白を取り去るかどうかを判定

StripEmbeddedWhitespace メソッドは、新しいパスワードから埋め込まれた空白を取り去るかどうかを判定するフラグを設定または取得します。

構文

StripEmbeddedWhitespace メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->StripEmbeddedWhitespace([stripEmbeddedFlag])
```


パラメータ

`StripEmbeddedWhitespace` メソッドは以下のパラメータを受け入れます。

stripEmbeddedFlag (int)

(オプション) 新しいパスワードから埋め込まれた空白を取り去るかどうかを指定します。

- 1 は埋め込まれた空白を取り去ります
- 0 は埋め込まれた空白を含めます

戻り値

`StripEmbeddedWhitespace` メソッドは、新規または既存のフラグ設定を返します。

StripLeadingWhitespace メソッド -- 新しいパスワードから先頭の空白を取り去るかどうかを判定

`StripLeadingWhitespace` メソッドは、新しいパスワードから先頭の空白を取り去るかどうかを判定するフラグを設定または取得します。

構文

`StripLeadingWhitespace` メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->StripLeadingWhitespace([stripLeadingFlag])
```

パラメータ

`StripLeadingWhitespace` メソッドは以下のパラメータを受け入れます。

stripLeadingFlag (int)

(オプション) パスワードから先頭の空白を取り去るかどうかを指定します。

- 1 は先頭の空白を取り去ります
- 0 は先頭の空白を含めます

戻り値

`StripLeadingWhitespace` メソッドは、新規または既存のフラグ設定を返します。

StripTrailingWhitespace メソッド -- 新しいパスワードから後続の空白を取り去るかどうかを判定

StripTrailingWhitespace メソッドは、新しいパスワードから後続の空白を取り去るかどうかを判定するフラグを設定または取得します。

構文

StripTrailingWhitespace メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->StripTrailingWhitespace([stripTrailingFlag])
```

パラメータ

StripTrailingWhitespace メソッドは以下のパラメータを受け入れます。

stripTrailingFlag (int)

(オプション) パスワードから後続の空白を取り去るかどうかを指定します。

- 1 は後続の空白を取り去ります
- 0 は後続の空白を含めます

戻り値

StripTrailingWhitespace メソッドは、新規または既存のフラグ設定を返します。

TrackLoginDetails メソッド -- 認証試行および成功したログインを追跡するかどうかを判定

TrackLoginDetails メソッドは、認証試行および成功したログインを追跡するかどうかを判定するフラグを設定または取得します。

構文

TrackLoginDetails メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->TrackLoginDetails([trackingFlag])
```

パラメータ

TrackLoginDetails メソッドは以下のパラメータを受け入れます。

trackingFlag (int)

(オプション) 以下のように、ログイン追跡を有効にするかどうかを指定します。

- 1 はログイン追跡を有効にします
- 0 はログイン追跡を無効にします

戻り値

TrackLoginDetails メソッドは、新規または既存のフラグ設定を返します。

UserDirClass メソッド -- パスワード ポリシーがディレクトリの一部に適用される場合のディレクトリ クラスを設定または取得

UserDirClass メソッドは、パスワードポリシーがディレクトリの一部に適用される場合のディレクトリ クラスを設定または取得します。

構文

UserDirClass メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->UserDirClass([path])
```

パラメータ

UserDirClass メソッドは以下のパラメータを受け入れます。

path (文字列)

(オプション) ディレクトリ クラスを指定します。

戻り値

UserDirClass メソッドは、新規または既存のディレクトリ クラスを返します。

UserDirectory メソッド -- パスワード ポリシーのユーザ ディレクトリを設定または取得

UserDirectory メソッドは、パスワード ポリシーのユーザ ディレクトリを設定または取得します。

構文

UserDirectory メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->UserDirectory([userDir])
```

パラメータ

UserDirectory メソッドは以下のパラメータを受け入れます。

userDir (PolicyMgtUserDir)

(オプション) パスワード ポリシーのユーザ ディレクトリを指定します。

戻り値

UserDirectory メソッドは、PolicyMgtUserDir オブジェクトを返します。

UserDirPath メソッド -- パスワード ポリシーがディレクトリの一部に適用される場合のディレクトリ パスを設定または取得

UserDirPath メソッドは、パスワード ポリシーがディレクトリの一部に適用される場合のディレクトリ パスを設定または取得します。

構文

UserDirPath メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPwdPolicy->UserDirPath([path])
```

パラメータ

UserDirPath メソッドは以下のパラメータを受け入れます。

path (タイプ)

(オプション) ディレクトリ パスを指定します。

戻り値

UserDirPath メソッドは、新規または既存のディレクトリ パスを返します。

CLI ポリシー メソッド

ActiveExpr メソッド -- ポリシーに関連付けられたアクティブな式を設定または取得

ActiveExpr メソッドは、ポリシーに関連付けられたアクティブな式を設定または取得します。

構文

ActiveExpr メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPolicy->ActiveExpr([activeExpr])
```

パラメータ

ActiveExpr メソッドは以下のパラメータを受け入れます。

activeExpr (文字列)

(オプション) 設定するアクティブな式を指定します。

戻り値

ActiveExpr メソッドは以下のいずれかの値を返します。

- 新規または既存のアクティブな式
- コールに失敗した場合は、**undef**

AddRule メソッド -- ポリシーにルールを追加

AddRule メソッドは、ポリシーにルールを追加します。

構文

AddRule メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPolicy->AddRule(rule)
```

パラメータ

AddRule メソッドは以下のパラメータを受け入れます。

rule (PolicyMgtRule)

追加するルールを指定します。

戻り値

AddRule メソッドは以下のいずれかの値を返します。

- コールに成功する場合は、0
- コールに失敗する場合は、-1

AddUser メソッド -- ポリシーにユーザを追加

AddUser メソッドは、ポリシーにユーザを追加します。

構文

AddUser メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPolicy->AddUser(user [, iExcludeUser] [, iRecursiveFlag] [, iANDUserFlag])
```

パラメータ

AddUser メソッドは、以下のパラメータを受け入れます。

user (PolicyMgtUser)

追加するユーザを指定します。

iExcludeUser (int)

(オプション) ユーザを除外するかどうかを指定します。

- 1 はユーザを除外します
- 0 はユーザを含めます

iRecursiveFlag (int)

(オプション) **AllowNested** フラグの設定を指定します。

- 1 は **AllowNested** フラグを設定します
- 0 は **AllowNested** フラグを無効にします

iANDUserFlag (int)

(オプション) **AND** フラグの設定を指定します。

- 1 は **AND** フラグを設定します
- 0 は **AND** フラグを無効にします

戻り値

AddUser メソッドは以下のいずれかの値を返します。

- コールに成功する場合は、0
- コールに失敗する場合は、-1

AllowNested メソッド -- AllowNested フラグを設定または取得

AllowNested メソッドは、**AllowNested** フラグを設定または取得します。

構文

AllowNested メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPolicy->AllowNested(user[, iRecursiveFlag])
```

パラメータ

`AllowNested` メソッドは、以下のパラメータを受け入れます。

`user` (PolicyMgtUser)

`AllowNested` フラグを設定または取得する場合のユーザを指定します。

`iRecursiveFlag` (int)

(オプション) 以下のように、`AllowNested` フラグの値を指定します。

- 非再帰的な評価の場合は 1
- 非再帰的な評価でない場合は 0

この値が渡されない場合、関数は `AllowNested` フラグの現在の値を返します。フラグは、特定のユーザ ディレクトリのポリシーに追加されるすべてのユーザに適用されます。

戻り値

`AllowNested` メソッドは以下のいずれかの値を返します。

- `AllowNested` フラグが正しく削除される場合は、0
- `AllowNested` フラグが正しく設定される場合は、1
- コールに失敗する場合は、-1

CreateIPHostName メソッド -- IP アドレス設定を作成

`CreateIPConfigHostName` メソッドは、メソッドに渡されるホスト名に基づいて IP アドレス設定を作成します。ポリシーを起動するには、要求を渡されたホスト名を持つマシンから行う必要があります。

構文

`CreateIPConfigHostName` メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPolicy->CreateIPConfigHostName(hostName)
```

パラメータ

`CreateIPConfigHostName` メソッドは以下のパラメータを受け入れます。

`hostName` (文字列)

ポリシーを起動するのに必要なホスト名を指定します。

戻り値

CreateIPConfigHostName メソッドは以下のいずれかの値を返します。

- PolicyMgtIPConfig オブジェクト
- コールに失敗した場合は、**undef**

CreateIPConfigRange メソッド -- IP アドレス設定を作成

CreateIPConfigRange メソッドは、メソッドに渡される IP アドレスの範囲に基づいて IP アドレス設定を作成します。ポリシーを起動するには、範囲に含まれる IP アドレスを持つマシンから行う必要があります。

構文

CreateIPConfigRange メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPolicy->CreateIPConfigRange(ipAddr1, ipAddr2)
```

パラメータ

CreateIPConfigRange メソッドは、以下のパラメータを受け入れます。

ipAddr1 (文字列)

受け入れられたアドレス範囲内の開始 IP アドレスを指定します。

ipAddr2 (文字列)

受け入れられたアドレス範囲内の終了 IP アドレスを指定します。

戻り値

CreateIPConfigRange メソッドは以下のいずれかの値を返します。

- PolicyMgtIPConfig オブジェクト
- コールに失敗した場合は、**undef**

CreateIPConfigSingleHost メソッド -- IP アドレス設定を作成

CreateIPConfigSingleHost メソッドは、メソッドに渡される IP アドレスに基づいて IP アドレス設定を作成します。ポリシーを起動するには、要求を渡された IP アドレスを持つマシンから行う必要があります。

構文

CreateIPConfigSingleHost メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPolicy->CreateIPConfigSingleHost(ipAddr)
```

パラメータ

CreateIPConfigSingleHost メソッドは以下のパラメータを受け入れます。

ipAddr (文字列)

ポリシーを起動するのに必要な IP アドレスを指定します。

戻り値

CreateIPConfigSingleHost メソッドは以下のいずれかの値を返します。

- PolicyMgtIPConfig オブジェクト
- コールに失敗した場合は、**undef**

CreateIPConfigSubnetMask メソッド -- IP アドレスおよびサブネット マスクに基づいて IP アドレス設定を作成

CreateIPConfigSubnetMask メソッドは、メソッドに渡される IP アドレスおよびサブネット マスクに基づいて IP アドレス設定を作成します。ポリシーを起動するには、要求を渡された IP アドレスおよびサブネット マスクから引き出されるサブネット アドレスから行う必要があります。

構文

CreateIPConfigSubnetMask メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPolicy->CreateIPConfigSubnetMask(ipAddr, subnetMask)
```

パラメータ

CreateIPConfigSubnetMask メソッドは、以下のパラメータを受け入れます。

ipAddr (文字列)

サブネット アドレスを引き出すために使用する IP アドレスを指定します。

subnetMask (署名なし long)

サブネット アドレスを引き出すために使用するサブネット マスクを指定します。

戻り値

CreateIPConfigSubnetMask メソッドは以下のいずれかの値を返します。

- PolicyMgtIPConfig オブジェクト
- コールに失敗した場合は、**undef**

注釈

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。たとえば、サブネット マスクが 255.255.255.128 であるとしします。バイナリ形式では以下のようになります。

```
11111111 11111111 11111111 10000000
```

左から右にカウントして、subnetMask で渡す数は 25 です。

DeleteIPConfig メソッド -- 指定された IP 設定 オブジェクトを削除

DeleteIPConfig メソッドは、指定された IP 設定オブジェクトを削除します。

構文

DeleteIPConfig メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPolicy->DeleteIPConfig(ipConfig)
```

パラメータ

DeleteIPConfig メソッドは、以下のパラメータを受け入れます。

IPConfig (PolicyMgtIPConfig)

削除する IP 設定オブジェクトを指定します。

戻り値

DeleteIPConfig メソッドは以下のいずれかの値を返します。

- 削除に成功する場合は、-0
- コールに失敗する場合は、-1

Description メソッド -- ポリシーの説明を設定または取得

Description メソッドは、ポリシーの説明を設定または取得します。

構文

Description メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPolicy->Description([policyDesc])
```

パラメータ

Description メソッドは以下のパラメータを受け入れます。

policyDesc (文字列)

設定する説明を指定します。

戻り値

Description メソッドは以下のいずれかの値を返します。

- 新規または既存のポリシーの説明
- コールに失敗した場合は、空

EnforceANDEvaluation メソッド -- ANDUser/Group フラグを設定または取得

EnforceANDEvaluation メソッドは、`iANDUserFlag` の値に応じて、ANDUser/Group フラグを設定または取得します。

構文

EnforceANDEvaluation メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPolicy->EnforceANDEvaluation(user[, iANDUserFlag])
```

パラメータ

EnforceANDEvaluation メソッドは、以下のパラメータを受け入れます。

`user` (PolicyMgtUser)

`iANDUserFlag` を設定または取得する場合のユーザを指定します。

`iANDUserFlag` (int)

(オプション) 以下のように、AND 評価を強制するかどうかを指定します。

AND 評価を強制するには 1

AND 評価を削除するには 0

この引数が渡されない場合、関数は `iANDUserFlag` の現在の値を返します。このフラグは、特定のユーザディレクトリのポリシーに追加されるすべてのユーザに適用されます。

戻り値

EnforceANDEvaluation メソッドは以下のいずれかの値を返します。

- ANDUser/Group フラグが正しく削除される場合は、0
- ANDUser/Group フラグが正しく設定される場合は、1
- コールに失敗する場合は、-1

ExcludeUser メソッド -- ポリシーからユーザを除外または含む

ExcludeUser メソッドは、*iExcludeFlag* の値に応じて、ユーザをポリシーから除外または含みます。

構文

ExcludeUser メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPolicy->ExcludeUser(user[, iExcludeFlag])
```

パラメータ

ExcludeUser メソッドは、以下のパラメータを受け入れます。

user (PolicyMgtUser)

除外または含むユーザを指定します。

iExcludeFlag (int)

(オプション) 以下のように、指定されたユーザを除外するかどうかを指定します。

- ユーザを除外するには 1
- ユーザを含むには 0

この引数が渡されない場合、関数は *iExcludeFlag* の現在の値を返します。

戻り値

ExcludeUser メソッドは以下のいずれかの値を返します。

- ユーザが正しく含まれる場合は、0
- ユーザが正しく除外される場合は、1
- コールに失敗する場合は、-1

GetAllIPConfigs メソッド -- ポリシー内のすべての IP アドレス制限オブジェクトを取得

GetAllIPConfigs メソッドは、ポリシー内のすべての IP アドレス制限オブジェクトを取得します。

構文

GetAllIPConfigs メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPolicy->GetAllIPConfigs( )
```

パラメータ

GetAllIPConfigs メソッドは、パラメータを受け入れません。

戻り値

GetAllIPConfigs メソッドは以下のいずれかの値を返します。

- PolicyMgtIPConfig オブジェクトの配列
- IP アドレス制限オブジェクトが見つからない場合は、**undef**

注釈

IP アドレス制限および IP アドレス制限タイプの詳細については、PolicyMgtIPConfig->GetType メソッドを参照してください。

GetAllRules メソッド -- ポリシーと関連付けられたすべてのルールを取得

GetAllRules メソッドは、ポリシーと関連付けられたすべてのルールを取得します。

構文

GetAllRules メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPolicy->GetAllRules()
```

パラメータ

GetAllRules メソッドは、パラメータを受け入れません。

戻り値

GetAllRules メソッドは以下のいずれかの値を返します。

- PolicyMgtRule オブジェクトの配列
- ルールが見つからない場合またはコールに失敗した場合は、**undef**

GetAllUsers メソッド -- ポリシーと関連付けられたすべてのユーザを取得

GetAllUsers メソッドは、ポリシーと関連付けられたすべてのユーザを取得します。ユーザディレクトリが指定される場合は、そのディレクトリと関連付けられたユーザのみが取得されます。

構文

GetAllUsers メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPolicy->GetAllUsers([userDir])
```

パラメータ

GetAllUsers メソッドは以下のパラメータを受け入れます。

userDir (PolicyMgtUserDir)

(オプション) このユーザディレクトリと関連付けられたユーザのみが取得されることを指定します。

戻り値

GetAllUsers メソッドは以下のいずれかの値を返します。

- PolicyMgtUser オブジェクトの配列
- ユーザが見つからなかった場合またはコールに失敗した場合は、**undef**

IsEnabled メソッド -- ポリシーを有効化または無効化

IsEnabled メソッドは、ポリシーを有効化または無効化します。

構文

IsEnabled メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPolicy->IsEnabled([enableFlag])
```

パラメータ

IsEnabled メソッドは以下のパラメータを受け入れます。

enableFlag (int)

(オプション) 以下のようにポリシーを有効または無効にするかどうかを指定します。

- 0 はポリシーを無効にします。
- 1 はポリシーを有効にします。

戻り値

IsEnabled メソッドは以下のいずれかの値を返します。

- ポリシーが有効な場合は、1。
- ポリシーが無効な場合は、0。
- コールに失敗する場合は、-1

Name メソッド -- ポリシー名を設定または取得

Name メソッドは、ポリシー名を設定または取得します。

構文

Name メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPolicy->Name([policyName])
```

パラメータ

Name メソッドは以下のパラメータを受け入れます。

policyName (文字列)

(オプション) ポリシーに割り当てる名前を指定します。

戻り値

Name メソッドは以下のいずれかの値を返します。

- 新規または既存のポリシー名
- コールに失敗した場合は、**undef**

RemoveResponse メソッド -- ポリシー内の設定されたルール用のレスポンスを削除

RemoveResponse メソッドは、ポリシー内の設定されたルール用のレスポンスを削除します。

構文

RemoveResponse メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPolicy->RemoveResponse(rule)
```

パラメータ

RemoveResponse メソッドは以下のパラメータを受け入れます。

rule (PolicyMgtRule)

そのレスポンスを削除するルールを指定します。

戻り値

RemoveResponse メソッドは以下のいずれかの値を返します。

- コールに成功する場合は、0
- コールに失敗する場合は、-1

RemoveRule メソッド -- ポリシーから指定されたルールを削除

RemoveRule メソッドは、ポリシーから指定されたルールを削除します。

構文

RemoveRule メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPolicy->RemoveRule(rule)
```

パラメータ

RemoveRule メソッドは以下のパラメータを受け入れます。

rule (PolicyMgtRule)

削除するルールを指定します。

戻り値

RemoveRule メソッドは以下のいずれかの値を返します。

- コールに成功する場合は、0
- コールに失敗する場合は、-1

RemoveUser メソッド -- ポリシーからユーザを削除

RemoveUser メソッドは、ポリシーからユーザを削除します。

構文

RemoveUser メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPolicy->RemoveUser(user)
```

パラメータ

RemoveUser メソッドは、以下のパラメータを受け入れます。

user (PolicyMgtUser)

削除するユーザを指定します。

戻り値

RemoveUser メソッドは以下のいずれかの値を返します。

- コールに成功する場合は、0
- コールに失敗する場合は、-1

SetResponse メソッド -- ポリシー内の設定されたルール用のレスポンスを設定

SetResponse メソッドは、ポリシー内の設定されたルール用のレスポンスを設定します。

構文

SetResponse メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtPolicy->SetResponse(rule, response)
```

パラメータ

SetResponse メソッドは、以下のパラメータを受け入れます。

rule (PolicyMgtRule)

そのレスポンスを設定するルールを指定します。

response (PolicyMgtResponse)

設定するレスポンスを指定します。

戻り値

SetResponse メソッドは以下のいずれかの値を返します。

- コールに成功する場合は、-0。
- コールに失敗する場合は、-1。

CLI ポリシー サーバ接続メソッド

GetPorts メソッド -- 非推奨

GetPorts メソッドは SiteMinder v6.0 で非推奨です。GetServerPort メソッドに置き換えられます。

GetServerAddress メソッド -- ポリシー サーバのホスト名または IP アドレスを取得

GetServerAddress メソッドは、ポリシー サーバのホスト名または IP アドレスを取得します。

構文

GetServerAddress メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtServer->GetServerAddress()
```

パラメータ

GetServerAddress メソッドは、パラメータを受け入れません。

戻り値

GetServerAddress メソッドは以下のいずれかの値を返します。

- ポリシー サーバ ホスト名または IP アドレスを表す文字列。
- コールに失敗した場合は、**undef**

GetServerPort メソッド -- ポリシー サーバまたはサーバ クラスタの TCP ポートを取得

GetServerPort メソッドは以下のいずれを返します。

- クラスタ化されたサーバの場合は、ポリシー サーバ ポートが含まれる配列を取得します。
- 非クラスタ化サーバの場合は、アカウントティング、認証および許可ポートの配列をその順序で取得します。

構文

GetServerPort メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtServer->GetServerPort()
```

パラメータ

GetServerPort メソッドは、パラメータを受け入れません。

戻り値

GetServerPort メソッドは以下のいずれかの値を返します。

- ホスト ポートの配列
- コールに失敗した場合は、**undef**

注釈

SiteMinder v6.0 に導入された単一プロセス ポリシー サーバが、もともとは別個の認証、許可およびアカウントティングプロセスを 1 つの結合プロセスに組み合わせることにより、複数の要求が 1 つの TCP ポートを通ります。その結果、配列で取得されるポート番号はすべて同じです。

CLI レルム メソッド

Agent メソッド -- レルムのエージェントを設定または取得

Agent メソッドは、レルムのエージェントを設定または取得します。

構文

Agent メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRealm->Agent([agent])
```

パラメータ

Agent メソッドは、以下のパラメータを受け入れます。

agent (PolicyMgtAgent)

(オプション) レルムに設定するエージェントを指定します。

戻り値

Agent メソッドは、コールが成功した場合に、新規または既存の PolicyMgtAgent オブジェクトか PolicyMgtGroup オブジェクトを返します。

コールが失敗した場合、このメソッドは **undef** を返します

AuthScheme メソッド -- レルムの認証方式を設定または取得

AuthScheme メソッドは、レルムの認証方式を設定または取得します。

構文

AuthScheme メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRealm->AuthScheme([authScheme])
```

パラメータ

AuthScheme メソッドは以下のパラメータを受け入れます。

authScheme (PolicyMgtAuthScheme)

(オプション) レルムに設定する認証方式を指定します。

戻り値

AuthScheme メソッドは以下のいずれかの値を返します。

- 新規または既存のレルム用の PolicyMgtAuthScheme オブジェクト
- コールに失敗した場合は、**undef**

AzUserDir メソッド -- レルムの許可ユーザ ディレクトリを設定または取得

AzUserDir メソッドは、レルムの許可ユーザ ディレクトリを設定または取得します。

構文

AzUserDir メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRealm->AzUserDir([dir])
```

パラメータ

AzUserDir メソッドは以下のパラメータを受け入れます。

dir (PolicyMgtUserDirectory)

(オプション) レルムに設定する許可ユーザ ディレクトリを指定します。

戻り値

AzUserDir メソッドは以下のいずれかの値を返します。

- 新規または既存のレルム用の PolicyMgtUserDir オブジェクト
- 存在しない場合、またはコールに失敗した場合は、**undef**

CreateChildRealm メソッド -- 子レalmを作成および設定

CreateChildRealm メソッドは、このメソッドのコール元レalm下のレalmディレクトリを作成および設定します。

構文

CreateChildRealm メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRealm->CreateChildRealm(realName, agent, authScheme [,  
realmDesc] [, resFilter] [, procAuthEvents] [, procAzEvents] [, protectAll] [,  
maxTimeout] [, idleTimeout] [, syncAudit] [, azUserDir] [, regScheme])
```

パラメータ

CreateChildRealm メソッドは、以下のパラメータを受け入れます。

realName (文字列)

レalmの名前を指定します。

agent (PolicyMgtAgent)

レalmのエージェントまたはエージェントのグループを指定します。

authScheme (PolicyMgtAuthScheme)

レalmと関連付ける認証方式を指定します。

realmDesc (文字列)

(オプション) レalmの説明を指定します。

resFilter (文字列)

(オプション) レalmのリソース フィルタを指定します。

procAuthEvents (int)

(オプション) 認証イベントを処理するためのフラグを指定します。
有効にするには **1** に設定し、無効にするには **0** に設定します。デフォルトは有効化です。

procAzEvents (int)

(オプション) 許可イベントを処理するためのフラグを指定します。
有効にするには **1** に設定し、無効にするには **0** に設定します。デフォルトは有効化です。

***protectAll* (int)**

(オプション) デフォルトのリソース保護をアクティブにするためのフラグを指定します。有効にするには **1** に設定し、無効にするには **0** に設定します。デフォルトは有効化です。

***maxTimeout* (int)**

(オプション) 再認証が必要になる前に、ユーザがレalmにアクセスできる最大時間を秒単位で指定します。デフォルトは **7200** です (2 時間)。

***idleTimeout* (int)**

(オプション) 再認証が必要になる前に、ユーザがレalmで非アクティブ状態を維持できる最大時間を指定します。デフォルトは **3600** です (1 時間)。

***syncAudit* (int)**

(オプション) 同期監査を有効にするフラグを指定します。有効にするには **1** に設定し、無効にするには **0** に設定します。このフラグが有効な場合は、リソースへのアクセスを許可する前に、**SiteMinder** はポリシー サーバおよびエージェント アクションをログ記録します。デフォルトは有効化です。

***azUserDir* (PolicyMgtUserDir)**

(オプション) レalm内のユーザが許可されるディレクトリを指定します。デフォルトは、デフォルト ディレクトリです。

***regScheme* (PolicyMgtRegScheme)**

(オプション) レalm内のリソースにアクセスする新しいユーザを登録するために使用される登録方式を指定します。

戻り値

CreateChildRealm メソッドは以下のいずれかの値を返します。

- PolicyMgtRealm オブジェクト
- コールに失敗した場合は、**undef**

注釈

このメソッドは、非永続セッション用に設定されるレルムを作成します。**SiteMinder 5.0** 永続セッション用のレルムを設定するには、管理 UI でレルムを編集します。

注: ポリシー管理 API は、以下のようにそのメソッドがコールされたオブジェクトの直接の子孫であるレルムのみを操作します。

- ドメイン下のレルムの場合。ドメイン オブジェクト内のトップレベルのレルムのみを操作できます。
- レルム下のレルムの場合。親レルムの直下のレルムのみを操作できます。

CreateRule メソッド --レルム下のルールを作成および設定

CreateRule メソッドは、レルム下のルールを作成および設定します。

構文

CreateRule メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRealm->CreateRule( ruleName [, ruleDesc] [, action] [, resource]  
[, allowAccess] [, regexMatch] [, activeExpr] [, isEnabled] )
```

パラメータ

CreateRule メソッドは、以下のパラメータを受け入れます。

ruleName (文字列)

 ルールの名前を指定します

ruleDesc (文字列)

 (オプション) ルールの説明を指定します。

action (文字列)

(オプション) ルールが実行されるアクションのタイプを指定します。以下のいずれかのアクション。

アクション タイプが **Web** エージェント アクションの場合は、以下の 1 つ以上の **HTTP** アクションを使用します。複数のアクションを区切るには、カンマを使用します。

- **GET**。 **HTTP** を使用して表示するためのリソースを取得します。
- **POST**。 **HTTP** を使用してユーザ提供の情報を **POST** 送信します。
- **PUT**。 レガシー **HTTP** アクションをサポートします。

アクション タイプが認証イベントの場合

- **OnAuthAccept**。 ユーザ認証が正常に行われたときに発生します。
- **OnAuthAttempt**。 ユーザ名が入力されていないために、ユーザ認証が失敗したときに発生します。
- **OnAuthChallenge**。 カスタム認証方式で使用し、レスポンスを発行することができます。
- **OnAuthReject**。 ユーザが認証に失敗したときに発生します。
- **OnAuthUserNotFound**。 アクティブ レスポンスをトリガするために使用します。

アクション タイプが許可イベントの場合

- **OnAccessAccept**。 ユーザに対してリソースへのアクセスが **SiteMinder** によって問題なく許可したときに発生します。
- **OnAccessReject**。 リソースへのアクセスが **SiteMinder** によって許可されなかったことが原因でユーザを拒否するときに発生します。

resource (文字列)

(オプション) ルールによって保護されるリソースを指定します。この値はアクション タイプ認証イベントには適用されません。

allowAccess (int)

(オプション) 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ためのフラグを指定します。1 はアクセスを許可し、0 はアクセスを拒否します。このフラグは、タイプ **GET**、**PUT** または **POST** (またはその両方) の **アクション** 値にのみ適用されます。デフォルトは 1 です。

regexMatch (int)

(オプション) **resource** フィールドでの正規表現パターン一致を許可するためのフラグを指定します。**1** は正規表現一致を許可します。**0** は正規表現一致を拒否します。このフラグはアクション タイプ 認証イベントには適用されません。デフォルトは **0** です。

activeExpr (文字列)

(オプション) ルールに関連付けられるアクティブな式を指定します。

isEnabled (int)

(オプション) ルールを有効または無効にするフラグを指定します。有効にするには **1**、無効にするには **0** を指定します。デフォルトは有効化です。

戻り値

CreateRule メソッドは以下のいずれかの値を返します。

- PolicyMgtRule オブジェクト
- コールに失敗した場合は、**undef**

DeleteChildRealm メソッド -- レルム内のトップレベルレルムを削除

DeleteChildRealm メソッドは、レルム内のトップレベル レルムを削除します。

構文

DeleteChildRealm メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRealm->DeleteChildRealm(realm)
```

パラメータ

DeleteChildRealm メソッドは以下のパラメータを受け入れます。

realm (PolicyMgtRealm)

削除する子レルムを指定します。

戻り値

DeleteChildRealm メソッドは以下のいずれかの値を返します。

- 成功した場合またはレルムが見つからなかった場合は 0
- コールに失敗する場合は、-1

DeleteRule メソッド -- レルム内の既存のルールを削除

DeleteRule メソッドは、レルム内の既存のルールを削除します。

構文

DeleteRule メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRealm->DeleteRule(rule)
```

パラメータ

DeleteRule メソッドは以下のパラメータを受け入れます。

rule (PolicyMgtRule)

削除するルールを指定します。

戻り値

DeleteRule メソッドは以下のいずれかの値を返します。

- 成功の場合は 0
- コールが失敗する場合、またはルールがルールを削除するために使用するレルムの一部でない場合は、-1

Description メソッド -- レルムの説明を設定または取得

Description メソッドは、レルムの説明を設定または取得します。

構文

Description メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRealm->Description([realmDesc])
```

パラメータ

Description メソッドは以下のパラメータを受け入れます。

realmDesc (文字列)

(オプション) レルムに割り当てる説明を指定します。

戻り値

Description メソッドは以下のいずれかの値を返します。

- 新規または既存のレルムの説明
- コールに失敗した場合は、空の文字列

Flush メソッド -- リソース キャッシュからレルムをフラッシュ

Flush メソッドは、リソース キャッシュからレルムをフラッシュします。

構文

Flush メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRealm->Flush()
```

パラメータ

Flush メソッドは、パラメータを受け入れません。

戻り値

Flush メソッドは以下のいずれかの値を返します。

- 成功の場合は 0
- コールに失敗する場合は、-1

GetAllChildRealms メソッド -- レルム内のすべてのトップレベル レルムを取得

GetAllChildRealms メソッドは、レルム内のすべてのトップレベル レルムを取得します。子のみを返します。

構文

GetAllChildRealms メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRealm->GetAllChildRealms()
```

パラメータ

GetAllChildRealms メソッドは、パラメータを受け入れません。

戻り値

GetAllChildRealms メソッドは以下のいずれかの値を返します。

- PolicyMgtRealm オブジェクトの配列
- コールに失敗した場合は、**undef**

GetAllRules メソッド -- レルムと関連付けられたルールを取得

GetAllRules メソッドは、レルムと関連付けられたルールを取得します。

構文

GetAllRules メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRealm->GetAllRules()
```

パラメータ

GetAllRules メソッドは、パラメータを受け入れません。

戻り値

GetAllRules メソッドは以下のいずれかの値を返します。

- PolicyMgtRule オブジェクトの配列
- コールに失敗した場合は、**undef**

GetChildRealm メソッド -- レalm下のトップレベルの子レalmを取得

GetChildRealm メソッドは、レalm下のトップレベルの子レalmを取得します。このメソッドは子レalmのみを検索します。

構文

GetChildRealm メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRealm->GetChildRealm( realmName)
```

パラメータ

GetChildRealm メソッドは以下のパラメータを受け入れます。

realmName (文字列)

子レalmを確認するレalmを指定します。

戻り値

GetChildRealm メソッドは以下のいずれかの値を返します。

- PolicyMgtRealm オブジェクト
- コールに失敗した場合またはレalmが存在しない場合は、**undef**

GetDomain メソッド -- レalmと関連付けられたドメインを取得

GetDomain メソッドは、レalmと関連付けられたドメインを取得します。

構文

GetDomain メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRealm->GetDomain()
```

パラメータ

GetDomain メソッドは、パラメータを受け入れません。

戻り値

GetDomain メソッドは以下のいずれかの値を返します。

- 既存のレルム用の **PolicyMgtDomain** オブジェクト
- コールに失敗した場合は、**undef**

GetRule メソッド -- レルム内の既存のルールを取得

GetRule メソッドは、レルム内の既存のルールを取得します。

構文

GetRule メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRealm->GetRule(ruleName)
```

パラメータ

GetRule メソッドは以下のパラメータを受け入れます。

ruleName (文字列)

取得するルールの名前を指定します。

戻り値

GetRule メソッドは以下のいずれかの値を返します。

- **PolicyMgtRule** オブジェクト
- コールに失敗した場合または指定されたルールが存在しない場合は、**undef**

IdleTimeout メソッド -- ユーザがレルムで非アクティブ状態を維持できる最大時間を設定または取得

IdleTimeout メソッドは、再認証が必要になる前に、ユーザがレルムで非アクティブ状態を維持できる最大時間を設定または取得します。

構文

IdleTimeout メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRealm->IdleTimeout([idleTimeout])
```

パラメータ

IdleTimeout メソッドは以下のパラメータを受け入れます。

idleTimeout (タイプ)

(オプション) アイドル タイムアウト値を秒単位で指定します。

戻り値

IdleTimeout メソッドは以下のいずれかの値を返します。

- 引数が指定されない場合は、既存のタイムアウト値
- *idleTimeout* が指定される場合は、新しいタイムアウト値
- コールに失敗する場合は、-1

MaxTimeout メソッド -- ユーザがレルムにアクセスできる最大時間を設定または取得

MaxTimeout メソッドは、再認証が必要になる前に、ユーザがレルムにアクセスできる最大時間を設定または取得します。

構文

MaxTimeout メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRealm->MaxTimeout([maxTimeout])
```

パラメータ

MaxTimeout メソッドは以下のパラメータを受け入れます。

maxTimeout (int)

(オプション) 最大タイムアウト値を秒単位で指定します。

戻り値

MaxTimeout メソッドは以下のいずれかの値を返します。

- 引数が指定されない場合は、既存の最大タイムアウト値
- *maxTimeout* が指定される場合は、新しい最大タイムアウト値。
- コールに失敗する場合は、-1

Name メソッド -- レルム名を設定または取得

Name メソッドは、レルム名を設定または取得します。

構文

Name メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRealm->Name([realmName])
```

パラメータ

Name メソッドは以下のパラメータを受け入れます。

realmName (文字列)

(オプション) レルムに割り当てる名前を指定します。

戻り値

Name メソッドは以下のいずれかの値を返します。

- 新規または既存のレルム名
- コールに失敗した場合は、**undef**

ProcessAuEvents メソッド -- レルム内の認証イベントフラグを設定または取得

ProcessAuEvents メソッドは、レルムの認証イベント フラグを設定または取得します。

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、このフラグを **0** に設定します。

構文

ProcessAuEvents メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRealm->ProcessAuEvents([authFlag])
```

パラメータ

ProcessAuEvents メソッドは以下のパラメータを受け入れます。

authFlag (int)

(オプション) 以下のように、認証イベントが処理されるかどうかを指定します。

- イベント処理を有効にするには **1**
- イベント処理を無効にするには **0**

戻り値

ProcessAuEvents メソッドは以下のいずれかの値を返します。

- 認証イベントを処理する場合は、**1**
- 認証イベントを処理しない場合は、**0**
- コールに失敗する場合は、**-1**

ProcessAzEvents メソッド -- レルム内の許可イベントフラグを設定または取得

ProcessAzEvents メソッドは、レルム内の許可イベント フラグを設定または取得します。

構文

ProcessAzEvents メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRealm->ProcessAzEvents([azFlag])
```

パラメータ

ProcessAzEvents メソッドは以下のパラメータを受け入れます。

azFlag (int)

(オプション) 許可イベント処理を有効にするかどうかを指定します。

- 1 はイベント処理を有効化
- 0 はイベント処理を無効化

戻り値

ProcessAzEvents メソッドは以下のいずれかの値を返します。

- 許可イベントを処理する場合は、1
- 許可イベントを処理しない場合は、0
- コールに失敗する場合は、-1

注釈

認可イベント処理はパフォーマンスに影響します。レルムのルールが許可イベントによってトリガされない場合は、このフラグを 0 に設定します。

ProtectResource メソッド -- 現在のリソース保護フラグの例を設定または取得

ProtectResource メソッドは、現在のリソース保護フラグを設定または取得します。

構文

ProtectResource メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRealm->ProtectResource([protectFlag])
```

パラメータ

ProtectResource メソッドは以下のパラメータを受け入れます。

protectFlag (int)

(オプション) 以下のように、リソース保護を有効にするかどうかを指定します。

- 1 はリソースを保護します
- 0 はリソースを未保護の状態にします

戻り値

ProtectResource メソッドは以下のいずれかの値を返します。

- 引数が指定されない場合は、既存のリソース保護状態 (0 または 1)。
- フラグ値がメソッドに渡される場合は、新しいリソース保護状態。
- コールに失敗した場合は、**undef**

RegScheme メソッド -- レルムの登録方式を設定または取得

RegScheme メソッドは、レルムの登録方式を設定または取得します。

構文

RegScheme メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRealm->RegScheme([regScheme])
```

パラメータ

RegScheme メソッドは以下のパラメータを受け入れます。

regScheme (PolicyMgtRegScheme)

(オプション) 設定する登録方式を指定します。

戻り値

RegScheme メソッドは以下のいずれかの値を返します。

- PolicyMgtRegScheme オブジェクト
- コールに失敗した場合または登録方式が存在しない場合は、**undef**

SessionDrift メソッド -- セッションドリフトを設定または取得

SessionDrift メソッドは、レルムのセッション ドリフトを設定または取得します。つまり、永続レルム上で有効な場合の検証期間（秒単位）を表します。

構文

SessionDrift メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRealm->SessionDrift([SessionDrift])
```

パラメータ

SessionDrift メソッドは以下のパラメータを受け入れます。

SessionDrift (int)

（オプション）新しい値を指定するか、または指定されていないときに現在値を返します。

戻り値

SessionDrift メソッドは以下のいずれかの値を返します。

- 新規または既存のセッション ドリフト値
- 失敗の場合は、Sm_PolicyApi_Status_t エラー コード

注: -1 (Sm_PolicyApi_Failure) は、セッション ドリフトが有効でないことを示す有効な戻り値です。

ResourceFilter メソッド -- レルム リソース フィルタを設定または取得

ResourceFilter メソッドは、レルム リソース フィルタを設定または取得します。

構文

ResourceFilter メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRealm->ResourceFilter([rFilter])
```


パラメータ

`ResourceFilter` メソッドは以下のパラメータを受け入れます。

rFilter (文字列)

(オプション) 設定するレルム リソース フィルタを指定します。

戻り値

`ResourceFilter` メソッドは以下のいずれかの値を返します。

- 新規または既存のレルム フィルタ
- コールに失敗した場合は、**undef**

SyncAudit メソッド -- 同期監査フラグを設定または取得

`SyncAudit` メソッドは、同期監査フラグを設定または取得します。このフラグが有効な場合は、リソースへのアクセスを許可する前に、**SiteMinder** はポリシー サーバおよびエージェント アクションをログ記録します。

構文

`SyncAudit` メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRealm->SyncAudit([syncFlag])
```

パラメータ

`SyncAudit` メソッドは以下のパラメータを受け入れます。

syncFlag (int)

(オプション) 以下のように、同期監査を有効にするかどうかを指定します。

- 1 は同期監査を有効にします
- 0 は同期監査を無効にします

戻り値

SyncAudit メソッドは以下のいずれかの値を返します。

- 引数が指定されない場合は、既存の同期監査値（0 または 1）。
- フラグ引数がメソッドに渡される場合は、新しい同期監査値。
- コールに失敗する場合は、-1

CLI 登録方式メソッド

Description メソッド -- 登録方式の説明を設定または取得

Description メソッドは、登録方式の説明を設定または取得します。

構文

Description メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRegScheme->Description([regDesc])
```

パラメータ

Description メソッドは以下のパラメータを受け入れます。

regDesc (文字列)

（オプション）登録方式の説明を指定します。

戻り値

Description メソッドは以下のいずれかの値を返します。

- 登録方式の新規または既存の説明。
- コールに失敗した場合は、空の文字列

EnableLogging メソッド -- 登録方式のログ記録を有効化または無効化

EnableLogging メソッドは、登録方式のログ記録を有効化または無効化します。

構文

EnableLogging メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRegScheme->EnableLogging([logFlag])
```

パラメータ

EnableLogging メソッドは以下のパラメータを受け入れます。

logFlag (int)

(オプション) 以下のように、登録方式のログ記録を有効にするかどうかを指定します。

- 1 はログ記録を有効にします
- 0 はログ記録を無効にします

戻り値

EnableLogging メソッドは以下のいずれかの値を返します。

- ログ記録が有効な場合は、1
- ログ記録が無効な場合は、0
- コールに失敗する場合は、-1

Name メソッド -- 登録方式名を設定または取得

Name メソッドは、登録方式名を設定または取得します。

構文

Name メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRegScheme->Name([regName])
```

パラメータ

Name メソッドは、以下のパラメータを受け入れます。

regName (文字列)

(オプション) 登録方式名を指定します。

戻り値

Name メソッドは以下のいずれかの値を返します。

- 新規または既存の登録方式名
- コールに失敗した場合は、**undef**

TemplatePath メソッド -- 登録方式のパスを設定または取得

TemplatePath メソッドは、登録方式テンプレートのパスを設定または取得します。

構文

TemplatePath メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRegScheme->TemplatePath([path])
```

パラメータ

TemplatePath メソッドは、以下のパラメータを受け入れます。

path (文字列)

(オプション) 登録方式テンプレートのパスを指定します。

戻り値

TemplatePath メソッドは以下のいずれかの値を返します。

- 新規または既存のテンプレート パス
- コールに失敗した場合は、**undef**

UserDirectory メソッド -- 登録方式のユーザ ディレクトリを設定または取得

UserDirectory メソッドは、登録方式のユーザ ディレクトリを設定または取得します。

構文

UserDirectory メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRegScheme->UserDirectory([userDir])
```

パラメータ

UserDirectory メソッドは、以下のパラメータを受け入れます。

userDir (PolicyMgtUserDir)

(オプション) 登録方式のユーザ ディレクトリを指定します。

戻り値

UserDirectory メソッドは以下のいずれかの値を返します。

- PolicyMgtUserDir オブジェクト
- コールが失敗した場合またはユーザ ディレクトリが存在しない場合は、**undef**

WelcomePageURL メソッド -- 登録方式のようこそページ URL を設定または取得

WelcomePageURL メソッドは、登録方式のようこそページ URL を設定または取得します。

構文

WelcomePageURL メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRegScheme->WelcomePageURL([URL])
```

パラメータ

WelcomePageURL メソッドは以下のパラメータを受け入れます。

URL (文字列)

(オプション) 登録方式のようこそページ URL を指定します。ユーザは正しく登録した後にこのページにリダイレクトされます。

形式: `http://my.acme.com/hr/welcome.htm`

戻り値

WelcomePageURL メソッドは以下のいずれかの値を返します。

- 新規または既存の URL
- コールに失敗した場合は、**undef**

CLI ポリシー管理レスポンス メソッド

CreateAttribute メソッド -- レスポンスのスタティックレスポンス属性を作成

CreateAttribute メソッドは、レスポンスのスタティック レスポンス属性を作成します。

構文

CreateAttribute メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtResponse->CreateAttribute(attrName, varValue [, TTL])
```

パラメータ

CreateAttribute メソッドは、以下のパラメータを受け入れます。

attrName (文字列)

作成する属性の名前を指定します。有効な属性名はレスポンスと関連付けられたエージェントのタイプに応じて変わります。

エージェントタイプは、レスポンスの作成時に表示される [SiteMinder レスポンス] ダイアログで指定されます。特定のエージェントタイプと関連付けられた属性のリストを確認するには、[SiteMinder レスポンス] ダイアログでエージェントタイプを選択し、[作成] をクリックします。その後に、SiteMinder レスポンス属性エディタの [属性] フィールドで選択肢を表示します。

たとえば、SiteMinder Web エージェントタイプでレスポンスを作成する場合は、以下のいずれかのレスポンス属性を作成できます。

- WebAgent-HTTP-Header-Variable
- WebAgent-HTTP-Cookie-Variable
- WebAgent-OnAccept-Redirect
- WebAgent-OnAccept-Text
- WebAgent-OnAuthAccept-Session-Idle-Timeout
- WebAgent-OnAuthAccept-Session-Max-Timeout
- WebAgent-OnReject-Redirect
- WebAgent-OnReject-Text

varValue (文字列)

スタティック属性の値を指定します。この値が [SiteMinder レスポンス] ダイアログの [値] 列に表示されます。値は変数か Cookie 値または名前/値ペアを表します。値と共に名前も指定する必要がある場合は、フォーム名 = 値を使用します。たとえば、属性 WebAgent-HTTP-Header-Variable には名前と値のペアが必要です。名前が show_content で、値がはいの場合は、varValue に show_content=yes を割り当てます。

TTL (int)

(オプション) レスポンス属性の値が再計算されるまでに経過できる秒単位の時間量を指定します。

戻り値

CreateAttribute メソッドは以下のいずれかの値を返します。

- PolicyMgtResponseAttr オブジェクト
- コールに失敗した場合は、**undef**

注釈

コマンドライン インターフェースでは、レスポンス属性としてユーザ属性または DN 属性を作成できません。

PolicyMgtResponse->CreateActiveAttribute メソッドと
PolicyMgtResponse->CreateVariableAttribute メソッドの説明も参照してください。

DeleteAttribute メソッド -- レスポンス内のレスポンス属性を削除

DeleteAttribute メソッドは、レスポンス内のレスポンス属性を削除します。

構文

DeleteAttribute メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtResponse->DeleteAttribute(respAttr)
```

パラメータ

DeleteAttribute メソッドは以下のパラメータを受け入れます。

respAttr (PolicyMgtResponseAttr)

削除するレスポンス属性を指定します。

戻り値

DeleteAttribute メソッドは以下のいずれかの値を返します。

- 成功の場合は 0
- コールに失敗する場合は、-1

Description メソッド -- レスポンスの説明を設定または取得

Description メソッドは、レスポンスの説明を設定または取得します。

構文

Description メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtResponse->Description([resDesc])
```

パラメータ

Description メソッドは以下のパラメータを受け入れます。

resDesc (文字列)

(オプション) レスポンスの説明を指定します。

戻り値

Description メソッドは以下のいずれかの値を返します。

- 新規または既存のレスポンスの説明
- コールに失敗した場合は、空の文字列

GetAllAttributes メソッド -- 設定されたレスポンス属性のリストを取得

GetAllAttributes メソッドは、設定されたレスポンス属性のリストを取得します。

構文

GetAllAttributes メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtResponse->GetAllAttributes()
```

パラメータ

GetAllAttributes メソッドは、パラメータを受け入れません。

戻り値

GetAllAttributes メソッドは以下のいずれかの値を返します。

- PolicyMgtResponseAttr オブジェクトの配列
- コールに失敗した場合は、**undef**

Name メソッド -- レスポンス名を設定または取得

Name メソッドは、レスポンス名を設定または取得します。

構文

Name メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtResponse->Name([resName])
```

パラメータ

Name メソッドは以下のパラメータを受け入れます。

resName (文字列)

(オプション) レスポンス名を指定します。

戻り値

Name メソッドは以下のいずれかの値を返します。

- 新規または既存のレスポンス名
- コールに失敗した場合は、**undef**

CLI レスpons属性メソッド

GetAgentTypeAttrName メソッド -- エージェント タイプ属性の名前を取得

GetAgentTypeAttrName メソッドは、このレスpons属性に関連付けられたエージェント タイプ属性の名前を取得します。

構文

GetAgentTypeAttrName メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtResponseAttr->GetAgentTypeAttrName()
```

パラメータ

GetAgentTypeAttrName メソッドは、パラメータを受け入れません。

戻り値

GetAgentTypeAttrName メソッドは以下のいずれかの値を返します。

- エージェント タイプ属性名 (たとえば、WebAgent-OnReject-Redirect)。
- コールに失敗した場合は、**undef**

GetTTL メソッド -- Time To Live (TTL) 設定を取得

GetTTL メソッドは、Time To Live (TTL) 設定を取得します。

構文

GetTTL メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtResponseAttr->GetTTL()
```

パラメータ

GetTTL メソッドは、パラメータを受け入れません。

戻り値

GetTTL メソッドは以下のいずれかの値を返します。

- 既存の TTL 設定
- コールに失敗した場合は、**undef**

GetValue メソッド -- レスポンス属性値を取得

GetValue メソッドは、レスポンス属性値を取得します。

構文

GetValue メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtResponseAttr->GetValue()
```

パラメータ

GetValue メソッドは、パラメータを受け入れません。

戻り値

GetValue メソッドは以下のいずれかの値を返します。

- レスポンス属性の既存の値。
- コールに失敗した場合は、**undef**

CLI ルール メソッド

AccessType メソッド -- 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を設定または取得

AccessType メソッドは、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を設定または取得します。

構文

AccessType メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRule->AccessType([allowAccess])
```

パラメータ

`AccessType` メソッドは以下のパラメータを受け入れます。

`allowAccess` (int)

(オプション) 以下のように、ルールでリソースへのアクセスを許可するかどうかを指定します。

- ルールでリソースへのアクセスを許可する場合は、1
- ルールでリソースへのアクセスを拒否する場合は、0

戻り値

`AccessType` メソッドは以下のいずれかの値を返します。

- ルールでリソースへのアクセスを許可する場合は、1
- ルールでリソースへのアクセスを拒否する場合は、0
- コールに失敗する場合は、-1

Action メソッド -- ルールのアクションを設定または取得

`Action` メソッドは、ルールのアクションを設定または取得します。

構文

`Action` メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRule->Action([action])
```

パラメータ

Action メソッドは以下のパラメータを受け入れます。

action (文字列)

(オプション) 以下のように、実行するアクションを指定します。

アクション タイプが **Web** エージェント アクションの場合は、以下の 1 つ以上の HTTP アクションを使用します。複数のアクションを区切るには、カンマを使用します。

- GET。HTTP を使用して表示するためのリソースを取得します。
- POST。HTTP を使用してユーザ提供の情報を POST 送信します。
- PUT。レガシー HTTP アクションをサポートします。

アクション タイプが認証イベントの場合

- OnAuthAccept。ユーザ認証が正常に行われたときに発生します。
- OnAuthAttempt。ユーザ名が入力されていないために、ユーザ認証が失敗したときに発生します。
- OnAuthChallenge。カスタム認証方式で使用し、レスポンスを発行することができます。
- OnAuthReject。ユーザが認証に失敗したときに発生します。
- OnAuthUserNotFound。アクティブ レスポンスをトリガするために使用します。

アクション タイプが許可イベントの場合

- OnAccessAccept。ユーザに対してリソースへのアクセスが SiteMinder によって問題なく許可したときに発生します。
- OnAccessReject。リソースへのアクセスが SiteMinder によって許可されなかったことが原因でユーザを拒否するときに発生します。

戻り値

Action メソッドは以下のいずれかの値を返します。

- 新規または既存のルール アクション
- コールに失敗した場合は、**undef**

ActiveExpr メソッド -- ルールのアクティブな式を設定または取得

ActiveExpr メソッドは、ルールのアクティブな式を設定または取得します。

構文

ActiveExpr メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRule->ActiveExpr([expr])
```

パラメータ

ActiveExpr メソッドは、以下のパラメータを受け入れます。

expr (文字列)

(オプション) 実行するアクティブな式を指定します。

戻り値

ActiveExpr メソッドは以下のいずれかの値を返します。

- 新規または既存のアクティブな式
- コールに失敗した場合は、**undef**

Agent メソッド -- グローバル ルールと関連付けられたエージェント オブジェクトまたはエージェント グループ オブジェクトを設定または取得

Agent メソッドは、グローバルルールと関連付けられたエージェント オブジェクトまたはエージェント グループ オブジェクトを設定または取得します。

構文

Agent メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRule->Agent(agentObject)
```

パラメータ

Agent メソッドは以下のパラメータを受け入れます。

agentObject (*objectType*)

ルールと関連付けるエージェント オブジェクトまたはエージェント グループ オブジェクトを指定します。*objectType* には **PolicyMgtAgent** または **PolicyMgtGroup** が有効です。

戻り値

Agent メソッドは新規または既存の **PolicyMgtAgent** オブジェクトか **PolicyMgtGroup** オブジェクトを返します。

注釈

ルールが作成された後、ルールと関連付けられたエージェントは、同じエージェント タイプ (**Web** エージェントなど) 内でのみ変更できます。

注: ドメイン範囲があるルールは、レلمを介してエージェントと間接的に関連付けられます。

Description メソッド -- ルールの説明を設定または取得

Description メソッドは、ルールの説明を設定または取得します。

構文

Description メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRule->Description([ruleDesc])
```

パラメータ

Description メソッドは以下のパラメータを受け入れます。

ruleDesc (文字列)

(オプション) ルールの説明を指定します。

戻り値

Description メソッドは以下のいずれかの値を返します。

- 新規または既存のルールの説明
- コールに失敗した場合は、空の文字列

IsEnabled メソッド -- ルールを有効化または無効化

IsEnabled メソッドは、ルールを有効化または無効化します。

構文

IsEnabled メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRule->IsEnabled([enableFlag])
```

パラメータ

IsEnabled メソッドは以下のパラメータを受け入れます。

enableFlag (タイプ)

(オプション) 以下のようにルールを有効にするかどうかを指定します。

- 1 はルールを有効にします。
- 0 はルールを無効にします。

戻り値

IsEnabled メソッドは以下のいずれかの値を返します。

- ルールが有効な場合は、1
- ルールが無効な場合は、0
- コールに失敗する場合は、-1

Name メソッド -- ルール名を設定または取得

Name メソッドは、ルール名を設定または取得します。

構文

Name メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRule->Name([ruleName])
```

パラメータ

Name メソッドは以下のパラメータを受け入れます。

ruleName (文字列)

 ルールの名前を指定します。

戻り値

Name メソッドは以下のいずれかの値を返します。

- 新規または既存のルール名
- コールに失敗した場合は、**undef**

RegexMatch メソッド -- 正規表現パターン一致が有効かどうかを判定

RegexMatch メソッドは、リソース一致操作の正規表現パターン一致が有効かどうかを判定するフラグを設定または取得します。

構文

RegexMatch メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRule->RegexMatch([enableFlag])
```

パラメータ

RegexMatch メソッドは、以下のパラメータを受け入れます。

enableFlag (int)

(オプション) 正規表現パターン一致を許可するかどうかを指定します。

- 1 はパターン一致を許可します
- 0 はパターン一致を許可しません

戻り値

RegexMatch メソッドは以下のいずれかの値を返します。

- 正規表現パターン一致が有効な場合は、1
- 正規表現パターン一致が無効な場合は、0
- コールに失敗する場合は、-1

Resource メソッド -- ルールによって保護されたリソースを設定または取得

Resource メソッドは、ルールによって保護されたリソースを設定または取得します。

構文

Resource メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtRule->Resource()
```

パラメータ

Resource メソッドは、パラメータを受け入れません。

戻り値

Resource メソッドは以下のいずれかの値を返します。

- コールに成功する場合は、保護されたリソース
- コールに失敗した場合は、**undef**

CLI SAML 2.0 アフィリエーション メソッド

GetAffiliatedSAMLAuthSchemes メソッド -- この SAML アフィリエーションと関連付けられた SAML 2.0 認証方式を取得

GetAffiliatedSAMLAuthSchemes メソッドは、この SAML アフィリエーションと関連付けられたすべての SAML 2.0 認証方式を取得します。

構文

GetAffiliatedSAMLAuthSchemes メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLAffiliation->GetAffiliatedSAMLAuthSchemes()
```

パラメータ

GetAffiliatedSAMLAuthSchemes メソッドは、パラメータを受け入れません。

戻り値

GetAffiliatedSAMLAuthSchemes メソッドは以下のいずれかの値を返します。

- SAML 2.0 テンプレートに基づく PolicyMgtAuthScheme オブジェクトの配列
- コールに失敗した場合は、**undef**

GetAffiliatedSAMLServiceProviders メソッド -- この SAML アフィリエーションと関連付けられた SAML 2.0 サービス プロバイダを取得

GetAffiliatedSAMLServiceProviders メソッドは、この SAML アフィリエーションと関連付けられたすべての SAML 2.0 サービス プロバイダを取得します。

構文

GetAffiliatedSAMLServiceProviders メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLAffiliation->GetAffiliatedSAMLServiceProviders()
```

パラメータ

GetAffiliatedSAMLServiceProviders メソッドは、パラメータを受け入れません。

戻り値

GetAffiliatedSAMLServiceProviders メソッドは以下のいずれかの値を返します。

- PolicyMgtSAMLServiceProvider オブジェクトの配列
- コールに失敗した場合は、**undef**

Property メソッド -- 指定された SAML 2.0 メタデータ プロパティを設定または取得

Property メソッド -- この SAML 2.0 アフィリエーションの指定された SAML 2.0 メタデータ プロパティを設定または取得します。

構文

Property メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLAffiliation->Property(name [, value])
```

パラメータ

Property メソッドは、以下のパラメータを受け入れます。

name (文字列)

設定または取得するプロパティを指定します。

value (文字列)

(オプション) 設定するプロパティの値を指定します。

戻り値

Property メソッドは以下のいずれかの値を返します。

- 新規または既存のプロパティ値
- コールに失敗した場合は、**undef**

注釈

アフィリエーション メタデータ プロパティのリストについては、[PolicyMgtSession->CreateSAMLAffiliation](#) (P. 475) メソッドの説明を参照してください。

注: このメソッドで 1 つ以上の既存のアフィリエーション プロパティを変更した後で、[PolicyMgtSAMLAffiliation->Save](#) (P. 421) をコールして変更をポリシー ストアに書き込みます。

Save メソッド -- この SAML 2.0 アフィリエーションの SAML 2.0 メタデータ プロパティの変更を保存

Save メソッドは、この SAML 2.0 アフィリエーションの SAML 2.0 メタデータ プロパティに対して行った変更を保存します。

構文

Save メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLAffiliation->Save()
```

パラメータ

Save メソッドは、パラメータを受け入れません。

戻り値

Save メソッドは以下のいずれかの値を返します。

- 成功の場合は 0
- コールに失敗する場合は、-1
- 変更を保存するためのユーザの権限が十分でない場合は、-4
- パスとクラスが空の場合は、-10

注釈

アフィリエーション プロパティを変更するには、[PolicyMgtSAMLAffiliation->Property](#) メソッドをコールします。

CLI SAML 2.0 インデックス化エンドポイント メソッド

GetACSIndex メソッド -- アサーション コンシューマ サービス オブジェクトのインデックス値を取得

GetACSIndex メソッドは、SAML サービス プロバイダ アサーション コンシューマ サービス オブジェクトのインデックス値を取得します。

構文

GetACSIndex メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLSPACS->GetACSIndex()
```

パラメータ

GetACSIndex メソッドは、パラメータを受け入れません。

戻り値

GetACSIndex メソッドは以下のいずれかの値を返します。

- Assertion_Consumer_Service_object_index_value
- コールに失敗した場合は、**undef**

GetACSBinding メソッド -- アサーション コンシューマ サービス オブジェクトのプロトコル バインディングを取得

GetACSBinding メソッドは、SAML サービス プロバイダ アサーション コンシューマ サービス オブジェクトのプロトコル バインディングを取得します。

構文

GetACSBinding メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLSPACS->GetACSBinding()
```

パラメータ

GetACSBinding メソッドは、パラメータを受け入れません。

戻り値

GetACSBinding メソッドは以下のいずれかの値を返します。

- Assertion_Consumer_Service_object_protocol_binding
- コールに失敗した場合は、**undef**

GetACSURL メソッド -- アサーション コンシューマ サービス オブジェクトの URL 値を取得

GetACSIndex メソッドは、SAML サービス プロバイダ アサーション コンシューマ サービス オブジェクトの URL 値を取得します。

構文

GetACSURL メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLSPACS->GetACSURL()
```

パラメータ

GetACSURL メソッドは、パラメータを受け入れません。

戻り値

GetACSURL メソッドは以下のいずれかの値を返します。

- Assertion_Consumer_Service_object_URL_value
- コールに失敗した場合は、**undef**

GetIsDefault メソッド -- アサーション コンシューマ サービス オブジェクトの IsDefault 値を取得

GetIsDefault メソッドは、SAML サービス プロバイダ アサーション コンシューマ サービス オブジェクトの IsDefault の値を取得します。

構文

GetIsDefault メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLSPACS->GetIsDefault()
```


パラメータ

GetIsDefault メソッドは、パラメータを受け入れません。

戻り値

GetIsDefault メソッドは以下のいずれかの値を返します。

- Assertion_Consumer_Service_object_IsDefault_value
- コールに失敗した場合は、**undef**

CLI SAML 2.0 リクエスト属性メソッド

GetAttrNameFormat メソッド -- SAML リクエスト属性の名前形式を取得

GetAttrNameFormat メソッドは、SAML リクエスト属性の名前形式を取得します。

構文

GetAttrNameFormat メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLRequesterAttr->GetAttrNameFormat()
```

パラメータ

GetAttrNameFormat メソッドは、パラメータを受け入れません。

戻り値

GetAttrNameFormat メソッドは、以下の値を返します。

- SAML_Requester_attribute_name_format

GetLocalName メソッド -- SAML リクエスト属性のローカル名を取得

GetLocalName メソッドは、SAML リクエスト属性のローカル名を取得します。

構文

GetLocalName メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLRequesterAttr->GetLocalName()
```

パラメータ

GetLocalName メソッドは、パラメータを受け入れません。

戻り値

GetLocalName メソッドは以下のいずれかの値を返します。

- SAML_Requester_attribute_local_name
- コールに失敗した場合は、**undef**

GetName メソッド -- SAML リクエスト属性の名前を取得

GetName メソッドは、SAML リクエスト属性の名前を取得します。

構文

GetName メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLRequesterAttr->GetName()
```

パラメータ

GetName メソッドは、パラメータを受け入れません。

戻り値

GetName メソッドは以下のいずれかの値を返します。

- SAML_Requester_attribute_name
- コールに失敗した場合は、**undef**

CLI SAML 2.0 サービス プロバイダ メソッド

以下のメソッドが `PolicyMgtSAMLServiceProvider` オブジェクトに有効です。

- `AddAssertionConsumerService` メソッド -- アサーション コンシューマ サービスを **SAML** サービス プロバイダ オブジェクトに追加
- `AddAttribute` メソッド -- 属性を **SAML 2.0** サービス プロバイダに追加
- `AddUser` メソッド -- **SAML** サービス プロバイダにユーザを追加
- `CreateIPConfigHostName` メソッド -- サービス プロバイダの IP 設定オブジェクトを作成
- `CreateIPConfigRange` -- サービス プロバイダの IP 設定オブジェクトを作成
- `CreateIPConfigSingleHost` -- サービス プロバイダの IP 設定オブジェクトを作成
- `CreateIPConfigSubnetMask` -- サービス プロバイダの IP 設定オブジェクトを作成
- `DeleteIPConfig` -- 指定された IP 設定 オブジェクトを削除
- `GetAllAttributes` メソッド -- **SAML 2.0** サービス プロバイダのすべての属性を取得
- `GetAllIPConfigs` メソッド -- すべての IP 設定オブジェクトを取得
- `GetAllAssertionConsumerServices` メソッド -- すべてのアサーション コンシューマ サービスを取得
- `GetAllUsers` メソッド -- すべてのユーザを取得
- `Property` メソッド -- メタデータ プロパティを設定または取得
- `RemoveAssertionConsumer` メソッド -- アサーション コンシューマ サービスを削除
- `RemoveAttribute Method` -- 指定された属性を削除
- `RemoveUser Method` -- 指定されたユーザを削除
- `Save` メソッド -- メタデータ プロパティに対して行った変更を保存

AddAssertionConsumerService メソッド -- アサーション コンシューマ サービスを SAML サービス プロバイダ オブジェクトに追加

AddAssertionConsumerService メソッドは、アサーション コンシューマ サービスを SAML サービス プロバイダ オブジェクトに追加します。

構文

AddAssertionConsumerService メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLServiceProvider->AddAssertionConsumerService(index,  
protocolBinding, URL)
```

パラメータ

AddAssertionConsumerService メソッドは、以下のパラメータを受け入れます。

index (int)

アサーション コンシューマ サービス インデックス化エンドポイント
インデックス値を指定します。

protocolBinding (文字列)

以下のいずれかのアサーション コンシューマ サービスのプロトコル
バインディングを指定します。

- SAMLSP_HTTP_Post
- SAMLSP_ACS_PROTOCOLBINDING_HTTP_Artifact

URL (文字列)

インデックス化エンドポイントの URL を指定します。

戻り値

AddAssertionConsumerService メソッドは以下のいずれかの値を返します。

- PolicyMgtSAMLSPACS オブジェクト
- コールに失敗した場合は、**undef**

AddAttribute メソッド -- 属性を SAML 2.0 サービス プロバイダに追加

AddAttribute メソッドは、属性を SAML 2.0 サービス プロバイダに追加します。

構文

AddAttribute メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLServiceProvider->AddAttribute(attrNameFormat, value,  
nEncrypted, nMode)
```

パラメータ

AddAttribute メソッドは、以下のパラメータを受け入れます。

attrNameFormat (int)

SAML 2.0 標準で定義されるように、以下のいずれかの属性形式を指定します。

- SAMLSP_UNSPECIFIED (値=0)
- SAMLSP_URI (値=1)。
- SAMLSP_BASIC (値=2)。

value (文字列)

属性の値指定情報を指定します。この値指定情報が、[SiteMinder SAML サービス プロバイダ プロパティ] ダイアログの [名前と値のペア] 列に表示されます。値指定情報の形式は、以下のように追加するアフィリエイト属性のタイプによって異なります -- スタティック、ユーザ属性または DN 属性。

- スタティック属性：
variableName=value
- ユーザ属性：
variableName=<%userattr="AttrName"%>

- DN 属性 :

`variableName=<#dn="DNSpec" attr="AttrName"##>`

SiteMinder で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、DNSpec の前に単符 (!) を付けます。たとえば、以下のように指定します。

`dn="!ou=People,o=security.com"`

nEncrypted (int)

属性が暗号化されるかどうかを指定します。ゼロ以外の場合、属性はアサーションに含まれた後に暗号化されます。

nMode (int)

以下のいずれかのこの属性の取得モードを指定します。

- SAMLSP_SSO
- SAMLSP_Attribute

戻り値

AddAttribute メソッドは以下のいずれかの値を返します。

- PolicyMgtSAMLSPAttr オブジェクト
- コールに失敗した場合は、**undef**

注釈

SAML 2.0 属性には、サービス プロバイダ上のリソースにアクセスしようとするプリンシパルに関する情報が含まれます。たとえば、プリンシパルのユーザ DN などです。

定義された属性は、このサービス プロバイダに対して生成されるすべての SAML 2.0 アサーション用の属性ステートメントに含まれます。

AddUser メソッド -- ユーザを SAML 2.0 サービス プロバイダに追加

AddUser メソッドは、ユーザを SAML サービス プロバイダに追加します。アサーションはサービス プロバイダと関連付けられたユーザに対して生成できます。

構文

AddUser メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLServiceProvider->AddUser(user)
```

パラメータ

AddUser メソッドは以下のパラメータを受け入れます。

user (PolicyMgtUser)

追加するユーザを指定します。

戻り値

AddUser メソッドは以下のいずれかの値を返します。

- 成功の場合は 0
- コールに失敗する場合は、-1

CreateIPConfigHostName メソッド -- サービス プロバイダの IP 設定オブジェクトを作成

CreateIPConfigHostName メソッドは、指定されたホスト名に基づいて、サービス プロバイダの IP 設定オブジェクトを作成します。

構文

CreateIPConfigHostName メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLServiceProvider->CreateIPConfigHostName(hostName)
```

パラメータ

CreateIPConfigHostName メソッドは、以下のパラメータを受け入れます。

hostName (文字列)

アサーションを発行する必要があるホスト名を指定します。

戻り値

CreateIPConfigHostName メソッドは以下のいずれかの値を返します。

- PolicyMgtIPConfig オブジェクト
- コールに失敗した場合は、**undef**

注釈

このメソッドは、アサーション生成ポリシー用の IP アドレス制限を作成します。このアドレス制限では、指定されたホストから生成されたアサーションのみが受け入れられます。

CreateIPConfigRange メソッド -- サービス プロバイダの IP 設定オブジェクトを作成

CreateIPConfigRange メソッドは、指定された IP アドレスの範囲に基づいて、サービス プロバイダの IP 設定オブジェクトを作成します。

構文

CreateIPConfigRange メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLServiceProvider->CreateIPConfigRange(ipAddr1, ipAddr2)
```

パラメータ

CreateIPConfigRange メソッドは、以下のパラメータを受け入れます。

ipAddr1 (文字列)

有効なアドレス範囲内の最初の IP アドレスを指定します。

ipAddr2 (文字列)

有効なアドレス範囲内の最後の IP アドレスを指定します。

戻り値

CreateIPConfigRange メソッドは以下のいずれかの値を返します。

- PolicyMgtIPConfig オブジェクト
- コールに失敗した場合は、**undef**

注釈

このメソッドは、アサーション生成ポリシー用の IP アドレス制限を作成します。このアドレス制限では、指定された IP アドレスの範囲から生成されたアサーションのみが受け入れられます。

CreateIPConfigSingleHost メソッド -- サービス プロバイダの IP 設定オブジェクトを作成

CreateIPConfigSingleHost メソッドは、指定された IP アドレスに基づいて、サービス プロバイダの IP 設定オブジェクトを作成します。

構文

CreateIPConfigSingleHost メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLServiceProvider->CreateIPConfigSingleHost(ipAddr)
```

パラメータ

CreateIPConfigSingleHost メソッドは以下のパラメータを受け入れます。

ipAddr (文字列)

アサーションを発行する必要がある IP アドレスを指定します。

戻り値

CreateIPConfigSingleHost メソッドは以下のいずれかの値を返します。

- PolicyMgtIPConfig オブジェクト
- コールに失敗した場合は、**undef**

注釈

このメソッドは、アサーション生成ポリシー用の IP アドレス制限を作成します。このアドレス制限では、指定された IP アドレスから生成されたアサーションのみが受け入れられます。

CreateIPConfigSubnetMask メソッド -- サービス プロバイダの IP 設定オブジェクトを作成

CreateIPConfigSubnetMask メソッドは、指定された IP アドレスおよびサブネットマスクに基づいて、サービス プロバイダの IP 設定オブジェクトを作成します。

構文

CreateIPConfigSubnetMask メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLServiceProvider->CreateIPConfigSubnetMask(ipAddr,  
subnetMask)
```

パラメータ

CreateIPConfigSubnetMask メソッドは、以下のパラメータを受け入れます。

ipAddr (文字列)

サブネットアドレスを引き出すために使用する IP アドレスを指定します。

subnetMask (署名なし long)

サブネットアドレスを引き出すために使用するサブネットマスクを指定します。

戻り値

CreateIPConfigSubnetMask メソッドは以下のいずれかの値を返します。

- PolicyMgtIPConfig オブジェクト
- コールに失敗した場合は、**undef**

注釈

このメソッドは、アサーション生成ポリシー用の IP アドレス制限を作成します。このアドレス制限では、サブネット マスクから生成されたアサーションのみが受け入れられます。サブネット アドレスは、渡された IP アドレスおよびサブネット マスクから引き出されます。サブネット マスク値の定義の詳細については、[PolicyMgtPolicy->CreateIPConfigSubnetMask](#) (P. 369) メソッドの説明を参照してください。

DeleteIPConfig メソッド -- 指定された IP 設定 オブジェクトを削除

DeleteIPConfig メソッドは、指定された IP 設定オブジェクトを削除します。

構文

DeleteIPConfig メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLServiceProvider->DeleteIPConfig(IPConfig)
```

パラメータ

DeleteIPConfig メソッドは以下のパラメータを受け入れます。

IPConfig (PolicyMgtIPConfig オブジェクト)

削除する IP 設定オブジェクトを指定します。

戻り値

DeleteIPConfig メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことを指定します。
- 値 = -1
メソッドが失敗したことを指定します。

GetAllAttributes メソッド -- SAML 2.0 サービス プロバイダのすべての属性を取得

GetAllAttributes メソッドは、SAML 2.0 サービス プロバイダ用に定義されたすべての属性を取得します。

構文

GetAllAttributes メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLServiceProvider->GetAllAttributes()
```

パラメータ

GetAllAttributes メソッドは、パラメータを受け入れません。

戻り値

GetAllAttributes メソッドは以下のいずれかの値を返します。

- PolicyMgtSAMLSPAttr (配列)
- コールに失敗した場合は、**undef**

GetAllIPConfigs メソッド -- すべての IP 設定オブジェクトを取得

GetAllIPConfigs メソッドは、SAML 2.0 サービス プロバイダのすべての IP 設定オブジェクトを取得します。

構文

GetAllIPConfigs メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLServiceProvider->GetAllIPConfigs()
```

パラメータ

GetAllIPConfigs メソッドは、パラメータを受け入れません。

戻り値

GetAllIPConfigs メソッドは以下のいずれかの値を返します。

- PolicyMgtIPConfig (配列)
- IP 設定オブジェクトが見つからない場合は、**undef**

GetAllAssertionConsumerServices メソッド -- すべてのアサーション コンシューマ サービスを取得

GetAllAssertionConsumerServices メソッドは、SAML 2.0 サービス プロバイダ オブジェクトからすべてのアサーション コンシューマ サービスを取得します。

構文

GetAllAssertionConsumerServices メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLServiceProvider->GetAllAssertionConsumerServices()
```

パラメータ

GetAllAssertionConsumerServices メソッドは、パラメータを受け入れません。

戻り値

GetAllAssertionConsumerServices メソッドは以下のいずれかの値を返します。

- PolicyMgtSAMLSPACS (配列)
- コールに失敗した場合は、**undef**

GetAllUsers メソッド -- すべてのユーザを取得

GetAllUsers メソッドは、SAML 2.0 サービス プロバイダと関連付けられたすべてのユーザを取得します。ユーザ ディレクトリが指定される場合は、指定されたディレクトリに属すユーザのみが返されます。

構文

GetAllUsers メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLServiceProvider->GetAllUsers([userDir])
```

パラメータ

GetAllUsers メソッドは以下のパラメータを受け入れます。

userDir (PolicyMgtUserDir オブジェクト)

(オプション) 取得されたすべてのユーザが属す必要があるユーザディレクトリを指定します。

戻り値

GetAllUsers メソッドは以下のいずれかの値を返します。

- PolicyMgtUser (配列)
- エラーが発生する場合またはユーザが見つからなかった場合は、**undef**

Property メソッド -- メタデータ プロパティを設定または取得

Property メソッド -- このサービス プロバイダ用に指定された SAML 2.0 メタデータ プロパティを設定または取得します。

注: このメソッドを使用して 1 つ以上のサービス プロバイダ プロパティを変更した後で、PolicyMgtSAMLServiceProvider->Save メソッドをコールして変更をポリシー ストアに書き込みます。

構文

Property メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLServiceProvider->Property(name[, value])
```

パラメータ

Property メソッドは、以下のパラメータを受け入れます。

name (文字列)

設定または取得するプロパティを指定します。

注: サービス プロバイダ メタデータ プロパティの全リストについては、メソッド [PolicyMgtAffDomain->CreateSAMLServiceProvider](#) (P. 174) を参照してください。

value (文字列)

(オプション) プロパティの新しい値を指定します。

戻り値

Property メソッドは以下のいずれかの値を返します。

- **property_value**
プロパティの新規または既存の値を指定します。
- **undef**
コールが失敗したことを指定します。

RemoveAssertionConsumer メソッド -- アサーション コンシューマ サービスを削除

RemoveAssertionConsumer メソッドは、SAML 2.0 サービス プロバイダから既存のアサーション コンシューマ サービスを削除します。

構文

RemoveAssertionConsumer メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLServiceProvider->RemoveAssertionConsumer(pSAMLSPACS)
```

パラメータ

RemoveAssertionConsumer メソッドは以下のパラメータを受け入れます。

pSAMLSPACS

削除するアサーション コンシューマ サービスを指定します。

戻り値

RemoveAssertionConsumer メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことを指定します。
- 値 = -1
メソッドが失敗したことを指定します。

RemoveAttribute Method -- 指定された属性を削除

RemoveAttribute メソッドは、SAML 2.0 サービス プロバイダから指定された属性を削除します。

構文

RemoveAttribute メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLServiceProvider->RemoveAttribute(SAMLSPAttr)
```

パラメータ

RemoveAttribute メソッドは以下のパラメータを受け入れます。

SAMLSPAttr (PolicyMgtSAMLSPAttr オブジェクト)

削除する属性を指定します。

戻り値

RemoveAttribute メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことを指定します。
- 値 = -1
メソッドが失敗したことを指定します。

RemoveUser Method -- 指定されたユーザを削除

RemoveUser メソッドは、SAML 2.0 サービス プロバイダから指定されたユーザを削除します。

構文

RemoveUser メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLServiceProvider->RemoveUser(user)
```


パラメータ

RemoveUser メソッドは以下のパラメータを受け入れます。

user (PolicyMgtUser オブジェクト)

削除するユーザを指定します。

戻り値

RemoveUser メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことを指定します。
- 値 = -1
メソッドが失敗したことを指定します。

Save メソッド -- メタデータ プロパティに対して行った変更を保存

Save メソッドは、サービス プロバイダの SAML 2.0 メタデータ プロパティに対して行った変更を保存します。SAML 2.0 サービス プロバイダに対してすべての変更を行った後で、このメソッドを 1 回コールします。変更を有効にするには、必ずこのメソッドをコールします。メタデータ プロパティを変更するには、PolicyMgtSAMLServiceProvider->Property メソッドをコールします。

構文

Save メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLServiceProvider->Save()
```

パラメータ

Save メソッドは、パラメータを受け入れません。

戻り値

Save メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことを指定します。
- 値 = -1
メソッドが失敗したことを指定します。
- 値 = -4
メタデータ プロパティを変更するのに必要な権限がユーザにないことを指定します。
- 値 = -10
パスとクラスが空であることを指定します。

CLI SAML 2.0 サービス プロバイダ属性メソッド

GetAttrNameFormat メソッド -- 属性名の形式を取得

GetAttrNameFormat メソッドは、SAML 2.0 サービス プロバイダで使用する属性名の形式を取得します。SAML 2.0 属性の詳細については、メソッド [PolicyMgtSAMLServiceProvider->AddAttribute](#) (P. 428) を参照してください。

構文

GetAttrNameFormat メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLSPAttr->GetAttrNameFormat()
```

パラメータ

GetAttrNameFormat メソッドは、パラメータを受け入れません。

戻り値

GetAttrNameFormat メソッドは以下のいずれかの値を返します。

- SAMLSP_UNSPECIFIED (値 = 0)
- SAMLSP_URI (値 = 1)
- SAMLSP_BASIC (値 = 2)

GetValue メソッド -- サービス プロバイダ属性値を取得

GetValue メソッドは、SAML 2.0 サービス プロバイダ属性の値を取得します。サービス プロバイダに関連付けられたすべての属性を取得するには、メソッド `PolicyMgtSAMLServiceProvider->GetAllAttributes` をコールします。SAML 2.0 属性の詳細については、メソッド [PolicyMgtSAMLServiceProvider->AddAttribute](#) (P. 428) を参照してください。

構文

GetValue メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSAMLSPAttr->GetValue()
```

パラメータ

GetValue メソッドは、パラメータを受け入れません。

戻り値

GetValue メソッドは以下のいずれかの値を返します。

- Service_Provider_attribute_value
- コールに失敗した場合は、**undef**

CLI セッション メソッド

AddAttributeToSAMLScheme メソッド -- 新しい属性を認証方式に追加

AddAttributeToSAMLScheme メソッドは、SAML 2.0 認証方式定義に新しい属性を追加します。

構文

AddAttributeToSAMLScheme メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->AddAttributeToSAMLScheme(scheme, AttrNameFormat,
LocalName, Name)
```

パラメータ

AddAttributeToSAMLScheme メソッドは、以下のパラメータを受け入れます。

scheme (PolicyMgtAuthScheme オブジェクト)

SAML 2.0 認証方式を指定します。

AttrNameFormat (int)

以下の属性タイプを指定します。

- SAMLSP_UNSPECIFIED
- SAMLSP_URI
- SAMLSP_BASIC

LocalName (文字列)

ローカルに使用されるとおりの属性の名前を指定します。

Name (文字列)

属性認証機関で定義されるとおりの属性の名前を指定します。

戻り値

AddAttributeToSAMLScheme メソッドは以下のいずれかの値を返します。

- PolicyMgtSAMLRequesterAttr (オブジェクト)
- コールに失敗した場合は、**undef**

AddTrustedHost メソッド -- トラストッド ホスト オブジェクトを作成または変更

AddTrustedHost メソッドは、ポリシー ストア内のトラストッド ホスト オブジェクトを作成または変更します。

構文

AddTrustedHost メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->AddTrustedHost(trustedHostName[,  
trustedHostDescription][, trustedHostIpAddress][, sharedSecret])
```

パラメータ

AddTrustedHost メソッドは、以下のパラメータを受け入れます。

trustedHostName (文字列)

トラストッド ホストの名前を指定します。

trustedHostDescription (文字列)

(オプション) トラストッド ホストの説明を指定します。

trustedHostIpAddress (文字列)

(オプション) トラストッド ホストの IP アドレスを指定します。

sharedSecret (文字列)

(オプション) 共有秘密キーを指定します。

注: また、**-sh** オプションを指定して **SiteMinder** ツール **smregghost** を実行することにより、ホスト設定ファイルで共有秘密キーを定義する必要があります。 **-sh** オプションを使用して共有秘密キーを指定しない場合は、**SiteMinder** によって自動的に生成されます。

戻り値

AddTrustedHost メソッドは以下のいずれかの値を返します。

- PolicyMgtTrustedHost (オブジェクト)
- トラストッド ホスト名がすでに存在する場合は、**undef**

注釈

AddTrustedHost メソッドを使用してトラステッドホストを登録する場合は、最初にポリシーサーバとエージェント間の接続を設定する必要はありません。このメソッドを使用してトラステッドホストを登録するときは、ホスト設定ファイルで共有秘密キーを定義するために **SiteMinder** ツール **smreghost** を実行する必要もあります。（ホスト設定ファイルはデフォルトでは **SmHost.conf** と命名されます）。**-sh** オプションおよび共有秘密キーを指定して **smreghost** を実行します。クリアテキスト内の共有秘密キーを取得するには、メソッド **PolicyMgtTrustedHost->GetSecret** をコールします。

または、**-sh** オプションなしでメソッド **CreateTrustedHost** をコールして **smreghost** を実行し、トラステッドホストを作成できます。この場合、**SiteMinder** は、インストール時にトラステッドホストを自動的に作成して設定します。

重要: **SiteMinder** はランダムな 128 バイト ASCII 共有秘密キーを生成します。共有秘密キーを作成するときは、任意の文字列値にできます。強力な共有秘密キーを作成するには、**sharedSecret** パラメータを空の文字列に設定した状態で **AddTrustedHost** メソッドをコールすることを強く推奨します。これにより、ランダムで長く推測困難な共有秘密キーが自動的に生成されます。

CreateAdmin メソッド -- システムレベル管理者を作成

CreateAdmin メソッドは、システムレベル管理者を作成および設定します。

構文

CreateAdmin メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->CreateAdmin(adminName[, adminDesc][, adminPwd][, userDir][, authScheme])
```

パラメータ

CreateAdmin メソッドは、以下のパラメータを受け入れます。

adminName (文字列)

管理者の名前を指定します。

adminDesc (文字列)

(オプション) 管理者の説明を指定します。

adminPwd (文字列)

(オプション) 管理者のパスワードを指定します。

userDir (PolicyMgtUserDir オブジェクト)

(オプション) 管理者が外部ディレクトリに保存される場合は、ユーザディレクトリを指定します。

authScheme (PolicyMgtAuthScheme オブジェクト)

(オプション) 管理者が外部ディレクトリに保存される場合は、使用する認証方式を指定します。

注: 外部ユーザディレクトリを指定する場合は、このパラメータが必要です。

戻り値

CreateAdmin メソッドは以下のいずれかの値を返します。

- PolicyMgtAdmin (オブジェクト)
- コールに失敗した場合または管理者名がすでに存在する場合は、**undef**

注釈

ポリシー管理 API では、特定のドメインの管理者を作成できません。ただし、メソッド **AddAdmin** をコールすることにより、特定のドメインに既存の管理者を追加できます。ドメイン権限がある管理者を作成するには、管理 UI を使用します。

CreateAffDomain メソッド -- アフィリエイトドメインを作成

CreateAffDomain メソッドは、アフィリエイトドメインを作成します。

構文

CreateAffDomain メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->CreateAffDomain(domName[, domDesc])
```

パラメータ

CreateAffDomain メソッドは、以下のパラメータを受け入れます。

domName (文字列)

アフィリエイトドメインの名前を指定します。

domDesc (文字列)

(オプション) アフィリエイトドメインの説明を指定します。

戻り値

CreateAffDomain メソッドは以下のいずれかの値を返します。

- PolicyMgtAffDomain (オブジェクト)
- コールに失敗した場合またはアフィリエイトドメイン名がすでに存在する場合は、**undef**

注釈

アフィリエイトドメインを実装するには、レガシーフェデレーションが必要です。

CreateAgent メソッド -- SiteMinder エージェントを作成

CreateAgent メソッドは、SiteMinder エージェントを作成および設定します。

構文

CreateAgent メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->CreateAgent(agentName, agentType[, agentDesc][,  
agentIP][, agentSecret][, realmHintAttrID])
```


パラメータ

CreateAgent メソッドは、以下のパラメータを受け入れます。

agentName (文字列)

エージェントの名前を指定します

agentType (PolicyMgtAgentType オブジェクト)

エージェントのタイプを指定します。

agentDesc (文字列)

(オプション) エージェントの説明を指定します。

agentIP (文字列)

(オプション) エージェントの IP アドレスを指定します。

注: このパラメータは、RADIUS エージェントにとって必須です。

agentSecret (文字列)

(オプション) 共有秘密キーを指定します。

注: v4.x エージェントを作成するには、共有秘密キーを指定します。

v5.x エージェントを作成するには、このパラメータを省略します。

realmHintAttrID (int)

(オプション) レルム ヒント属性 ID を指定します。

注: このパラメータは RADIUS エージェントにのみ適用されます。

戻り値

CreateAgent メソッドは以下のいずれかの値を返します。

- PolicyMgtAgent (オブジェクト)
- コールに失敗した場合または SiteMinder エージェント名がすでに存在する場合は、**undef**

CreateAgentConfig メソッド -- エージェント設定オブジェクトを作成

CreateAgentConfig メソッドは、エージェント設定オブジェクトを作成します。

構文

CreateAgentConfig メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->CreateAgentConfig(agentConfigName[,  
AgentConfigDesc])
```

パラメータ

CreateAgentConfig メソッドは、以下のパラメータを受け入れます。

agentConfigName (文字列)

エージェント設定の名前を指定します。

AgentConfigDesc (文字列)

(オプション) エージェント設定の説明を指定します。

戻り値

CreateAgentConfig メソッドは以下のいずれかの値を返します。

- PolicyMgtAgentConfig (オブジェクト)
- コールに失敗した場合またはエージェント設定名がすでに存在する場合は、**undef**

CreateAgentGroup メソッド -- エージェント グループを作成

CreateAgentGroup メソッドは、エージェント グループを作成します。

構文

CreateAgentGroup メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->CreateAgentGroup(agentGroupName, agentType[,  
groupDesc])
```

パラメータ

CreateAgentGroup メソッドは、以下のパラメータを受け入れます。

agentGroupName (文字列)

エージェント グループの名前を指定します。

agentType (PolicyMgtAgentType オブジェクト)

エージェント グループと関連付けられたエージェントのタイプを指定します。

注: このメソッドのエージェント タイプを取得するには、メソッド PolicyMgtSession->GetAgentType をコールします。

groupDesc (文字列)

(オプション) エージェント グループの説明を指定します。

戻り値

CreateAgentGroup メソッドは以下のいずれかの値を返します。

- PolicyMgtGroup (オブジェクト)
- エージェント グループ名がすでに存在する場合は、**undef**

CreateAuthAzMap メソッド -- ディレクトリ マッピング オブジェクトを作成

CreateAuthAzMap メソッドは、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。

構文

CreateAuthAzMap メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->CreateAuthAzMap(authDir, azDir, mapType)
```

パラメータ

CreateAuthAzMap メソッドは、以下のパラメータを受け入れます。

authDir (PolicyMgtUserDir オブジェクト)

ユーザを認証するときに使用するユーザ ディレクトリを指定します。

azDir (PolicyMgtUserDir オブジェクト)

ユーザを許可するときに使用するユーザ ディレクトリを指定します。

mapType (int)

ディレクトリ マッピングのタイプを指定します。

- AUTHAZMAPTYPE_DN (値=1)

DN に基づいてマッピングを指定します。

- AUTHAZMAPTYPE_UNIVERSALID (値=2)

ユニバーサル識別子に基づいてマッピングを指定します。

- AUTHAZMAPTYPE_ATTR (値=3)

ユーザ ディレクトリ内の属性に基づいてマッピングを指定します。

戻り値

CreateAuthAzMap メソッドは以下のいずれかの値を返します。

- PolicyMgtAuthAzMap (オブジェクト)
- コールに失敗した場合は、**undef**

注釈

SiteMinder は、ユーザを認証し認可するために同じユーザ ディレクトリを使用します。また、SiteMinder では、認証用の 1 つのユーザ ディレクトリおよび許可用の別のユーザ ディレクトリを指定できます。この機能をディレクトリ マッピングといいます。認証情報がセントラルディレクトリに保存され、許可情報が複数のディレクトリに保存されて、それぞれが特定のアプリケーションに関連付けられる場合に、ディレクトリ マッピングは特に役立ちます。

CreateAuthScheme メソッド -- 認証方式を作成

CreateAuthScheme メソッドは、認証方式を作成および設定します。

構文

CreateAuthScheme メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->CreateAuthScheme(schemeName, schemeTemplate[,  
schemeDesc][, protLevel][, schemeLib][, schemeParam][, secret][, isTemplate][,  
isUsedByAdmin][, saveCreds][, isRadius][, ignorePwd])
```

パラメータ

CreateAuthScheme メソッドは、以下のパラメータを受け入れます。

schemeName (文字列)

認証方式の名前を指定します。

schemeTemplate (PolicyMgtAuthScheme オブジェクト)

(オプション) 認証方式のベースにするテンプレートを指定します。

注: テンプレートのリストを表示するには、メソッド
PolicyMgtSession->GetAuthScheme を参照してください。

schemeDesc (文字列)

(オプション) 認証方式タイプの説明を指定します。

protLevel (int)

(オプション) 認証方式の保護レベルを指定します。

範囲: 1 ~ 1000

注: 保護レベルの値が高いほど、認証方式のセキュリティ レベルが高くなります。

schemeLib (文字列)

(オプション) 各タイプの認証方式と共にリリースされるデフォルト
ライブラリの代わりに使用する、カスタム ライブラリの名前を指定し
ます。

schemeParam (文字列)

(オプション) 認証方式に渡すパラメータ文字列を指定します。

注: パラメータ文字列を構築しやすくするため、管理 UI で「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」タブに移動します。認証方式タイプを選択し、フィールドに値を入力して、「詳細」タブ上で結果を確認します。

secret (文字列)

(オプション) 認証方式の共有秘密キーを指定します。

isTemplate (int)

(オプション) 認証方式が他の認証方式用のテンプレートかどうかを指定します。

デフォルト: ゼロ (0) 値は、認証方式がテンプレートでないことを指定します。

注: このパラメータは SiteMinderv6.0 SP3 の時点で非推奨です。

isUsedByAdmin (int)

(オプション) 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します。

saveCreds (int)

(オプション) ユーザ認証情報を保存するかどうかを指定します。

isRadius (int)

(オプション) 認証方式タイプが RADIUS かどうかを指定します。

ignorePwd (int)

(オプション) パスワード ポリシーを無視するかどうかを指定します。

戻り値

CreateAuthScheme メソッドは以下のいずれかの値を返します。

- PolicyMgtAuthScheme (オブジェクト)
- コールに失敗した場合または認証方式名がすでに存在する場合は、**undef**

CreateCustomCertMap メソッド -- カスタム証明書マップを作成

CreateCustomCertMap メソッドは、カスタム証明書マップを作成します。カスタム証明書マップは、証明書の **Subject DN** で定義されたユーザ属性名と、ユーザディレクトリ内の対応するユーザ属性名を関連付けます。認証に成功するには、マップされたユーザ属性ペアの値が一致する必要があります。マップされる属性名を定義するために **AttributeMap** パラメータを使用します。

構文

CreateCustomCertMap メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->CreateCustomCertMap(IssuerDN, AttributeMap[,  
DirectoryType])
```

パラメータ

CreateCustomCertMap メソッドは、以下のパラメータを受け入れます。

IssuerDN (文字列)

証明書発行者の識別名を指定します。

AttributeMap (文字列)

ユーザディレクトリ内の属性名に証明書の **Subject DN** 内の属性名をマップする式を指定します。

構文 :

```
UserAttrName1=%{CertAttrName1},UserAttrName2=%{CertAttrName2},...  
UserAttrName#=%{CertAttrName#}
```

例 :

証明書の **Subject DN** の内容 : CN=John Smith, UID=JSMITH,
OU=Development, O=CompanyA

AttributeMap の内容 : CN=%{UID}, OU=%{OU}, O=%{O}

ユーザディレクトリで一致するユーザ DN : CN=JSMITH,
OU=Development, O=CompanyA

DirectoryType (int)

(オプション) 以下のように、認証ディレクトリとして指定されるユーザディレクトリのタイプを指定します。

- Sm_PolicyApi_DirType_LDAP
- 注: これがデフォルトです。
- Sm_PolicyApi_DirType_WinNT
- Sm_PolicyApi_DirType_ODBC

戻り値

CreateCustomCertMap メソッドは以下のいずれかの値を返します。

- PolicyMgtCertMap (オブジェクト)
- コールに失敗した場合は、**undef**

注釈

証明書マップが作成されると、以下のフラグがデフォルト値である **false** に設定されます。

- certificate_required_flag
- use_distributionpoints_flag
- verify_signature_flag
- check_certificate_revocation_list_flag
- cache_certificate_revocation_list_entries_flag

これらのフラグの値の変更の詳細については、メソッド [PolicyMgtSession->CreateExactCertMap](#) (P. 457) を参照してください。

CreateDomain メソッド -- ポリシードメイン オブジェクトを作成

CreateDomain メソッドは、ポリシードメイン オブジェクトを作成します。

構文

CreateDomain メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->CreateDomain(domName[, domDesc][, globalPoliciesApply])
```


パラメータ

CreateDomain メソッドは、以下のパラメータを受け入れます。

domName (文字列)

ドメインの名前を指定します。

domDesc (文字列)

(オプション) ドメインの説明を指定します。

globalPoliciesApply (int)

(オプション) ドメインでグローバル ポリシーを受け入れられるかどうかを指定します。

- 値 = 1 (デフォルト)

ドメインでグローバル ポリシーを受け入れられることを指定します。

- 値 = 0

ドメインでグローバル ポリシーを受け入れられないことを指定します。

戻り値

CreateDomain メソッドは以下のいずれかの値を返します。

- PolicyMgtDomain (オブジェクト)
- コールに失敗した場合またはポリシー ドメイン名がすでに存在する場合は、**undef**

CreateExactCertMap メソッド -- ユーザ ディレクトリ属性に一致する証明書マップを作成

CreateExactCertMap メソッドは、その Subject DN 属性が対応するユーザ ディレクトリ属性に正確に一致する証明書マップオブジェクトを作成します。証明書マップオブジェクトが作成されるときに、以下のフラグが FALSE に設定されます。

- 証明書は必要

注: このフラグの値を変更するには、メソッド **PolicyMgtCertMap->CertRequired** をコールします。

- 配布ポイントを使用

注: このフラグの値を変更するには、メソッド **PolicyMgtCertMap->UseDistributionPoints** をコールします。

- 署名を確認

注: このフラグの値を変更するには、メソッド **PolicyMgtCertMap->VerifySignature** をコールします。

- (CRL) をチェック

注: このフラグの値を変更するには、メソッド **PolicyMgtCertMap->EnableCRL** をコールします。

- CRL エントリをキャッシュに入れる

注: このフラグの値を変更するには、メソッド **PolicyMgtCertMap->CacheCRL** をコールします。

構文

CreateExactCertMap メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->CreateExactCertMap(IssuerDN[, DirectoryType])
```

パラメータ

CreateExactCertMap メソッドは、以下のパラメータを受け入れます。

IssuerDN (文字列)

証明書発行者の識別名を指定します。

DirectoryType (int)

(オプション) 認証に使用する以下のいずれかのタイプのユーザ ディレクトリを指定します。

- Sm_PolicyApi_DirType_LDAP (デフォルト)
- Sm_PolicyApi_DirType_WinNT
- Sm_PolicyApi_DirType_ODBC

戻り値

CreateExactCertMap メソッドは以下のいずれかの値を返します。

- PolicyMgtCertMap (オブジェクト)
- コールに失敗した場合は、**undef**

CreateGlobalPolicy メソッド -- グローバル ポリシーを作成

CreateGlobalPolicy メソッドは、グローバル範囲を持つポリシーを作成します。

構文

CreateGlobalPolicy メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->CreateGlobalPolicy(policyName[, enableFlag][, activeExpr][, policyDesc])
```

パラメータ

CreateGlobalPolicy メソッドは、以下のパラメータを受け入れます。

policyName (文字列)

グローバル ポリシーの名前を指定します。

enableFlag (タイプ)

(オプション) グローバル ポリシーを有効にするかどうかを指定します。

- 値 = 1 (デフォルト)

グローバル ポリシーを有効にすることを指定します。

- 値 = 0

グローバル ポリシーを無効にすることを指定します。

activeExpr (文字列)

(オプション) ... を指定します。

policyDesc (文字列)

(オプション) グローバル ポリシーの説明を指定します。

戻り値

CreateGlobalPolicy メソッドは以下のいずれかの値を返します。

- PolicyMgtPolicy (オブジェクト)
- コールに失敗した場合は、**undef**

CreateGlobalResponse メソッド -- グローバル レスポンスを作成

CreateGlobalResponse メソッドは、グローバル範囲を持つレスポンスを作成します。

構文

CreateGlobalResponse メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->CreateGlobalResponse(respName, agentType[, respDesc])
```

パラメータ

CreateGlobalResponse メソッドは、以下のパラメータを受け入れます。

respName (文字列)

グローバル レスポンスの名前を指定します。

agentType (PolicyMgtAgentType オブジェクト)

グローバル レスポンスと関連付けられたエージェントのタイプを指定します。

注: エージェント タイプ オブジェクトを取得するには、メソッド PolicyMgtSession->GetAgentType をコールします。

respDesc (文字列)

(オプション) グローバル レスポンスの説明を指定します。

戻り値

CreateGlobalResponse メソッドは以下のいずれかの値を返します。

- PolicyMgtResponse (オブジェクト)
- コールに失敗した場合は、**undef**

CreateGlobalResponseGroup メソッド -- ドメイン固有のルール グループを作成

CreateGlobalResponseGroup メソッドは、特定のドメインに固有なルールグループを作成します。

構文

CreateGlobalResponseGroup メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->CreateGlobalResponseGroup(groupName, agentType, domain)
```

パラメータ

CreateGlobalResponseGroup メソッドは、以下のパラメータを受け入れます。

groupName (文字列)

グローバル ルール グループ名を指定します。

agentType (PolicyMgtAgentType)

エージェントのタイプを指定します。

domain (PolicyMgtDomain)

ルール グループを適用するためのドメインを指定します。

戻り値

CreateGlobalResponseGroup メソッドは以下のいずれかの値を返します。

- PolicyMgtGroup (オブジェクト)
- コールに失敗した場合は、**undef**

CreateGlobalRule メソッド -- グローバル ルールを作成

CreateGlobalRule メソッドは、グローバル範囲を持つルールを作成します。

構文

CreateGlobalRule メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->CreateGlobalRule(ruleName, resource, event, agent[,  
ruleDesc][, allowAccess][, regexMatch][, activeExpr][, isEnabled])
```

パラメータ

CreateGlobalRule メソッドは、以下のパラメータを受け入れます。

ruleName (文字列)

グローバル ルールの名前を指定します。

resource (文字列)

グローバル ルールが保護しているリソース用のフィルタを指定します。

event (文字列)

(オプション) ルールが実行されるイベントのタイプを指定します。

agent (PolicyMgtAgent | PolicyMgtGroup)

グローバル ルールと関連付けるエージェントまたはエージェント グループを指定します。

ruleDesc (文字列)

(オプション) グローバル ルールの説明を指定します。

allowAccess (int)

(オプション) 以下のように、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かどうかを指定します。

- 値 = 1 (デフォルト)
アクセスを許可することを指定します。
- 値 = 0
アクセスを拒否することを指定します。

regexMatch (int)

(オプション) 正規表現パターン一致を実行するかどうかを指定します。

- 値 = 1
正規表現パターン一致を実行することを指定します。
- 値 = 0 (デフォルト)
正規表現パターン一致を実行しないことを指定します。

activeExpr (文字列)

(オプション) グローバル ルールのアクティブな式を指定します。

isEnabled (int)

(オプション) 以下のようにグローバル ルールを有効または無効にするかどうかを指定します。

- 値 = 1 (デフォルト)
グローバル ルールを有効にすることを指定します。
- 値 = 0
グローバル ルールを無効にすることを指定します。

戻り値

CreateGlobalRule メソッドは以下のいずれかの値を返します。

- PolicyMgtRule (オブジェクト)
- コールに失敗した場合は、**undef**

CreateGlobalRuleGroup メソッド -- ドメイン固有のルール グループを作成

CreateGlobalRuleGroup メソッドは、特定のドメインに固有なルール グループを作成します。

構文

CreateGlobalRuleGroup メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->CreateGlobalRuleGroup(groupName, agentType, domain)
```

パラメータ

CreateGlobalRuleGroup メソッドは、以下のパラメータを受け入れます。

groupName (文字列)

グローバル ルール グループ名を指定します。

agentType (PolicyMgtAgentType)

エージェントのタイプを指定します。

domain (PolicyMgtDomian)

ルール グループを適用するためのドメインを指定します。

戻り値

CreateGlobalRuleGroup メソッドは以下のいずれかの値を返します。

- PolicyMgtGroup (オブジェクト)
- コールに失敗した場合は、**undef**

CreateHostConfig メソッド -- ホスト設定オブジェクトを作成

CreateHostConfig メソッドは、ホスト設定オブジェクトを作成します。

構文

CreateHostConfig メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->CreateHostConfig(hostConfigName[, hostConfDesc][, enableFailover][, maxSocketsPerPort][, minSocketsPerPort][, newSocketstep][, requestTimeout])
```


パラメータ

CreateHostConfig メソッドは、以下のパラメータを受け入れます。

hostConfigName (文字列)

ホスト設定オブジェクトの名前を指定します。

hostConfDesc (文字列)

(オプション) ホスト設定オブジェクトの説明を指定します。

EnableFailover (int)

(オプション) 以下のように、ポリシー サーバとエージェント間に、フェールオーバーまたはラウンド ロビン通信を使用するかどうかを指定します。

- 値 = 1

フェールオーバー通信を指定します。

- 値 = 0

ラウンド ロビン通信を指定します。

maxSocketsPerPort (int)

(オプション) エージェントとポリシー サーバ間で開くことができる TCP/IP ソケットの最大数を指定します。

minSocketsPerPort (int)

(オプション) エージェントとポリシー サーバ間で開くことができる TCP/IP ソケットの最小数を指定します。

newSocketStep (int)

(オプション) 追加のソケットが必要なときに開くソケットの数を指定します。

requestTimeout (int)

(オプション) 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できる時間の長さを秒単位で指定します。

戻り値

CreateHostConfig メソッドは以下のいずれかの値を返します。

- PolicyMgtHostConfig (オブジェクト)
- コールに失敗した場合またはホスト設定名がすでに存在する場合は、**undef**

CreateODBCQueryScheme メソッド -- ODBC クエリ方式を作成

CreateODBCQueryScheme メソッドは、ODBC クエリ方式を作成および設定します。ODBC クエリ方式は SQL クエリ方式とも呼ばれます。

注: ODBC クエリ方式ごとに一意のデータ ソースを作成します。

構文

CreateODBCQueryScheme メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->CreateODBCQueryScheme(schemeName[, schemeDesc][,
queryEnumerate][, queryGetObjInfo][, queryLookup][, queryInitUser][,
queryAuthenticateUser][, queryGetUserProp][, querySetUserProp][,
queryGetUserProps][, queryLookupUser][, queryGetGroups][, queryIsGroupMember][,
queryGetGroupProp][, querySetGroupProp][, queryGetGroupProps][, queryLookupGroup][,
querySetPassword])
```

パラメータ

CreateODBCQueryScheme メソッドは、以下のパラメータを受け入れます。

schemeName (文字列)

ODBC クエリ方式の名前を指定します。

schemeDesc (文字列)

(オプション) ODBC クエリ方式の説明を指定します。

queryEnumerate (文字列)

(オプション) ディレクトリ内のユーザ オブジェクトの名前をリストするクエリを指定します。

注: 詳細については、メソッド

[PolicyMgtODBCQueryScheme->QueryEnumerate](#) (P. 310) を参照してください。

queryGetObjInfo (文字列)

(オプション) オブジェクトのクラスをフェッチするクエリを指定します。

注: 詳細については、メソッド

[PolicyMgtODBCQueryScheme->QueryGetObjInfo](#) (P. 314) を参照してください。

queryLookup (文字列)

(オプション) グループ テーブル内の属性の値に基づいてオブジェクトを返すクエリを指定します。

注: 詳細については、メソッド

[PolicyMgtODBCQueryScheme->QueryLookup](#) (P. 318) を参照してください。

queryInitUser (文字列)

(オプション) データベースに特定の名前のユーザが存在するかどうかを判定するクエリを指定します。

注: 詳細については、メソッド

[PolicyMgtODBCQueryScheme->QueryInitUser](#) (P. 316) を参照してください。

queryAuthenticateUser (文字列)

(オプション) ユーザのパスワードを取得するクエリを指定します。

注: 詳細については、メソッド

[PolicyMgtODBCQueryScheme->QueryAuthenticateUser](#) (P. 309) を参照してください。

queryGetUserProp (文字列)

(オプション) ユーザ プロパティの値を取得するクエリを定義します。

注: プロパティは *queryGetUserProps* パラメータ文字列でリストされる必要があります。詳細については、メソッド

[PolicyMgtODBCQueryScheme->QueryGetUserProp](#) (P. 315) を参照してください。

querySetUserProp (文字列)

(オプション) ユーザ プロパティの値を設定するクエリを定義します。

注: プロパティは *queryGetUserProps* パラメータ文字列でリストされる必要があります。詳細については、メソッド

[PolicyMgtODBCQueryScheme->QuerySetUserProp](#) (P. 323) を参照してください。

queryGetUserProps (文字列)

(オプション) 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指定します。

注: 詳細については、メソッド

[PolicyMgtODBCQueryScheme->QueryGetUserProps](#) (P. 316) を参照してください。

queryLookupUser (文字列)

(オプション) ユーザテーブルの属性を通じて、ユーザ名を取得するクエリを指定します。

注: 詳細については、メソッド

[PolicyMgtODBCQueryScheme->QueryLookupUser](#) (P. 320) を参照してください。

queryGetGroups (文字列)

(オプション) ユーザが属すグループの名前を取得するクエリを指定します。

注: 詳細については、メソッド

[PolicyMgtODBCQueryScheme->QueryGetGroups](#) (P. 313) を参照してください。

queryIsGroupMember (文字列)

(オプション) 特定のユーザがグループのメンバかどうかを判定するクエリを指定します。

注: 詳細については、メソッド

[PolicyMgtODBCQueryScheme->QueryIsGroupMember](#) (P. 317) を参照してください。

queryGetGroupProp (文字列)

(オプション) グループプロパティの値を返すクエリを定義します。

注: プロパティは *queryGetGroupProps* パラメータ文字列でリストされる必要があります。詳細については、メソッド

[PolicyMgtODBCQueryScheme->QueryGetGroupProp](#) (P. 311) を参照してください。

querySetGroupProp (文字列)

(オプション) グループ プロパティの値を設定するクエリを定義します。

注: プロパティは ***queryGetGroupProps*** パラメータ文字列でリストされる必要があります。詳細については、メソッド

[PolicyMgtODBCQueryScheme->QuerySetGroupProp](#) (P. 321) を参照してください。

queryGetGroupProps (文字列)

(オプション) グループ属性のカンマ区切りリストを指定します。

注: 詳細については、メソッド

[PolicyMgtODBCQueryScheme->QueryGetGroupProps](#) (P. 312) を参照してください。

queryLookupGroup (文字列)

(オプション) グループ テーブルの属性を通じて、グループ名を取得するクエリを指定します。

注: 詳細については、メソッド

[PolicyMgtODBCQueryScheme->QueryLookupGroup](#) (P. 319) を参照してください。

querySetPassword (文字列)

(オプション) ユーザのパスワードを変更するクエリを指定します。

注: 詳細については、メソッド

[PolicyMgtODBCQueryScheme->QuerySetPassword](#) (P. 322) を参照してください。

戻り値

CreateODBCQueryScheme メソッドは以下のいずれかの値を返します。

- PolicyMgtODBCQueryScheme (オブジェクト)
- コールに失敗した場合または ODBC クエリ方式名がすでに存在する場合は、**undef**

CreatePwdPolicy メソッド -- パスワード ポリシーを作成

CreatePwdPolicy メソッドは、パスワード ポリシーを作成および設定します。

構文

CreatePwdPolicy メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->CreatePwdPolicy(pwdPolName, userDir[, pwdPolDesc][, enabledFlag][, entireDirFlag][, path][, class][, allowNestedGroups][, maxLoginFailures][, maxLoginInactive][, expDelay][, expWarningDays][, dicName][, dicMatchLength][, userwait][, pwdSvcRedirect][maxPwdLength][, minPwdLength][, maxPwdRepeatChar][, minPwdAlphaNum][, minPwdAlpha][, minPwdNonAlpha][, minPwdNonPrint][, minPwdNum][, minPwdPunc][, pwdReuseCount][, pwdReuseDelay][, pwdPctDiff][, pwdIgnoreSeq][, profileAttrMatch])
```

パラメータ

CreatePwdPolicy メソッドは、以下のパラメータを受け入れます。

pwdPolName (文字列)

パスワード ポリシーの名前を指定します。

userDir (PolicyMgtUserDir オブジェクト)

パスワード ポリシーが適用されるユーザ ディレクトリを指定します。

pwdPolDesc (文字列)

(オプション) パスワード ポリシーの説明を指定します。

enabledFlag (int)

(オプション) パスワード ポリシーを有効にするかどうかを指定します。

entireDirFlag (int)

(オプション) LDAP ディレクトリ全体またはその一部のみにパスワード ポリシーを適用するかどうかを指定します。

■ 値 = 1

パスワード ポリシーが LDAP ディレクトリ全体に適用されることを指定します。

■ 値 = 0

パスワード ポリシーが LDAP ディレクトリの一部のみに適用されることを指定します。

注: LDAP ディレクトリの一部の場合は、*path* パラメータでディレクトリパスを指定し、*class* パラメータでクラスを指定します。

path (文字列)

(オプション) パスワード ポリシーが適用されるディレクトリの一部を指定します。

注: *entireDirFlag* パラメータが 0 に設定されるときに、このパラメータを含めます。

class (文字列)

(オプション) パスワード ポリシーが適用されるクラスを指定します。

注: *entireDirFlag* パラメータが 0 に設定されるときに、このパラメータを含めます。

allowNestedGroups (int)

(オプション) パスワード ポリシーが、LDAP ディレクトリ内のネストされたグループに関連付けられるかを指定します。

注: *entireDirFlag* パラメータが 0 に設定されるときに、このパラメータを含めます。

maxLoginFailures (int)

(オプション) ユーザのアカウントが無効になるまでに許可される最大ログイン失敗数を指定します。

maxLoginInactive (int)

(オプション) ユーザのパスワードが期限切れになる前に許可される非アクティブ状態の最大日数を設定

expDelay (int)

(オプション) 期限切れになるまでにパスワードを未変更にできる日数を指定します。

expWarningDays (int)

(オプション) パスワードが期限切れになることをユーザに事前に通知する場合の日数を設定します。

dicName (文字列)

(オプション) パスワードに使用できない語をリスト表示するディクショナリ ファイルの場所を指定します。

dicMatchLength (int)

(オプション) ディクショナリ確認に必要な文字の最小数を指定します。

userwait (int)

(オプション) アカウントが有効になりユーザが再度ログイン試行を許可されるまでにアカウントが無効になる時間数(分)を指定します。

pwdSvcRedirect (文字列)

(オプション) 無効なパスワードが入力されたときにユーザがリダイレクトされる場合の URL を指定します。

注: 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。

maxPwdLength (int)

(オプション) ユーザ パスワードの最大長を指定します。

注: この値はパラメータ *minPwdLength* によって指定された値よりも大きくする必要があります。

minPwdLength (int)

(オプション) ユーザ パスワードの最小長を指定します。

maxPwdRepeatChar (int)

(オプション) パスワード内で連続して使用可能な同一文字の最大数を指定します。

minPwdAlphaNum (int)

(オプション) パスワードに使用する必要がある英数字 (A ~ Z、a ~ z、0 ~ 9) の最小数を指定します。

minPwdAlpha (int)

(オプション) パスワードに使用する必要があるアルファベット文字 (A ~ Z、a ~ z) の最小数を指定します。

***minPwdNonAlpha* (int)**

(オプション) パスワードに使用する必要がある英数字以外の文字最小数を指定します。

注: 次の例は英数字以外の文字の例です。「@」、「\$」および「*」。

***minPwdNonPrint* (int)**

(オプション) パスワードに使用する必要がある印刷不能文字の最小数を指定します。

注: 印刷不能文字はコンピュータ画面に表示されません。

***minPwdNum* (int)**

(オプション) パスワードに使用する必要がある数字 (0 ~ 9) の最小数を指定します。

***minPwdPunc* (int)**

(オプション) パスワードに使用する必要がある句読点の最小数を指定します。

注: 句読点にはピリオド、カンマ、感嘆符、スラッシュ、ハイフン、ダッシュが含まれます。

***pwdReuseCount* (int)**

(オプション) 古いパスワードを再利用するまでに使用する必要がある新しいパスワードの数を指定します。

***pwdReuseDelay* (int)**

(オプション) パスワードを再利用する前にユーザが待機する必要がある日数を指定します。

***pwdPctDiff* (int)**

(オプション) 以前のパスワード内の文字と異なる必要がある新しいパスワード内の文字のパーセンテージを指定します。

注: 値 100 は、新しいパスワードには古いパスワード内の文字を使用できないことを指定します。詳細は、パラメータ *pwdIgnoreSeq* を参照してください。

***pwdIgnoreSeq* (int)**

(オプション) 新しいパスワードが古いパスワードと比較され、異なる文字のパーセンテージが計算されるときに、文字位置が無視されるかどうかを指定します。

■ 値 = 1

文字シーケンスが無視されることを指定します。

■ 値 = 0

文字シーケンスが考慮されることを指定します。

例：文字「c」が新しいパスワードと古いパスワードの両方にあるが、各パスワード内のその位置が異なる場合、パーセンテージが計算されるときに、2つの別の文字と見なされます。

***profileAttrMatch* (int)**

(オプション) **SiteMinder** がユーザのディレクトリ エントリ内の属性とパスワードを照合するときの最小文字シーケンスを指定します。

戻り値

CreatePwdPolicy メソッドは以下のいずれかの値を返します。

- **PolicyMgtPwdPolicy** (オブジェクト)
- コールに失敗した場合またはパスワード ポリシー名がすでに存在する場合は、**undef**

CreateRegScheme メソッド -- 登録方式を作成

CreateRegScheme メソッドは、登録方式を作成および設定します。

構文

CreateRegScheme メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->CreateRegScheme(regName, userDir[, regDesc][, welcomeURL][, templatePath][, enableLogging])
```

パラメータ

CreateRegScheme メソッドは、以下のパラメータを受け入れます。

regName (文字列)

登録方式の名前を指定します。

userDir (文字列)

登録方式に関連付けられたユーザ ディレクトリを指定します。

regDesc (文字列)

(オプション) 登録方式の説明を指定します。

welcomeURL (文字列)

(オプション) ようこそページの URL を指定します。

注: ユーザは正しく登録した後にこのページにリダイレクトされます。

例: `http://my.acme.com/hr/welcome.htm`

templatePath (文字列)

(オプション) 登録テンプレートがあるパスを指定します。

注: *templatePath* パラメータの詳細については、「注釈」を参照してください。

enableLogging (int)

(オプション) ログ記録を有効にするかどうかを指定します。

- 値 = 1

ログ記録の有効化を指定します。

- 値 = 0 (デフォルト)

ログ記録の無効化を指定します。

戻り値

CreateRegScheme メソッドは以下のいずれかの値を返します。

- PolicyMgtRegScheme (オブジェクト)
- コールに失敗した場合または登録方式名がすでに存在する場合は、**undef**

注釈

SiteMinder Web エージェントをインストールすると、デフォルトでは Web エージェント インストール ディレクトリのサブディレクトリ `samples/selfreg` に登録テンプレートがインストールされます。SiteMinder インストール時に、仮想ディレクトリ `/siteminderagent` が作成され、Web エージェント インストール ディレクトリ内のサンプル ディレクトリが示されます。したがって、デフォルトディレクトリを使用するときは、次のように *templePath* を指定します。`/siteminderagent/selfreg` (最後のスラッシュなし)。

登録用の SSL を使用している場合は、登録テンプレート用の絶対パスを指定する必要があります。デフォルトパスは以下のとおりです。

- Windows プラットフォーム : `install-dir¥Netegrity¥Siteminder Web Agent¥Samples¥SelfReg¥`
- Solaris/Hpux プラットフォーム : `install-dir/netegrity/siteminder/webagent/samples/selfreg/`

CreateSAMLAffiliation メソッド -- SAML 2.0 アフィリエーション オブジェクトを作成

CreateSAMLAffiliation メソッドは、SAML 2.0 アフィリエーション オブジェクトを作成します。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。既存の SAML アフィリエーションのプロパティを変更するには、メソッド `PolicyMgtSAMLAffiliation->Property` をコールします。

構文

CreateSAMLAffiliation メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->CreateSAMLAffiliation(propsHash_ref)
```

パラメータ

CreateSAMLAffiliation メソッドは以下のパラメータを受け入れます。

propsHash_ref (ハッシュ)

SAML 2.0 アフィリエーションのメタデータ プロパティのハッシュテーブルの参照を指定します。

例 : `¥%myhash`

戻り値

CreateSAMLAffiliation メソッドは以下のいずれかの値を返します。

- PolicyMgtSAMLAffiliation (オブジェクト)
- コールに失敗した場合は、**undef**

注釈

SAML 2.0 アフィリエーション プロパティは、以下のようにグループ化されます。

一般的なプロパティ

SAML_NAME
SAML_DESCRIPTION
SAML_KEY_AFFILIATION_ID
SAML_MAJOR_VERSION
SAML_MINOR_VERSION
SAML_OID

名前 ID プロパティ

SAML_SP_NAMEID_FORMAT
SAML_SP_NAMEID_TYPE
SAML_SP_NAMEID_STATIC
SAML_SP_NAMEID_ATTRNAME
SAML_SP_NAMEID_DNSPEC

ユーザ プロパティ

SAML_IDP_XPATH
SAML_IDP_LDAP_SEARCH_SPEC
SAML_IDP_ODBC_SEARCH_SPEC
SAML_IDP_WINNT_SEARCH_SPEC
SAML_IDP_CUSTOM_SEARCH_SPEC
SAML_IDP_AD_SEARCH_SPEC

詳細については、このガイドの「**SAML 2.0** プロパティ リファレンス」を参照してください。

CreateSAMLAuthScheme メソッド -- SAML 認証方式オブジェクトを作成

CreateSAMLAuthScheme メソッドは、SAML 2.0 認証方式オブジェクトをそのプロパティが指定された値に設定された状態で作成します。オブジェクトと関連付けられた次の 2 つのタイプのプロパティがあります。認証方式プロパティおよびメタデータ プロパティ。

認証方式プロパティ

認証方式プロパティは、SAML 2.0 テンプレートに基づき、以下の初期値があります。

- ライブラリ
デフォルト : smauthsaml
- Is_Template
デフォルト : FALSE
- Is_Used_by_Administrator
デフォルト : FALSE
- Save_Credentials
デフォルト : FALSE
- Is_RADIUS
デフォルト : FALSE
- Ignore_Password_Check
デフォルト : TRUE
- Protection_Level
デフォルト : 5

注: デフォルト保護レベルを変更するには、CreateSAMLAuthScheme メソッドをコールし、オプションの *protLevel* パラメータを新しい値に設定した状態にします。

メタデータプロパティ

メタデータ プロパティは、SAML 2.0 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のプロパティで、認証方式と共に保存されます。それらを指定するには、メタデータ プロパティのハッシュテーブルへの参照を *propsHash_ref* パラメータ内の `CreateSAMLAuthScheme` メソッドに渡します。既存の SAML 2.0 認証方式を更新するには、メソッド `PolicyMgtSession->SAMLAuthSchemeProperties` をコールします。

構文

`CreateSAMLAuthScheme` メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->CreateSAMLAuthScheme(schemeName, propsHash_ref[,  
schemeDesc][, protLevel])
```

パラメータ

`CreateSAMLAuthScheme` メソッドは、以下のパラメータを受け入れます。

schemeName (文字列)

認証方式の名前を指定します。

propsHash_ref (ハッシュ)

SAML 2.0 認証方式に関連付けられたメタデータ プロパティのハッシュテーブルの参照を指定します。

例：`¥%myhash`

注: メタデータ プロパティの全リストについては、「注釈」を参照してください。

schemeDesc (文字列)

(オプション) 認証方式の説明を指定します。

protLevel (int)

(オプション) 認証方式の保護レベルを指定します。

戻り値

`CreateSAMLAuthScheme` メソッドは以下のいずれかの値を返します。

- `PolicyMgtAuthScheme` (オブジェクト)
- コールに失敗した場合または SAML 認証方式名がすでに存在する場合は、`undef`

注釈

SAML 2.0 認証方式と関連付けられたメタデータ プロパティを以下に示します。

一般的なプロパティ

SAML_NAME

SAML_DESCRIPTION

方式セットアップ プロパティ

SAML_IDP_SPID

SAML_KEY_IDPID

SAML_MAJOR_VERSION

SAML_MINOR_VERSION

SAML_SKEWTIME

SAML_DISABLE_SIGNATURE_PROCESSING

SAML_DSIG_VERINFO_ISSUER_DN

SAML_DSIG_VERINFO_SERIAL_NUMBER

ユーザ プロパティ

SAML_IDP_XPATH

SAML_IDP_LDAP_SEARCH_SPEC

SAML_IDP_ODBC_SEARCH_SPEC

SAML_IDP_WINNT_SEARCH_SPEC

SAML_IDP_CUSTOM_SEARCH_SPEC

SAML_IDP_AD_SEARCH_SPEC

SAML_AFFILIATION

SSO プロパティ

SAML_IDP_SSO_REDIRECT_MODE

SAML_IDP_SSO_DEFAULT_SERVICE

SAML_AUDIENCE

SAML_IDP_SSO_TARGET

SAML_ENABLE_SSO_ARTIFACT_BINDING

SAML_KEY_IDP_SOURCEID

SAML_IDP_ARTIFACT_RESOLUTION_DEFAULT_SERVICE

SAML_IDP_BACKCHANNEL_AUTH_TYPE

SAML_IDP_SPNAME

SAML_IDP_PASSWORD

SAML_ENABLE_SSO_POST_BINDING

SAML_IDP_SSO_ENFORCE_SINGLE_USE_POLICY

SAML_SSOECPPROFILE

SAML_IDP_SIGN_AUTHNREQUESTS

SLO プロパティ

SAML_SLO_REDIRECT_BINDING

SAML_SLO_SERVICE_VALIDITY_DURATION

SAML_SLO_SERVICE_URL

SAML_SLO_SERVICE_RESPONSE_URL

SAML_SLO_SERVICE_CONFIRM_URL

暗号化プロパティ

SAML_IDP_REQUIRE_ENCRYPTED_ASSERTION

SAML_IDP_REQUIRE_ENCRYPTED_NAMEID

属性プロパティ

SAML_IDP_SAMLREQ_ENABLE

SAML_IDP_SAMLREQ_REQUIRE_SIGNED_ASSERTION

SAML_IDP_SAMLREQ_ATTRIBUTE_SERVICE

SAML_IDP_SAMLREQ_GET_ALL_ATTRIBUTES

名前 ID プロパティ

SAML_IDP_SAMLREQ_NAMEID_FORMAT

SAML_IDP_SAMLREQ_NAMEID_TYPE

SAML_IDP_SAMLREQ_NAMEID_STATIC

SAML_IDP_SAMLREQ_NAMEID_ATTR_NAME

SAML_IDP_SAMLREQ_NAMEID_DN_SPEC

SAML_IDP_SAMLREQ_NAMEID_ALLOW_NESTED

拡張プロパティ

SAML_SP_PLUGIN_CLASS
SAML_SP_PLUGIN_PARAMS
SAML_IDP_REDIRECT_URL_USER_NOT_FOUND
SAML_IDP_REDIRECT_MODE_USER_NOT_FOUND
SAML_IDP_REDIRECT_URL_FAILURE
SAML_IDP_REDIRECT_MODE_FAILURE
SAML_IDP_REDIRECT_URL_INVALID
SAML_IDP_REDIRECT_MODE_INVALID

CreateSingleCertMap メソッド -- 単一属性証明書マップを作成

CreateSingleCertMap メソッドは、証明書の Subject DN 内の単一属性と、ユーザ ディレクトリ内の対応するユーザ属性間の証明書マップを作成します。認証を成功させるには、Subject DN 内の属性の値と、ユーザ ディレクトリ内の対応するユーザ属性の値が一致する必要があります。

構文

CreateSingleCertMap メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->CreateSingleCertMap(IssuerDN, Attribute[,  
DirectoryType])
```

パラメータ

CreateSingleCertMap メソッドは、以下のパラメータを受け入れます。

IssuerDN (文字列)

証明書発行者の識別名を指定します。

Attribute (文字列)

証明書の Subject DN とユーザ ディレクトリ内でその値が一致する必要がある属性の名前を指定します。

構文: %{attribute_name}

例: %{uid}

DirectoryType (int)

(オプション) 以下のように、認証用に指定されるユーザ ディレクトリのタイプを指定します。

- `Sm_PolicyApi_DirType_LDAP` (デフォルト)
- `Sm_PolicyApi_DirType_WinNT`
- `Sm_PolicyApi_DirType_ODBC`

戻り値

`CreateSingleCertMap` メソッドは以下のいずれかの値を返します。

- `PolicyMgtCertMap` (オブジェクト)
- コールに失敗した場合は、**undef**

注釈

証明書マップが作成されると、以下のフラグがデフォルト値である `false` に設定されます。

- `certificate_required_flag`
- `use_distributionpoints_flag`
- `verify_signature_flag`
- `check_certificate_revocation_list_flag`
- `cache_certificate_revocation_list_entries_flag`

これらのフラグの値の変更の詳細については、メソッド [PolicyMgtSession->CreateExactCertMap](#) (P. 457) を参照してください。

CreateTrustedHost メソッド -- トラストド ホスト オブジェクトを作成

CreateSAMLAuthScheme メソッドは、ポリシー ストアでトラストド ホスト オブジェクトを作成し、ポリシー サーバにトラストド ホストを登録します。さらに、登録が成功すると、ローカル登録ファイルを作成します。ポリシー サーバがエージェント ホストに接続されるときに、このメソッドを使用します。ポリシー サーバとエージェント ホスト間に接続がないときは、代わりにメソッド `PolicyMgtSession->AddTrustedHost` をコールします。

構文

CreateTrustedHost メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->CreateTrustedHost(trustedHostName [, ipAddress][, adminName][, adminPassword][, hostConfigName][, registrationDataFileName])
```

パラメータ

CreateTrustedHost メソッドは、以下のパラメータを受け入れます。

trustedHostName (文字列)

トラストド ホストの名前を指定します。

ipAddress (文字列)

(オプション) ポリシー サーバの IP アドレスを指定します。

adminName (文字列)

(オプション) ポリシー サーバ管理者の名前を指定します。

adminPassword (文字列)

(オプション) 管理者のパスワードを指定します。

hostConfigName (文字列)

(オプション) ホスト設定オブジェクトの名前を指定します。

registrationDataFileName (文字列)

(オプション) ホストがポリシー サーバに正しく登録されると、登録データが書き込まれるファイルの名前を指定します。

注: このファイル名はエージェント API メソッドの `Connect` をコールすることによって指定されます。ファイルは `SiteMinder` によって保存され管理されます。

戻り値

CreateTrustedHost メソッドは以下のいずれかの値を返します。

- PolicyMgtTrustedHost (オブジェクト)
- コールが失敗した場合またはトラステッド ホスト名がすでに存在する場合は、**undef**

CreateUserDir メソッド -- ユーザ ディレクトリ オブジェクトを作成

CreateUserDir メソッドは、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。

構文

CreateUserDir メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->CreateUserDir(dirName, namespace, server[,  
ODBCQueryScheme][, domDesc][, searchRoot][, usrLookStart][, usrLookEnd][,  
username][, password][, searchResults][, searchScope][, searchTimeout][,  
secureConn][, requireCreds][, disabledAttr][, UIDAttr][, anonID][, pwdData][,  
pwdAttr][, emailAttr][, chalRespAttr])
```

パラメータ

CreateUserDir メソッドは、以下のパラメータを受け入れます。

dirName (文字列)

ユーザ ディレクトリ オブジェクトの名前を指定します。

namespace (文字列)

以下のように、ユーザ ディレクトリのネームスペースを指定します。

- LDAP
- AD
- ODBC
- Windows NT
- カスタム

server (文字列)

以下のいずれかのディレクトリ従属値を指定します。

LDAP と AD

LDAP サーバの IP アドレスおよびポート番号を指定します。

構文: IP_address:port_number

注: デフォルト ポート番号は 389 です。

ODBC

データ ソース名を指定します。

Windows NT

ドメイン名を指定します。

カスタム

カスタム ディレクトリに対応するライブラリの名前を指定します。

ODBCQueryScheme (PolicyMgtODBCQueryScheme オブジェクト)

(オプション) ODBC ディレクトリのクエリを実行するために SiteMinder が使用する一連のクエリを指定します。

注: ユーザ ディレクトリが ODBC ディレクトリでない場合、このパラメータの値は *undef* です。

domDesc (文字列)

(オプション) ユーザ ディレクトリの説明を指定します。

searchRoot (文字列)

(オプション) 以下のいずれかのディレクトリ従属値を指定します。

LDAP

ディレクトリ接続の開始ポイントである LDAP ツリー内の場所を指定します。たとえば、組織 (o) または組織単位 (ou) などです。検索ルートと呼ばれるこの場所は、ポリシー サーバがユーザに対して検索を開始するポイントです。

注: このパラメータの詳細については、パラメータ *searchScope* を参照してください。

カスタム

カスタム ライブラリに渡すパラメータ文字列を指定します。

usrLookStart (文字列)

(オプション) LDAP ディレクトリ内のユーザ DN 検索用の開始値を指定します。

usrLookEnd (文字列)

(オプション) LDAP ディレクトリ内のユーザ DN 検索用の終了値を指定します。

注: ユーザ DN 検索開始ポイントおよびエンドポイントの値を指定すると、認証時に DN 文字列の一部を入力できます。以下の例で、ユーザはログインするときに DN 文字列全体ではなく、文字列「JSmith」のみを指定する必要があります。

- `DN = "uid=JSmith,ou=marketing,o=myorg.org"`
- `starting_point = "uid="`
- `endpoint = ",ou=marketing,o=myorg.org"`
- `login = "JSmith"`

username (文字列)

(オプション) ユーザディレクトリにアクセスするために必要なユーザ名を指定します。

注: このパラメータを使用するときは、*requireCreds* を 1 に設定します。

password (文字列)

(オプション) ユーザディレクトリにアクセスするために必要なパスワードを指定します。

注: このパラメータを使用するときは、*requireCreds* を 1 に設定します。

searchResults (int)

(オプション) LDAP またはカスタム ディレクトリの検索から返される結果の最大数を指定します。

searchScope (int)

(オプション) 以下のように、LDAP ディレクトリ内のユーザまたはユーザ グループを探すときに、SiteMinder 検索のレベル数を指定します。

- `USERDIR_SCOPE_SUBTREE`

ルートおよびその下のすべてのレベルの検索を指定します。

- **USERDIR_SCOPE_ONELEVEL**

ルートおよびその 1 つ下のレベルの検索を指定します。

注: 詳細については、*searchRoot* パラメータを参照してください。

searchTimeout (int)

(オプション) LDAP またはカスタム ディレクトリの検索に許容できる最大時間を秒単位で指定します。

secureConn (int)

(オプション) 以下のように、LDAP またはカスタム ユーザ ディレクトリ接続が SSL によってセキュア保護されるかどうかを指定します。

- 値 = 1

SSL によってセキュア保護される接続を指定します。

- 値 = 0 (デフォルト)

セキュア保護されない接続を指定します。

注: このフラグが有効な場合、SiteMinder 認証はセキュアであり、転送は暗号化されます。SSL を使用するときこのフラグを有効にします。

requireCreds (int)

(オプション) 以下のように、ユーザ認証情報が認証に必要なかどうかを指定します。

- 値 = 1

認証情報が必要であることを指定します。

- 値 = 0 (デフォルト)

認証情報が必要でないことを指定します。

disabledAttr (文字列)

(オプション) ユーザの無効化状態が含まれているユーザ ディレクトリ属性の名前を指定します。

注: このパラメータは、LDAP ディレクトリ、ODBC ディレクトリおよびいくつかのカスタム ディレクトリに適用されます。

UIDAttr (文字列)

(オプション) ユーザ ディレクトリのユニバーサル ID 属性の名前を指定します。

注: 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。このパラメータは、LDAP ディレクトリ、ODBC ディレクトリ、WinNT ディレクトリおよびいくつかのカスタム ディレクトリに適用されます。

anonID (文字列)

(オプション) ユーザ ディレクトリの匿名ユーザ DN 属性の名前を指定します。

注: 匿名の認証方式で定義される DN は、匿名の認証方式によって保護されたりソースへの匿名ユーザ アクセス権を与えます。このパラメータは、LDAP ディレクトリおよびいくつかのカスタム ディレクトリに適用されます。

pwdData (文字列)

(オプション) ユーザ ディレクトリのパスワードデータ属性の名前を指定します。

注: このパラメータは、LDAP ディレクトリ、ODBC ディレクトリおよびいくつかのカスタム ディレクトリに適用されます。

pwdAttr (文字列)

(オプション) ユーザ ディレクトリのパスワード属性の名前を指定します。

注: このパラメータは、LDAP ディレクトリ、ODBC ディレクトリおよびいくつかのカスタム ディレクトリに適用されます。

emailAttr (文字列)

注: このオプションのパラメータは将来の使用のために予約されています。

chalRespAttr (文字列)

(オプション) ユーザ ディレクトリのチャレンジ/レスポンス属性の名前を指定します。

例: チャレンジ/レスポンスは、ユーザがパスワードを忘れたときに、SiteMinder がユーザに送信するヒントにできます。

注: このパラメータは、LDAP ディレクトリおよびいくつかのカスタム ディレクトリに適用されます。

戻り値

CreateUserDir メソッドは以下のいずれかの値を返します。

- PolicyMgtUserDir (オブジェクト)
- コールに失敗した場合は、**undef**

CreateWSFEDAuthScheme メソッド -- WS-Federation 認証方式を作成

CreateWSFEDAuthScheme メソッドは、WS-Federation 認証方式のインスタンスを作成し、認証方式のプロパティを設定します。

構文

CreateWSFEDAuthScheme メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->CreateWSFEDAuthScheme(name, propsHash[, desc][, level])
```

パラメータ

CreateWSFEDAuthScheme メソッドは、以下のパラメータを受け入れます。

name (文字列)

WS-Federation 認証方式の名前を指定します。

propsHash (ハッシュ テーブル)

設定する WS-Federation 認証方式プロパティのハッシュテーブルの参照を指定します。

注: WS-Federation 認証方式プロパティの全リストについては、「注釈」を参照してください。

desc (文字列)

(オプション) 認証方式の説明を指定します。

level (int)

(オプション) 認証方式レベルを指定します。

戻り値

CreateWSFEDAuthScheme メソッドは以下のいずれかの値を返します。

- PolicyMgtAuthScheme (オブジェクト)
- コールに失敗した場合は、**undef**

注釈

WS-Federation 認証方式プロパティは、FSS 管理 UI で以下のようにグループ化されます。

一般的なプロパティ

WSFED_NAME

WSFED_DESCRIPTION

[方式のセットアップ]タブ

WSFED_KEY_APID

WSFED_RPID

WSFED_SKEW_TIME

WSFED_DISABLE_SIGNATURE_PROCESSING

WSFED_DSIG_VERINFO_ALIAS

追加設定、[ユーザ]タブ

WSFED_AP_XPATH

WSFED_AP_LDAP_SEARCH_SPEC

WSFED_AP_ODBC_SEARCH_SPEC

WSFED_AP_WINNT_SEARCH_SPEC

WSFED_AP_CUSTOM_SEARCH_SPEC

WSFED_AP_ADD_SEARCH_SPEC

追加設定、[SSO]タブ

WSFED_AP_SSO_REDIRECT_MODE

WSFED_AP_SSO_DEFAULT_SERVICE

WSFED_AP_SSO_TARGET

WSFED_ENFORCE_SINGLE_USE_POLICY

追加設定、[サインアウト]タブ

WSFED_AP_SLO_ENABLED

WSFED_AP_SIGNOUT_URL

追加設定、[詳細]タブ

WSFED_AP_PLUGIN_CLASS

WSFED_AP_PLUGIN_PARAMS

WSFED_AP_USER_NOT_FOUND_REDIRECT_URL

WSFED_AP_USER_NOT_FOUND_REDIRECT_MODE

WSFED_AP_FAILURE_REDIRECT_URL

WSFED_AP_FAILURE_REDIRECT_MODE

WSFED_AP_INVALID_REDIRECT_URL

WSFED_AP_INVALID_REDIRECT_MODE

DeleteAdmin メソッド -- 管理者を削除

DeleteAdmin メソッドは、ポリシー ストアから管理者を削除します。

構文

DeleteAdmin メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->DeleteAdmin(admin)
```

パラメータ

DeleteAdmin メソッドは以下のパラメータを受け入れます。

admin (PolicyMgtAdmin オブジェクト)

削除する管理者オブジェクトを指定します。

戻り値

DeleteAdmin メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことまたは管理者が見つからないことを指定します。
- 値 = -1
メソッドが失敗したことを指定します。

注釈

特定のドメインから管理者を削除するには、メソッド
[PolicyMgtAffDomain->RemoveAdmin](#) (P. 189) を参照してください。

DeleteAffDomain メソッド -- アフィリエイトドメインを削除

DeleteAffDomain メソッドは、アフィリエイト ドメインを削除します。

構文

DeleteAffDomain メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->DeleteAffDomain(affDomain)
```

パラメータ

DeleteAffDomain メソッドは以下のパラメータを受け入れます。

affDomain (PolicyMgtAffDomain オブジェクト)

削除するアフィリエイト ドメイン オブジェクトを指定します。

戻り値

DeleteAffDomain メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことまたはアフィリエイト ドメインが見つからないことを指定します。
- 値 = -1
メソッドが失敗したことを指定します。

DeleteAgent メソッド -- エージェントを削除

DeleteAgent メソッドは、エージェントを削除します。

構文

DeleteAgent メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->DeleteAgent(agent)
```

パラメータ

DeleteAgent メソッドは以下のパラメータを受け入れます。

agent (PolicyMgtAgent オブジェクト)

削除するエージェント オブジェクトを指定します。

戻り値

DeleteAgent メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことまたはエージェントが見つからないことを指定します。
- 値 = -1
メソッドが失敗したことを指定します。

DeleteAgentConfig メソッド -- エージェント設定オブジェクトを削除

DeleteAgentConfig メソッドは、エージェント設定オブジェクトを削除します。

構文

DeleteAgentConfig メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->DeleteAgentConfig(AgentConfig)
```

パラメータ

DeleteAgentConfig メソッドは以下のパラメータを受け入れます。

AgentConfig (PolicyMgtAgentConfig オブジェクト)

削除するエージェント設定オブジェクトを指定します。

戻り値

DeleteAgentConfig メソッドは以下のいずれかの値を返します。

- 値 = 0

メソッドが成功したことまたはエージェント設定オブジェクトが見つからないことを指定します。

- 値 = -1

メソッドが失敗したことを指定します。

DeleteAuthAzMap メソッド -- 認証および許可マップを削除

DeleteAuthAzMap メソッドは、認証および許可マップを削除します。

構文

DeleteAuthAzMap メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->DeleteAuthAzMap(map)
```

パラメータ

DeleteAuthAzMap メソッドは以下のパラメータを受け入れます。

map (PolicyMgtAuthAzMap オブジェクト)

削除する認証および許可マップ オブジェクトを指定します。

戻り値

DeleteAuthAzMap メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことまたは認証および許可マップ オブジェクトが見つからないことを指定します。
- 値 = -1
メソッドが失敗したことを指定します。

DeleteAuthScheme メソッド -- 認証方式を削除

DeleteAuthScheme メソッドは、認証方式を削除します。

構文

DeleteAuthScheme メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->DeleteAuthScheme(authScheme)
```

パラメータ

DeleteAuthScheme メソッドは以下のパラメータを受け入れます。

authScheme (PolicyMgtAuthScheme オブジェクト)

削除する認証方式オブジェクトを指定します。

戻り値

DeleteAuthScheme メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことまたは認証方式が見つからないことを指定します。
- 値 = -1
メソッドが失敗したことを指定します。

DeleteCertMap メソッド -- 証明書マップを削除

DeleteCertMap メソッドは、証明書マップを削除します。

構文

DeleteCertMap メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->DeleteCertMap(map)
```

パラメータ

DeleteCertMap メソッドは以下のパラメータを受け入れます。

map (PolicyMgtCertMap オブジェクト)

削除する証明書マップ オブジェクトを指定します。

戻り値

DeleteCertMap メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことまたは証明書マップが見つからないことを指定します。
- 値 = -1
メソッドが失敗したことを指定します。

DeleteDomain メソッド -- ポリシー ドメインを削除

DeleteDomain メソッドは、ポリシー ドメインを削除します。

構文

DeleteDomain メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->DeleteDomain(domain)
```

パラメータ

DeleteDomain メソッドは以下のパラメータを受け入れます。

domain (PolicyMgtDomain オブジェクト)

削除するドメイン オブジェクトを指定します。

戻り値

DeleteDomain メソッドは以下のいずれかの値を返します。

- 値 = 0

メソッドが成功したことまたはドメインが見つからないことを指定します。

- 値 = -1

メソッドが失敗したことを指定します。

DeleteGlobalPolicy メソッド -- グローバル ポリシーを削除

DeleteGlobalPolicy メソッドは、グローバル ポリシーを削除します。

構文

DeleteGlobalPolicy メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->DeleteGlobalPolicy(policy)
```

パラメータ

DeleteGlobalPolicy メソッドは以下のパラメータを受け入れます。

policy (PolicyMgtPolicy オブジェクト)

削除するグローバル ポリシー オブジェクトを指定します。

戻り値

DeleteGlobalPolicy メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことを指定します。
- 値 = -1
メソッドが失敗したことを指定します。

DeleteGlobalResponse メソッド -- グローバル レスポンスを削除

DeleteGlobalResponse メソッドは、グローバル レスポンスを削除します。

構文

DeleteGlobalResponse メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->DeleteGlobalResponse(response)
```

パラメータ

DeleteGlobalResponse メソッドは以下のパラメータを受け入れます。

response (PolicyMgtResponse オブジェクト)

削除するグローバル レスポンス オブジェクトを指定します。

戻り値

DeleteGlobalResponse メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことを指定します。
- 値 = -1
メソッドが失敗したことを指定します。

DeleteGlobalRule メソッド -- グローバル ルールを削除

DeleteGlobalRule メソッドは、グローバル ルールを削除します。

構文

DeleteGlobalRule メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->DeleteGlobalRule(rule)
```

パラメータ

DeleteGlobalRule メソッドは以下のパラメータを受け入れます。

rule (PolicyMgtRule オブジェクト)

削除するグローバル ルール オブジェクトを指定します。

戻り値

DeleteGlobalRule メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことを指定します。
- 値 = -1
メソッドが失敗したことを指定します。

DeleteGroup メソッド -- エージェント グループを削除

DeleteGroup メソッドは、エージェント グループを削除します。

構文

DeleteGroup メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->DeleteGroup(group)
```

パラメータ

DeleteGroup メソッドは以下のパラメータを受け入れます。

group (PolicyMgtGroup オブジェクト)

削除するエージェント グループ オブジェクトを指定します。

戻り値

DeleteGroup メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことまたはエージェント グループが見つからないことを指定します。
- undef
メソッドが失敗したことを指定します。

DeleteHostConfig メソッド -- ホスト設定オブジェクトを削除

DeleteHostConfig メソッドは、ホスト設定オブジェクトを削除します。

構文

DeleteHostConfig メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->DeleteHostConfig(HostConfig)
```

パラメータ

DeleteHostConfig メソッドは以下のパラメータを受け入れます。

HostConfig (PolicyMgtHostConfig オブジェクト)

削除するホスト設定オブジェクトを指定します。

戻り値

DeleteHostConfig メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことまたはホスト設定オブジェクトが見つからないことを指定します。
- 値 = -1
メソッドが失敗したことを指定します。

DeleteODBCQueryScheme メソッド -- ODBC クエリ方式を削除

DeleteODBCQueryScheme メソッドは、ODBC クエリ方式を削除します。

構文

DeleteODBCQueryScheme メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->DeleteODBCQueryScheme(scheme)
```

パラメータ

DeleteODBCQueryScheme メソッドは以下のパラメータを受け入れます。

scheme (PolicyMgtODBCQueryScheme オブジェクト)

削除する ODBC クエリ方式オブジェクトを指定します。

戻り値

DeleteODBCQueryScheme メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことまたは ODBC クエリ方式が見つからないことを指定します。
- 値 = -1
メソッドが失敗したことを指定します。

DeletePwdPolicy メソッド -- パスワード ポリシーを削除

DeletePwdPolicy メソッドは、パスワード ポリシーを削除します。

構文

DeletePwdPolicy メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->DeletePwdPolicy(pwdPolicy)
```

パラメータ

DeletePwdPolicy メソッドは以下のパラメータを受け入れます。

pwdPolicy (PolicyMgtPwdPolicy オブジェクト)

削除するパスワード ポリシー オブジェクトを指定します。

戻り値

DeletePwdPolicy メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことまたはパスワード ポリシーが見つからないことを指定します。
- 値 = -1
メソッドが失敗したことを指定します。

DeleteRegScheme メソッド -- 登録方式を削除

DeleteRegScheme メソッドは、登録方式を削除します。

構文

DeleteRegScheme メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->DeleteRegScheme(regScheme)
```

パラメータ

DeleteRegScheme メソッドは以下のパラメータを受け入れます。

regScheme (PolicyMgtRegScheme オブジェクト)

削除する登録方式オブジェクトを指定します。

戻り値

DeleteRegScheme メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことまたは登録方式が見つからないことを指定します。
- 値 = -1
メソッドが失敗したことを指定します。

DeleteSAMLAffiliation メソッド -- SAML アフィリエーションを削除

DeleteSAMLAffiliation メソッドは、SAML 2.0 アフィリエーション オブジェクトを削除します。

構文

DeleteSAMLAffiliation メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->DeleteSAMLAffiliation(SAMLAffil)
```

パラメータ

DeleteSAMLAffiliation メソッドは以下のパラメータを受け入れます。

SAMLAffil (PolicyMgtSAMLAffiliation オブジェクト)

削除する SAML 2.0 アフィリエーション オブジェクトを指定します。

戻り値

DeleteSAMLAffiliation メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことまたは SAML アフィリエーション オブジェクトが見つからないことを指定します。
- 値 = -1
メソッドが失敗したことを指定します。

DeleteTrustedHost メソッド -- トラストッド ホストを削除

DeleteTrustedHost メソッドは、トラストッド ホストを削除します。

構文

DeleteTrustedHost メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->DeleteTrustedHost(TrustedHost)
```

パラメータ

DeleteTrustedHost メソッドは以下のパラメータを受け入れます。

TrustedHost (PolicyMgtTrustedHost オブジェクト)

削除するトラストッド ホスト オブジェクトを指定します。

戻り値

DeleteTrustedHost メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことまたはトラストッド ホストが見つからないことを指定します。
- 値 = -1
メソッドが失敗したことを指定します。

DeleteUserDir メソッド -- ユーザ ディレクトリを削除

DeleteUserDir メソッド

構文

DeleteUserDir メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->DeleteUserDir(userdir)
```

パラメータ

DeleteUserDir メソッドは以下のパラメータを受け入れます。

userDir (PolicyMgtUserDir オブジェクト)

削除するユーザ ディレクトリ オブジェクトを指定します。

戻り値

DeleteUserDir メソッドは以下のいずれかの値を返します。

- 値 = 0

メソッドが成功したことまたはユーザ ディレクトリが見つからないことを指定します。

- 値 = -1

メソッドが失敗したことを指定します。

GetAdmin メソッド -- 管理者を取得

GetAdmin メソッドは、指定された管理者を取得します。

構文

GetAdmin メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetAdmin(adminName)
```

パラメータ

GetAdmin メソッドは以下のパラメータを受け入れます。

adminName (文字列)

取得する管理者の名前を指定します。

戻り値

GetAdmin メソッドは以下のいずれかの値を返します。

- コールに成功する場合は、PolicyMgtAdmin オブジェクト
- コールに失敗した場合または指定された管理者が存在しない場合は、**undef**

GetAffDomain メソッド -- アフィリエイトドメインを取得

GetAffDomain メソッドは、指定されたアフィリエイト ドメインを取得します。

構文

GetAffDomain メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetAffDomain(domName)
```

パラメータ

GetAffDomain メソッドは以下のパラメータを受け入れます。

domName (文字列)

取得するアフィリエイト ドメインの名前を指定します。

戻り値

GetAffDomain メソッドは以下のいずれかの値を返します。

- PolicyMgtAffDomain オブジェクト
- コールに失敗した場合または指定されたアフィリエイト ドメインが存在しない場合は、**undef**

GetAgent メソッド -- エージェントを取得

GetAgent メソッドは、指定されたエージェントを取得します。

構文

GetAgent メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetAgent(agentName)
```

パラメータ

GetAgent メソッドは以下のパラメータを受け入れます。

agentName (文字列)

取得するエージェントの名前を指定します。

戻り値

GetAgent メソッドは以下のいずれかの値を返します。

- コールに成功する場合は、PolicyMgtAgent オブジェクト
- コールに失敗した場合または指定されたエージェントが存在しない場合は、**undef**

GetAgentConfig メソッド -- エージェント設定オブジェクトを取得

GetAgentConfig メソッドは、指定されたエージェント設定オブジェクトを取得します。

構文

GetAgentConfig メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetAgentConfig(acName)
```

パラメータ

GetAgentConfig メソッドは以下のパラメータを受け入れます。

acName (文字列)

取得するエージェント設定オブジェクトの名前を指定します。

戻り値

GetAgentConfig メソッドは以下のいずれかの値を返します。

- コールに成功する場合は、PolicyMgtAgentConfig オブジェクト
- コールに失敗した場合または指定されたエージェント設定オブジェクトが存在しない場合は、**undef**

GetAgentGroup メソッド -- エージェント グループを取得

GetAgentGroup メソッドは、指定されたエージェント グループを取得します。

構文

GetAgentGroup メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetAgentGroup(agentGroup)
```

パラメータ

GetAgentGroup メソッドは以下のパラメータを受け入れます。

agentGroup (文字列)

取得するエージェント グループの名前を指定します。

戻り値

GetAgentGroup メソッドは以下のいずれかの値を返します。

- PolicyMgtGroup オブジェクト
- コールに失敗した場合または指定されたエージェント グループが存在しない場合は、**undef**

GetAgentType メソッド -- エージェント タイプを取得

GetAgentType メソッドは、指定されたエージェント タイプを取得します。

構文

GetAgentType メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetAgentType(agentTypeName)
```

パラメータ

GetAgentType メソッドは以下のパラメータを受け入れます。

agentTypeName (文字列)

取得する以下のいずれかの事前定義済みエージェント タイプを指定します。

- 3Com (RADIUS エージェント)
- アフィリエイト エージェント (SiteMinder Affiliate エージェント)
- AffiliateMinder (AffiliateMinder エージェント)
- Ascend (RADIUS エージェント)
- Bay Networks (RADIUS エージェント)
- CheckPoint (RADIUS エージェント)
- Cisco (RADIUS エージェント)
- EJB エージェント (SiteMinder EJB エージェント)
- Generic RADIUS (RADIUS エージェント)
- Livingston (RADIUS エージェント)
- Network Associates Sniffer (RADIUS エージェント)
- Servlet エージェント (SiteMinder Servlet エージェント)
- Shiva (RADIUS エージェント)
- TeleBit (RADIUS エージェント)
- U.S. Robotics (RADIUS エージェント)
- Web エージェント (SiteMinder Web エージェント)

戻り値

GetAgentType メソッドは以下のいずれかの値を返します。

- コールに成功する場合は、PolicyMgtAgentType オブジェクト
- コールに失敗した場合または指定されたエージェント タイプが存在しない場合は、**undef**

GetAllAdmins メソッド -- すべての管理者のリストを取得

GetAllAdmins メソッドは、ポリシー サーバ上で設定されているすべての管理者のリストを取得します。

構文

GetAllAdmins メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetAllAdmins()
```

パラメータ

GetAllAdmins メソッドは、パラメータを受け入れません。

戻り値

GetAllAdmins メソッドは以下のいずれかの値を返します。

- PolicyMgtAdmin (配列)
- コールに失敗した場合または管理者が存在しない場合は、**undef**

GetAllAffDomains メソッド -- すべてのアフィリエイトドメインのリストを取得

GetAllAffDomains メソッドは、設定されたすべてのアフィリエイト ドメインのリストを取得します。

構文

GetAllAffDomains メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetAllAffDomains()
```

パラメータ

GetAllAffDomains メソッドは、パラメータを受け入れません。

戻り値

GetAllAffDomains メソッドは以下のいずれかの値を返します。

- PolicyMgtAffDomain (配列)
- コールに失敗した場合またはアフィリエイト ドメインが存在しない場合は、**undef**

GetAllAgentConfigs メソッド -- すべてのエージェント設定オブジェクトのリストを取得

GetAllAgentConfigs メソッドは、すべてのエージェント設定オブジェクトのリストを取得します。

構文

GetAllAgentConfigs メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetAllAgentConfigs()
```

パラメータ

GetAllAgentConfigs メソッドは、パラメータを受け入れません。

戻り値

GetAllAgentConfigs メソッドは以下のいずれかの値を返します。

- PolicyMgtAgentConfig (配列)
- コールに失敗した場合またはエージェント設定オブジェクトが存在しない場合は、**undef**

GetAllAgentGroups メソッド -- すべてのエージェント グループ オブジェクトのリストを取得

GetAllAgentGroups メソッドは、すべてのエージェント グループ オブジェクトのリストを取得します。

構文

GetAllAgentGroups メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetAllAgentGroups()
```

パラメータ

GetAllAgentGroups メソッドは、パラメータを受け入れません。

戻り値

GetAllAgentGroups メソッドは以下のいずれかの値を返します。

- PolicyMgtGroup (配列)
- コールに失敗した場合は、**undef**

GetAllAgents メソッド -- すべてのエージェントのリストを取得

GetAllAgents メソッドは、ポリシー サーバ上で設定されているすべてのエージェントのリストを取得します。

構文

GetAllAgents メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetAllAgents()
```

パラメータ

GetAllAgents メソッドは、パラメータを受け入れません。

戻り値

GetAllAgents メソッドは以下のいずれかの値を返します。

- PolicyMgtAgent (配列)
- コールに失敗した場合またはエージェントが存在しない場合は、**undef**

GetAllAuthAzMaps メソッド -- すべての AuthAz マップのリストを取得

GetAllAuthAzMaps メソッドは、すべての認証および許可マップのリストを取得します。

構文

GetAllAuthAzMaps メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetAllAuthAzMaps()
```

パラメータ

GetAllAuthAzMaps メソッドは、パラメータを受け入れません。

戻り値

GetAllAuthAzMaps メソッドは以下のいずれかの値を返します。

- PolicyMgtAuthAzMap (配列)
- コールに失敗した場合または認証および許可マップが存在しない場合は、**undef**

GetAllAuthSchemes メソッド -- 認証方式のリストを取得

GetAllAuthSchemes メソッドは、ポリシー サーバ上で設定されているすべての認証方式のリストを取得します。

構文

GetAllAuthSchemes メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetAllAuthSchemes([showTemplates])
```

パラメータ

GetAllAuthSchemes メソッドは以下のパラメータを受け入れます。

showTemplates (int)

(オプション) 認証方式のリストにテンプレート方式を含めるかどうかを指定します。

- 値 = 0

認証方式のリストにテンプレート方式を含めないことを指定します。

- 値 = 1

認証方式のリストにテンプレート方式を含めることを指定します。

戻り値

GetAllAuthSchemes メソッドは以下のいずれかの値を返します。

- PolicyMgtAuthScheme (配列)
- コールに失敗した場合または認証方式が存在しない場合は、**undef**

GetAllCertMaps メソッド -- 証明書マッピング オブジェクトのリストを取得

GetAllCertMaps メソッドは、すべての証明書マッピング オブジェクトのリストを取得します。

構文

GetAllCertMaps メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetAllCertMaps()
```

パラメータ

GetAllCertMaps メソッドは、パラメータを受け入れません。

戻り値

GetAllCertMaps メソッドは以下のいずれかの値を返します。

- PolicyMgtCertMap (配列)
- コールに失敗した場合または証明書マッピング オブジェクトが存在しない場合は、**undef**

GetAllDomains メソッド -- すべてのドメインのリストを取得

GetAllDomains メソッドは、ポリシー サーバ上で設定されているすべてのドメインのリストを取得します。

構文

GetAllDomains メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetAllDomains()
```

パラメータ

GetAllDomains メソッドは、パラメータを受け入れません。

戻り値

GetAllDomains メソッドは以下のいずれかの値を返します。

- PolicyMgtDomain (配列)
- コールに失敗した場合またはドメインが存在しない場合は、**undef**

GetAllGlobalPolicies メソッド -- グローバル ポリシー オブジェクトのリストを取得

GetAllGlobalPolicies メソッドは、すべてのグローバル ポリシー オブジェクトのリストを取得します。

構文

GetAllGlobalPolicies メソッドの形式は以下のとおりです。

```
Netegrity::GetAllGlobalPolicies()
```

パラメータ

GetAllGlobalPolicies メソッドは、パラメータを受け入れません。

戻り値

GetAllGlobalPolicies メソッドは以下のいずれかの値を返します。

- PolicyMgtPolicy (配列)
- コールに失敗した場合は、**undef**

GetAllGlobalResponses メソッド -- すべてのグローバル レスポンス オブジェクトのリストを取得

GetAllGlobalResponses メソッドは、すべてのグローバル レスポンス オブジェクトのリストを取得します。

構文

GetAllGlobalResponses メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetAllGlobalResponses()
```

パラメータ

GetAllGlobalResponses メソッドは、パラメータを受け入れません。

戻り値

GetAllGlobalResponses メソッドは以下のいずれかの値を返します。

- PolicyMgtResponse (配列)
- コールに失敗した場合は、**undef**

GetAllGlobalRules メソッド -- すべてのグローバル ルール オブジェクトのリストを取得

GetAllGlobalRules メソッドは、すべてのグローバル ルール オブジェクトのリストを取得します。

構文

GetAllGlobalRules メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetAllGlobalRules()
```

パラメータ

GetAllGlobalRules メソッドは、パラメータを受け入れません。

戻り値

GetAllGlobalRules メソッドは以下のいずれかの値を返します。

- PolicyMgtRule (配列)
- コールに失敗した場合は、**undef**

GetAllHostConfigs メソッド -- すべてのホスト設定オブジェクトのリストを取得

GetAllHostConfigs メソッドは、すべてのホスト設定オブジェクトのリストを取得します。

構文

GetAllHostConfigs メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetAllHostConfigs()
```

パラメータ

GetAllHostConfigs メソッドは、パラメータを受け入れません。

戻り値

GetAllHostConfigs メソッドは以下のいずれかの値を返します。

- PolicyMgtHostConfig (配列)
- コールに失敗した場合またはホスト設定オブジェクトが存在しない場合は、**undef**

GetAllODBCQuerySchemes メソッド -- すべての ODBC クエリ方式のリストを取得

GetAllODBCQuerySchemes メソッドは、ポリシー サーバ上のすべての ODBC クエリ方式のリストを取得します。

構文

GetAllODBCQuerySchemes メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetAllODBCQuerySchemes()
```

パラメータ

GetAllODBCQuerySchemes メソッドは、パラメータを受け入れません。

戻り値

GetAllODBCQuerySchemes メソッドは以下のいずれかの値を返します。

- PolicyMgtODBCQueryScheme (配列)
- コールに失敗した場合または ODBC クエリ方式が存在しない場合は、**undef**

GetAllPwdPolicies メソッド -- すべてのパスワード ポリシーのリストを取得

GetAllPwdPolicies メソッドは、設定されたすべてのパスワード ポリシーのリストを取得します。

構文

GetAllPwdPolicies メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetAllPwdPolicies()
```

パラメータ

GetAllPwdPolicies メソッドは、パラメータを受け入れません。

戻り値

GetAllPwdPolicies メソッドは以下のいずれかの値を返します。

- PolicyMgtPwdPolicy (配列)
- コールに失敗した場合またはパスワード ポリシーが存在しない場合は、**undef**

GetAllRegSchemes メソッド -- すべての登録方式のリストを取得

GetAllRegSchemes メソッドは、ポリシー サーバ上で設定されているすべての登録方式のリストを取得します。

構文

GetAllRegSchemes メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetAllRegSchemes()
```

パラメータ

GetAllRegSchemes メソッドは、パラメータを受け入れません。

戻り値

GetAllRegSchemes メソッドは以下のいずれかの値を返します。

- PolicyMgtRegScheme (配列)
- コールに失敗したか、登録方式が存在しない場合は、**undef**

GetAllSAMLAffiliations メソッド -- すべての SAML 2.0 アフィリエーションのリストを取得

GetAllSAMLAffiliations メソッドは、すべての SAML 2.0 アフィリエーションのリストを取得します。

構文

GetAllSAMLAffiliations メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetAllSAMLAffiliations()
```

パラメータ

GetAllSAMLAffiliations メソッドは、パラメータを受け入れません。

戻り値

GetAllSAMLAffiliations メソッドは以下のいずれかの値を返します。

- PolicyMgtSAMLAffiliation (配列)
- コールに失敗した場合は、**undef**

GetAllSAMLSchemeAttributes メソッド -- すべてのリクエスト属性のリストを取得

GetAllSAMLSchemeAttributes メソッドは、定義されたすべての SAML 2.0 リクエスト属性のリストを取得します。

構文

GetAllSAMLSchemeAttributes メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetAllSAMLSchemeAttributes(scheme)
```

パラメータ

GetAllSAMLSchemeAttributes メソッドは以下のパラメータを受け入れます。

scheme (PolicyMgtAuthScheme オブジェクト)

SAML 2.0 認証方式オブジェクトを指定します。

戻り値

GetAllSAMLSchemeAttributes メソッドは以下のいずれかの値を返します。

- PolicyMgtSAMLRequesterAttr (配列)
- コールに失敗した場合は、**undef**

GetAllTrustedHosts メソッド -- すべてのトラステッド ホスト オブジェクトのリストを取得

GetAllTrustedHosts メソッドは、すべてのトラステッド ホスト オブジェクトのリストを取得します。

構文

GetAllTrustedHosts メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetAllTrustedHosts()
```

パラメータ

GetAllTrustedHosts メソッドは、パラメータを受け入れません。

戻り値

GetAllTrustedHosts メソッドは以下のいずれかの値を返します。

- PolicyMgtTrustedHost (配列)
- コールに失敗した場合またはトラステッド ホスト オブジェクトが存在しない場合は、**undef**

GetAllUserDirs メソッド -- すべてのユーザ ディレクトリのリストを取得

GetAllUserDirs メソッドは、ポリシー サーバと関連付けられたすべてのユーザ ディレクトリのリストを取得します。

構文

GetAllUserDirs メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetAllUserDirs()
```

パラメータ

GetAllUserDirs メソッドは、パラメータを受け入れません。

戻り値

GetAllUserDirs メソッドは以下のいずれかの値を返します。

- PolicyMgtUserDir (配列)
- コールに失敗した場合またはユーザ ディレクトリが存在しない場合は、**undef**

GetAllVariableTypes メソッド -- すべての変数タイプ オブジェクトのリストを取得

GetAllVariableTypes メソッドは、ポリシー サーバ上で設定されているすべての変数タイプ オブジェクトのリストを取得します。

構文

GetAllVariableTypes メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetAllVariableTypes()
```

パラメータ

GetAllVariableTypes メソッドは、パラメータを受け入れません。

戻り値

GetAllVariableTypes メソッドは以下のいずれかの値を返します。

- PolicyMgtVariableType (配列)
- コールに失敗した場合または変数タイプ オブジェクトが存在しない場合は、**undef**

GetAuthScheme メソッド -- 認証方式オブジェクトを取得

GetAuthScheme メソッドは、指定された認証方式オブジェクトを取得します。既存の認証方式は名前によって指定されます。新しい認証方式を作成するには、このメソッドを使用して、新しい方式のベースにする認証方式オブジェクトまたはテンプレートのタイプを取得します。次に、結果として得られたオブジェクトを、[PolicyMgtSession->CreateAuthScheme](#) (P. 452) メソッドに *schemeTemplate* パラメータで渡します。SAML 2.0 認証方式の作成については、メソッド [PolicyMgtSession->CreateSAMLAuthScheme](#) (P. 477) を参照してください。

構文

GetAuthScheme メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetAuthScheme(schemeName)
```

パラメータ

GetAuthScheme メソッドは以下のパラメータを受け入れます。

schemeName (文字列)

以下のいずれかを指定します。

- 既存の認証方式の名前を指定します。
- 作成する認証方式のタイプは以下のとおりです。
 - 匿名テンプレート
 - SSL を介した基本テンプレート
 - 基本テンプレート
 - カスタム テンプレート
 - HTML フォーム テンプレート
 - インパーソネーションテンプレート
 - RADIUS CHAP/PAP テンプレート
 - RADIUS サーバ テンプレート
 - SafeWord HTML フォーム テンプレート
 - SafeWord テンプレート
 - SAML Artifact テンプレート

- SAML POST テンプレート
- SAML 2.0 テンプレート
- SecurID HTML フォーム テンプレート
- SecurID テンプレート
- TeleID テンプレート
- Windows 認証テンプレート
- X509 クライアント証明書および基本テンプレート
- X509 クライアント証明書およびフォーム テンプレート
- X509 クライアント証明書または基本テンプレート
- X509 クライアント証明書またはフォーム テンプレート
- X509 クライアント証明書テンプレート

戻り値

GetAuthScheme メソッドは以下のいずれかの値を返します。

- PolicyMgtAuthScheme (オブジェクト)
- コールに失敗した場合または指定された認証方式が存在しない場合は、**undef**

GetCertMap メソッド -- 証明書マッピング オブジェクトを取得

GetCertMap メソッドは、証明書発行者の DN によって指定された証明書マッピング オブジェクトを取得します。

構文

GetCertMap メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetCertMap(issuerDN)
```

パラメータ

GetCertMap メソッドは以下のパラメータを受け入れます。

issuerDN (文字列)

証明書発行者の DN を指定します。

戻り値

GetCertMap メソッドは以下のいずれかの値を返します。

- PolicyMgtCertMap (オブジェクト)
- コールに失敗した場合または指定された発行者の DN が存在しない場合は、**undef**

GetDomain メソッド -- ドメイン オブジェクトを取得

GetDomain メソッドは、指定されたポリシー ドメイン オブジェクトを取得します。

構文

GetDomain メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetDomain(domName)
```

パラメータ

GetDomain メソッドは以下のパラメータを受け入れます。

domName (文字列)

取得するドメインの名前を指定します。

戻り値

GetDomain メソッドは以下のいずれかの値を返します。

- PolicyMgtDomain (オブジェクト)
- コールに失敗した場合または指定されたドメインが存在しない場合は、**undef**

GetGlobalPolicy メソッド -- グローバル ポリシー オブジェクトを取得

GetGlobalPolicy メソッドは、指定されたグローバル ポリシー オブジェクトを取得します。

構文

GetGlobalPolicy メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetGlobalPolicy(policyName)
```

パラメータ

GetGlobalPolicy メソッドは以下のパラメータを受け入れます。

policyName (文字列)

取得するグローバル ポリシーの名前を指定します。

戻り値

GetGlobalPolicy メソッドは以下のいずれかの値を返します。

- PolicyMgtPolicy (オブジェクト)
- コールに失敗した場合または指定されたグローバル ポリシーが存在しない場合は、**undef**

GetGlobalResponse メソッド -- グローバル レスポンス オブジェクトを取得

GetGlobalResponse メソッドは、指定されたグローバル レスポンス オブジェクトを取得します。

構文

GetGlobalResponse メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetGlobalResponse(responseName)
```

パラメータ

GetGlobalResponse メソッドは以下のパラメータを受け入れます。

responseName (文字列)

取得するグローバル レスポンスの名前を指定します。

戻り値

GetGlobalResponse メソッドは以下のいずれかの値を返します。

- PolicyMgtResponse (オブジェクト)
- コールに失敗した場合または指定されたグローバル レスポンスが存在しない場合は、**undef**

GetGlobalRule メソッド -- グローバル ルール オブジェクトを取得

GetGlobalRule メソッドは、指定されたグローバル ルール オブジェクトを取得します。

構文

GetGlobalRule メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetGlobalRule(ruleName)
```

パラメータ

GetGlobalRule メソッドは以下のパラメータを受け入れます。

ruleName (文字列)

取得するグローバル ルールの名前を指定します。

戻り値

GetGlobalRule メソッドは以下のいずれかの値を返します。

- PolicyMgtRule (オブジェクト)
- コールに失敗した場合または指定されたグローバル ルールが存在しない場合は、**undef**

GetHostConfig メソッド -- ホスト設定オブジェクトを取得

GetHostConfig メソッドは、指定されたホスト設定オブジェクトを取得します。

構文

GetHostConfig メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetHostConfig(hcName)
```

パラメータ

GetHostConfig メソッドは以下のパラメータを受け入れます。

hcName (文字列)

取得するホスト設定オブジェクトの名前を指定します。

戻り値

GetHostConfig メソッドは以下のいずれかの値を返します。

- PolicyMgtHostConfig (オブジェクト)
- コールに失敗した場合または指定されたホスト設定オブジェクトが存在しない場合は、**undef**

GetODBCQueryScheme メソッド -- ODBC クエリ方式オブジェクトを取得

GetODBCQueryScheme メソッドは、指定された ODBC クエリ方式オブジェクトを取得します。

構文

GetODBCQueryScheme メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetODBCQueryScheme(schemeName)
```

パラメータ

GetODBCQueryScheme メソッドは以下のパラメータを受け入れます。

schemeName (文字列)

取得する ODBC クエリ方式を指定します。

戻り値

GetODBCQueryScheme メソッドは以下のいずれかの値を返します。

- PolicyMgtODBCQueryScheme (オブジェクト)
- コールに失敗した場合または指定された ODBC クエリ方式が存在しない場合は、**undef**

GetPwdPolicy メソッド -- パスワード ポリシー オブジェクトを取得

GetPwdPolicy メソッドは、指定されたパスワード ポリシー オブジェクトを取得します。

構文

GetPwdPolicy メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetPwdPolicy(pwdPolicyName)
```

パラメータ

GetPwdPolicy メソッドは以下のパラメータを受け入れます。

pwdPolicyName (文字列)

取得するパスワード ポリシーの名前を指定します。

戻り値

GetPwdPolicy メソッドは以下のいずれかの値を返します。

- PolicyMgtPwdPolicy (オブジェクト)
- コールに失敗した場合または指定されたパスワード ポリシーが存在しない場合は、**undef**

GetRegScheme メソッド -- 登録方式オブジェクトを取得

GetRegScheme メソッドは、指定された登録方式オブジェクトを取得します。

構文

GetRegScheme メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetRegScheme(schemeName)
```

パラメータ

GetRegScheme メソッドは以下のパラメータを受け入れます。

schemeName (文字列)

取得する登録方式の名前を指定します。

戻り値

GetRegScheme メソッドは以下のいずれかの値を返します。

- PolicyMgtRegScheme (オブジェクト)
- コールに失敗した場合または指定された登録方式が存在しない場合は、**undef**

GetSAMLAffiliation メソッド -- SAML 2.0 アフィリエーション オブジェクトを取得

GetSAMLAffiliation メソッドは、指定された SAML 2.0 アフィリエーション オブジェクトを取得します。

構文

GetSAMLAffiliation メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetSAMLAffiliation(affilName)
```

パラメータ

GetSAMLAffiliation メソッドは以下のパラメータを受け入れます。

affilName (文字列)

取得する SAML アフィリエーションの名前または **OID** を指定します。

注: **OID** を指定するときは、その前に「@」文字を付けることができます。

戻り値

GetSAMLAffiliation メソッドは以下のいずれかの値を返します。

- PolicyMgtSAMLAffiliation (オブジェクト)
- コールに失敗した場合または指定された SAML アフィリエーションが存在しない場合は、**undef**

GetSAMLAffiliationById メソッド -- ID により SAML 2.0 アフィリエーション オブジェクトを取得

GetSAMLAffiliationById メソッドは、メソッドに渡されるアフィリエーション ID によって指定された SAML 2.0 アフィリエーション オブジェクトを取得します。

構文

GetSAMLAffiliationById メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetSAMLAffiliationById(affilID)
```

パラメータ

GetSAMLAffiliationById メソッドは以下のパラメータを受け入れます。

affilID (文字列)

取得する SAML アフィリエーションのアフィリエーション ID を指定します。

戻り値

GetSAMLAffiliationById メソッドは以下のいずれかの値を返します。

- PolicyMgtSAMLAffiliation (オブジェクト)
- コールに失敗した場合または指定された SAML アフィリエーションが存在しない場合は、**undef**。

GetSharedSecretPolicy メソッド -- 共有秘密キー ポリシー オブジェクトを取得

GetSharedSecretPolicy メソッドは、指定された共有秘密キー ロールオーバー ポリシー オブジェクトを取得します。それぞれのポリシー ストアドメインには、1 つの共有秘密キー ロールオーバー ポリシーのみが存在できるので、ポリシーの名前をこのメソッドに渡す必要はありません。

構文

GetSharedSecretPolicy メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetSharedSecretPolicy()
```

パラメータ

GetSharedSecretPolicy メソッドは、パラメータを受け入れません。

戻り値

GetSharedSecretPolicy メソッドは、以下の値を返します。

- PolicyMgtSharedSecretPolicy (オブジェクト)

GetTrustedHost メソッド -- トラストド ホスト オブジェクトを取得

GetTrustedHost メソッドは、指定されたトラストド ホスト オブジェクトを取得します。

構文

GetTrustedHost メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetTrustedHost(thName)
```

パラメータ

GetTrustedHost メソッドは以下のパラメータを受け入れます。

thName (文字列)

取得するトラステッドホストの名前を指定します。

戻り値

GetTrustedHost メソッドは以下のいずれかの値を返します。

- PolicyMgtTrustedHost (オブジェクト)
- コールに失敗した場合または指定されたトラステッドホストが存在しない場合は、**undef**

GetUserDir メソッド -- ユーザ ディレクトリ オブジェクトを取得

GetUserDir メソッドは、指定されたユーザ ディレクトリ オブジェクトを取得します。

構文

GetUserDir メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetUserDir(dirName)
```

パラメータ

GetUserDir メソッドは以下のパラメータを受け入れます。

dirName (文字列)

取得するユーザ ディレクトリの名前を指定します。

戻り値

GetUserDir メソッドは以下のいずれかの値を返します。

- PolicyMgtUserDir (オブジェクト)
- コールに失敗した場合または指定されたユーザ ディレクトリが存在しない場合は、**undef**

GetVariableType メソッド -- 変数タイプ オブジェクトを取得

GetVariableType メソッドは、指定された変数タイプ オブジェクトを取得します。指定されたタイプの新しい変数オブジェクトを作成するには、結果として得られる変数タイプ オブジェクトを **CreateVariable** メソッドに *varType* パラメータで渡します。

構文

GetVariableType メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->GetVariableType(varTypeName)
```

パラメータ

GetVariableType メソッドは以下のパラメータを受け入れます。

varTypeName (文字列)

以下のいずれかの変数タイプ名を指定します。

SiteMinder 変数タイプ

ポスト

UserContext

RequestContext

スタティック

WebService

TransactionMinder 変数タイプ

XMLBody

XMLAgent

XMLEnvelopeHeader

トランスポート

SAMLEAssertion

注: 変数タイプ名は大文字と小文字を区別し、スペースを含めることはできません。

戻り値

GetVariableType メソッドは以下のいずれかの値を返します。

- PolicyMgtVariableType (オブジェクト)
- コールに失敗した場合は、**undef**

注釈

コマンドライン インターフェースでは、TransactionMinder 変数を作成できません。TransactionMinder と Option Pack がインストールされている場合は、管理 UI で TransactionMinder 変数を作成できます。

RemoveAttributeFromSAMLScheme メソッド -- SAML 方式から属性を削除

RemoveAttributeFromSAMLScheme メソッドは、SAML 2.0 認証方式から属性を削除します。

構文

RemoveAttributeFromSAMLScheme メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->RemoveAttributeFromSAMLScheme(scheme,  
pSAMLRequesterAttr)
```

パラメータ

RemoveAttributeFromSAMLScheme メソッドは、以下のパラメータを受け入れます。

scheme (PolicyMgtAuthScheme オブジェクト)

属性を削除する SAML 2.0 認証方式を指定します。

pSAMLRequesterAttribute (文字列)

削除する属性を指定します。

戻り値

`RemoveAttributeFromSAMLScheme` メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことを指定します。
- 値 = -1
メソッドが失敗したことを指定します。

SAMLAuthSchemeProperties メソッド -- SAML メタデータ プロパティを設定または取得

`SAMLAuthSchemeProperties` メソッドは、既存の SAML 2.0 認証方式に存在する SAML 2.0 メタデータ プロパティを設定または取得します。SAML 2.0 メタデータ プロパティの完全なリストについては、メソッド [PolicyMgtSession->CreateSAMLAuthScheme](#) (P. 477) を参照してください。

構文

`SAMLAuthSchemeProperties` メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->SAMLAuthSchemeProperties(scheme, propsHash_ref)
```

パラメータ

`SAMLAuthSchemeProperties` メソッドは、以下のパラメータを受け入れます。

scheme (PolicyMgtAuthScheme オブジェクト)

そのメタデータ プロパティを設定または取得する認証方式を指定します。

propsHash_ref (ハッシュ)

設定または取得するメタデータ プロパティのハッシュテーブルの参照を指定します。

戻り値

SAMLAuthSchemeProperties メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことを指定します。
- 値 = -1
メソッドが失敗したことを指定します。

注釈

ハッシュテーブルが空のとき、SAMLAuthSchemeProperties メソッドはすべてのメタデータ プロパティを取得します。以下のように空のハッシュテーブルを定義できます。

```
%myhash=();
```

次に、以下のように空のハッシュテーブルを参照できます。

```
%myhash
```

最後に、ハッシュテーブルの参照を SAMLAuthSchemeProperties メソッドに *propsHash_ref* パラメータを通じて渡します。

WSFEDAuthSchemeProperties メソッド -- WS-Federation プロパティを設定または取得

WSFEDAuthSchemeProperties メソッドは、既存の WS-Federation 認証方式内で WS-Federation メタデータ プロパティを設定または取得します。WS-Federation メタデータ プロパティの全リストについては、メソッド [PolicyMgtSession->CreateWSFEDAuthScheme](#) (P. 489) を参照してください。

構文

WSFEDAuthSchemeProperties メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSession->WSFEDAuthSchemeProperties(scheme, propsHash_ref)
```

パラメータ

`WSFEDAuthSchemeProperties` メソッドは、以下のパラメータを受け入れます。

scheme (PolicyMgtAuthScheme オブジェクト)

その WS-Federation メタデータ プロパティが設定または取得される認証方式を指定します。

propsHash_ref (ハッシュ)

設定または取得するメタデータ プロパティのハッシュテーブルの参照を指定します。

戻り値

`WSFEDAuthSchemeProperties` メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことを指定します。
- 値 = -1
メソッドが失敗したことを指定します。

注釈

ハッシュテーブルが空のとき、`WSFEDAuthSchemeProperties` メソッドはすべてのメタデータ プロパティを取得します。以下のように空のハッシュテーブルを定義できます。

```
%myhash=();
```

次に、以下のように空のハッシュテーブルを参照できます。

```
%myhash
```

最後に、ハッシュテーブルの参照を `WSFEDAuthSchemeProperties` メソッドに *propsHash_ref* パラメータを通じて渡します。

CLI 共有秘密キーのロールオーバー メソッド

Enabled メソッド -- ポリシーのロールオーバー有効化フラグを設定または取得

Enabled メソッドは、共有秘密キー ロールオーバー ポリシーを有効にするかどうかを指定するフラグを設定または取得します。

構文

Enabled メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSharedSecretPolicy->Enabled([enableFlag])
```

パラメータ

Enabled メソッドは以下のパラメータを受け入れます。

enableFlag (int)

(オプション) 有効化フラグの新しい値を指定します。

- 値 = 1

共有秘密キー ロールオーバー ポリシーを有効にすることを指定します。

- 値 = 0

共有秘密キー ロールオーバー ポリシーを無効にすることを指定します。

戻り値

Enabled メソッドは、有効化フラグの新規または既存の値を返します。

- 値 = 1

共有秘密キーのロールオーバー ポリシーが有効であることを指定します。

- 値 = 0

共有秘密キーのロールオーバー ポリシーが無効であることを指定します。

注釈

共有秘密キーのロールオーバー ポリシーが有効な場合は、その共有秘密キーをロールオーバー ポリシーの共有秘密キーと同期する必要があるトラステッドホストに対しても、ロールオーバーを有効にする必要があります。トラステッドホスト オブジェクト用のロールオーバーを有効にするには、メソッド `PolicyMgtTrustedHost->RolloverEnabled` をコールします。

RolloverFrequency メソッド -- ポリシーのロールオーバー頻度を設定または取得

`RolloverFrequency` メソッドは、ポリシーのロールオーバー頻度を設定または取得します。この値は、メソッド `PolicyMgtSharedSecretPolicy->RolloverPeriod` によって指定された期間に共有秘密キーが自動的に更新される頻度を決定します。

構文

`RolloverFrequency` メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSharedSecretPolicy->RolloverFrequency([rollFreq])
```

パラメータ

`RolloverFrequency` メソッドは以下のパラメータを受け入れます。

rollFreq (int)

(オプション) ロールオーバー頻度の新しい値を指定します。

範囲 : *rollFreq* >= 1

戻り値

`RolloverFrequency` メソッドは、以下の値を返します。

- `rollover_frequency` (int)

ロールオーバー頻度の新規または既存の値を指定します。

RolloverPeriod メソッド -- ポリシーのロールオーバー期間を設定または取得

RolloverPeriod メソッドは、ロールオーバー ポリシーのロールオーバー期間を設定または取得します。ロールオーバー期間には次の 4 つの値が有効です。1 時間ごと、日単位、週単位、または月単位。共有秘密キーが自動的に変更される頻度を指定するには、ロールオーバー期間をロールオーバー頻度とともに使用します。たとえば、ロールオーバー頻度が 2 で、ロールオーバー期間が毎週の場合、共有秘密キーは 2 週間ごとに自動的に変更されます。ロールオーバー頻度を設定するには、PolicyMgtSharedSecretPolicy->RolloverFrequency メソッドをコールします。

構文

RolloverPeriod メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSharedSecretPolicy->RolloverPeriod([rollPeriod])
```

パラメータ

RolloverPeriod メソッドは以下のパラメータを受け入れます。

rollPeriod (int)

(オプション) ロールオーバー期間の新しい値を指定します。

- 値 = 0
ロールオーバー期間が 1 時間ごとであることを指定します。
- 値 = 1
ロールオーバー期間が日単位であることを指定します。
- 値 = 2
ロールオーバー期間が週単位であることを指定します。
- 値 = -3
ロールオーバー期間が月単位であることを指定します。

戻り値

RolloverPeriod メソッドは以下のいずれかの値を返します。

- rollover_period (int)
ロールオーバー期間の新規または既存の値を指定します。
範囲 : 0-3
- 値 = -1
戻り値が 0-3 の範囲にないことを指定します。

Save メソッド -- 共有秘密キー ポリシー オブジェクトを保存

Save メソッドは、共有秘密キー ポリシー オブジェクトをポリシー ストアに保存します。共有秘密キー ポリシー オブジェクトに変更をすべて加えた後、このメソッドを 1 回コールします。変更を有効にするには、必ずこのメソッドをコールします。

構文

Save メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtSharedSecretPolicy->Save()
```

パラメータ

Save メソッドは、パラメータを受け入れません。

戻り値

Save メソッドは以下のいずれかの値を返します。

- 値 = 0
コールが成功したことを指定します。
- 値 = -1
コールが失敗したことを指定します。

CLI トラストッド ホスト メソッド

GetDescription メソッド -- トラストッド ホストの説明を取得

GetDescription メソッドは、トラストッド ホストの説明を取得します。

構文

GetDescription メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtTrustedHost->GetDescription()
```

パラメータ

GetDescription メソッドは、パラメータを受け入れません。

戻り値

GetDescription メソッドは、以下の値を返します。

- trusted_host_description

GetIPAddress メソッド -- トラストッド ホストの IP アドレスを取得

GetIPAddress メソッドは、トラストッド ホストの IP アドレスを取得します。

構文

GetIPAddress メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtTrustedHost->GetIPAddress()
```

パラメータ

GetIPAddress メソッドは、パラメータを受け入れません。

戻り値

GetIPAddress メソッドは、以下の値を返します。

- trusted_host_ip_address

GetName メソッド -- トラストド ホストの名前を取得

GetName メソッドは、トラストド ホストの名前を取得します。

構文

GetName メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtTrustedHost->GetName()
```

パラメータ

GetName メソッドは、パラメータを受け入れません。

戻り値

GetName メソッドは、以下の値を返します。

- trusted_host_name

GetSecret メソッド -- トラストド ホストの共有秘密キーを取得

GetSecret メソッドは、クリア テキスト内のトラストド ホストの共有秘密キーを取得します。

構文

GetSecret メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtTrustedHost->GetSecret()
```

パラメータ

GetSecret メソッドは、パラメータを受け入れません。

戻り値

GetSecret メソッドは以下のいずれかの値を返します。

- trusted_host_shared_secret
- コールに失敗した場合は、**undef**

RolloverEnabled メソッド -- 共有秘密キーのロールオーバー フラグを設定するまたは取得

RolloverEnabled メソッドは、このトラストド ホストに対して共有秘密キー ロールオーバーを有効にするかどうかを指定する共有秘密キー ロールオーバー フラグを設定または取得します。

構文

RolloverEnabled メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtTrustedHost->RolloverEnabled([rolloverEnabled])
```

パラメータ

RolloverEnabled メソッドは以下のパラメータを受け入れます。

rolloverEnabled (int)

(オプション) 共有秘密キー ロールオーバー フラグの新しい値を指定します。

- 値 = 1

このトラストド ホストに対して共有秘密キー ロールオーバーが有効であることを指定します。

- 値 = 0

このトラストド ホストに対して共有秘密キー ロールオーバーが有効でないことを指定します。

戻り値

RolloverEnabled メソッドは、以下のように共有秘密キー ロールオーバー フラグの新規または既存の値を返します。

- 値 = 1

このトラストド ホストに対して共有秘密キー ロールオーバーが有効であることを指定します。

- 値 = 0

このトラステッド ホストに対して共有秘密キー ロールオーバーが有効でないことを指定します。

- undef

コールが失敗したことを指定します。

注釈

このトラステッド ホストに対して共有秘密キー ロールオーバーが有効である場合、トラステッド ホストが登録されているポリシー ストア ドメインの **PolicyMgtSharedSecretPolicy** オブジェクト内でも共有秘密キー ロールオーバーが有効である必要があります。このオブジェクトで共有秘密キー ロールオーバーが有効でない場合は、**PolicyMgtSharedSecretPolicy->Enabled** メソッドをコールして有効にします。

SetSecret メソッド -- トラストッド ホストの共有秘密キーを設定

SetSecret メソッドは、トラステッド ホストの共有秘密キーを設定します。

構文

SetSecret メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtTrustedHost->SetSecret([sharedSecret])
```

パラメータ

SetSecret メソッドは以下のパラメータを受け入れます。

sharedSecret (文字列)

(オプション) トラストッド ホスト用に設定する共有秘密キーを指定します。

注: 共有秘密キーが指定されていない場合、**SiteMinder** はトラステッド ホスト用のランダムな 128 バイト ASCII 共有秘密キーを生成します。

戻り値

SetSecret メソッドは以下のいずれかの値を返します。

- shared_secret (文字列)
(オプション) トラストッド ホスト用の新しい共有秘密キーを指定します。
- "" (空の文字列)
コールが失敗したことを指定します。

注釈

このメソッドを使用して共有秘密キーを設定するときは、ホスト設定ファイルで共有秘密キーを定義するために SiteMinder ツール smregghost を実行する必要もあります (ホスト設定ファイルはデフォルトでは SmHost.conf と命名されます)。-sh オプションを指定して smregghost を実行します。詳細については、メソッド [PolicyMgtSession->AddTrustedHost](#) (P. 444) を参照してください。

注: 共有秘密キー ロールオーバーを自動的に発生するようにスケジュールできます。この機能の詳細については、「ポリシー サーバ設定ガイド」を参照してください。

CLI ユーザ メソッド

DisableByAdmin メソッド -- Disabled-by-Administrator フラグを設定または取得

DisableByAdmin メソッドは、ユーザ アカウントが管理者によって無効化されるかどうかを指定する disabled-by-administrator フラグを設定または取得します。

構文

DisableByAdmin メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUser->DisableByAdmin([disableFlag])
```

パラメータ

DisableByAdmin メソッドは以下のパラメータを受け入れます。

disableFlag (int)

(オプション) disabled-by-administrator フラグの新しい値を指定します。

- 値 = 1

ユーザ アカウントが管理者によって無効化されることを指定します。

- 値 = 0

ユーザ アカウントが管理者によって無効化されないことを指定します。

注: ユーザ アカウントは他の理由で無効になる場合があります。詳細については、「注釈」を参照してください。

戻り値

DisableByAdmin メソッドは、以下のように disabled-by-administrator フラグの新規または既存の値を返します。

- 値 = 1

ユーザ アカウントが管理者によって無効化されることを指定します。

- 値 = 0

ユーザ アカウントが管理者によって無効化されないことを指定します。

注: ユーザ アカウントは他の理由で無効になる場合があります。詳細については、「注釈」を参照してください。

- 値 = -1

コールが失敗したことを指定します。

注釈

ユーザ アカウントは以下の 1 つ以上の理由で無効になる場合があります。

- 管理者がユーザ アカウントを無効にした。
- アカウント非アクティブ状態が許可されている時間を超えました。

詳細については、メソッド [PolicyMgtUser->DisableInactive](#) (P. 549) を参照してください。

- ログイン失敗の数が許可されている最大数を超えた。

詳細については、メソッド [PolicyMgtUser->DisableMaxLoginFail](#) (P. 551) を参照してください。

- パスワードが期限切れになった。

詳細については、メソッド [PolicyMgtUser->DisablePwdExpired](#) (P. 552) を参照してください。

DisableInactive メソッド -- Disabled-by-Inactivity フラグを設定または取得

DisableInactive メソッドは、アカウントの非アクティブ状態が許可された時間を超えたことを理由にユーザ アカウントが無効化されるかどうかを指定する、**disabled-by-inactivity** フラグを設定または取得します。

構文

DisableInactive メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUser->DisableInactive([disableFlag])
```

パラメータ

DisableInactive メソッドは以下のパラメータを受け入れます。

disableFlag (int)

(オプション) **disabled-by-inactivity** フラグの新しい値を指定します。

- 値 = 1

ユーザ アカウントが非アクティブ状態のために無効化されることを指定します。

- 値 = 0

ユーザ アカウントが非アクティブ状態のために無効化されないことを指定します。

注: ユーザ アカウントは他の理由で無効になる場合があります。詳細については、「注釈」を参照してください。

戻り値

`DisableInactive` メソッドは、以下のように `disabled-by-inactivity` フラグの新規または既存の値を返します。

- 値 = 1

ユーザ アカウントが非アクティブ状態のために無効化されることを指定します。

- 値 = 0

ユーザ アカウントが非アクティブ状態のために無効化されないことを指定します。

注: ユーザ アカウントは他の理由で無効になる場合があります。詳細については、「注釈」を参照してください。

- 値 = -1

コールが失敗したことを指定します。

注釈

ユーザ アカウントは以下の 1 つ以上の理由で無効になる場合があります。

- 管理者がユーザ アカウントを無効にした。

詳細については、メソッド [PolicyMgtUser->DisableByAdmin](#) (P. 547) を参照してください。

- アカウント非アクティブ状態が許可されている時間を超えました。

- ログイン失敗の数が許可されている最大数を超えた。

詳細については、メソッド [PolicyMgtUser->DisableMaxLoginFail](#) (P. 551) を参照してください。

- パスワードが期限切れになった。

詳細については、メソッド [PolicyMgtUser->DisablePwdExpired](#) (P. 552) を参照してください。

DisableMaxLoginFail メソッド -- Disabled-by-Max-Login-Failure フラグを設定または取得

DisableMaxLoginFail メソッドは、`disabled-by-max-login-failure` フラグを設定または取得します。このフラグは、ログイン失敗の回数が許可されている最大数を超えたことを理由にユーザ アカウントが無効化されるかどうかを指定します。

構文

DisableMaxLoginFail メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUser->DisableMaxLoginFail([disableFlag])
```

パラメータ

DisableMaxLoginFail メソッドは以下のパラメータを受け入れます。

disableFlag (int)

(オプション) `disabled-by-max-login-failure` フラグの新しい値を指定します。

- 値 = 1

ログイン失敗の回数が許可されている最大数を超えたことを理由にユーザ アカウントが無効化されることを指定します。

- 値 = 0

ログイン失敗の回数が許可されている最大数を超えてもユーザ アカウントが無効化されないことを指定します。

注: ユーザ アカウントは他の理由で無効になる場合があります。詳細については、「注釈」を参照してください。

戻り値

DisableMaxLoginFail メソッドは、以下のように `disabled-by-max-login-failure` フラグの新規または既存の値を返します。

- 値 = 1

ログイン失敗の回数が許可されている最大数を超えたことを理由にユーザ アカウントが無効化されることを指定します。

- 値 = 0

ログイン失敗の回数が許可されている最大数を超えてもユーザ アカウントが無効化されないことを指定します。

注: ユーザ アカウントは他の理由で無効になる場合があります。詳細については、「注釈」を参照してください。

- 値 = -1

コールが失敗したことを指定します。

注釈

ユーザ アカウントは以下の 1 つ以上の理由で無効になる場合があります。

- 管理者がユーザ アカウントを無効にした。

詳細については、メソッド [PolicyMgtUser->DisableByAdmin](#) (P. 547) を参照してください。

- アカウント非アクティブ状態が許可されている時間を超えました。

詳細については、メソッド [PolicyMgtUser->DisableInactive](#) (P. 549) を参照してください。

- ログイン失敗の数が許可されている最大数を超えた。

- パスワードが期限切れになった。

詳細については、メソッド [PolicyMgtUser->DisablePwdExpired](#) (P. 552) を参照してください。

DisablePwdExpired メソッド -- Disabled-by-Password-Expired フラグを設定または取得

DisablePwdExpired メソッドは、ユーザ アカウントがパスワードの期限切れを理由に無効化されるかどうかを指定する disabled-by-password-expired フラグを設定または取得します。

構文

DisablePwdExpired メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUser->DisablePwdExpired([disableFlag])
```

パラメータ

`DisablePwdExpired` メソッドは以下のパラメータを受け入れます。

`disableFlag` (int)

(オプション) `disabled-by-password-expired` フラグの新しい値を指定します。

- 値 = 1

ユーザ アカウントがパスワードの期限切れを理由に無効化されることを指定します。

- 値 = 0

ユーザ アカウントがパスワードの期限切れを理由に無効化されないことを指定します。

- 注: ユーザ アカウントは他の理由で無効になる場合があります。詳細については、「注釈」を参照してください。

戻り値

`DisablePwdExpired` メソッドは、以下のように `disabled-by-password-expired` フラグの新規または既存の値を返します。

- 値 = 1

ユーザ アカウントがパスワードの期限切れを理由に無効化されることを指定します。

- 値 = 0

ユーザ アカウントがパスワードの期限切れを理由に無効化されないことを指定します。

注: ユーザ アカウントは他の理由で無効になる場合があります。詳細については、「注釈」を参照してください。

- 値 = -1

コールが失敗したことを指定します。

注釈

ユーザ アカウントは以下の 1 つ以上の理由で無効になる場合があります。

- 管理者がユーザ アカウントを無効にした。

詳細については、メソッド [PolicyMgtUser->DisableByAdmin](#) (P. 547) を参照してください。

- アカウント非アクティブ状態が許可されている時間を超えました。

詳細については、メソッド [PolicyMgtUser->DisableInactive](#) (P. 549) を参照してください。

- ログイン失敗の数が許可されている最大数を越えた。

詳細については、メソッド [PolicyMgtUser->DisableMaxLoginFail](#) (P. 551) を参照してください。

- パスワードが期限切れになった。

ForcePwdChange メソッド -- Force-Password-Change フラグを設定または取得

ForcePwdChange メソッドは、次回のユーザ ログイン時にパスワード変更を強制するかどうかを指定する `force-password-change` フラグを設定または取得します。

構文

ForcePwdChange メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUser->ForcePwdChange([forceFlag])
```

パラメータ

ForcePwdChange メソッドは以下のパラメータを受け入れます。

forceFlag (int)

(オプション) 次回のユーザ ログイン時にパスワード変更を強制するかどうかを指定します。

- 値 = 1

次回のユーザ ログイン時にパスワード変更を強制することを指定します。

- 値 = 0

次回のユーザ ログイン時にパスワード変更を強制しないことを指定します。

戻り値

ForcePwdChange メソッドは、force-password-change フラグの新規または既存の値を返します。

- 値 = 1

次回のユーザ ログイン時にパスワード変更を強制することを指定します。

- 値 = 0

次回のユーザ ログイン時にパスワード変更を強制しないことを指定します。

- 値 = -1

コールが失敗したことを指定します。

GetClass メソッド -- ユーザ クラスを取得

GetClass メソッドは、ユーザ クラスを取得します。

構文

GetClass メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUser->GetClass()
```

パラメータ

GetClass メソッドは、パラメータを受け入れません。

戻り値

GetClass メソッドは以下のいずれかの値を返します。

- user_class
例: "organization"
- コールに失敗した場合は、**undef**

SetPassword メソッド -- 新しいパスワードを設定

SetPassword メソッドは、ユーザの新しいパスワードを設定します。

構文

SetPassword メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUser->SetPassword(newPwd[, oldPwd])
```

パラメータ

SetPassword メソッドは、以下のパラメータを受け入れます。

newPwd (文字列)

新しいパスワードを指定します。

oldPwd (文字列)

(オプション) 変更する古いパスワードを指定します。

注: 指定する場合、この値はユーザ ディレクトリ内の既存のパスワードに一致する必要があります。

戻り値

SetPassword メソッドは以下のいずれかの値を返します。

- 値 = 0
パスワードの変更が成功したことを指定します。
- 値 = -1
パスワードの変更が失敗したことを指定します。

UserPasswordState メソッド -- パスワード状態オブジェクトを設定または取得

UserPasswordState メソッドは、現在のユーザのパスワード状態オブジェクトを設定または取得します。新しいパスワード状態オブジェクトを設定すると、オブジェクトの属性が更新され、実行した変更が反映されます。**empty-history** フラグで指定された場合、このメソッドはパスワード履歴もクリアします。

構文

UserPasswordState メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUser->UserPasswordState([pPwState][, emptyHistoryFlag])
```

パラメータ

UserPasswordState メソッドは、以下のパラメータを受け入れます。

pPwState (PolicyMgtUserPasswordState)

(オプション) 設定する新しいパスワード状態オブジェクトを指定します。

emptyHistoryFlag (int)

(オプション) パスワード履歴をクリアするかどうかを指定します。

- 値 = 0 (デフォルト)
パスワード履歴をクリアしないことを指定します。

- 値 = 1

パスワード履歴をクリアすることを指定します。

注: パスワード履歴をクリアすると、`last-password-change-time` 属性が 0 に設定されます。詳細については、メソッド

[PolicyMgtUserPasswordState->LastPWChangeTime](#) (P. 582) を参照してください。

戻り値

UserPasswordState メソッドは以下のいずれかの値を返します。

- PolicyMgtUserPasswordState (オブジェクト)
- コールに失敗した場合は、**undef**

ValidatePassword メソッド -- パスワードを検証

ValidatePassword メソッドは、ユーザパスワードがパスワードポリシーに適合するかどうかを判断します。SetPassword メソッドをコールする前に、ValidatePassword をコールします。

構文

ValidatePassword メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUser->ValidatePassword(password)
```

パラメータ

ValidatePassword メソッドは、以下のパラメータを受け入れます。

password (文字列)

(オプション) 検証するパスワードを指定します。

戻り値

ValidatePassword メソッドは以下のいずれかの値を返します。

- 値 = 0
パスワードが有効であることを指定します。
- 値 = -1
パスワードが有効でないことを指定します。

CLI ユーザ ディレクトリ メソッド

AnonymousIDAttr メソッド -- 匿名の DN 名を設定または取得

AnonymousIDAttr メソッドは、ユーザ ディレクトリの匿名ユーザ DN 属性の名前を設定または取得します。匿名の認証方式で定義される DN は、匿名の認証方式によって保護されたリソースへの匿名ユーザ アクセス権を与えます。LDAP ディレクトリおよびいくつかのカスタム ディレクトリと共に AnonymousIDAttr メソッドを使用できます。

構文

AnonymousIDAttr メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserDir->AnonymousIDAttr([anonIDAttr])
```

パラメータ

AnonymousIDAttr メソッドは以下のパラメータを受け入れます。

anonIDAttr (文字列)

(オプション) 匿名ユーザ DN 属性の新しい名前を指定します。

戻り値

AnonymousIDAttr メソッドは以下のいずれかの値を返します。

- **anonymous_user_dn_attribute_name** (文字列)
匿名ユーザ DN 属性の新規または既存の名前を指定します。
- **undef**
コールが失敗したことを指定します。

ChalRespAttr メソッド -- チャレンジ/レスポンス名を設定または取得

ChalRespAttr メソッドは、ユーザ ディレクトリのチャレンジ/レスポンス属性の名前を設定または取得します。LDAP ディレクトリおよびいくつかのカスタム ディレクトリと共に ChalRespAttr メソッドを使用できます。

構文

ChalRespAttr メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserDir->ChalRespAttr([chalRespAttr])
```

パラメータ

ChalRespAttr メソッドは以下のパラメータを受け入れます。

chalRespAttr (文字列)

(オプション) ユーザ ディレクトリのチャレンジ/レスポンス属性の新しい名前を指定します。

戻り値

ChalRespAttr メソッドは以下のいずれかの値を返します。

- **challenge_response_attribute_name** (文字列)
ユーザ ディレクトリのチャレンジ/レスポンス属性の新規または既存の名前を指定します。
- **undef**
コールが失敗したことを指定します。

Description メソッド -- ユーザ ディレクトリの説明を設定または取得

Description メソッドは、ユーザ ディレクトリの説明を設定または取得します。

構文

Description メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserDir->Description([userDirDesc])
```

パラメータ

Description メソッドは以下のパラメータを受け入れます。

userDirDesc (文字列)

(オプション) ユーザ ディレクトリの新しい説明を指定します。

戻り値

Description メソッドは以下のいずれかの値を返します。

- *user_directory_description* (文字列)
ユーザ ディレクトリの新規または既存の説明を指定します。
- "" (空の文字列)
コールが失敗したことを指定します。

DisabledAttr メソッド -- 無効化属性の名前を設定または取得

DisabledAttr メソッドは、ユーザの無効化状態が含まれているユーザ ディレクトリ属性の名前を設定または取得します。このメソッドは、LDAP ディレクトリ、ODBC ディレクトリおよびいくつかのカスタム ディレクトリに適用されます。

構文

DisabledAttr メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserDir->DisabledAttr([disabledAttr])
```

パラメータ

DisabledAttr メソッドは以下のパラメータを受け入れます。

disabledAttr (文字列)

(オプション) ユーザの無効化状態が含まれているユーザ ディレクトリ属性の新しい名前を指定します。

戻り値

DisabledAttr メソッドは以下のいずれかの値を返します。

- **disabled_attribute_name** (文字列)
ユーザの無効化状態が含まれているユーザ ディレクトリ属性の新規または既存の名前を指定します。
- **undef**
コールが失敗したことを指定します。

EmailAttr メソッド -- 電子メール属性名を設定または取得

EmailAttr メソッドは、電子メール属性の名前を設定または取得します。

注: このメソッドは将来の使用のために予約されています。

構文

EmailAttr メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserDir->EmailAttr([emailAttr])
```

パラメータ

EmailAttr メソッドは以下のパラメータを受け入れます。

emailAttr (文字列)

(オプション) 電子メール属性の新しい名前を指定します。

戻り値

EmailAttr メソッドは以下のいずれかの値を返します。

- **email_attribute_name** (文字列)
新規または既存の電子メール属性の名前を指定します。
- **undef**
コールが失敗したことを指定します。

EnableSecurityContext メソッド -- セキュリティ コンテキスト フラグを設定または取得

EnableSecurityContext メソッドは、セキュリティ コンテキストが有効かどうかを指定するユーザ ディレクトリ フラグを設定または取得します。

構文

EnableSecurityContext メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserDir->EnableSecurityContext([securityctxflag])
```

パラメータ

EnableSecurityContext メソッドは以下のパラメータを受け入れます。

securityctxflag (int)

(オプション) 以下のように、ユーザ ディレクトリのセキュリティ コンテキスト フラグの新しい値を指定します。

- 値 = 1 (有効化)
- 値 = 0 (無効化)

戻り値

EnableSecurityContext メソッドは、以下のようにセキュリティ コンテキスト フラグの新規または既存の値を返します。

- 値 = 1
セキュリティ コンテキストが有効であることを指定します。
- 値 = 0
セキュリティ コンテキストが無効であることを指定します。
- Sm_PolicyApi_Failure
コールが失敗したことを指定します。

GetContents メソッド -- ユーザ ディレクトリのすべてのユーザを取得

GetContents メソッドは、ユーザ ディレクトリのすべてのユーザを取得します。

構文

GetContents メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserDir->GetContents()
```

パラメータ

GetContents メソッドは、パラメータを受け入れません。

戻り値

GetContents メソッドは以下のいずれかの値を返します。

- PolicyMgtUser (配列)
- コールに失敗した場合は、**undef**

GetNamespace メソッド -- ユーザ ディレクトリ ネームスペースを取得

GetNamespace メソッドは、ユーザ ディレクトリ ネームスペースを取得します。

構文

GetNamespace メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserDir->GetNamespace()
```

パラメータ

GetNamespace メソッドは、パラメータを受け入れません。

戻り値

GetNamespace メソッドは以下のいずれかの値を返します。

- user_directory_namespace
- コールに失敗した場合は、**undef**

IsSecure メソッド -- セキュア認証フラグを設定または取得

IsSecure メソッドは、SiteMinder が LDAP またはカスタム ユーザ ディレクトリ用にセキュア認証を実行するかどうかを指定するフラグを設定または取得します。このフラグが有効な場合、SiteMinder 認証はセキュアであり、転送は暗号化されます。SSL を使用するときこのフラグを有効にします。

構文

IsSecure メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserDir->IsSecure([secureFlag])
```

パラメータ

IsSecure メソッドは以下のパラメータを受け入れます。

secureFlag (int)

(オプション) SiteMinder がセキュア認証を実行するかどうかを指定します。

- 値 = 1 (セキュア認証は有効)
- 値 = 0 (セキュア認証は無効)

戻り値

IsSecure メソッドは、以下のようにセキュア認証フラグの新規または既存の値を返します。

- 値 = 1
セキュア認証が有効であることを指定します。
- 値 = 0
セキュア認証が無効であることを指定します。
- 値 = -1
コールが失敗したことを指定します。

LookupEntry メソッド -- 指定されたパターンに一致するユーザを取得

LookupEntry メソッドは、指定された検索パターンに一致するユーザ ディレクトリのユーザ（複数可）を取得します。

構文

LookupEntry メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserDir->LookupEntry(srchPattern)
```

パラメータ

LookupEntry メソッドは以下のパラメータを受け入れます。

srchPattern（文字列）

ユーザ ディレクトリのユーザを検索するときに一致すべきパターンを指定します。

戻り値

LookupEntry メソッドは以下のいずれかの値を返します。

- PolicyMgtUser（配列）
- コールに失敗した場合は、**undef**

MaxResults メソッド -- 最大検索結果を設定または取得

MaxResults メソッドは、LDAP またはカスタム ユーザ ディレクトリの検索から返される検索結果の最大数を設定または取得します。

構文

MaxResults メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserDir->MaxResults([nResults])
```

パラメータ

MaxResults メソッドは以下のパラメータを受け入れます。

nResults (int)

(オプション) ユーザ ディレクトリの検索から返される最大結果の新しい数を指定します。

戻り値

MaxResults メソッドは以下のいずれかの値を返します。

- **maximum_results** (int)

ユーザ ディレクトリの検索から返される結果の新規または既存の最大数を指定します。

- **値 = -1**

コールが失敗したことを指定します。

Name メソッド -- ユーザ ディレクトリ名を設定または取得

Name メソッドは、ユーザ ディレクトリの名前を設定または取得します。

構文

Name メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserDir->Name([userDirName])
```

パラメータ

Name メソッドは以下のパラメータを受け入れます。

userDirName (文字列)

(オプション) ユーザ ディレクトリの新しい名前を指定します。

戻り値

Name メソッドは以下のいずれかの値を返します。

- **user_directory_name** (文字列)
ユーザ ディレクトリの新規または既存の名前を指定します。
- **undef**
コールが失敗したことを指定します。

ODBCQueryScheme メソッド -- ODBC クエリ方式を設定または取得

ODBCQueryScheme メソッドは、ユーザ ディレクトリ用の ODBC クエリ方式を設定または取得します。

構文

ODBCQueryScheme メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserDir->ODBCQueryScheme([odbcScheme])
```

パラメータ

ODBCQueryScheme メソッドは、以下のパラメータを受け入れます。

odbcScheme (PolicyMgtODBCQueryScheme)

(オプション) ユーザ ディレクトリの新しい ODBC クエリ方式を指定します。

戻り値

ODBCQueryScheme メソッドは以下のいずれかの値を返します。

- **odbcScheme** (PolicyMgtODBCQueryScheme)
- 方式が存在しない場合、またはコールに失敗した場合は、**undef**

Password メソッド -- ユーザのパスワードを設定または取得

Password メソッドはユーザ ディレクトリへのアクセスのためのユーザ パスワードを設定または取得します。

構文

Password メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserDir->Password([pwd])
```

パラメータ

Password メソッドは以下のパラメータを受け入れます。

pwd (文字列)

(オプション) ユーザ ディレクトリへのアクセス用の新しいユーザ パスワードを指定します。

戻り値

Password メソッドは以下のいずれかの値を返します。

- **password** (文字列)
新規または既存のユーザ パスワードを指定します。
- **undef**
コールが失敗したことを指定します。

PwdAttr メソッド -- パスワード属性名を設定または取得

PwdAttr メソッドは、ユーザ ディレクトリのパスワード属性の名前を設定または取得します。

構文

PwdAttr メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserDir->PwdAttr([pwdAttr])
```

パラメータ

PwdAttr メソッドは以下のパラメータを受け入れます。

pwdAttr (文字列)

(オプション) ユーザ ディレクトリのパスワード属性の新しい名前を指定します。

戻り値

PwdAttr メソッドは以下のいずれかの値を返します。

- **password_attribute_name** (文字列)
ユーザ ディレクトリのパスワード属性の新規または既存の名前を指定します。
- **undef**
コールが失敗したことを指定します。

PwdDataAttr メソッド -- パスワード データ属性名を設定または取得

PwdDataAttr メソッドは、ユーザ ディレクトリのパスワード データ属性の名前を設定または取得します。

構文

PwdDataAttr メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserDir->PwdDataAttr([pwdDataAttr])
```

パラメータ

PwdDataAttr メソッドは以下のパラメータを受け入れます。

pwdDataAttr (文字列)

(オプション) ユーザ ディレクトリのパスワード データ属性の新しい名前を指定します。

戻り値

PwdDataAttr メソッドは以下のいずれかの値を返します。

- **password_data_attribute_name** (文字列)
ユーザ ディレクトリのパスワードデータ属性の新規または既存の名前を指定します。
- **undef**
コールが失敗したことを指定します。

RequireCredentials メソッド -- 認証情報が必要かどうかを設定または取得

RequireCredentials メソッドは、SiteMinder でユーザ認証情報の確認が必要かどうかを指定するフラグを設定または取得します。

構文

RequireCredentials メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserDir->RequireCredentials([credFlag])
```

パラメータ

RequireCredentials メソッドは以下のパラメータを受け入れます。

credFlag (int)

(オプション) SiteMinder がユーザ認証情報の確認を要求されるかどうかを指定します。

- 値 = 1 (認証情報が必要)
- 値 = 0 (認証情報が不要)

戻り値

`RequireCredentials` メソッドは、認証情報要求フラグの新規または既存の値を返します。

- 値 = 1
認証情報が必要であることを指定します。
- 値 = 0
認証情報が必要でないことを指定します。
- 値 = -1
コールが失敗したことを指定します。

SearchRoot メソッド -- ディレクトリ検索ルートを設定または取得

`SearchRoot` メソッドは、以下のようにディレクトリ タイプごとに異なる値を設定または取得します。

LDAP ディレクトリ

`SearchRoot` メソッドは、たとえば、ディレクトリ接続の開始ポイントである LDAP ツリー内の場所として、たとえば、組織 (o)、組織単位 (ou) などを設定または取得します。検索ルートと呼ばれるこの場所は、ポリシー サーバがユーザに対して検索を開始するポイントです。

カスタム ディレクトリ

`SearchRoot` メソッドは、カスタム ライブラリに渡すパラメータ文字列を設定または取得します。

構文

`SearchRoot` メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserDir->SearchRoot([srchRoot])
```

パラメータ

`SearchRoot` メソッドは以下のパラメータを受け入れます。

srchRoot (文字列)

LDAP ディレクトリ用の新しい検索ルートまたはカスタム ディレクトリ用のパラメータ文字列を指定します。

戻り値

SearchRoot メソッドは以下のいずれかの値を返します。

- **search_root** (文字列)
LDAP ディレクトリ用の新規または既存の検索ルートかカスタム ディレクトリ用のパラメータ文字列を指定します。
- **undef**
コールが失敗したことを指定します。

SearchScope メソッド -- LDAP ディレクトリ検索範囲を設定または取得

SearchScope メソッドは、LDAP ユーザ ディレクトリの検索範囲を設定または取得します。 検索範囲は、LDAP ディレクトリ内のユーザまたはユーザグループを SiteMinder で検索するレベル数を指定します。

構文

SearchScope メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserDir->SearchScope([searchScope])
```

パラメータ

SearchScope メソッドは以下のパラメータを受け入れます。

searchScope (int)

(オプション) LDAP ユーザ ディレクトリの新しい検索範囲を指定します。

- **USERDIR_SCOPE_SUBTREE**
ルートおよびその下のすべてのレベルの検索を指定します。
- **USERDIR_SCOPE_ONELEVEL**
ルートおよびその 1 つ下のレベルの検索を指定します。

戻り値

SearchScope メソッドは以下のいずれかの値を返します。

- **USERDIR_SCOPE_SUBTREE**
ルートおよびその下のすべてのレベルの検索を指定します。

- **USERDIR_SCOPE_ONELEVEL**

ルートおよびその 1 つ下のレベルの検索を指定します。

- **値 = -1**

コールが失敗したことを指定します。

SearchTimeout メソッド -- 最大ディレクトリ検索時間を設定または取得

SearchTimeout メソッドは、LDAP またはカスタム ユーザ ディレクトリの検索に許容できる秒単位の最大時間を設定または取得します。

構文

SearchTimeout メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserDir->SearchTimeout([maxTimeout])
```

パラメータ

SearchTimeout メソッドは以下のパラメータを受け入れます。

maxTimeout (int)

(オプション) LDAP またはカスタム ユーザ ディレクトリの検索に許容できる新しい最大時間を秒単位で指定します。

戻り値

SearchTimeout メソッドは以下のいずれかの値を返します。

- **maximum_time_allowed** (int)

LDAP またはカスタム ユーザ ディレクトリの検索に許容できる新規または既存の最大時間を秒単位で指定します。

- **値 = -1**

コールが失敗したことを指定します。

Server メソッド -- ディレクトリ従属値を設定または取得

Server メソッドは、値を設定または取得します。値のタイプは、以下のよう
にユーザ ディレクトリのタイプに依存します。

LDAP および AD ディレクトリ

Server メソッドは、LDAP サーバの IP アドレスおよびポート番号を設定
または取得します。

ODBC ディレクトリ

Server メソッドは、データ ソース名を設定または取得します。

WinNT ディレクトリ

Server メソッドは、ドメイン名を設定または取得します。

カスタム ディレクトリ

Server メソッドは、ライブラリ名を設定または取得します。

構文

Server メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserDir->Server([server])
```

パラメータ

Server メソッドは以下のパラメータを受け入れます。

server (文字列)

(オプション) 以下のいずれかのタイプのディレクトリの新しい値を
指定します。

- LDAP および AD ディレクトリ

LDAP サーバの新しい IP アドレスおよびポート番号を指定します。

形式 : IP_address:port_number

デフォルト ポート番号 : 389

- ODBC ディレクトリ

新規のデータ ソース名を指定します。

- WinNT ディレクトリ
新規のドメイン名を指定します。
- カスタム ディレクトリ
新規のライブラリ名を指定します。

戻り値

Server メソッドは以下のいずれかの値を返します。

- value (文字列)
ユーザ ディレクトリの新規または既存の値を指定します。
- undef
コールが失敗したことを指定します。

SetPath メソッド--ユーザ パスの設定

構文

SetPath メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUser->SetPath(newPath, policy)
```

パラメータ

SetPath メソッドは、以下のパラメータを受け入れます。

newPath(文字列)

新しいパスを指定します。

policy(PolicyMgtPolicy オブジェクト)

ポリシー オブジェクトを指定します。

戻り値

SetPath メソッドは以下のいずれかの値を返します。

- 値 = 0
パスが正常に設定されたことを示します。
- 値 = -1
パスが正常に設定されなかったことを示します。

UIDAttr メソッド -- ユニバーサル ID 属性名を設定または取得

UIDAttr メソッドは、ユーザ ディレクトリのユニバーサル ID 属性の名前を設定または取得します。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。このメソッドは、LDAP ディレクトリ、ODBC ディレクトリ、WinNT ディレクトリおよびいくつかのカスタム ディレクトリに適用されます。

構文

UIDAttr メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserDir->UIDAttr([uidAttr])
```

パラメータ

UIDAttr メソッドは以下のパラメータを受け入れます。

uidAttr (文字列)

(オプション) ユニバーサル ID 属性の新しい名前を指定します。

戻り値

UIDAttr メソッドは以下のいずれかの値を返します。

- **uid_attribute_name** (文字列)
新規または既存のユニバーサル ID 属性の名前を指定します。
- **undef**
コールが失敗したことを指定します。

UserLookupEnd メソッド -- ユーザ DN 検索エンドポイントを設定または取得

UserLookupEnd メソッドは、LDAP ディレクトリ内の DN ユーザ検索用のエンドポイントを設定または取得します。

構文

UserLookupEnd メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserDir->UserLookupEnd([lookupEnd])
```

パラメータ

UserLookupEnd メソッドは以下のパラメータを受け入れます。

lookupEnd (文字列)

(オプション) ユーザ DN 検索エンドポイントの新しい値を指定します。

戻り値

UserLookupEnd メソッドは以下のいずれかの値を返します。

- **user_dn_lookup_endpoint** (文字列)
新規または既存のユーザ DN 検索エンドポイントを指定します。
- **undef**
コールが失敗したことを指定します。

注釈

ユーザ DN 検索開始ポイントおよびエンドポイントの値を指定すると、認証時に DN 文字列の一部を入力できます。以下の例で、ユーザはログインするときに DN 文字列全体ではなく、文字列「JSmith」のみを指定する必要があります。

- **DN** = "uid=JSmith,ou=marketing,o=myorg.org"
- **starting_point** = "uid="
- **endpoint** = ",ou=marketing,o=myorg.org"
- **login** = "JSmith"

UserLookupStart メソッド -- ユーザ DN 検索開始ポイントを設定または取得

UserLookupStart メソッドは、LDAP ディレクトリ内の DN ユーザ検索用の開始ポイントを設定または取得します。

構文

UserLookupStart メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserDir->UserLookupStart([lookupStart])
```

パラメータ

UserLookupStart メソッドは以下のパラメータを受け入れます。

lookupStart (文字列)

(オプション) ユーザ DN 検索開始ポイントの新しい値を指定します。

戻り値

UserLookupStart メソッドは以下のいずれかの値を返します。

- **user_dn_lookup_starting_point** (文字列)
新規または既存のユーザ DN 検索開始ポイントを指定します。
- **undef**
コールが失敗したことを指定します。

注釈

ユーザ DN 検索開始ポイントおよびエンドポイントの値を指定すると、認証時に DN 文字列の一部を入力できます。以下の例で、ユーザはログインするときに DN 文字列全体ではなく、文字列「JSmith」のみを指定する必要があります。

- DN = "uid=JSmith,ou=marketing,o=myorg.org"
- starting_point = "uid="
- endpoint = ",ou=marketing,o=myorg.org"
- login = "JSmith"

Username メソッド -- ユーザ名を設定または取得

Username メソッドは、ユーザ ディレクトリにアクセスするためのユーザ名を設定または取得します。RequireCredentials メソッドが 1 の値を返す場合にのみ、ユーザ名を設定します。

構文

Username メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserDir->Username([username])
```

パラメータ

Username メソッドは、以下のパラメータを受け入れます。

username (文字列)

(オプション) ユーザの新しい名前を指定します。

戻り値

Username メソッドは以下のいずれかの値を返します。

- **user_name** (文字列)
ユーザの新規または既存の名前を指定します。
- **undef**
コールが失敗したことを指定します。

ValidateEntry メソッド -- ユーザ ディレクトリ エントリを検証

ValidateEntry メソッドは、ユーザ ディレクトリ エントリを検証します。

構文

ValidateEntry メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserDir->ValidateEntry(path)
```

パラメータ

ValidateEntry メソッドは以下のパラメータを受け入れます。

path (文字列)

検証するユーザまたはユーザ グループのパスを指定します。

戻り値

ValidateEntry メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことを指定します。
- 値 = -1
メソッドが失敗したことを指定します。

CLI ユーザ パスワード状態メソッド

DisabledTime メソッド -- オブジェクトが無効化された時間を設定または取得

DisabledTime メソッドは、ユーザ オブジェクトが無効化された時間を設定または取得します。時間は、システム間で変化する特定の時点以来の秒数として表されます。一般的な表現は、00:00:00 1/1/1970 UTC（協定世界時）以来の経過秒数です。

構文

DisabledTime メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserPasswordState->DisabledTime([time])
```

パラメータ

DisabledTime メソッドは以下のパラメータを受け入れます。

time (文字列)

(オプション) ユーザ オブジェクトが無効化されたときの新しい時間を指定します。

戻り値

DisabledTime メソッドは、以下の値を返します。

- time (文字列)
(オプション) ユーザ オブジェクトが無効化された新規または既存の時間を指定します。

LastPWChangeTime メソッド -- パスワードが最後に変更された時間を設定または取得

LastPWChangeTime メソッドは、ユーザのパスワードが最後に変更された時間を設定または取得します。時間は、システム間で変化する特定の時点以来の秒数として表されます。一般的な表現は、00:00:00 1/1/1970 UTC (協定世界時) 以来の経過秒数です。

構文

LastPWChangeTime メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserPasswordState->LastPWChangeTime([time])
```

パラメータ

LastPWChangeTime メソッドは以下のパラメータを受け入れます。

time (文字列)

ユーザのパスワードが最後に変更されたときを表す新規の時間を指定します。

戻り値

LastPWChangeTime メソッドは以下のいずれかの値を返します。

- time (文字列)
(オプション) ユーザのパスワードが変更された新規または既存の時間を指定します。
- 値 = 0
ユーザがパスワードを変更し始めたが手順を完了しなかったことを指定します。

LastLoginTime メソッド -- 最後のログイン時間を設定または取得

LastLoginTime メソッドは、ユーザが最後に正しくログインした時間を設定または取得します。時間は、システム間で変化する特定の時点以来の秒数として表されます。一般的な表現は、00:00:00 1/1/1970 UTC（協定世界時）以来の経過秒数です。

構文

LastLoginTime メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserPasswordState->LastLoginTime([time])
```

パラメータ

LastLoginTime メソッドは以下のパラメータを受け入れます。

time（文字列）

（オプション）ユーザが最後に正しくログインしたときの新しい時間を指定します。

戻り値

LastLoginTime メソッドは、以下の値を返します。

- *time*（文字列）

ユーザが最後に正しくログインした新規または既存の時間を指定します。

LoginFailures メソッド -- ログインの失敗数を設定または取得

LoginFailures ユーザの最後の正しいログイン以後に、ユーザがログインに失敗した回数を設定または取得します。

構文

LoginFailures メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtUserPasswordState->LoginFailures([count])
```


パラメータ

LoginFailures メソッドは以下のパラメータを受け入れます。

count (int)

(オプション) ログイン失敗回数の新しい値を指定します。

戻り値

LoginFailures メソッドは以下のいずれかの値を返します。

- *count* (int)

ユーザの最後の正しいログイン以後のログイン失敗の新規または既存の回数を指定します。

CLI 変数タイプ メソッド

GetName メソッド -- 変数タイプ オブジェクトの名前を取得

GetName メソッドは、変数タイプ オブジェクトの名前を取得します。変数タイプ オブジェクトは読み取り専用です。GetName で返すことができる変数タイプ オブジェクト名のリストについては、[PolicyMgtSession->GetVariableType](#) (P. 534) メソッドを参照してください。

構文

GetName メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtVariableType->GetName()
```

パラメータ

GetName メソッドは、パラメータを受け入れません。

戻り値

GetName メソッドは以下のいずれかの値を返します。

- *variable_type_object_name* (文字列)
- コールに失敗した場合は、**undef**

WS-フェデレーション リソース パートナー メソッド

AddAttribute メソッド -- リソース パートナーに属性を追加

AddAttribute メソッドは、WS-Federation リソース パートナーに属性を追加します。

構文

AddAttribute メソッドの形式は以下のとおりです。

```
Netgerity::PolicyMgtWSFEDResourcePartner->AddAttribute(attrNameFormat, value)
```

パラメータ

AddAttribute メソッドは、以下のパラメータを受け入れます。

attrNameFormat (int)

以下のいずれかの属性タイプを指定します。

- WSFEDRP_EMAILADDRESS
- WSFEDRP_UPN
- WSFEDRP_COMMON
- WSFEDRP_GROUP
- WSFEDRP_NAMEVALUE

value (文字列)

以下のいずれかの形式で属性値を指定します。

- スタティック : *variableName* = *value*

注: タイプが WSFEDRP_NAMEVALUE でない限り、値の形式は属性のタイプに一致する必要があります。この場合、値は任意の形式にできます。

- ユーザ属性 : *variableName* = <%userattr="AttrName"%>
- DN 属性 : *variableName* = <#dn="DNSpec" attr="AttrName"#>

注: SiteMinder で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、DNSpec の前に簡単符を付けます。次のようにします。dn="!ou=People,o=security.com"

- アクティブ レスポンス

戻り値

AddAttribute メソッドは以下のいずれかの値を返します。

- PolicyMgtWSFEDRAttr (オブジェクト)
- コールに失敗した場合は、**undef**

AddUser メソッド -- リソース パートナーにユーザを追加

AddUser メソッドは、WS-Federation リソース パートナーにユーザを追加します。

構文

AddUser メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtWSFEDResourcePartner->AddUser(user)
```

パラメータ

AddUser メソッドは以下のパラメータを受け入れます。

user (PolicyMgtUser オブジェクト)

リソース パートナーに追加するユーザを指定します。

戻り値

AddUser メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことを指定します。
- 値 = -1
メソッドが失敗したことを指定します。

CreateIPConfigHostName メソッド -- 指定されたホストに基づいてオブジェクトを作成

CreateIPConfigHostName メソッドは、指定されたホスト名に基づいて、WS-Federation リソース パートナー用の IP 設定オブジェクトを作成します。このメソッドは、アサーション生成ポリシー用の IP アドレス制限を作成します。このアドレス制限では、指定されたホストから生成されたアサーションのみが受け入れられています。

構文

CreateIPConfigHostName メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtWSFEDResourcePartner->CreateIPConfigHostName(hostName)
```

パラメータ

CreateIPConfigHostName メソッドは以下のパラメータを受け入れます。

hostName (文字列)

アサーションを発行する必要があるホストの名前を指定します。

戻り値

CreateIPConfigHostName メソッドは以下のいずれかの値を返します。

- PolicyMgtIPConfig (オブジェクト)
- コールに失敗した場合は、**undef**

CreateIPConfigSingleHost メソッド -- 1 つのアドレスに基づいてオブジェクトを作成

CreateIPConfigSingleHost メソッドは、指定された IP アドレスに基づいて、WS-Federation リソース パートナー用の IP 設定オブジェクトを作成します。このメソッドは、アサーション生成ポリシー用の IP アドレス制限を作成します。このアドレス制限では、指定された IP アドレスから生成されたアサーションのみが受け入れられています。

構文

メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtWSFEDResourcePartner->CreateIPConfigSingleHost(ipAddr)
```

パラメータ

CreateIPConfigSingleHost メソッドは以下のパラメータを受け入れます。

ipAddr (文字列)

アサーションを発行する必要がある IP アドレスを指定します。

戻り値

CreateIPConfigSingleHost メソッドは以下のいずれかの値を返します。

- PolicyMgtIPConfig (オブジェクト)
- コールに失敗した場合は、**undef**

CreateIPConfigSubnetMask メソッド -- サブネット アドレスに基づいてオブジェクトを作成

CreateIPConfigSubnetMask メソッドは、指定された IP アドレスおよびサブネット マスクに基づいて、**WS-Federation** リソース パートナー用の IP 設定オブジェクトを作成します。このメソッドは、アサーション生成ポリシー用の IP アドレス制限を作成します。このアドレス制限では、サブネット アドレスから生成されたアサーションのみが受け入れられます。サブネット アドレスは、指定された IP アドレスおよびサブネット マスクから引き出されます。

構文

CreateIPConfigSubnetMask メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtWSFEDResourcePartner->CreateIPConfigSubnetMask(ipAddr,  
subnetMask)
```

パラメータ

CreateIPConfigSubnetMask メソッドは、以下のパラメータを受け入れます。

ipAddr (文字列)

サブネット アドレスを引き出すために使用する IP アドレスを指定します。

subnetMask (署名なし long)

サブネット アドレスを引き出すために使用するサブネット マスクを指定します。

注: サブネット マスクの詳細については、メソッド

[PolicyMgtPolicy->CreateIPConfigSubnetMask](#) (P. 369) を参照してください。

戻り値

CreateIPConfigSubnetMask メソッドは以下のいずれかの値を返します。

- PolicyMgtIPConfig (オブジェクト)
- コールに失敗した場合は、**undef**

DeleteIPConfig メソッド -- 指定された IP 設定 オブジェクトを削除

DeleteIPConfig メソッドは、指定された IP 設定オブジェクトを削除します。

構文

DeleteIPConfig メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtWSFEDResourcePartner->DeleteIPConfig(IPConfig)
```

パラメータ

DeleteIPConfig メソッドは以下のパラメータを受け入れます。

IPConfig (PolicyMgtIPConfig オブジェクト)

削除する IP 設定オブジェクトを指定します。

戻り値

DeleteIPConfig メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことを指定します。
- 値 = -1
メソッドが失敗したことを指定します。

GetAllAttributes メソッド -- リソース パートナーのすべての属性を取得

GetAllAttributes メソッドは、WS-Federation リソース パートナー用に定義されたすべての属性を取得します。

構文

GetAllAttributes メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtWSFEDResourcePartner->GetAllAttributes()
```

パラメータ

GetAllAttributes メソッドは、パラメータを受け入れません。

戻り値

GetAllAttributes メソッドは以下のいずれかの値を返します。

- PolicyMgtWSFEDRPAttr (配列)
- コールに失敗した場合は、**undef**

GetAllIPConfigs メソッド -- サービス プロバイダのすべての IP 設定オブジェクトを取得

GetAllIPConfigs メソッドは、サービス プロバイダのすべての IP 設定オブジェクトを取得します。

構文

GetAllIPConfigs メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtWSFEDResourcePartner->GetAllIPConfigs()
```

パラメータ

GetAllIPConfigs メソッドは、パラメータを受け入れません。

戻り値

GetAllIPConfigs メソッドは以下のいずれかの値を返します。

- PolicyMgtIPConfig (配列)
- IP 設定オブジェクトが見つからない場合は、**undef**

GetAllUsers メソッド -- リソース パートナーと関連付けられたすべてのユーザを取得

GetAllUsers メソッドは、WS-Federation リソース パートナーと関連付けられたすべてのユーザを取得します。ユーザ ディレクトリが指定される場合、このメソッドは単に指定されたディレクトリと関連付けられたユーザを返します。

構文

GetAllUsers メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtWSFEDResourcePartner->GetAllUsers([userDir])
```

パラメータ

GetAllUsers メソッドは以下のパラメータを受け入れます。

userDir (PolicyMgtUserDir オブジェクト)

(オプション) ユーザ ディレクトリと関連付けられたユーザのみを指定します。

戻り値

GetAllUsers メソッドは以下のいずれかの値を返します。

- PolicyMgtUser (配列)
- ユーザが見つからなかった場合またはエラーが発生した場合は、**undef**

Property メソッド -- リソース パートナー プロパティを設定または取得

Property メソッドは、指定された **WS-Federation** リソース パートナー プロパティの値を設定または取得します。 メタデータ プロパティのリストについては、このガイドで **WS-Federation** プロパティ リファレンス」を参照してください。

注: 1 つ以上のプロパティを変更した後に、**Save** をコールしてポリシー ストアに変更を書き込みます。

構文

Property メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtWSFEDResourcePartner->Property(name, [newvalue])
```

パラメータ

Property メソッドは、以下のパラメータを受け入れます。

name (文字列)

設定または取得するプロパティを指定します。

newvalue (文字列)

(オプション) リソース パートナー プロパティの新しい値を指定します。

戻り値

Property メソッドは以下のいずれかの値を返します。

- **value**

プロパティの新規または既存の値を指定します。

- **undef**

コールが失敗したことを指定します。

RemoveAttribute メソッド -- WSFED リソース パートナーから属性を削除

RemoveAttribute メソッドは、WS-Federation リソース パートナーから属性を削除します。

構文

RemoveAttribute メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtWSFEDResourcePartner->RemoveAttribute(attrName)
```

パラメータ

RemoveAttribute メソッドは以下のパラメータを受け入れます。

attrName (PolicyMgtWSFEDRPAttr)

削除する属性を指定します。

戻り値

RemoveAttribute メソッドは以下のいずれかの値を返します。

- 成功の場合は 0
- 失敗の場合は -1

RemoveUser メソッド -- リソース パートナーから指定されたユーザを削除

RemoveUser メソッドは、WS-Federation リソース パートナーから指定されたユーザを削除します。

構文

RemoveUser メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtWSFEDResourcePartner->RemoveUser(user)
```

パラメータ

RemoveUser メソッドは以下のパラメータを受け入れます。

user (PolicyMgtUser オブジェクト)

リソース パートナーから削除するユーザを指定します。

戻り値

RemoveUser メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことを指定します。
- 値 = -1
メソッドが失敗したことを指定します。

Save メソッド -- リソース パートナーのメタデータを保存

Save メソッドは、ポリシー ストアに WS-Federation リソース パートナーのメタデータを書き込みます。メタデータを変更するには、PolicyMgtWSFEDResourcePartner->Property メソッドをコールします。次に、Save をコールして変更を保存します。

構文

Save メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtWSFEDResourcePartner->Save()
```

パラメータ

Save メソッドは、パラメータを受け入れません。

戻り値

Save メソッドは以下のいずれかの値を返します。

- 値 = 0
メソッドが成功したことを指定します。
- 値 = -1
メソッドが失敗したことを指定します。

- 値 = -4
変更を保存するためのユーザの権限が十分でないことを指定します。
- 値 = -10
パスとクラスが空であることを指定します。

CLI WS-フェデレーション リソース パートナー属性メソッド

以下のメソッドが PolicyMgtWSFEDResourcePartnerAttr オブジェクトに有効です。

- GetAttrNameFormat メソッド -- 属性名の形式を取得
- GetValue メソッド -- 属性値を取得

GetAttrNameFormat メソッド -- 属性名の形式を取得

GetAttrNameFormat メソッドは、この WS-Federation リソース パートナーと共に使用する属性名の形式を取得します。

構文

GetAttrNameFormat メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtWSFEDRPartner->GetAttrNameFormat()
```

パラメータ

GetAttrNameFormat メソッドは、パラメータを受け入れません。

戻り値

GetAttrNameFormat メソッドは以下のいずれかの整数値を返します。

- WSFEDRP_EMAILADDRESS (値 = 0)
- WSFEDRP_UPN (値 = 1)
- WSFEDRP_COMMON (値 = 2)
- WSFEDRP_GROUP (値 = 3)
- WSFEDRP_NAMEVALUE (値 = 4)

GetValue メソッド -- 属性値を取得

GetValue メソッドは、WS-Federation リソース パートナー属性の値を取得します。

構文

GetValue メソッドの形式は以下のとおりです。

```
Netegrity::PolicyMgtWSFEDRPAAttr->GetValue()
```

パラメータ

GetValue メソッドは、パラメータを受け入れません。

戻り値

GetValue メソッドは以下のいずれかの値を返します。

- attribute_value
- コールに失敗した場合は、**undef**

第 8 章: CLI ポリシー管理操作

このセクションには、以下のトピックが含まれています。

- [セッションの初期化](#) (P. 597)
- [システム オブジェクトの作成および管理](#) (P. 599)
- [ドメイン範囲を持つオブジェクト](#) (P. 601)
- [ドメイン範囲またはグローバル範囲を持つオブジェクト](#) (P. 604)
- [許可変数](#) (P. 606)
- [オブジェクトへの変更の保存](#) (P. 612)
- [パスワード ポリシーの変更](#) (P. 613)
- [パスワード状態の管理](#) (P. 613)
- [レスポンスとレスポンス属性の作成](#) (P. 616)
- [新しい認証方式によるレルムの更新](#) (P. 617)
- [認証方式テンプレートのデフォルト値の表示](#) (P. 618)
- [認証方式の作成](#) (P. 619)
- [共有秘密キー ロールオーバー ポリシーの変更](#) (P. 620)
- [ファイルへのドメインおよびレルム レポートの書き込み](#) (P. 621)
- [認証および許可イベント処理の無効化](#) (P. 622)
- [ポリシー サーバ負荷分散の管理](#) (P. 623)

セッションの初期化

セッションを作成すると、多くのセッション初期化フラグがそのデフォルト値に設定されます。以下の表は、PolicyMgtAPI オブジェクト内の初期化メソッドおよびそれらのデフォルト値の一覧です。

メソッド	デフォルト	説明
DisableAudit()	0 (監査は有効)	有効化または無効化 ■ 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）。 ■ ユーザセッションの監視。

メソッド	デフォルト	説明
DisableManagement WatchDog()	0 (ウォッチドッグは有効)	SiteMinder 管理ウォッチドッグを有効または無効にします。ウォッチドッグは内部的に使用され無効にはしません。
無効化 Validation()	0 (検証は有効)	ポリシーストアオブジェクトの検証を有効または無効にします。
LoadAgentType Dictionary()	0 (事前ロード ディクショナリ は無効)	エージェントタイプ辞書の事前ロードを有効または無効にします。
PreLoadCache()	0 (事前ロード キャッシュは無効)	SiteMinder キャッシュの事前ロードを有効または無効にします。

注: CreateSession() の後にコールした場合、これらのメソッドは効果を示しません。

例: セッションの初期化操作

以下の例は、デフォルトでは有効にされないセッション初期化操作をすべて有効にします。セッションが正しく初期化されると、スクリプトは以下のように初期化フラグを表示します。

```
use Netegrity::PolicyMgtAPI;

$username = "adminid";
$password = "adminpwd";

print "%nInitializing and connecting to PolicyMgtAPI...%n";
$policyapi = Netegrity::PolicyMgtAPI->New();
```

```
$policyapi->PreLoadCache(1);
$policyapi->LoadAgentTypeDictionary(1);

die "ERROR: Couldn't create session¥n" unless ($session != undef);

print "Initialization settings:¥n";
print "  Preload cache flag: ".$policyapi->PreLoadCache()."¥n";
print "  Disable validation flag: " .
    $policyapi->DisableValidation()."¥n";
print "  Load agent type dictionary flag: " .
    $policyapi->LoadAgentTypeDictionary()."¥n";
print "  Disable audit flag: ".$policyapi->DisableAudit()."¥n";
print "  Disable watchdog flag: " .
    $policyapi->DisableManagementWatchDog()."¥n";
```

システム オブジェクトの作成および管理

ポリシー サーバへの接続を初期化するときに、セッション オブジェクト (PolicyMgtSession) を作成します。[SiteMinder 管理] ウィンドウの [システム] タブにリスト表示される、システム レベルオブジェクトを作成、取得、削除するためにセッション オブジェクトを使用します。

システム オブジェクト (グローバル オブジェクトとも呼ばれる) の範囲はグローバルです。つまり、それらはポリシー ストア内のすべてのドメインにわたって表示できます。セッション オブジェクトから作成できるシステム オブジェクトには以下のものが含まれます。

- 管理者
- エージェントおよびエージェント タイプ
- 認証方式
- 認証および許可マップ
- ドメイン
- ODBC クエリ方式
- パスワード ポリシー
- 登録方式
- ユーザ ディレクトリ

エージェント オブジェクトの作成

エージェント オブジェクトはグローバル範囲を持つシステム オブジェクトです。以下の例は、セッション オブジェクトを使用してエージェントを作成および設定し、次に、ポリシー ストアで設定されているエージェント名をすべて出力します。

```
use Netegrity::PolicyMgtAPI;

$policyapi = Netegrity::PolicyMgtAPI->New();
$session = $policyapi->CreateSession("adminid", "adminpwd");

$ip="127.0.0.1";
$secret="oursecret";

$agentType=$session->GetAgentType("Web Agent");
$session->CreateAgent("agent1", $agentType, "", $ip, $secret);

@agents=$session->GetAllAgents();
foreach $agent(@agents) {
    print $agent->Name()."\n";
}
```

注: この例では **v4.x** エージェントを作成します。**v5.x** または **v6.x** エージェントを作成するには、共有秘密キーを指定しません。

オブジェクト プロパティの表示および変更

オブジェクトを作成した後（またはオブジェクトを 管理 UI で作成した後）は、オブジェクトの **get** および **set** メソッドをコールして、作成されたオブジェクトの個々のプロパティを表示および変更できます。

通常の場合、オブジェクトの **get** および **set** メソッド名は、取得または変更されるプロパティの名前を表します。また、メソッドは単一のオプションの引数をとります。引数を指定する場合は、オブジェクトのプロパティを設定します。引数を省略すると、プロパティ値が変更されずに取得されます。メソッドは、新規または既存のプロパティ値を返します。

以下の例では、それぞれのユーザディレクトリがその **Maximum Results** プロパティに関してチェックされます。つまり、ディレクトリ検索の後に返される検索結果の最大数を指定する値です。取得された数が 25 でない場合、スクリプトは以下のようにプロパティを 25 に設定します。

```
use Netegrity::PolicyMgtAPI;

$policyapi = Netegrity::PolicyMgtAPI->New();
$session = $policyapi->CreateSession("adminid", "adminpwd");

$max="25";

@userdirs=$session->GetAllUserDirs();
foreach $userdir(@userdirs) {
    print "\nMax results for directory " . $userdir->Name()."\n";
    if ($userdir->MaxResults() != $max) {
        print " Updating from " . $userdir->MaxResults(). " to " .
                                                    $max . "\n";
        $userdir->MaxResults($max);
    }
    else {
        print " Max results are correct.\n";
    }
}
```

ドメイン範囲を持つオブジェクト

作成できるシステム オブジェクトの 1 つはドメイン オブジェクト (**PolicyMgtDomain**) です。ドメイン オブジェクト自体にはグローバル範囲がありますが、ドメイン オブジェクトを通じて作成し取得するオブジェクトには **ドメイン範囲**があります。つまり、それらはドメイン内でのみ表示でき、ドメイン間では共有できません。

ドメイン オブジェクトは [SiteMinder 管理] ウィンドウの [ドメイン] タブにリスト表示されます。

ドメイン範囲を持つオブジェクトには以下のものがあります。

- ポリシー
- レルム
- レスポンスとレスポンス属性
- ルール
- ユーザ ポリシー

1 つのオブジェクトを取得して別のオブジェクトを作成

オブジェクトを操作する前に、より高いレベルのオブジェクトを取得することが必要な場合があります。つまり、ドメイン範囲があるオブジェクトの場合です。

ドメイン範囲を持つポリシーを作成する例を以下に示します。ポリシーオブジェクトを作成する前に、ポリシーを配置するドメインオブジェクトを取得することに注意してください。

```
use Netegrity::PolicyMgtAPI;

$policyapi = Netegrity::PolicyMgtAPI->New();
$session = $policyapi->CreateSession("adminid", "adminpwd");
$domain=$session->GetDomain("engineering");
$policy=$domain->CreatePolicy("Payroll Policy");
if($policy == undef) {
    print "Couldn't create the policy.¥n";
}
else {
    print "Successfully created policy " . $policy->Name() . ".¥n";
}
```

オブジェクトのプロパティの管理

グローバル範囲を持つオブジェクトの場合と同じように、オブジェクトの **get** および **set** メソッドを使用して、ドメイン範囲を持つオブジェクトのプロパティを表示および設定します。

オブジェクトのプロパティの値は、文字列または数値ではなく別のオブジェクトである場合があります。たとえば、ポリシーでは、ポリシーにルールおよびユーザを追加し、ポリシー内のルール用のレスポンスを設定します。これらの値はすべてオブジェクトとして提供されます。

例：ポリシーへのオブジェクトの追加

以下の例で、ポリシーはユーザ、ルールおよびレスポンスと共に設定されます。

```
use Netegrity::PolicyMgtAPI;

$policyapi = Netegrity::PolicyMgtAPI->New();
$session = $policyapi->CreateSession("adminid", "adminpwd");
$domain=$session->GetDomain("engineering");
$policy=$domain->GetPolicy("Payroll Policy");

# Add a user to the policy
$userdir=$session->GetUserDir("Acme North Directory");

@users=$userdir->LookupEntry("uid=ppaycheck");
foreach $user(@users) {
    if ($user->GetPath()=="uid=ppaycheck,ou=HR,o=security.com") {
        $userResult=$policy->AddUser($user);
        $thisUser=$user;
    }
}

if ($userResult != 0) {
    print "Error adding user to policy.¥n";
}

# Add a rule to the policy
$realm=$domain->GetRealm("HR");
$rule=$realm->GetRule("Payroll Rule");
$ruleResult=$policy->AddRule($rule);
if ($ruleResult != 0) {
    print "Error adding rule to policy.¥n";
}
else
{
    # Set a response for the rule
    $response=$domain->GetResponse("Welcome to Payroll");
    $respResult=$policy->SetResponse($rule,$response);
    if ($respResult != 0) {
        print "Error adding response to policy.¥n"
    }
}

print "¥nAdded these objects to the policy:¥n";
print "  User DN: ".$thisUser->GetPath()."¥n" unless $userResult!=0;
print "  Rule: ".$rule->Name()."¥n" unless $ruleResult!=0;
print "  Response: ".$response->Name()."¥n" unless $respResult!=0;
```

ドメイン範囲またはグローバル範囲を持つオブジェクト

いくつかのオブジェクトはドメイン範囲またはグローバル範囲のどちらでも作成できます。それらのオブジェクトは以下のとおりです。

- PolicyMgtPolicy
 - ドメイン： PolicyMgtDomain->CreatePolicy()
 - グローバル： PolicyMgtSession->CreateGlobalPolicy()
- PolicyMgtResponse
 - ドメイン： PolicyMgtDomain->CreateResponse()
 - グローバル： PolicyMgtSession->CreateGlobalResponse()
- PolicyMgtRule
 - ドメイン： PolicyMgtRealm->CreateRule()
 - グローバル： PolicyMgtSession->CreateGlobalRule()
- PolicyMgtGroup
 - ドメイン： PolicyMgtDomain->CreateResponseGroup()
 - ドメイン： PolicyMgtDomain->CreateRuleGroup()
 - グローバル： PolicyMgtSession->CreateAgentGroup()

以下のテーブルでは、ドメイン範囲およびグローバル範囲があるときのポリシー、レスポンスおよびルール オブジェクトを比較します。

オブジェクト	ドメイン範囲	グローバル範囲
ポリシー	特定のユーザまたはグループのユーザにバインドされます。	すべてのユーザにバインドされる。
	ポリシーに対して個々のユーザを含めたり除外したりできます。	ユーザを個別に含めたり除外したりはできません。
	ドメイン固有のルールとルールグループ、ドメイン固有のレスポンスとレスポンスグループ、およびグローバルレスポンスを使用します。	グローバルルールおよびグローバルレスポンスのみを使用します。
	変数式を使用できます。	変数式を使用できません。

オブジェクト	ドメイン範囲	グローバル範囲
Response	ドメイン固有のポリシーで使用されます。	グローバル固有またはドメイン固有のポリシーで使用されます。
	ドメイン固有レスポンス グループのメンバにできます。	ドメイン固有レスポンス グループのメンバにできます。グローバル レスポンス グループはサポートされていません。
	変数ベースの属性を使用できる。	変数ベースの属性は使用できない。
ルール	ドメイン固有のポリシーで使用されます。	グローバル ポリシーで使用されます。
	レルムを介してエージェントと関連付けられます。	特定のエージェントまたはエージェント グループと関連付けられます。エージェントまたはエージェント グループは、グローバル ルールの作成時に指定されます。
	リソース フィルタは特定のレルム（ルール フィルタとレルム フィルタ）にバインドされます。	リソース フィルタは絶対フィルタです（つまり、レルムにバインドされません）。
	特定のドメイン内で定義されたリソースの場合のみ起動します。	グローバル ポリシー処理が有効なドメイン内で定義されているリソースの場合に起動します。
	アクセスルールまたはイベントルールとして定義できます。	イベントルールとしてのみ定義できます（認証および許可イベント）。
	ドメイン固有ルール グループのメンバにできます。	ドメイン固有ルール グループのメンバにできます。グローバル ルール グループはサポートされていません。
すべて	特定のドメインのコンテキストでドメイン管理者によって作成されます。	システム レベルでシステム管理者によって作成されます。

許可変数

許可変数は、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。

許可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。その後、値は評価され、ユーザを許可すべきかどうかに関する決定に使用されます。

たとえば、銀行のクレジット カード申込用紙を保護するポリシーに、**Credit Rating** 変数および **Salary** 変数を使用したアクティブな式が含まれると仮定します。ユーザがフォームにアクセスしようとする、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、レスポンス属性はリモート **Web** サービスから取得されたトランザクションのトラッキング番号を返すように設定できます。

許可変数を使用するには、**SiteMinder** オプション パックをインストールする必要があります。

特定の変数タイプに応じた変数の設定

PolicyMgtDomain オブジェクト用の **CreateVariable()** をコールして、変数を作成および設定します。

このメソッドの引数の1つは **定義**です。この引数の値は、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 **定義** 引数の説明を以下に示します。

- **ポスト**

定義 引数には、**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

定義引数には以下の XML コードが含まれます。

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

変数値は、ItemName エlement 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**。可変値は、リクエストで指定されたアクションのタイプです（たとえば GET または POST）。
- **Resource**。変数値はターゲット リソースです（たとえば、/directory_name/）。
- **Server**。変数値は、要求で指定された完全サーバ名です（たとえば、server.company.com）。

■ スタティック

定義引数には、ランタイムにユーザ提供の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 VAR_RETTYPE_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

定義引数には以下のいずれかまたはすべての XML コードが含まれます。

```
<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 Element ItemName 内で使用されます。

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。

有効な **ItemName** 値の全リストについては、「[ポリシー管理API リファレンス \(PolicyMgtAPI.htm\)](#)」での **CreateVariable()** の説明を参照してください。

■ **WebService**

定義引数には以下の XML 基本構造が含まれます。

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

要素	説明
RemoteURL	WebService 変数を解決する Web サービスの URL。

要素	説明
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。
AuthCredentials	オプションで、以下の要素によってユーザの Web サービス認証情報を指定します。 ■ Username ■ 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) 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。
Document	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。 ■ Envelope。 SOAP ネームスペースは次のとおりです： <code>http://schemas.xmlsoap.org/soap/envelope</code> ■ 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。 ■ Body。 ユーザ定義の SOAP 本文。 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。 ネストされた変数は、以下のように指定します。 <code>\$variable-name\$</code>

注: 上に表示された XML エlement 構造は、読みやすいようにフォーマットされます。定義引数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 属性用の RequestContext 変数は定義を以下のように渡されます。

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

以下の情報が `CreateVariable()` のコールに必要です。

- ユーザ定義の変数名。
- 変数タイプ -- たとえば Static または ResourceContext。
- 変数定義。
- 変数値のデータ型。有効なデータ型値は以下のとおりです。
 - VAR_RETTYPE_BOOLEAN
 - VAR_RETTYPE_NUMBER
 - VAR_RETTYPE_STRING
 - VAR_RETTYPE_DATE

オプションの TransactionMinder 製品とオプション パックの両方がインストールされている場合、以下のタイプの変数を使用できます。

- SAMLAssertion
- トランスポート
- XMLAgent
- XMLBody
- XMLEnvironment

コマンドライン インターフェースではこれらのタイプの変数を作成できません。管理 UI を使用してのみ作成できます。

例：ResourceContext 変数の作成

以下の例では、ResourceContext 変数として変数 `MyVar` を作成します。変数値は保護されているリソースです（たとえば、`/directory_name/`）。

```
use Netegrity::PolicyMgtAPI;  
  
$pmgtapi=Netegrity::PolicyMgtAPI->New();  
$session=$pmgtapi->CreateSession("adminid", "adminpwd");
```

```
$dom=$session->GetDomain("MyDomain");
$varName="MyVar";
$varType=$session->GetVariableType("RequestContext");
$varDef="<RequestContextVariableDef><ItemName>Resource</ItemName>
</RequestContextVariableDef>";

$vr=$dom->CreateVariable($varName,$varType,$varDef,
    VAR_RETTYPE_STRING);

if ($vr==undef) {
    print "Create operation failed.";
}
else {
    print "Created variable " . $varName;
}
```

オブジェクトへの変更の保存

以下のオブジェクトのインスタンスに変更を加えるときは、オブジェクトの `Save()` メソッドをコールして変更を保存する必要があります。これらのオブジェクトに変更を加えた後は、`Save()` を 1 回コールします。

- `PolicyMgtAffiliate`
- `PolicyMgtAuthScheme`
- `PolicyMgtPwdPolicy`
- `PolicyMgtSAMLAffiliation`
- `PolicyMgtSAMLServiceProvider`
- `PolicyMgtSharedSecretPolicy`

以下の例では、認証方式オブジェクトのプロパティを変更します。

```
use Netegrity::PolicyMgtAPI;

$policyapi = Netegrity::PolicyMgtAPI->New();
$session = $policyapi->CreateSession("adminid", "adminpwd");
$authscheme=$session->GetAuthScheme("HTML Authentication");

#Specify the attribute list
$attrList="AL=PASSWORD,SSN,age,zipcode;";
#Specify the new credentials collector
$target="http://my.server.com/siteminderagent/forms/customlogin.fcc";

#Make the changes and save them
$authscheme->ProtectionLevel(10);
$authscheme->CustomParam($attrList.$target);
$authscheme->Save();
```

パスワードポリシーの変更

会社サイトに定義された異なるパスワードポリシーがあり、特定のパスワードポリシー要件を標準化することを決定したと想定します。以下のスクリプトは各サイトで新しいパスワードポリシー定義を設定するために実行できます。

```
use Netegrity::PolicyMgtAPI;

$policymgtapi = Netegrity::PolicyMgtAPI->New();
$session = $policymgtapi->CreateSession("adminid", "adminpwd");

@pwdpols=$session->GetAllPwdPolicies();
foreach $pwdpol(@pwdpols) {
    $pwdpol->Description("Standardized settings 02/04/15");
    print "%n%nPwd Policy: " . $pwdpol->Name();
    print "%nUpdated the following settings:";
    print "%n  Minimum length:%t". $pwdpol->PwdMinLength(6);
    print "%n  Maximum length:%t". $pwdpol->PwdMaxLength(18);
    print "%n  Minimum digits:%t". $pwdpol->PwdMinNumbers(3);
    print "%n  Allowable failures:%t". $pwdpol->MaxLoginFailures(3);
    print "%n  Days before reuse:%t". $pwdpol->PwdReuseDelay(360);
    print "%n  Changes before reuse:%t". $pwdpol->PwdReuseCount(6);
    $pwdpol->Save();
}
```

パスワード状態の管理

パスワード状態は、たとえばパスワードが最後に変更されたとき、およびパスワードが最後にユーザをログインするために使用されたときなど、特定のユーザのパスワードに関連するアクティビティを示しています。

ユーザ用の既存の **PolicyMgtUserPasswordState** オブジェクトを取得するか、任意の属性変更で新しいパスワード状態オブジェクトを設定するには、**PolicyMgtUser->UserPasswordState()** をコールします。

特定のユーザに対してアクセスできるパスワード状態属性および属性値の設定または取得に使用されるメソッドを以下のテーブルにリスト表示します。特に明記されない場合、すべてのメソッドはオブジェクト `PolicyMgtUserPasswordState` にあります。

パスワード状態属性	メソッド	説明
ログインの失敗	<code>LoginFailures()</code>	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設定または取得します。
最終ログイン時間	<code>LastLoginTime()</code>	ユーザが最後に正常にログインした時間を設定または取得します。
前のログイン時間	<code>PrevLoginTime()</code>	ユーザの最後から 2 番目の正常なログインの時間を設定または取得します。
無効化時間	<code>DisabledTime()</code>	ユーザ オブジェクトが無効化された時間を設定または取得します。
パスワード履歴	<code>PolicyMgtUser->UserPasswordState()</code>	オプションで、ユーザのパスワード状態オブジェクトを設定するときに、ユーザのパスワード履歴をクリアします。 パスワード履歴を取得したり、パスワード履歴エントリを設定することはできません。
前回のパスワード変更時間	<code>LastPWChangeTime()</code>	ユーザ パスワードが最後に変更された時間を設定または取得します。

パスワード状態属性を変更する場合、変更は現在のパスワード状態オブジェクトのみに適用されます。続けて取得される可能性のあるパスワード状態オブジェクトに変更を適用するには、現在のパスワード状態オブジェクトを `PolicyMgtUser->UserPasswordState()` の呼び出しで受け渡します。このメソッドは、メソッドに渡される属性値を含む新規パスワード状態オブジェクトを設定します。

たとえば、以下のコードフラグメントは以下の操作を実行します。

1. パスワード状態オブジェクト、`$passwordstate` を現在のユーザ、`$user[0]` に対して取得します。
2. パスワード状態オブジェクトのこのインスタンスでログイン失敗属性を 3 に設定します。
3. `UserPasswordState()` をコールして、ユーザのパスワード履歴をクリアし、新しい履歴およびログイン失敗属性でユーザ用の新しいパスワード状態オブジェクトを設定します。

```
$passwordstate = $user[0]->UserPasswordState();  
$passwordstate->LoginFailures(3);  
$user[0]->UserPasswordState($passwordstate, 1);
```


レスポンスとレスポンス属性の作成

以下のスクリプトはレスポンスとレスポンスの属性を作成します。
CreateResponse() へのコール用のエージェント タイプ オブジェクトを取得するために **GetAgentType()** をコールする必要があることに注意します。

```
use Netegrity::PolicyMgtAPI;

$policyapi = Netegrity::PolicyMgtAPI->New();
$session = $policyapi->CreateSession("adminid", "adminpwd");

$domain = $session->GetDomain("Acme North Domain");
$agenttype=$session->GetAgentType("Web Agent");
$response=$domain->CreateResponse("Welcome to Payroll",$agenttype);
if ($response==undef) {
    print "\nCouldn't create the response.";
}
else {
    $attr=$response->CreateAttribute("WebAgent-HTTP-Cookie-Variable",
                                    "cookieName=mycookie");

    if ($attr==undef) {
        print "\nCouldn't create attribute for ".$response->Name();
    }
    else {
        print "\nCreated response " . $response->Name();
        print "\nCreated attribute " . $attr->GetValue();
    }
}
```

新しい認証方式によるレルムの更新

以下の例は、Acme North Domain ドメイン内のすべてのレルムに認証方式 HTML Authentication を割り当てます。

```
use Netegrity::PolicyMgtAPI;

$policyapi = Netegrity::PolicyMgtAPI->New();
$session = $policyapi->CreateSession("adminid", "adminpwd");

$domain = $session->GetDomain("Acme North Domain");
$authscheme=$session->GetAuthScheme("HTML Authentication");

@realms=$domain->GetAllRealms();
print "Updating these realms to auth scheme ".$authscheme->Name();
foreach $realm(@realms) {
    $name=$realm->Name();
    if ($realm->AuthScheme($authscheme)==undef) {
        print "%n Couldn't update realm " . $name;
    }
    else {
        print "%n Realm ". $name;
    }
}
```

認証方式テンプレートのデフォルト値の表示

以下のスクリプトは、認証スキーム テンプレートである **HTML Form Template** 用のデフォルト値を表示します。

認証方式を作成することは必要ではありません。 スクリプトは、以下のよう
にテンプレート用の認証方式オブジェクトを取得するために **GetAuthScheme()** を使用し、次にオブジェクトのデフォルト プロパティを出力します。

```
use Netegrity::PolicyMgtAPI;

$policyapi = Netegrity::PolicyMgtAPI->New();
$session = $policyapi->CreateSession("adminid", "adminpwd");

#Retrieve the object for the authentication scheme template
$template=$session->GetAuthScheme("HTML Form Template");

print "\nDefault values for template " . $template->Name();
print "\n  Type:%t" . $template->Type()->Name();
print "\n  Description:%t" . $template->Description();
print "\n  Protection level:%t" . $template->ProtectionLevel();
print "\n  Library:%t" . $template->CustomLib();
print "\n  Parameter:%t" . $template->CustomParam();
print "\n  Shared secret:%t" . $template->CustomSecret();
print "\n  Is template?:%t" . $template->IsTemplate();
print "\n  Is used by admin?:%t" . $template->IsUsedByAdmin();
print "\n  Save credentials?:%t" . $template->SaveCredentials();
print "\n  Is Radius:%t" . $template->IsRadius();
print "\n  Ignore pwd ck?:%t" . $template->IgnorePwd();
```

認証方式の作成

認証方式を作成するときに、方式のベースとして認証方式テンプレートを 사용합니다。そうするには、まず既存のテンプレートを取得し、次に、`CreateAuthScheme()` へのコールでテンプレートオブジェクトを指定します。

以下の例は、テンプレート [HTML フォーム テンプレート] に基づいて認証方式を作成し、デフォルトをすべて受け入れます。

```
use Netegrity::PolicyMgtAPI;

$policyapi = Netegrity::PolicyMgtAPI->New();
$session = $policyapi->CreateSession("adminid", "adminpwd");

#Retrieve the object for the authentication scheme template
$template=$session->GetAuthScheme("HTML Form Template");

#Create the authentication scheme
$scheme=$session->CreateAuthScheme("HTML Authentication",$template);
if ($scheme == undef) {
    print "\nCouldn't create the authentication scheme.";
}
else {
    print "\nCreated authentication scheme ".$scheme->Name();
}
```

共有秘密キー ロールオーバー ポリシーの変更

共有秘密キーは、信頼されたホスト、およびホストが登録されているポリシー ストア ドメインにのみ既知のテキスト文字列です。共有秘密キーは、ポリシー サーバへのセキュア接続を行うときに、トラステッドホストの ID を認証するために使用されます。

共有秘密キーのロールオーバー機能は、共有秘密キーの定期的な変更を自動的に行うためのメカニズムを提供します。

Scripting Interface for Perl を使用して、次のことができます。

- 既存の共有秘密キーのロールオーバー ポリシー オブジェクトを取得するために `GetSharedSecretPolicy()` をコールする。
- 取得された `PolicyMgtSharedSecretPolicy` オブジェクトを以下のように変更する。
 - ロールオーバー頻度を変更する。この設定は、ロールオーバーが指定されたロールオーバー期間に行われる頻度を指定します（次の項目を参照）。
 - ロールオーバー期間（1 時間ごと、日単位、週単位、月単位）を変更する。たとえば、ロールオーバー頻度が 3 で、日単位ロールオーバー期間の場合、共有秘密キーは 3 日おきに自動的に変更されます。
 - ロールオーバー機能を有効または無効にする。共有秘密キーのロールオーバーポリシーが有効な場合は、その共有秘密キーをロールオーバーポリシーの共有秘密キーと同期する必要があるトラステッドホストに対しても、ロールオーバーを有効にする必要があります。`RolloverEnabled()` をコールして、トラステッドホストオブジェクト用のロールオーバーを有効にできます。

ファイルへのドメインおよびレルム レポートの書き込み

以下のスクリプトは、テキスト ファイルにすべてのポリシー ストア ドメインおよびトップレベルのレルムの名前を書き込みます。

```
use Netegrity::PolicyMgtAPI;

$destFile="DomainsRealms.txt";
open(DEST,">".$destFile) || die "Open file error: $!";
print DEST "Domains and Domain Realms for Acme North Site\n";
print DEST "Printed " . scalar(localtime)."\n";

$policyapi = Netegrity::PolicyMgtAPI->New();
$session = $policyapi->CreateSession("adminid", "adminpwd");

@domains=$session->GetAllDomains();
foreach $domain(@domains) {
    print DEST "%n%nDomain " . $domain->Name() . ":";
    @realms=$domain->GetAllRealms();
    foreach $realm(@realms) {
        print DEST "%n  Realm " . $realm->Name();
    }
}

print "%nDomain and realm report written to " . $destFile."%n";
```

認証および許可イベント処理の無効化

以下のスクリプトは、ポリシー ストアの各レルムのルールをすべて確認します。レルムのルールが認証イベントによってトリガされない場合、スクリプトは認証イベント処理がレルムで有効かどうかを確認します。その場合、スクリプトはレルム用の認証イベント処理を無効にします。スクリプトは、許可イベントに対しても同じ確認を実行します。

例を簡略化するために、ドメイン内のレルムには子レルムがないものとします。

```
use Netegrity::PolicyMgtAPI;

$policyapi = Netegrity::PolicyMgtAPI->New();
$session = $policyapi->CreateSession("adminid", "adminpwd");

$auAction=0; # Initialize flag for authentication actions
$azAction=0; # Initialize flag for authorization actions
$auChange="";# Realms with a changed auth event processing property
$azChange="";# Realms with a changed az event processing property

@domains=$session->GetAllDomains();
foreach $domain(@domains) {
    @realms=$domain->GetAllRealms();
    foreach $realm(@realms) {
        @rules=$realm->GetAllRules();
        foreach $rule(@rules) {
            if ($rule->Action()=~ /OnAuth./ ) {
                $auAction=1;
            }

            if ($rule->Action()=~ /OnAccess./ ) {
                $azAction=1;
            }
        }
        if($auAction==0) {
            if($realm->ProcessAuEvents()==1) {
                $realm->ProcessAuEvents(0);
                $auChange=$auChange.$domain->Name().": ";
                $auChange=$auChange.$realm->Name()."%n";
            }
        }
        else { $auAction=0; }
        if($azAction==0) {
            if($realm->ProcessAzEvents()==1) {
                $realm->ProcessAzEvents(0);
                $azChange=$azChange.$domain->Name().": ";
                $azChange=$azChange.$realm->Name()."%n";
            }
        }
    }
}
```

```
}  
    else { $azAction=0; }  
}  
}  
  
if ($auChange ne "") {  
    print "Stopped auth event processing for these realms:%n";  
    print $auChange . "%n%n";  
}  
if ($auChange ne "") {  
    print "Stopped az event processing for these realms:%n";  
    print $azChange . "%n";  
}
```

ポリシー サーバ負荷分散の管理

サービス中断を防ぎやすくするため、**SiteMinder** にはフェールオーバー機能が含まれます。主ポリシー サーバに障害が起きて、フェールオーバーが有効化されていると、バックアップ ポリシー サーバがポリシー操作を引き継ぎます。**SiteMinder v6.0** 以降では、フェールオーバーをポリシー サーバ間だけでなく、ポリシー サーバのグループまたはクラスタ間に適用することができます。

また、クラスタ機能は、クラスタ内のサーバ間に動的な負荷分散機能を提供して、サーバのパフォーマンスを改善します。動的な負荷分散では、各サーバのパフォーマンス能力に応じてポリシー操作がクラスタ内の使用可能サーバ間で自動的に分散されます。

ホスト設定は、どのようにホスト設定オブジェクトを定義するかに応じて、以下のように 1 つ以上のポリシー サーバまたはポリシー サーバの 1 つ以上のクラスタと関連付けることができます。

■ クラスタ化サーバ

- ホスト設定で作成するクラスタごとに **AddCluster()** をコールします。
- **AddCluster()** から返される **PolicyMgtCluster** オブジェクトで、**AddServer()** をコールしてクラスタにサーバを適用します。

動作：複数のクラスタが定義される場合、フェールオーバーがサーバのクラス間で行われます。また、クラスタ内のサーバへの要求は、**Agent API v6.0** に導入された、機能強化パフォーマンスベース負荷分散テクニックにしたがって送信されます。

- **非クラスタ化サーバ**

- ホスト設定で作成する非クラスタ化サーバごとに `AddServer()` をコールします。

動作：動作は v5.x インストールの場合と同じです、つまり、ホスト設定と関連付けられたサーバ間でフェールオーバーを有効にするか

(`EnableFailover()` を 1 に設定)、サーバ間でラウンドロビン動作を有効にできます (`EnableFailover()` を 0 に設定)。

ラウンドロビン動作が有効なとき、エージェント API に導入された、機能強化パフォーマンスベース負荷分散テクニックが使用されます。

注：ホスト設定内でクラスタ化されたサーバとクラスタ化されていないサーバを混在させることはできません。

クラスタの設定

クラスタはホスト設定オブジェクトに保存されます。クラスタ フェールオーバーは、`PolicyMgtHostConfig` で設定された以下の設定値に従って行われます。

- **フェールオーバーしきい値**。ポリシー サーバ要求に対して利用可能である必要があるクラスタ内のサーバの最小の割合 (%)。利用可能なサーバの数がしきい値以下になると、次のクラスタへのフェールオーバーが行われます。

フェールオーバーしきい値パーセンテージはホスト設定オブジェクトと関連付けられたすべてのクラスタに適用されます。

パーセンテージが特定のクラスタ内で表すサーバの数を判定するには、しきい値パーセンテージにクラスタ内のサーバ数を乗算して、次の最も高い整数まで丸めます。例：

- 5 つのサーバのクラスタの 60 パーセント フェールオーバーしきい値では、クラスタ内の利用可能なサーバの数が 3 を下回ると、次のクラスタへのフェールオーバーが行われます。
- 同じクラスタの 61 パーセント フェールオーバーしきい値では、クラスタ内の利用可能なサーバの数が 4 を下回ると、フェールオーバーが行われます。

次のメソッドによって設定。 `FailoverThreshold()`。

- **サーバタイムアウト。** エージェントがサーバからのレスポンスを待機する最大時間。待機時間がサーバタイムアウト値を超える場合、サーバは非アクティブであると見なされ、次のサーバへのフェールオーバーが行われます。

サーバタイムアウトがクラスタ内で発生し、タイムアウトによってクラスタのフェールオーバーしきい値を超える場合は、次のクラスタへのフェールオーバーが行われます。

次のメソッドによって設定。 `RequestTimeout()`。

- **クラスタ フェールオーバーのシーケンス。** クラスタ フェールオーバーが発生するとき、`SiteMinder` は後続のポリシー サーバ要求をクラスタ シーケンス内の次のクラスタに送信します。クラスタ シーケンスはクラスタ配列内のクラスタ オブジェクトの順序によって決定されます。

次のメソッドによってクラスタを追加: `AddCluster()`。追加された新しいクラスタは、クラスタ配列の終わりに追加されます。

次のメソッドによってクラスタ配列を取得。 `GetAllClusters()`。

ホスト設定オブジェクトにクラスタを追加する順序によって、フェールオーバー シーケンスが決定されます。追加する最初のクラスタ（つまりクラスタ配列内の最初のクラスタ）は、*プライマリ クラスタ*です。これは、クラスタ内の利用可能なサーバの数がフェールオーバーしきい値未満にならない限り使用されるクラスタです。プライマリ クラスタに使用できる十分なサーバがないと、次のクラスタへのフェールオーバーが行われます。つまり、ホスト設定オブジェクトに追加された 2 番目のクラスタが対象になります。そのクラスタも失敗する場合、ホスト設定オブジェクトに追加された 3 番目のクラスタへのフェールオーバーが行われます（以後同じ要領）。

クラスタがすべて失敗するとき

利用可能なサーバの数がホスト設定内のすべてのクラスタでフェールオーバーしきい値未満になった場合、ポリシー操作は停止しません。要求は、少なくとも 1 つの利用可能なサーバがあるクラスタ シーケンスの最初のクラスタに送信されます。

たとえば、ホスト設定に 2 つのクラスタがあると想定します。3 つのサーバが含まれる C1 と 5 つのサーバが含まれる C2 です。ホスト設定用のフェールオーバーしきい値は 60 パーセントに設定されます。以下の表は、各クラスタ内で利用可能である必要があるサーバの最小数を示しています。

クラスタ	クラスタ内のサーバ数	パーセンテージ フェールオーバーしきい値。	数値フェールオーバーしきい値 (最小利用可能サーバ数)
C1	3	60	1
C2	5	60	3

利用可能なサーバの数が各クラスタ内でしきい値未満になり、その結果、C1 には利用可能なサーバがなく、C2 に 2 つのみ利用可能サーバがあるという状態になった場合、着信する次の要求は最良のレスポンス時間を持つ C2 サーバに送信されます。少なくとも 3 つの C1 サーバのうちの 2 つが修復された後は、後続の要求が利用可能な C1 サーバ間で負荷分散されます。

エージェント API v6 は、エージェント API v5 と下位互換性があり、v5/v6 エージェントおよび v5/v6 エージェント API との完全な相互操作性が実現されます。

例：ホスト設定の作成

以下のコードは、ホスト設定オブジェクトを作成し、多くのクラスタおよびサーバを追加します。

```
use Netegrity::PolicyMgtAPI;

# Initialize the Policy Management API and create a Host Config object.
$pmgtapi = Netegrity::PolicyMgtAPI->New();
$session = $pmgtapi->CreateSession("SiteMinder", "password");
$hostconf = $session->CreateHostConfig("host", "description",
                                     false, 2, 2, 1, 30);
```

```
# Add two non-cluster servers. The Az, Auth and Acct ports are
# specified.
$hostconf->AddServer("1.1.1.1", 44443, 44442, 44441);
$hostconf->AddServer("2.2.2.2", 44443, 44442, 44441);

# Add two clusters with two servers in each cluster. One Policy
# Server port number is specified.
$clusterconf1 = $hostconf->AddCluster();
$clusterconf1->AddServer("1.1.1.1", 44443);
$clusterconf1->AddServer("2.2.2.2", 44443);

$clusterconf2 = $hostconf->AddCluster();
$clusterconf2->AddServer("3.3.3.3", 44443);
$clusterconf2->AddServer("4.4.4.4", 44443);

# Print configuration of all non-cluster servers in the Host
# Config object
@servers = $hostconf->GetAllServers();
foreach $server (@servers) {
    $address = $server->GetServerAddress();
    @ports = $server->GetServerPort();
    print("Server: $address,@ports[0],@ports[1],@ports[2]¥n");
}

# Print all cluster servers
@clusters = $hostconf->GetAllClusters();
foreach $cluster (@clusters) {
    $num++;
    foreach $server ($cluster->GetAllServers()) {
        $address = $server->GetServerAddress();
        $port = $server->GetServerPort();
        print("Cluster $num Server: $address,$port¥n");
    }
}

# Remove all clusters and non-cluster servers from host configuration
$hostconf->RemoveAllClusters();
$hostconf->RemoveAllServers();
```

第 9 章: コマンドライン インターフェース制限

この付録では、コマンドライン インターフェースでは実行できないが、管理 UI で実行できる操作の概要について説明します。

コマンドライン インターフェースでは以下を実行できません。

- 解決済みまたは未解決のデータ リストに関してカスタム エージェント サポートを提供する。この機能は **SiteMinder v5.x** で導入されました。
- 特定のドメインの管理者を作成します。Perl スクリプト作成インターフェースで作成する管理者にはシステム全体の範囲があります。
- 永続セッション機能用に設定されたレルムを作成する。この機能は **SiteMinder v5.x** で導入されました。
- 別の子レルム下で子レルムを作成する。ドメイン下のレルムおよびレルム下の **1** レベルの子レルムを作成できます。
- ユーザ属性タイプまたは DN 属性タイプのレスポンス属性を作成する。スタティック レスポンス属性は **CreateAttribute** メソッドへのコールで作成でき、アクティブ レスポンスは **CreateActiveAttribute** メソッドへのコールで作成できます。
- ポリシーがいつアクティブかを指定する 1 週間の時間グリッド内の期間を定義する。コマンドライン インターフェースによってポリシーを作成すると、ポリシーはデフォルトにより週全体にわたってアクティブになります。

第 10 章: CLI プロパティリファレンス

SAML 2.0 プロパティ

このセクションには、Perl ポリシー管理 API でサポートされる SAML 2.0 メタデータ プロパティのアルファベット順に並べられた参照が含まれます。

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、SAML アサーション内の情報に基づいてプリンシパルを透過的に検証します。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。

参照メモ

- すべてのプロパティ値は文字列として提供されます。
- 特に指定されない限り、プロパティには以下の最大長があります。
 - 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_AFFILIATION

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。 **IdentityProvider** は、アフィリエーションにわたってユーザ特定プロパティを共有します。

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。

アイデンティティ プロバイダはその関連の **SAML 2.0** 認証方式によってアフィリエーションに割り当てられます。

SAML 2.0 アフィリエーションの詳細については、 **CreateSAMLAffiliation** メソッドの説明を参照してください。

SAML_AUDIENCE

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

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**必須**

いいえ

デフォルト

0

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。通常の実行時は、このプロパティを **0** に設
定します（署名処理を有効にする）。

有効な値： **0** (false) および **1** (true)。

SAML_DSIG_ALGO**必須**

いいえ

デフォルト

1

説明

以下のいずれかの値を持った **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

2 = RSAwithSHA256

SAML_DSIG_VERINFO_ISSUER_DN**必須**

SAML 2.0 認証方式の場合

SAML_DISABLE_SIGNATURE_PROCESSING が **0** で、以下のいずれかまたは
両方が **1** の場合にのみ必要です。

- SAML_SLO_REDIRECT_BINDING
- SAML_ENABLE_SSO_POST_BINDING

サービス プロバイダの場合

SAML_DISABLE_SIGNATURE_PROCESSING が 0 で、以下のいずれかまたは両方が 1 の場合にのみ必要です。

- SAML_SLO_REDIRECT_BINDING
- SAML_SP_REQUIRE_SIGNED_AUTHNREQUESTS

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

必須

SAML 2.0 認証方式の場合

SAML_DISABLE_SIGNATURE_PROCESSING が 0 で、以下のいずれかまたは両方が 1 の場合にのみ必要です。

- SAML_SLO_REDIRECT_BINDING
- SAML_ENABLE_SSO_POST_BINDING

サービス プロバイダの場合

SAML_DISABLE_SIGNATURE_PROCESSING が 0 で、以下のいずれかまたは両方が 1 の場合にのみ必要です。

- SAML_SLO_REDIRECT_BINDING
- SAML_SP_REQUIRE_SIGNED_AUTHNREQUESTS

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**必須**

いいえ

デフォルト

0

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。

有効な値： 0 (false) および 1 (true)。

SAML_ENABLE_SSO_POST_BINDING**必須**

いいえ

デフォルト

0

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。

有効な値： 0 (false) および 1 (true)。

SAML_DSIG_VERINFO_ISSUER_DN および
SAML_DSIG_VERINFO_SERIAL_NUMBER も参照してください。

SAML_ENABLED**必須**

いいえ

デフォルト

1

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。

有効な値： 0 (false) および 1 (true)。

SAML_IDP_AD_SEARCH_SPEC**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**必須**

SAML_ENABLE_SSO_ARTIFACT_BINDING が 1 の場合は、はい

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**必須**

いいえ

デフォルト

0

説明

バック チャ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. Basic** - 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. Client Cert** - キーストア内の証明書を検索す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**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**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**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

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] でのみ使用されます。

SAML_IDP_PLUGIN_CLASS**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: 特定が行われるユーザ ディレクトリ (たとえば LDAP ディレクトリ用の `SAML_IDP_LDAP_SEARCH_SPEC`) 用に検索指定情報が提供されない場合、プラグインはこの段階でコールされません。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**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**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。認証方式に渡される 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**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。認証方式に渡される 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**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。認証方式に渡される 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**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**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**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**必須**

いいえ

デフォルト

0

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 1 で、認証方式に暗号化されていないアサーションが渡された場合、アサーションは認証できません。

有効な値： 0 (false) および 1 (true)。

SAML_IDP_REQUIRE_ENCRYPTED_NAMEID

必須

いいえ

デフォルト

0

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 1 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。

有効な値： 0 (false) および 1 (true)。

SAML_IDP_SAMLREQ_ATTRIBUTE_SERVICE

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

必須

はい

デフォルト

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。

有効な値： 0 (false) および 1 (true)。

SAML_IDP_SAMLREQ_GET_ALL_ATTRIBUTES**必須**

いいえ

デフォルト

0

説明

属性認証機関に送信されるクエリに属性が含まれるべきかどうかを示します。これは、属性認証機関で定義されたすべての属性を返すための簡略表記です。

SAML_IDP_SAMLREQ_NAMEID_ALLOW_NESTED**必須**

いいえ

デフォルト

0

説明

名前識別子用に DN 属性を選択するとき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。

有効な値： 0 (false) および 1 (true)。

SAML_IDP_SAMLREQ_NAMEID_ATTR_NAME**必須**

NameIdType が 1 または 2 に設定されているときは、はい

デフォルト

なし

説明

NameIdType が 1 または 2 に設定されるときに識別子名を保持する属性名（ユーザまたは DN）。

SAML_IDP_SAMLREQ_NAMEID_DN_SPEC

必須

NamedIdType が 2 に設定されているときは、はい

デフォルト

なし

説明

NamedIdType が 2 に設定されているときに使用する DN 指定情報。

SAML_IDP_SAMLREQ_NAMEID_FORMAT

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

必須

NamedIdType が 0 に設定されているときは、はい

デフォルト

なし

説明

NamedIdType が 0 に設定されているときに使用するスタティック テキスト。

SAML_IDP_SAMLREQ_NAMEID_TYPE**必須**

いいえ

デフォルト

1 (ユーザ属性)

説明

名前識別子のタイプを表します。

有効な値： 0 (静的テキスト)、1 (ユーザ属性) および 2 (DN 属性)。

SAML_IDP_SAMLREQ_REQUIRE_SIGNED_ASSERTION**必須**

いいえ

デフォルト

0

説明

<AttributeQuery> に応じて返されたアサーションに署名が必要かどうかを示します。

有効な値： 0 (false) および 1 (true)。

SAML_IDP_SAMLREQ_SIGN_ATTRIBUTE_QUERY**必須**

いいえ

デフォルト

0

説明

属性クエリに署名が必要かどうかを示します。

有効な値： 0 (false) および 1 (true)。

SAML_IDP_SIGN_AUTHNREQUESTS

必須

いいえ

デフォルト

0

説明

認証要求が署名されるかどうかを指定します。

有効な値： 0 (false) および 1 (true)。

SAML_IDP_SPID

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

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**必須**

はい

デフォルト

なし

説明

以下のような、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。

<http://mysite.netegrity.com/affwebservices/public/saml2sso>

SAML_IDP_SSO_ENFORCE_SINGLE_USE_POLICY**必須**

いいえ

デフォルト

1

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 1（デフォルト）に設定することで、SAML POST 固有の処理ルールに従って、2 番目のセッションを確立するためにサービス プロバイダ サイトにアサーションを「再生」できないようにします。

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。

有効な値： 0（false） および 1（true）。

SAML_IDP_SSO_REDIRECT_MODE**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とき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。

レスポンスでは、ユーザ属性、DN 属性、スタティック テキスト、またはポリシー サーバからのカスタマイズされたアクティブ レスポンスを、エージェントがログインまたは許可要求を発行した後に SiteMinder エージェントに渡します。これらのレスポンス属性の詳細については、`CreateAttribute()` を参照してください。

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 データなし - 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。

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**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**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**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**必須**

はい

デフォルト

なし

説明

アフィリエーションの 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ーション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、以下のようになります。

- 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ーション 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ーション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ーション 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ーション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ーション 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ーションに属することを確認します。

アフィリエーション 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

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

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

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

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

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**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**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的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**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を表す文字列である必要があります。

SAML_SLO_REDIRECT_BINDING**必須**

いいえ

デフォルト

0

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。

有効な値： 0 (false) および 1 (true)。

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**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**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**必須**

SAML_SLO_REDIRECT_BINDING が 1 の場合は、はい

デフォルト

なし

説明

HTTP-リダイレクト バインディングでは、シングル ログアウト 要求が送信されるアイデンティティ プロバイダの URL。

SAML_SLO_SERVICE_VALIDITY_DURATION**必須**

いいえ

デフォルト

60 (値が指定されず、SAML_SLO_REDIRECT_BINDING が 1 の場合に適用)。

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を表す文字列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

必須

いいえ

デフォルト

FORM（値が指定されず、SAML_ENABLE_SSO_ARTIFACT_BINDING 1 の場合に適用）。

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

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、以下のようになります。

<http://mysite.netegrity.com/affwebservices/public/saml2assertionconsumer>

SAML_SP_AUTHENTICATION_LEVEL**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_ATTRSVC_AD_SEARCH_SPEC**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**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

必須

いいえ

デフォルト

0

説明

属性認証機関が有効かどうかを示します。

有効な値： 0 (false) および 1 (true)。

SAML_SP_ATTRSVC_LDAP_SEARCH_SPEC

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

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**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指定します。

SAML_SP_ATTRSVC_SIGN_ASSERTION**必須**

いいえ

デフォルト

0

説明

SAML アサーションに署名が必要かどうかを示します。

有効な値： 0 (false) および 1 (true)。

SAML_SP_ATTRSVC_SIGN_RESPONSE**必須**

いいえ

デフォルト

0

説明

SAML レスポンスに署名が必要かどうかを示します。

有効な値： 0 (false) および 1 (true)。

SAML_SP_ATTRSVC_VALIDITY_DURATION

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。

SAML_SP_ATTRSVC_WINNT_SEARCH_SPEC

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_URL

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

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

必須

SAML_SP_ENABLE_IPD が 1 の場合は、はい

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

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

必須

いいえ

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

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 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。

有効な値： 0 (false) および 1 (true)。

SAML_SP_ENCRYPT_ASSERTION**必須**

いいえ

デフォルト

0

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。

有効な値： 0 (false) および 1 (true)。

SAML_SP_ENCRYPT_BLOCK_ALGO**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**必須**

以下のいずれかの状況では、はい。

以下のいずれかが 1 の場合

- **SAML_SP_ENCRYPT_ID**
- **SAML_SP_ENCRYPT_ASSERTION**

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。

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、インラインで指定されない場合に、キーストア内のサービス プロバイダの証明書を検索するために SAML_SP_ENCRYPT_CERT_SERIAL_NUMBER と共に使用されます。

SAML_SP_ENCRYPT_CERT_SERIAL_NUMBER

必須

以下のいずれかの状況では、はい。

以下のいずれかが 1 の場合

- SAML_SP_ENCRYPT_ID
- SAML_SP_ENCRYPT_ASSERTION

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。

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、インラインで指定されない場合に、キーストア内のサービス プロバイダの証明書を検索するために SAML_SP_ENCRYPT_CERT_ISSUER_DN と共に使用されます。

SAML_SP_ENCRYPT_ID**必須**

いいえ

デフォルト

0

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。

有効な値： 0 (false) および 1 (true)。

SAML_SP_ENCRYPT_KEY_ALGO**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**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）。

このプロパティに時間を割り当てやすくするには、**Perl time()** メソッドを使用します。時間値は文字列として保存されます。例：

```
$SAML_SP_ENDTIME=SAML_SP_ENDTIME;  
$time=time() + 20;  
$serviceProvider->Property($SAML_SP_ENDTIME,"$time");
```

アサーションの生成の時間制限を定義するには、このプロパティを **SAML_SP_STARTTIME** と共に使用します。

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**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**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**必須**

いいえ

デフォルト

0

説明

アイデンティティ プロバイダが受信 AuthnRequest メッセージ内の "RequestedAuthnContext" を無視するか（値 1）、または無視しないか（値 0）を指定します。

SAML_SP_IPD_SERVICE_URL**必須**

SAML_SP_ENABLE_IPD が 1 の場合は、はい

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**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

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。

SAML_SP_NAMEID_FORMAT

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 情報を使用します。

SAML_SP_NAMEID_STATIC

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

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

必須

いいえ

デフォルト

False

説明

アサーションに、アサーションを 1 回だけ使用する必要があるかどうかを示すエレメントが含まれるかどうかを指定します。

SAML_SP_PASSWORD

必須

SAML_ENABLE_SSO_ARTIFACT_BINDING が 1 の場合は、はい

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

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 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 1 の場合にのみ適用します。

有効な値： 0 (false) および 1 (true)。

SAML_SP_PLUGIN_CLASS**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、「SiteMinder Java API ドキュメント」を参照してください。

SAML_SP_PLUGIN_PARAMS**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**必須**

いいえ

デフォルト

0

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。

有効な値： 0 (false) および 1 (true)。

SAML_SP_REUSE_SESSION_INDEX

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_SESSION_NOTORAFTER_TYPE

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** パラメータのカスタム値を指定できます。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_SP_STARTTIME

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。

このプロパティに時間を割り当てやすくするには、Perl `time()` メソッドを使用します。時間値は文字列として保存されます。例：

```
$SAML_SP_STARTTIME=SAML_SP_STARTTIME;  
$time=time() + 10;  
$serviceProvider->Property($SAML_SP_STARTTIME,"$time");
```

アサーションの生成の時間制限を定義するには、このプロパティを **SAML_SP_ENDTIME** と共に使用します。

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

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を表す文字列である必要があります。

SAML_SKEWTIME も参照してください。

SAML_SSOECPPROFILE**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。

有効な値： 0 (false) および 1 (true)。

SAML2_CUSTOM_ENABLE_INVALID_REQUEST_URL**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**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**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**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**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**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**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**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

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。

WSFED プロパティ

このセクションでは、**WS**-フェデレーション メタデータ プロパティごとの名前、タイプおよび説明を提供します。

以下のプロパティは、リソース パートナーかアカウント パートナーまたはその両方を定義するために使用します。

WSFED_AP_ADD_SEARCH_SPEC

必須

いいえ

タイプ

文字列

説明

1 個の AD ディレクトリの検索指定情報。

WSFED_AP_CUSTOM_SEARCH_SPEC

必須

いいえ

タイプ

文字列

説明

カスタム ディレクトリの検索指定情報です。

WSFED_AP_FAILURE_REDIRECT_MODE

必須

いいえ

タイプ

0/1

説明

- 0 - フェデレーション メッセージを渡さない (デフォルト) Http 302 リダイレクト。
- 1 - Http フォーム ポスト リダイレクト。

WSFED_AP_FAILURE_REDIRECT_URL

必須

いいえ

タイプ

文字列

説明

アサーション処理が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。

WSFED_APID

必須

はい

タイプ

文字列

説明

アカウント パートナーの ID。

WSFED_AP_INVALID_REDIRECT_MODE

必須

いいえ

タイプ

0/1

説明

- 0 - フェデレーション メッセージを渡さない (デフォルト) Http 302 リダイレクト。
- 1 - Http フォーム ポスト リダイレクト。

WSFED_AP_INVALID_REDIRECT_URL

必須

いいえ

タイプ

文字列

説明

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。

WSFED_AP_LDAP_SEARCH_SPEC

必須

いいえ

タイプ

文字列

説明

LDAP ディレクトリの検索指定情報。

WSFED_AP_ODBC_SEARCH_SPEC

必須

いいえ

タイプ

文字列

説明

1 個の ODBC ディレクトリの検索指定情報。

WSFED_AP_PLUGIN_CLASS

必須

いいえ

タイプ

文字列

説明

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。

WSFED_AP_PLUGIN_PARAMS

必須

いいえ

タイプ

文字列

説明

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。

WSFED_AP_SIGNOUT_URL

必須

いいえ

タイプ

文字列

説明

アカウント パートナーのサインアウト URL。WSFED_AP_SLO_ENABLED が true の場合は、このプロパティが必要です。

WSFED_AP_SLO_ENABLED

必須

いいえ

タイプ

ブール値

説明

サインアウトがアカウント パートナーに有効かどうかを示します。アカウント パートナー作成時に指定されなかった場合、デフォルトでは、有効化されません。

WSFED_AP_SSO_DEFAULT_SERVICE

必須

いいえ

タイプ

文字列

説明

シングルサインオン サービスのデフォルトの場所。

WSFED_AP_SSO_REDIRECT_MODE

必須

いいえ

タイプ

整数

説明

アサーション属性のリダイレクト モード。有効な値は以下のとおりです。

- 0. 302 データなし - レスポンス属性は渡されません（デフォルト）。
- 1. 302 Cookie データなし - 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。

WSFED_AP_SSO_TARGET

必須

いいえ

タイプ

文字列

説明

送信先サイトのターゲット リソース。

WSFED_AP_USER_NOT_FOUND_REDIRECT_MODE

必須

いいえ

タイプ

0/1

説明

- 0 - フェデレーション メッセージを渡さない (デフォルト) Http 302 リダイレクト。
- 1 - Http フォーム ポスト リダイレクト。

WSFED_AP_USER_NOT_FOUND_REDIRECT_URL

必須

いいえ

タイプ

文字列

説明

以下のいずれかの場合に使用されるオプションのリダイレクト URL が含まれています。

- 認証方式で、指定の設定済みクエリ文字列に従って、フェデレーション メッセージからログイン ID を取得できないとき。
- 認証方式で、指定の設定済みユーザ ストア検索文字列に従って、指定されたユーザ ディレクトリからユーザを検出できないとき。

WSFED_AP_WINNT_SEARCH_SPEC

必須

いいえ

タイプ

文字列

説明

1 個の WinNT ディレクトリの検索指定情報。

WSFED_AP_XPATH

必須

いいえ

タイプ

文字列

説明

プリンシパルの特定を行うための XPath クエリ。

WSFED_DESCRIPTION

必須

いいえ

タイプ

文字列

説明

プロバイダの簡単な説明。

WSFED_DISABLE_SIGNATURE_PROCESSING

必須

いいえ

タイプ

ブール値

説明

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役立ちます。アカウントパートナーが稼働中のときは、この設定を **false** に設定して、セキュリティの含意を回避する必要があります。デフォルト値はゼロです。

WSFED_DSIG_VERINFO_ALIAS

必須

いいえ

タイプ

文字列

説明

インラインで指定されない場合にキー ストア内でプロバイダの証明書を検索します。

WSFED_ENABLED

必須

いいえ

タイプ

ブール

説明

リソース パートナーが有効かどうかを示します。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、プロパティ収集に対して物理的に保存されませんが、基礎となるポリシーを有効にするために使用されます。

WSFED_ENFORCE_SINGLE_USE_POLICY

必須

いいえ

タイプ

ブール値

説明

1 の値に設定されている場合は、WS-フェデレーション アサーション用の単一使用ポリシーが適用されます。0 の値に設定されている場合は、アサーション用の単一使用ポリシーは適用されません。デフォルトは 1 です。

WSFED_KEY_APID

必須

はい

タイプ

文字列

説明

アカウント パートナーの識別子。これは、1024 文字未満の URI である必要があります また、これはアカウント パートナーと関連付けられたプロパティを検索できるキーです。

WSFED_KEY_RPID

必須

はい

タイプ

文字列

説明

リソース パートナーの ID。これは、1024 文字未満の URI である必要があります また、これはリソース パートナーと関連付けられたプロパティを検索できるキーです。

WSFED_MAJOR_VERSION

必須

いいえ

タイプ

整数

説明

このプロバイダによってサポートされている WS-フェデレーション
プロトコルのバージョン。このプロパティの値は **1** である必要があり
ます。

WSFED_MINOR_VERSION

必須

いいえ

タイプ

整数

説明

このプロバイダによってサポートされている WS-フェデレーション
プロトコルのバージョン。このプロパティの値は **0** に設定する必要が
あります。

WSFED_NAME

必須

はい

タイプ

文字列

説明

プロバイダの名前。

WSFED_RPID

必須

はい

タイプ

文字列

説明

リソース パートナーの識別子。

WSFED_RP_ASSERTION_CONSUMER_DEFAULT_URL

必須

はい

タイプ

文字列

説明

デフォルトのアサーション コンシューマの URL。

WSFED_RP_AUTHENTICATION_LEVEL

必須

いいえ

タイプ

整数

説明

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 パートナーが作成されるときに指定されない場合、デフォルトは 5 です。

WSFED_RP_AUTHENTICATION_METHOD

必須

いいえ

タイプ

文字列

説明

アサーションで使用する認証方式。これは、通常の場合 **WS-フェデレーション** 指定情報に基づくいずれかの認証方式値になります。

WSFED_RP_AUTHENTICATION_URL

必須

はい

タイプ

文字列

説明

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。

WSFED_RP_DOMAIN

必須

はい

タイプ

OID

説明

このプロバイダが定義されるリソース パートナー ドメイン。

WSFED_RP_ENDTIME

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）。

このプロパティに時間を割り当てやすくするには、**Perl time()** メソッドを使用します。時間値は文字列として保存されます。例：

```
$WSFED_RP_ENDTIME=$WSFED_RP_ENDTIME;  
$time=time() + 20;  
$ResourcePartner->Property($WSFED_RP_ENDTIME,"$time");
```

アサーションの生成の時間制限を定義するには、このプロパティを **WSFED_RP_STARTTIME** と共に使用します。

時間制限をすぐに終了するには、**WSFED_RP_ENDTIME** を **0** に設定します。

WSFED_RP_NAMEID_ALLOWED_NESTED

必須

いいえ

タイプ

ブール値

説明

名前識別子用に **DN** 属性を選択するとき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。デフォルトは **0** です。

WSFED_RP_NAMEID_ATTR_NAME

必須

いいえ

タイプ

文字列

説明

NameIdType が 1 または 2 に割り当てられるとき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 に設定される場合は、このプロパティに値が必要です。

WSFED_RP_NAMEID_DN_SPEC

必須

いいえ

タイプ

文字列

説明

NameIdType が 2 に割り当てられるときに使用する DN 指定情報。**NameIdType** が 2 に割り当てられる場合は、このプロパティに値が必要です。

WSFED_RP_NAMEID_FORMAT

必須

いいえ

タイプ

文字列

説明

WS-フェデレーション名識別子の URI。

WSFED_RP_NAMEID_TYPE

必須

いいえ

タイプ

整数

説明

以下のいずれかのタイプの名前識別子。

- Static Text (0)
- ユーザ属性 (1、デフォルト)
- DN Attribute (2)

WSFED_RP_NAMEID_STATIC

必須

いいえ

タイプ

文字列

説明

NameIdType が 0 に割り当てられるときに名前識別子として使用するスタティック テキスト。このプロパティに対して指定された値がなく、**NameIdType** が 0 に割り当てられる場合は、エラーが返されます。

WSFED_RP_PLUGIN_CLASS

必須

いいえ

タイプ

文字列

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。

WSFED_RP_PLUGIN_PARAMS

必須

いいえ

タイプ

文字列

説明

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。

WSFED_RP_SIGNOUT_CLEANUP_URL

必須

いいえ

タイプ

文字列

説明

リソース パートナーのサインアウト クリーンアップ URL。サインアウトが有効な場合は、このプロパティが必要です。

WSFED_RP_SIGNOUT_CONFIRM_URL

必須

いいえ

タイプ

文字列

説明

サインアウトが完了し、要求に返信クエリ パラメータがない場合にユーザがリダイレクトされる URL。このプロパティがリソース パートナー オブジェクトの一部でも、これはアカウント パートナーでのサインアウトが完了するときにユーザがリダイレクトされる URL です。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 Signout Confirm URL が使用されます。デフォルトは無効化です。

WSFED_RP_SLO_ENABLED

必須

いいえ

タイプ

ブール値

説明

サインアウトがリソース パートナーに有効かどうかを示します。

WSFED_RP_STARTTIME

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。

このプロパティに時間を割り当てやすくするには、Perl `time()` メソッドを使用します。時間値は文字列として保存されます。例：

```
$WSFED_RP__STARTTIME=WSFED_RP_STARTTIME;  
$time=time() + 10;  
$ResourcePartner->Property($WSFED_RP_STARTTIME, "$time");
```

アサーションの生成の時間制限を定義するには、このプロパティを **WSFED_RP_ENDTIME** と共に使用します。

時間制限をすぐに開始するには **WSFED_RP_STARTTIME** を 0 に設定します。

WSFED_RP_VALIDITY_DURATION

必須

いいえ

タイプ

整数

説明

生成されたアサーションが有効な秒数。リソース パートナー作成時に指定しない場合のデフォルトは 60 秒です。

WSFED_SAML_MAJOR_VERSION

必須

いいえ

タイプ

整数

説明

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。値は 1 です。

WSFED_SAML_MINOR_VERSION

必須

いいえ

タイプ

整数

説明

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。値は 1 です。

WSFED_SKEW_TIME

必須

いいえ

タイプ

文字列

説明

コンシューマとプロデューサ側間の秒単位のスキュー時間。この値はアサーションおよびサインアウト要求の有効期間を計算するために使用されます。デフォルト値は **30** です。