



Sun Java™ System

Portal Server Secure Remote Access 6 管理ガイド

2005Q1

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819-1202

Copyright © 2005 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

Sun Microsystems, Inc. は、この製品に含まれるテクノロジーに関する知的所有権を保持しています。特に限定されることなく、これらの知的所有権は <http://www.sun.com/patents> に記載されている 1 つ以上の米国特許および米国およびその他の国における 1 つ以上の追加特許または特許出願中のものが含まれている場合があります。

このソフトウェアは SUN MICROSYSTEMS, INC. の機密情報と企業秘密を含んでいます。SUN MICROSYSTEMS, INC. の書面による許諾を受けることなく、このソフトウェアを使用、開示、複製することは禁じられています。

U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

この配布には、第三者が開発したソフトウェアが含まれている可能性があります。

本製品の一部は、カリフォルニア大学からライセンスされている Berkeley BSD システムに基づいていることがあります。UNIX は、X/Open Company, Ltd が独占的にライセンスしている米国およびその他の国における登録商標です。

Sun、Sun Microsystems、Sun のロゴマーク、Java、Solaris、JDK、Java Naming and Directory Interface、JavaMail、JavaHelp、J2SE、iPlanet、Duke のロゴマーク、Java Coffee Cup のロゴ、Solaris のロゴ、SunTone 認定ロゴマークおよび Sun ONE ロゴマークは、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標もしくは登録商標です。

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャに基づくものです。

Legato および Legato のロゴマークは Legato Systems, Inc. の商標であり、Legato NetWorker は同社の商標または登録商標です。

Netscape Communications Corp のロゴマークは Netscape Communications Corporation の商標または登録商標です。

OPEN LOOK および Sun Graphical User Interface は、米国 Sun Microsystems 社が自社のユーザーおよびライセンス実施権者向けに開発しました。米国 Sun Microsystems 社は、コンピュータ産業用のビジュアルまたはグラフィカルユーザーインターフェースの概念の研究開発における米国 Xerox 社の先駆者としての成果を認めるものです。米国 Sun Microsystems 社は米国 Xerox 社から Xerox Graphical User Interface の非独占的ライセンスを取得しており、このライセンスは米国 Sun Microsystems 社のライセンス実施権者にも適用されます。

この製品は、米国の輸出規制に関する法規の適用および管理下にあり、また、米国以外の国の輸出および輸入規制に関する法規の制限を受ける場合があります。核、ミサイル、生物化学兵器もしくは原子力船に関連した使用またはかかる使用者への提供は、直接的にも間接的にも、禁止されています。このソフトウェアを、米国の輸出禁止国へ輸出または再輸出すること、および米国輸出制限対象リスト (輸出が禁止されている個人リスト、特別に指定された国籍者リストを含む) に指定された、法人、または団体に輸出または再輸出することは一切禁止されています。

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されない、明示的であるか黙示的であるかを問わない、なんらの保証も行われぬものとします。

目次

図目次	11
表目次	13
手順一覧	15
はじめに	19
対象読者	20
本書の構成	20
本書で使用する規則	22
表記上の規則	22
デフォルトのパスとファイル名	23
Linux の使用	24
Linux を使用する場合の制限	24
Solaris と Linux のパス名の比較	24
関連情報	25
このドキュメントセット内のマニュアル	25
その他の Portal Server ドキュメント	26
その他のサーバー関連ドキュメント	26
オンラインの Sun リソースへのアクセス	27
Sun 技術サポートへの問い合わせ	27
関連するサードパーティの Web サイト	27
Sun へのコメント	28
第 1 章 Portal Server Secure Remote Access について	29
SRA ソフトウェアの概要	29
オープンモード	30
セキュアモード	31

SRA サービス	33
ゲートウェイ	33
リライタ	33
NetFile	34
Netlet	34
プロキシレット	34
SRA 製品の管理	34
SRA の属性の設定	35
競合解決の設定	36
サポートされるアプリケーション	37
第 2 章 ゲートウェイ	39
ゲートウェイの概要	40
ゲートウェイプロファイルの作成	40
platform.conf ファイルの概要	42
ゲートウェイのインスタンスの作成	51
マルチホームゲートウェイのインスタンスの作成	52
同じ LDAP を使用するゲートウェイインスタンスの作成	53
chroot 環境でのゲートウェイの実行	54
chroot 環境でのゲートウェイの再起動	57
ゲートウェイの起動と停止	57
ゲートウェイの再起動	59
仮想ホストの指定	60
Access Manager へアクセスするプロキシの指定	60
Web プロキシの使用	61
自動プロキシ設定の使用	67
Netlet プロキシの使用	70
Netlet プロキシのインスタンスの作成	73
Netlet プロキシの有効化	74
Netlet プロキシの再起動	74
リライタプロキシの使用	75
リライタプロキシのインスタンスの作成	76
リライタプロキシの有効化	77
リライタプロキシの再起動	77
ゲートウェイでの逆プロキシの使用	78
クライアント情報の取得	79
認証連鎖の使用	81
ワイルドカード証明書の使用	82
ブラウザキャッシングの無効化	82
ゲートウェイサービスのユーザーインターフェースのカスタマイズ	83
連携管理の使用	84
連携管理の例	84
連携管理リソースの設定	85

第3章 プロキシレットとリライタ	91
プロキシレットの概要	92
プロキシレットを使用する利点	93
プロキシレットの設定	93
リライタの概要	93
文字セットのエンコーディング	94
リライタの使用例	94
URL スクレイパー	94
ゲートウェイ	95
ルールセットの記述	95
パブリックインタフェース (ルールセット DTD)	96
XML DTD の例	99
ルールの記述手順	100
ルールセットのガイドライン	101
ルールセットのルート要素の定義	102
再帰機能の使用	102
言語ベースのルールの定義 (ルールの定義)	103
HTML コンテンツのルール	103
JavaScript コンテンツのルール	110
XML コンテンツのルール	124
カスケードスタイルシートのルール	127
WML のルール	127
再帰機能の使用	127
ゲートウェイサービスのリライタの設定	128
基本タスク	128
高度なタスク	133
デバッグログを使用した障害追跡	137
リライタのデバッグレベルの設定	137
デバッグファイル名	138
サンプルの操作	140
HTML コンテンツのサンプル	141
JavaScript コンテンツのサンプル	150
XML 属性のサンプル	170
ケーススタディ	172
6.x と 3.0 のルールセットのマッピング	177
第4章 NetFile	179
NetFile の概要	179
サポートされるファイルアクセスプロトコル	180
NetFile のデバッグの有効化	182
NetFile のロギングの有効化	182

第 5 章 Netlet	183
Netlet の概要	183
Netlet のコンポーネント	184
Netlet の使用例	186
Netlet の操作	186
リモートホストからのアプレットのダウンロード	187
Netlet ルールの定義	187
ルールのタイプ	192
Netlet ルールの例	195
Netlet ルールの例	200
Netlet ロギングの有効化	205
デバッグロギングの有効化	206
Sun Ray 環境での Netlet の実行	206
新しい HTML ファイル	206
変更前の HTML ファイル	208
第 6 章 Netlet での PDC の使用	209
PDC 用の Netlet の設定	209
第 7 章 証明書	211
SSL 証明書の概要	212
証明書ファイル	212
証明書の信頼属性	214
CA の信頼属性	215
certadmin スクリプト	218
自己署名証明書の生成	219
証明書署名要求 (CSR) の生成	221
ルート CA 証明書の追加	223
証明書認証局から届いた SSL 証明書のインストール	224
CA への証明書の要求	224
CA から届いた証明書のインストール	225
証明書の削除	226
証明書の信頼属性の変更	228
ルート CA 証明書のリスト表示	230
すべての証明書のリスト表示	231
証明書の出力	232
第 8 章 URL アクセス制御の設定	235
拒否される URL リストの設定	236
許可される URL リストの設定	236
シングルサインオンの管理	237

第 9 章 ゲートウェイの設定	239
コアタブ	240
HTTP 接続と HTTPS 接続の有効化	241
リライタプロキシの有効化とリストの作成	242
Netlet の有効化	243
Netlet プロキシの有効化とリストの作成	244
プロキシレットの有効化	245
Cookie 管理の有効化	246
HTTP 基本認証の有効化	247
持続 HTTP 接続の有効化	248
持続接続 1 つあたりの最大要求数の指定	249
持続ソケット接続のタイムアウトの指定	249
回復時間に必要な正常なタイムアウトの指定	250
Cookie を転送する URL のリストの作成	250
最大接続キューの指定	252
ゲートウェイタイムアウトの指定	252
最大スレッドプールサイズの指定	253
キャッシュされたソケットのタイムアウトの指定	254
Portal Server のリストの作成	254
サーバーの再試行間隔の指定	255
外部サーバー Cookie の格納の有効化	256
URL からのセッションを取得するには	256
安全な Cookie としてマークする	257
プロキシタブ	258
Web プロキシ使用の有効化	258
Web プロキシを使用する URL のリストの作成	259
Web プロキシを使用しない URL のリストの作成	260
ドメインとサブドメインのプロキシのリストの作成	260
プロキシのパスワードリストの作成	261
自動プロキシ設定サポートの有効化	262
自動プロキシ設定ファイルの場所の指定	263
Web プロキシを通じた Netlet トンネリングの有効化	263
セキュリティタブ	264
非認証 URL のリストの作成	264
証明書が有効なゲートウェイホストのリストの作成	265
40 ビット暗号化接続の許可	266
SSL Version 2.0 の有効化	267
SSL 符号化選択の有効化	267
SSL Version 3.0 の有効化	268
Null 暗号化方式の有効化	269
信頼されている SSL ドメインのリストの作成	269
PDC (Personal Digital Certificate) 認証の設定	270
リライタタブ	273

すべての URL のリライトの有効化	273
URI とルールセットのマッピングリストの作成	274
パースする MMI タイプリストの作成	276
リライトしない URI のリストの作成	277
デフォルトドメインの指定	278
MIME 推測の有効化	278
パースする URI マッピングリストの作成	279
マスキングの有効化	280
マスキングのためのシード文字列の指定	280
マスクしない URI のリストの作成	281
ゲートウェイプロトコルと元の URI プロトコルの同一化	282
ロギングタブ	283
ロギングの有効化	283
Netlet ロギングの有効化	284
第 10 章 NetFile の設定	287
ホストタブ	288
OS の文字セットの指定	288
ホスト検出順序の指定	289
共通ホストのリストの設定	289
デフォルトドメインの指定	291
Microsoft Windows のドメイン / ワークグループの指定	292
デフォルトの WINS/DNS サーバーの指定	293
異なるタイプのホストへのアクセスの指定	293
「許可されるホスト」リストの設定	294
「拒否されるホスト」リストの設定	295
権限タブ	296
表示タブ	298
NetFile のウィンドウサイズの指定	298
NetFile ウィンドウの位置の指定	299
操作タブ	300
一時ファイルディレクトリの指定	300
ファイルアップロードサイズの制限の設定	301
検索ディレクトリ制限の指定	302
圧縮の指定	303
一般タブ	304
MIME タイプ設定ファイルの場所の指定	304
第 11 章 Netlet の設定	305
ユーザーへの Netlet サービスの割り当て	306
Netlet ルールの追加	307
既存の Netlet ルールの変更	308

Netlet ルールの削除	309
デフォルトの暗号化方式の指定	309
デフォルトループバックポートの割り当て	310
接続の再認証の有効化	311
接続の警告ポップアップを表示	311
「ポート警告ダイアログにチェックボックスを表示」の有効化	312
キーペアライブ間隔の設定	313
「Portal のログアウト時に Netlet を終了」オプションの設定	313
Netlet ルールへのアクセスの定義	314
Netlet ルールへのアクセスの拒否	315
ホストへのアクセスの許可	316
ホストへのアクセスの拒否	317
プロキシの設定	318
第 12 章 プロキシレットの設定	319
プロキシレットの設定	319
第 13 章 SSL アクセラレータの設定	321
概要	321
Sun Crypto Accelerator 1000	321
Sun Crypto Accelerator 1000 の有効化	322
Sun Crypto Accelerator 1000 の設定	322
Sun Crypto Accelerator 4000	325
Sun Crypto Accelerator 4000 の有効化	325
Sun Crypto Accelerator 4000 の設定	326
外部 SSL デバイスとプロキシアクセラレータ	328
外部 SSL デバイスアクセラレータの有効化	328
外部 SSL デバイスアクセラレータの設定	329
付録 A ログファイル	331
付録 B 設定属性	333
アクセスリストサービス	333
ゲートウェイサービス	334
コア	334
プロキシ	337
セキュリティ	338
リライター	340
ロギング	342
NetFile サービス	343
ホスト	344
権限	345

表示	346
操作	346
一般	348
Netlet サービス	348
プロキシレットサービス	351
付録 C 国コード	353
用語集	363
索引	365

図目次

図 1-1	オープンモードの Portal Server	31
図 1-2	セキュアモードの Portal Server (SRA ソフトウェアを使用)	32
図 2-1	Web プロキシの管理	62
図 2-2	Netlet プロキシの実装	72
図 5-1	Netlet のコンポーネント	184

表目次

表 1	本書の構成	20
表 2	表記上の規則	22
表 3	デフォルトのパスとファイル名	23
表 4	Solaris と Linux のパス名の比較	24
表 A-1	情報ファイルとデバッグファイル	331
表 B-1	アクセスリストサービスの属性	333
表 B-2	ゲートウェイサービスのコア属性	334
表 B-3	ゲートウェイサービスのプロキシ属性	337
表 B-4	ゲートウェイサービスのリライター属性 - 基本	340
表 B-5	ゲートウェイサービスのリライター属性 - 詳細	341
表 B-6	ゲートウェイサービスロギングの属性	342
表 B-7	NetFile サービスのホスト設定属性	344
表 B-8	NetFile サービスのホストアクセス属性	345
表 B-9	NetFile サービスの権限属性	345
表 B-10	NetFile サービスの表示属性	346
表 B-11	NetFile サービスの操作トラフィック属性	347
表 B-12	NetFile サービスの操作検索属性	347
表 B-13	NetFile サービスの操作圧縮属性	348
表 B-14	NetFile サービスの一般属性	348
表 B-15	Netlet サービスの属性	348
表 B-16	プロキシレットサービスの属性	351
表 C-1	2 文字の国コード	353

手順一覧

競合の解決レベルを設定する手順	36
ゲートウェイプロファイルを作成するには	41
chroot をインストールするには	54
chroot 環境でゲートウェイを再起動するには	57
ゲートウェイを起動するには	57
ゲートウェイを停止するには	58
別のプロファイルでゲートウェイを再起動するには	59
ゲートウェイを再起動するには	59
ゲートウェイ watchdog を設定するには	59
仮想ホストを指定するには	60
プロキシを指定するには	60
Netlet プロキシを再起動するには	74
Netlet プロキシの watchdog を設定するには	74
リライタプロキシを再起動するには	77
リライタプロキシの watchdog を設定するには	77
逆プロキシを有効化するには	78
既存の PDC インスタンスに認証モジュールを追加するには	81
ブラウザキャッシングを無効にする手順	83
ゲートウェイによるすべての URL のリライトを有効にするには	128
URI をルールセットにマッピングするには	130
MIME のマッピングを指定するには	131
リライトしない URI を指定するには	131
デフォルトドメインを指定するには	132
MIME 推測を有効にするには	133
パーサーを URI にマッピングするには	133
マスキングを有効にするには	134
マスキングのためのシード文字列を指定するには	135
マスクしない URI のリストを作成するには	135
ゲートウェイプロトコルと元の URI プロトコルを同一化するには	136
リライタのデバッグレベルを設定するには	137
HTML 属性のサンプルを使用するには	141

HTML JavaScript トークンのサンプルを使用するには	143
フォームのサンプルを使用するには	146
アプレットのサンプルを使用するには	148
JavaScript の URL 変数のサンプルを使用するには	150
JavaScript の EXPRESSION 変数のサンプルを使用するには	153
JavaScript の DHTML 変数のサンプルを使用するには	155
JavaScript の DJS 変数のサンプルを使用するには	158
JavaScript の SYSTEM 変数のサンプルを使用するには	160
JavaScript の URL 関数のサンプルを使用するには	162
JavaScript の EXPRESS 関数のサンプルを使用するには	164
JavaScript の DHTML 関数のサンプルを使用するには	166
JavaScript の DJS 関数のサンプルを使用するには	168
XML 属性のサンプルを使用するには	170
OWA のルールセットを設定するには	177
ルールの追加後に Netlet を実行するには	199
Netlet を PDC 用に設定するには	209
インストール後に自己署名証明書を生成するには	219
CSR を生成するには	221
ルート CA 証明書を追加するには	223
CA に証明書を要求するには	224
CA から届いた証明書をインストールするには	225
証明書を削除するには	226
証明書の信頼属性を変更するには	228
ルート CA 証明書をリスト表示するには	230
すべての証明書をリスト表示するには	231
証明書を出力するには	232
URL アクセス制御を設定するには	235
拒否される URL リストを設定するには	236
許可される URL リストを設定するには	236
ホストのシングルサインオンを無効にするには	238
セッションごとのシングルサインオンを有効にするには	238
認証レベルを指定するには	238
ゲートウェイの属性を設定するには	239
HTTP モードまたは HTTPS モードで実行するようにゲートウェイを設定するには	241
リライタプロキシを有効化し、リライタプロキシリストを作成するには	242
Netlet を有効にするには	243
Netlet プロキシを有効化し、Netlet プロキシリストを作成するには	244
プロキシレットを有効にするには	245
Cookie の管理を有効にするには	246
HTTP 基本認証を有効にするには	247
持続 HTTP 接続を有効にするには	248
持続接続 1 つあたりの最大要求数を指定するには	249
持続ソケット接続のタイムアウトを指定するには	249

回復時間に必要な正常なタイムアウトを指定するには	250
Cookie を転送する URL を追加するには	251
最大接続キューを指定するには	252
ゲートウェイタイムアウトを指定するには	252
最大スレッドプールサイズを指定するには	253
キャッシュされたソケットのタイムアウトを指定するには	254
Portal Server を指定するには	254
サーバーの再試行間隔を指定するには	255
外部サーバー Cookie を格納するには	256
URL からのセッションを取得するには	256
安全な Cookie としてマークするには	257
Web プロキシの使用を有効にするには	258
Web プロキシを使用する URL を指定するには	259
Web プロキシを使用しない URL を指定するには	260
ドメインとサブドメインのプロキシを指定するには	260
プロキシパスワードを指定するには	261
自動プロキシ設定サポートを有効にするには	262
自動プロキシ設定ファイルの場所を指定するには	263
Web プロキシを通じての Netlet トンネリングを有効にするには	263
非認証 URL パスを指定するには	264
証明書が有効なゲートウェイホストのリストにゲートウェイを追加するには	265
40 ビット暗号化接続を許可するには	266
SSL Version 2.0 を有効にするには	267
暗号化方式の個別選択を有効にするには	267
SSL Version 3.0 を有効にするには	268
Null 暗号化方式を有効にするには	269
信頼されている SSL ドメインのリストを作成するには	269
PDC とコード化されたデバイスを設定するには	270
ゲートウェイによるすべての URL のリライトを有効にするには	273
URI をルールセットにマッピングするには	274
OWA のルールセットを設定するには	276
MIME のマッピングを指定するには	277
リライトしない URI を指定するには	277
デフォルトドメインを指定するには	278
MIME 推測を有効にするには	279
パーサーを URI にマッピングするには	279
マスキングを有効にするには	280
マスキングのためのシード文字列を指定するには	281
マスクしない URI を作成するには	281
ゲートウェイプロトコルと元の URI プロトコルを同一化するには	282
ゲートウェイのロギングを有効にするには	283
Netlet ロギングを有効にするには	284
NetFile の属性を設定するには	287

OS の文字セットを指定するには	288
ホスト検出順序を指定するには	289
共通ホストのリストを設定するには	290
デフォルトドメインを指定するには	291
デフォルトの Microsoft Windows ドメインまたはワークグループを指定するには	292
デフォルトの WINS/DNS サーバーを指定するには	293
異なるタイプのホストへのアクセスを指定するには	293
許可されたホストリストを作成するには	294
拒否されたホストリストを作成するには	295
アクセス権を有効化または無効化するには	297
NetFile ウィンドウのサイズを指定するには	298
NetFile ウィンドウの位置を指定するには	299
一時ディレクトリを指定するには	300
ファイルアップロードサイズの制限を設定するには	301
ディレクトリ検索の制限を指定するには	302
デフォルトの圧縮タイプを指定するには	303
MIME タイプ設定ファイルの場所を指定するには	304
Netlet ルールを追加するには	307
Netlet ルールを変更するには	308
Netlet ルールを削除するには	309
デフォルトの暗号化方式を指定するには	309
デフォルトループバックポートを割り当てるには	310
接続の再認証を有効にするには	311
接続の警告ポップアップを有効にするには	311
ユーザーによるポート警告ダイアログの非表示を許可するには	312
キーブライブ間隔を設定するには	313
「ポータルのログアウト時に Netlet を終了」オプションを設定するには	314
Netlet ルールへのアクセスを定義するには	314
Netlet ルールへのアクセスを拒否するには	315
ホストへのアクセスを許可するには	316
ホストへのアクセスを拒否するには	317
プロキシレットの属性を設定するには	319
Sun Crypto Accelerator 1000 を設定するには	322
Sun Crypto Accelerator 4000 を設定するには	326
外部 SSL デバイスアクセラレータの有効化	328
外部 SSL デバイスアクセラレータを設定するには	329

はじめに

このガイドでは、Sun Java™ System Portal Server Secure Remote Access の管理方法について説明します。

Sun Java System Portal Server Secure Remote Access (SRA) は、リモートユーザーがインターネットを通じて社内のネットワークおよびサービスに安全にアクセスできる環境を提供します。また、従業員、ビジネスパートナー、一般ユーザーなど、あなたの会社のインターネットポータルを使用する誰もがコンテンツやアプリケーション、データに安全にアクセスできるようになります。

SRA は、Solaris™ オペレーティングシステム 8.0 以降および Linux プラットフォームで稼動します。このガイドには、SRA を設定および管理するための手順が記載されています。

この章は以下の各節で構成されます。

- [対象読者](#)
- [本書の構成](#)
- [本書で使用する規則](#)
- [デフォルトのパスとファイル名](#)
- [オンラインの Sun リソースへのアクセス](#)
- [Sun 技術サポートへの問い合わせ](#)
- [関連するサードパーティの Web サイト](#)
- [Sun へのコメント](#)

対象読者

この管理ガイドは、SRA を設定し、管理するユーザーを対象としています。

この管理ガイドは、UNIX® システムと TCP/IP ネットワークの管理に熟練したネットワーク管理者またはシステム管理者を想定して作成されています。SRA の各種のコンポーネントをインストールする場合、必要なマシンに root でアクセスする必要はありません。ユーザーとサービスの設定など、その他の操作を実行するのに必要な管理権限が必要です。

Portal Server Secure Remote Access の管理者は、次のテクノロジーを理解している必要があります。

- Solaris™ オペレーティングシステムの基本管理手順
- Lightweight Directory Access Protocol (LDAP)
- Sun Java System Directory Server
- Sun Java System Web Server
- Sun Java System Portal Server

また、リライタ規則を記述するために、次の内容についても理解している必要があります。

- HTML (Hypertext Markup Language) と HTML タグの理解
- JavaScript の正しい知識
- XML (Extensible Markup Language) の基本的な知識

本書の構成

次の表は、本書の内容についてまとめたものです。

表 1 本書の構成

章	説明
29 ページの第 1 章「Portal Server Secure Remote Access について」	この章では SRA ソフトウェア、および Portal Server 製品と SRA ソフトウェアコンポーネントの関係について説明します。SRA ソフトウェアの管理と設定に関する情報も記載されています。
39 ページの第 2 章「ゲートウェイ」	この章では、ゲートウェイのスムーズな実行に必要な、ゲートウェイに関連する概念と情報について説明します。

表 1 本書の構成 (続き)

章	説明
91 ページの第 3 章「プロキシレットとリライタ」	この章では、プロキシレットとリライタについて説明します。リライタについては、サンプルルールと最良の実行方法を提示します。
179 ページの第 4 章「NetFile」	NetFile とその操作について説明します。
183 ページの第 5 章「Netlet」	ユーザーのリモート標準ポータルデスクトップとイントラネット上のアプリケーションを実行しているサーバーとの間で、Netlet を使用してアプリケーションを安全に実行する方法について説明します。
209 ページの第 6 章「Netlet での PDC の使用」	Netlet で PDC を使用できるように、クライアントブラウザの Java プラグインを設定する方法について説明します。
211 ページの第 7 章「証明書」	証明書の管理、および自己署名証明書または認証局からの証明書をインストールする方法について説明します。
235 ページの第 8 章「URL アクセス制御の設定」	特定の URL に対するゲートウェイ経由のエンドユーザーからのアクセスを許可または拒否する方法について説明します。
239 ページの第 9 章「ゲートウェイの設定」	Access Manager 管理コンソールからゲートウェイの属性を設定する方法について説明します。
287 ページの第 10 章「NetFile の設定」	Access Manager 管理コンソールから NetFile を設定する方法について説明します。
305 ページの第 11 章「Netlet の設定」	Access Manager 管理コンソールから Netlet の属性を設定する方法について説明します。
319 ページの第 12 章「プロキシレットの設定」	Access Manager 管理コンソールからプロキシレットを設定する方法について説明します。
321 ページの第 13 章「SSL アクセラレータの設定」	Portal Server Secure Remote Access の各種アクセラレータを設定する方法について説明します。
付録 A 「ログファイル」	Portal Server Secure Remote Access のログファイルの一覧を示し、それぞれのログファイルについて説明します。
付録 B 「設定属性」	Access Manager 管理コンソールで Portal Server Secure Remote Access に対して設定する属性の一覧を示します。

表 1 本書の構成 (続き)

章	説明
付録 C 「国コード」	認証管理の際に指定する 2 文字の国コードの一覧を示します。
用語集	Sun Java System 全体の用語集へのリンクが含まれます。

本書で使用する規則

この節の各表では、本書で使用する各種の規則について説明します。

表記上の規則

次の表は、本書で使用する表記上の規則について説明したものです。

表 2 表記上の規則

表記	意味	例
AaBbCc123 (モノスペース)	API および言語の構成要素、HTML タグ、Web サイトの URL、コマンド名、ファイル名、ディレクトリパス名、コンピュータからの画面出力、サンプルコード。	.login ファイルを編集します。 すべてのファイルを一覧表示するには <code>ls -a</code> を使用します。 % You have mail.
AaBbCc123 (太字のモノスペース)	ユーザーが入力するテキストを強調し、コンピュータからの画面出力と区別します。	% su Password:
<i>AaBbCc123</i> (イタリック)	実際の名前または値によって置き換えられるコマンドまたはパス名の可変部分。	これらを <i>class</i> オプションと呼びます。 ファイルは <i>install-dir/bin</i> ディレクトリにあります。

デフォルトのパスとファイル名

次の表は、本書で使用するデフォルトのパスとファイル名について説明したものです。

表 3 デフォルトのパスとファイル名

パス名	説明
<code>/etc/opt/SUNWps/platform.conf.default</code>	すべての <code>platform.conf.*</code> ファイルの場所
<code>gateway-install-root/SUNWps/bin/gateway</code>	gateway-instance の名前場所
<code>/etc/opt/SUNWam/config/</code>	AMConfig-instance-name.properties の場所
<code>portal-server-install-root/SUNWps/locale</code>	srapGateway.properties の場所
<code>/var/opt/SUNWps/debug</code>	ゲートウェイデバッグのログファイルの場所
<code>/var/opt/SUNWam/debug</code>	Access Manager 管理コンソールからデバッグを開始した場合のサービスログのデフォルトのデバッグディレクトリ。
<code>/var/opt/SUNWam/logs/srapNetFile</code>	ログファイルの場所
<code>/etc/opt/SUNWps/cert/default/gateway-profile-name</code>	証明書関連ファイルの場所
<code>/opt/S1PS62/SUNWps/samples/config/netfile</code>	MIME タイプ設定ファイルの場所

Linux の使用

Sun Java™ System Portal Server は RedHat 3.0 Linux プラットフォームをサポートしています。ただし、以下に説明する Solaris と Linux プラットフォームの相違に注意してください。

Linux を使用する場合の制限

- Portal Server と Access Manager は同じサーバー上になければなりません。
- サンプルのポータルは Linux プラットフォームをサポートしていません。
- IBM と BEA の Web コンテナはサポートされません。

設定ファイル、配備、およびアプリケーションプログラミングインタフェースは Solaris と Linux で同じです。

Solaris と Linux のパス名の比較

表 4 Solaris と Linux のパス名の比較

Solaris のパス名	Linux のパス名
/opt/SUNWps (デフォルト)	/opt/sun/portal (デフォルト)
/etc/opt/SUNWps (設定)	/etc/opt/sun/portal (設定)
/var/opt/SUNWps (データ)	/var/opt/sun/portal (データ)

関連情報

<http://docs.sun.com>SM Web サイトでは、Sun の技術文書にオンラインでアクセスできます。アーカイブを参照するか、個々の書名または件名を検索できます。

このドキュメントセット内のマニュアル

次の表は、Portal Server Secure Remote Access コアドキュメントセットに含まれるマニュアルについてまとめたものです。

マニュアル名	説明
『Portal Server 配備計画ガイド』 http://docs.sun.com/db/doc/819-1206?l=ja	Portal Server ソフトウェアを計画し、配備する方法について説明します。
Portal Server 管理ガイド http://docs.sun.com/db/doc/819-1198?l=ja	Access Manager 管理コンソールおよびコマンド行を使用して Portal Server を管理する方法について説明します。
Portal Server Secure Remote Access 管理ガイド http://docs.sun.com/db/doc/819-1202?l=ja	Portal Server Secure Remote Access を管理する方法について説明します。
『Portal Server リリースノート』 http://docs.sun.com/db/doc/819-1494?l=ja	製品のリリース後に提供されます。最新リリースの新機能、既知の問題と制限、インストールの注意事項、ソフトウェアまたはドキュメントに関する問題の報告方法などの最新情報が収められています。
『Portal Server Technical Reference Guide』 http://docs.sun.com/app/docs/coll/PortalServer_05q1?l=ja	Portal Server の技術概念 (表示プロファイル、リライタなど)、コマンド行ユーティリティ、(ソフトウェア内の) タグライブラリ、ファイル (テンプレートや JSP など) についての詳細な情報を提供します。このガイドは、これらの重要な背景情報の唯一の提供源です。

その他の Portal Server ドキュメント

その他の Portal Server ドキュメントには、次のものがあります。

- 『Portal Server Desktop Customization Guide』
http://docs.sun.com/app/docs/coll/PortalServer_05q1?l=ja
- 『Portal Server Developer's Guide』
http://docs.sun.com/app/docs/coll/PortalServer_05q1?l=ja
- 『Portal Server Mobile Access Developer's Guide』
http://docs.sun.com/app/docs/coll/PortalServer_05q1?l=ja
- 『Portal Server Mobile Access Developer's Reference』
http://docs.sun.com/app/docs/coll/PortalServer_05q1?l=ja
- 『Portal Server Mobile Access 配備計画ガイド』
<http://docs.sun.com/doc/819-1961?l=ja>
- 『Portal Server Mobile Access Tag Library Reference』
http://docs.sun.com/app/docs/coll/PortalServer_05q1?l=ja

その他のサーバー関連ドキュメント

サーバー関連のその他のドキュメントを次に示します。

- Directory Server ドキュメント
http://docs.sun.com/coll/DirectoryServer_04q2
- Web Server ドキュメント
http://docs.sun.com/coll/S1_websvr61_en
- Application Server ドキュメント
http://docs.sun.com/coll/s1_asseu3_en
- Web Proxy Server ドキュメント
<http://docs.sun.com/prod/s1.webproxys#hic>

オンラインの Sun リソースへのアクセス

製品のダウンロード、プロフェッショナルサービス、パッチとサポート、その他の開発関連情報については、次の各サイトをご覧ください。

- ダウンロードセンター
<http://www.sun.com/software/download/>
- プロフェッショナルサービス
<http://www.sun.com/service/sunps/sunone/index.html>
- Sun エンタープライズサービス、Solaris オペレーティングシステムのパッチ、サポート
<http://sunsolve.sun.com/>
- 開発者向け情報
<http://developers.sun.com/prodtech/index.html>

Sun 技術サポートへの問い合わせ

製品のドキュメントで解決できない、本製品に関する技術的な質問の問い合わせ先については、<http://www.sun.com/service/contacting> を参照してください。

関連するサードパーティの Web サイト

Sun は、このマニュアルに記載されているサードパーティ Web サイトの利用について責任を負いません。Sun は、このようなサイトまたはリソースで得られるあらゆる内容、広告、製品、およびその他素材を保証するものではなく、責任または義務を負いません。Sun は、このようなサイトまたはリソースで得られるあらゆるコンテンツ、製品、またはサービスによって生じる、または生じたと主張される、または使用に関連して生じる、または信頼することによって生じる、いかなる損害または損失についても責任または義務を負いません。

Sun へのコメント

Sun ではドキュメントの改善に取り組んでおり、コメントや提案を歓迎します。

コメントをお送りになる場合は、<http://docs.sun.com/app/docs?l=ja> にアクセスして「コメントの送信」をクリックしてください。オンラインフォームで、ドキュメントのタイトルと部品番号を入力します。部品番号は、マニュアルの表紙またはドキュメントの先頭に記載されている 7 桁または 9 桁の数字です。たとえば、本書のタイトルは『Portal Server Secure Remote Access 2005Q1 管理ガイド』であり、部品番号は 819-1202 です。

Portal Server Secure Remote Access について

この章では、Sun Java™ System Portal Server Secure Remote Access について、および Sun Java System Portal Server (Portal Server) ソフトウェアと Sun Java System Portal Server Secure Remote Access (SRA) コンポーネントの関係について説明します。

この章で説明する内容は次のとおりです。

- [SRA ソフトウェアの概要](#)
- [SRA サービス](#)
- [SRA 製品の管理](#)
- [SRA の属性の設定](#)
- [サポートされるアプリケーション](#)

SRA ソフトウェアの概要

SRA ソフトウェアは、リモートユーザーがインターネットを通じて社内のネットワークおよびサービスに安全にアクセスできる環境を提供します。また、従業員、ビジネスパートナー、一般ユーザーなど、あなたの会社のインターネットポータルを使用する誰もがコンテンツやアプリケーション、データに安全にアクセスできるようになります。

リモートデバイスからポータルコンテンツおよびサービスにアクセスする場合、SRA ソフトウェアはブラウザによるセキュアリモートアクセスを提供します。SRA は、Java™ テクノロジーに対応したブラウザを使用するすべてのデバイスからアクセス可能な安全なアクセスソリューションであり、クライアントソフトウェアを使用しません。Portal Server に統合すると、アクセス権のあるコンテンツおよびサービスに対して暗号化された安全なアクセスが保証されます。

SRA ソフトウェアは、安全性の高いリモートアクセスポータルを配備する企業を対象に設計されています。このようなポータルは、イントラネットリソースのセキュリティ、保護、およびプライバシーに重点が置かれています。SRA のアーキテクチャは、これらのタイプのポータルによく適しています。ユーザーは SRA ソフトウェアを利用することにより、イントラネットリソースをインターネットに公開することなく、これらのリソースにインターネットを通じて安全にアクセスできます。

Portal Server は、次の 2 つのモードで動作します。

- オープンモード
- セキュアモード

オープンモード

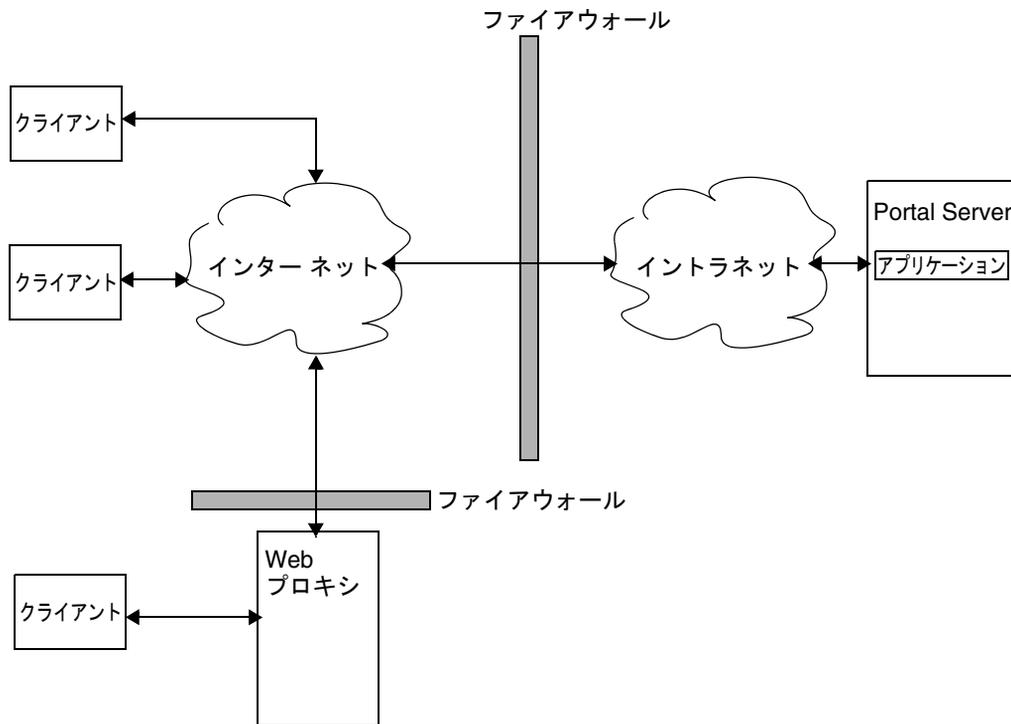
オープンモードの場合、Portal Server のインストール時に SRA ソフトウェアはインストールされません。このモードでの HTTPS 通信は可能ですが、セキュアリモートアクセスは使用できません。つまり、セキュリティ保護されたリモートファイルシステムとアプリケーションにはアクセスできません。

オープンポータルとセキュアポータルの主な違いは、オープンポータルを通じて提供されるサービスは、通常は保護されたイントラネット内ではなく非武装ゾーン (DMZ) 内に存在する点にあります。DMZ は一般のインターネットと私的なイントラネットの間に存在する保護付きの小規模ネットワークで、通常は両端のファイアウォールで境界が定められます。

ポータルに機密情報が含まれていない場合 (公開情報の配布や無償アプリケーションへのアクセス許可)、大量のアクセス要求への応答は、セキュアモードに比べて速くなります。

図 1-1 は、オープンモードの Portal Server を示しています。この例では、Portal Server はファイアウォールの背後にある単一のサーバーにインストールされています。複数のクライアントが単一のファイアウォールを経由して、インターネット上の Portal Server にアクセスしています。

図 1-1 オープンモードの Portal Server



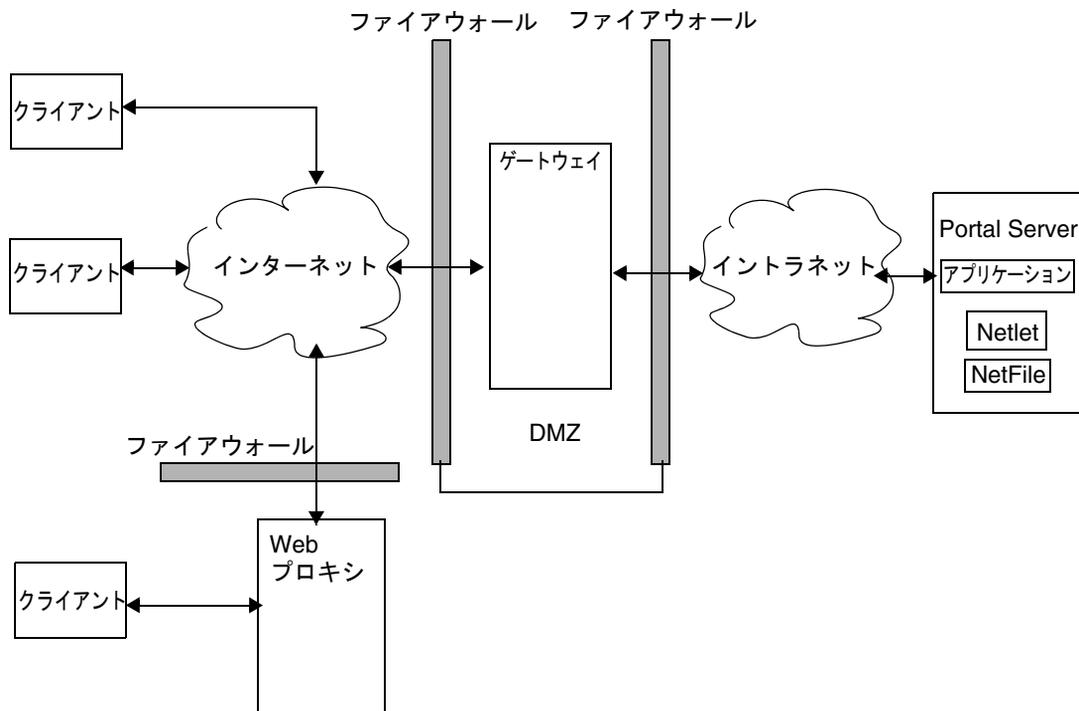
セキュアモード

セキュアモードは、必要とされるイントラネットファイルシステムとアプリケーションへのセキュリティ保護されたリモートアクセスを可能にします。

ゲートウェイは非武装ゾーン (DMZ) に常駐します。ゲートウェイはすべてのイントラネット URL とアプリケーションへの単一のセキュアアクセスポイントとして機能し、ファイアウォールに開かれるポートの数は減ります。その他のセッション、認証、および標準のポータルデスクトップなどの Portal Server サービスはすべて、保護されたイントラネットの DMZ の背後で実行されます。クライアントブラウザからゲートウェイへの通信は、SSL (Secure Socket Layer) を使った HTTP を使って暗号化されます。ゲートウェイからサーバーおよびイントラネットリソースへの通信には HTTP か HTTPS が使用されます。

図 1-2 は、Portal Server と SRA ソフトウェアを示しています。SSL はクライアントとゲートウェイの接続をインターネット上で暗号化するために使用されます。また、SSL はゲートウェイとサーバー間の接続の暗号化にも使用されます。イントラネットとインターネット間にゲートウェイが存在することで、クライアントと Portal Server 間のパスの安全性が強化されます。

図 1-2 セキュアモードの Portal Server (SRA ソフトウェアを使用)



サーバーとゲートウェイをさらに追加して、サイトを拡張することができます。SRA ソフトウェアは、ビジネスの要件に基づいてさまざまな方法で構成することができます。

SRA サービス

SRA ソフトウェアには、次に示す 5 つの主要なコンポーネントがあります。

- [ゲートウェイ](#)
- [リライタ](#)
- [NetFile](#)
- [Netlet](#)
- [プロキシレット](#)

ゲートウェイ

SRA のゲートウェイは、インターネットから送信されるリモートユーザーセッションと企業イントラネットの間のインタフェースおよびセキュリティバリアとして機能します。ゲートウェイはリモートユーザーとの単一のインタフェースを通じて、内部 Web サーバーとアプリケーションサーバーのコンテンツを安全に提供します。

Web サーバーは、クライアントとゲートウェイの間の通信に HTML、JavaScript、XML などの Web ベースのリソースを使用します。リライタは、Web コンテンツを使用できるようにするためのゲートウェイコンポーネントです。

アプリケーションサーバーは、クライアントとゲートウェイの間の通信に telnet や FTP などのバイナリプロトコルを使用します。ゲートウェイに常駐する Netlet は、この目的で使用されます。詳細については、[第 2 章「ゲートウェイ」](#)を参照してください。

リライタ

リライタは、エンドユーザーのイントラネット参照を可能にし、またそのページ上のリンクや URL へのリンクが正しく機能するようにします。リライタは Web ブラウザのロケーションフィールドにゲートウェイ URL を追加して、ゲートウェイを通じてコンテンツ要求をリダイレクトします。詳細については、[第 3 章「プロキシレットとリライタ」](#)を参照してください。

NetFile

NetFile はファイルシステムとディレクトリのリモートアクセスおよびリモート操作を可能にする、ファイルマネージャアプリケーションです。NetFile には Java ベースのユーザーインターフェースが含まれます。これは、Java1 と Java2 で使用できます。詳細については、第 4 章「NetFile」を参照してください。

Netlet

Netlet は一般的なアプリケーション、または企業独自のアプリケーションをリモートデスクトップで安全に、効率的に実行できるようにします。サイトに Netlet を実装すると、Telnet や SMTP などの共通の TCP/IP サービスや、pcANYWHERE または Lotus Notes などの HTTP ベースのアプリケーションを安全に実行できます。詳細については、第 5 章「Netlet」を参照してください。

プロキシレット

プロキシレットは、クライアントマシン上で稼動する動的なプロキシサーバーです。プロキシレットは URL をゲートウェイにリダイレクトします。クライアントマシン上のプロキシレットは、この機能を実現するために、ブラウザのプロキシ設定を読み込んでから、ローカルプロキシサーバー (プロキシレット) をポイントするように変更します。

SRA 製品の管理

SRA ソフトウェアには、管理のための次の 2 つのインターフェースがあります。

- Access Manager 管理コンソール
- コマンド行

管理作業の大半は、Web ベースの Sun Java System Access Manager 管理コンソールを通じて行います。管理コンソールにはローカルにアクセスできます。また、Web ブラウザからのリモートアクセスも可能です。ただし、ファイルの修正などの管理作業には UNIX コマンド行インターフェースを使用します。

SRA の属性の設定

ほとんどの属性は、Access Manager の「アイデンティティ管理」タブまたは「サービス設定」タブで設定できます。このサービス設定レベルの属性は、テンプレートとして機能します。組織またはユーザーが新規に作成されると、デフォルトでこれらの値を継承します。

SRA に関連する属性は、組織、ロール、ユーザーのレベルで設定できます。ただし、次の例外が適用されます。

- 競合の解決レベルはユーザーレベルでは設定できず、「サービス設定」タブからは設定できません。36 ページの「競合解決の設定」を参照してください。
- MIME タイプ設定ファイルの場所は、組織レベルだけで設定可能です。304 ページの「MIME タイプ設定ファイルの場所の指定」を参照してください。

組織レベルで設定した値は、その組織に属するすべてのロールとユーザーにも継承されます。ユーザーレベルで設定された値は、組織レベルまたはロールレベルで設定された値よりも優先されます。

属性の値は「サービス設定」タブで変更できます。新しい値は、組織を新たに追加した場合にだけ、適用されます。「サービス設定」タブでの属性値の変更は、既存の組織またはユーザーに影響しません。詳細については、『Access Manager 管理ガイド』を参照してください。

SRA の属性は、Access Manager 管理コンソールの「SRA 設定」の下にある次のサービスを使用して設定します。

- **アクセスリスト**
特定の URL へのアクセスを許可または制限し、シングルサインオン機能を管理する場合に使用します。詳細については、第 8 章「URL アクセス制御の設定」を参照してください。
- **ゲートウェイ**
プロキシ管理、Cookie 管理、ロギング、リライタ管理、暗号化などのゲートウェイに関連したすべての属性を設定する場合に使用します。詳細については、第 9 章「ゲートウェイの設定」を参照してください。
- **NetFile**
共通ホスト、MIME タイプ、異なる種類のホストへのアクセスなど、NetFile 関連のすべての属性を設定する場合に使用します。詳細については、第 10 章「NetFile の設定」を参照してください。
- **Netlet**
Netlet ルール、必須ルールへのアクセス、組織およびホスト、デフォルトアルゴリズムなどの Netlet に関連したすべての属性を設定する場合に使用します。詳細については、第 11 章「Netlet の設定」を参照してください。

- プロキシレット

「プロキシレットアプレットのバインド IP」アドレスやポート番号など、プロキシレットに関連する属性を設定する場合に使用します。詳細については、[第 12 章「プロキシレットの設定」](#)を参照してください。

警告

ゲートウェイの実行中に行われた属性変更は、ゲートウェイに通知されません。更新された (ゲートウェイまたはその他のサービスに属する) プロファイル属性をゲートウェイで確実に使用するようには、ゲートウェイを再起動します。[81 ページの「認証連鎖の使用」](#)を参照してください。

競合解決の設定

▶ 競合の解決レベルを設定する手順

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」 ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に表示されます。
5. 「表示」 ドロップダウンリストから「サービス」を選択します。
6. 「SRA 設定」の下で、適切なサービス (アクセスリスト、NetFile、Netlet、またはプロキシレット) の隣の矢印をクリックします。
7. 「競合の解決レベル」 ドロップダウンリストから適切なレベルを選択します。
8. 「保存」をクリックし、変更内容を記録します。

サポートされるアプリケーション

SRA ソフトウェアは、次のアプリケーションをサポートします。

- MS Exchange 2000 SP3 および MS Exchange 2003 (Outlook Web Access (OWA) 経由)
- exchange_2003_owa_ruleset という名前の OWA ルールセット
- iNotes - Notes 5.0.11
- Sun Java System Calendar Server Release 5.1.1 以降
- Sun Java System Messenger Express 6 2005Q1 - Sun Java System Messaging Server 5.2 以降
- Sun Java System Communications Express 6 2005Q1

サポートされるアプリケーション

ゲートウェイ

この章では、ゲートウェイのスムーズな実行に必要な、ゲートウェイに関連する概念と情報について説明します。ゲートウェイの設定については、[第9章「ゲートウェイの設定」](#)を参照してください。

この章で説明する内容は次のとおりです。

- [ゲートウェイの概要](#)
- [ゲートウェイプロファイルの作成](#)
- [platform.conf ファイルの概要](#)
- [chroot 環境でのゲートウェイの実行](#)
- [chroot 環境でのゲートウェイの再起動](#)
- [ゲートウェイの起動と停止](#)
- [ゲートウェイの起動と停止](#)
- [ゲートウェイの再起動](#)
- [仮想ホストの指定](#)
- [Access Manager へアクセスするプロキシの指定](#)
- [Web プロキシの使用](#)
- [自動プロキシ設定の使用](#)
- [Netlet プロキシの使用](#)
- [リライタプロキシの使用](#)
- [ゲートウェイでの逆プロキシの使用](#)
- [クライアント情報の取得](#)
- [認証連鎖の使用](#)
- [ワイルドカード証明書の使用](#)

- [ブラウザキャッシングの無効化](#)
- [ゲートウェイサービスのユーザーインターフェースのカスタマイズ](#)
- [連携管理の使用](#)

ゲートウェイの概要

ゲートウェイは、インターネットから送信されるリモートユーザーセッションと企業イントラネットの間のインターフェースおよびセキュリティバリアとして機能します。ゲートウェイはリモートユーザーとの単一のインターフェースを通じて、内部 Web サーバーとアプリケーションサーバーのコンテンツを安全に提供します。

各ゲートウェイに対して、次のことを実行する必要があります。

- ゲートウェイプロファイルを作成します。[40 ページの「ゲートウェイプロファイルの作成」](#)を参照してください。
- ゲートウェイのインスタンスを作成します。[51 ページの「ゲートウェイのインスタンスの作成」](#)を参照してください。
- ゲートウェイを設定します。[239 ページの第 9 章「ゲートウェイの設定」](#)を参照してください。

ゲートウェイプロファイルの作成

ゲートウェイプロファイルには、ゲートウェイが待機するポート、SSL オプション、プロキシオプションなどのゲートウェイの設定に関連したすべての情報が収められています。

ゲートウェイをインストールする場合、デフォルトの値を選択すると「default」というデフォルトゲートウェイプロファイルが作成されます。デフォルトプロファイルに相当する設定ファイルは、次の場所にあります。

```
/etc/opt/SUNWps/platform.conf.default
```

/etc/opt/SUNWps は、すべての platform.conf.* ファイルが格納されるデフォルトの場所です。

platform.conf ファイルの内容についての詳細は、[42 ページの「platform.conf ファイルの概要」](#)を参照してください。

次の処理を実行できます。

- 複数のプロファイルを作成して、各プロファイルに属性を定義します。また、必要に応じてこれらのプロファイルを異なる複数のゲートウェイに割り当てます。

- 同じプロファイルを、複数のマシン上にあるゲートウェイに割り当てます。
- 異なる複数のプロファイルを、同じマシン上で稼動している単一のゲートウェイの複数のインスタンスに割り当てます。

警告

同じマシン上で稼動するゲートウェイの複数のインスタンスに同じプロファイルを割り当てないでください。このような方法で割り当てると、ポート番号が同じになることにより衝突が発生します。

また、同じゲートウェイに作成された複数のプロファイルに、同じポート番号を指定しないでください。同じゲートウェイの複数のインスタンスを同じポートで実行すると、衝突が発生します。

▶ **ゲートウェイプロファイルを作成するには**

1. Sun Java™ System Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
右の区画に「ゲートウェイ」ページが表示されます。
4. 「新規」をクリックします。
「新規ゲートウェイプロファイルを作成」ページが表示されます。
5. 新規ゲートウェイプロファイル名を入力します。
6. ドロップダウンリストから、新規プロファイルの作成に使用するプロファイルを選択します。

デフォルトでは、新規プロファイルはパッケージ内の「default」プロファイルに基づいて作成されます。カスタムプロファイルを作成している場合、ドロップダウンリストからそのプロファイルを選択できます。新しいプロファイルは、選択したプロファイルのすべての属性を継承します。

新規プロファイル用にコピーした既存のプロファイルは同じポートをコピーします。このため、既存のプロファイルと競合しないように、新規プロファイルのポートを変更する必要があります。

7. 「作成」をクリックします。
新規プロファイルが作成され、「ゲートウェイ」ページに戻ります。このページには、新しいプロファイルが表示されます。
8. gwmultiinstance スクリプトを実行して、ゲートウェイのインスタンスを作成します。57 ページの「ゲートウェイの起動と停止」を参照してください。
9. 変更を有効にするには、このゲートウェイプロファイル名でゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

57 ページの「ゲートウェイの起動と停止」を参照してください。ゲートウェイの設定については、第9章「ゲートウェイの設定」を参照してください。

platform.conf ファイルの概要

platform.conf ファイルは、デフォルトで次の場所にあります。

```
/etc/opt/SUNWps
```

platform.conf ファイルには、ゲートウェイが必要とする詳細情報が収められています。ここでは、サンプルの platform.conf ファイルを提示し、すべてのエントリについて説明します。

マシン固有の詳細を設定ファイルにすべて格納しているため、複数のマシンで実行するゲートウェイが共通のプロファイルを共有できるという利点があります。

次に例を示します。

```
#
# Copyright 11/28/00 Sun Microsystems, Inc. All Rights Reserved.
# "(#)platform.conf1.38 00/11/28 Sun Microsystems"
#
gateway.user=noaccess
gateway.jdk.dir=/usr/java_1.3.1_06
gateway.dsame.agent=http://pserv2.iportal.com:8080/sunportal/Remote
ConfigServlet
portal.server.protocol=http
portal.server.host=pserv2.iportal.com
portal.server.port=8080
gateway.protocol=https
gateway.host=siroe.india.sun.com
gateway.port=333
gateway.trust_all_server_certs=true
gateway.trust_all_server_cert_domains=false
gateway.virtualhost=siroe1.india.sun.com 10.13.147.81
gateway.virtualhost.defaultOrg=o=root,dc=test,dc=com
```

```
gateway.notification.url=/notification
gateway.retries=6
gateway.debug=error
gateway.debug.dir=/var/opt/SUNWps/debug
gateway.logdelimiter=&&
gateway.external.ip=10.12.147.71
gateway.certdir=/etc/opt/SUNWps/cert/portal
gateway.allow.client.caching=true
gateway.userProfile.cacheSize=1024
gateway.userProfile.cacheSleepTime=60000
gateway.userProfile.cacheCleanupTime=300000
gateway.bindipaddress=10.12.147.71
gateway.sockretries=3
gateway.enable.accelerator=false
gateway.enable.customurl=false
gateway.httpurl=http://siroe.india.sun.com
gateway.httpsurl=https://siroe.india.sun.com
gateway.favicon=https://siroe.india.sun.com
gateway.logging.password=ALKJDF123SFLKJJSDFU
portal.server.instance=
gateway.cdm.cacheSleepTime=60000
gateway.cdm.cacheCleanUpTime=300000
netletproxy.port=10555
rewriterproxy.port=10556
```

表 2-1 は、platform.conf ファイルのすべてのフィールドと、その説明を示しています。

表 2-1 platform.conf ファイルのプロパティ

エントリ	デフォルト値	説明
gateway.user	noaccess	ゲートウェイは、このユーザーとして実行されます。 ゲートウェイは root として起動する必要があり、初期化の後、 root 権限を失いこのユーザーになります。
gateway.jdk.dir		ゲートウェイが使用する JDK ディレクトリの場所。
gateway.dsame.agent		ゲートウェイが起動中にそのプロファイルを取得するために通信する Access Manager の URL。
portal.server.protocol portal.server.host portal.server.port		デフォルトの Portal Server が使用しているプロトコル、ホスト、ポート。
gateway.protocol gateway.host gateway.port		ゲートウェイのプロトコル、ホスト、ポート。これらの値はインストール時に指定したモードおよびポートと同じです。これらの値は通知 URL の作成に使用されます。
gateway.trust_all_server_certs	true	ゲートウェイがすべてのサーバーの証明書を信頼する必要があるか、ゲートウェイ認証データベースの証明書のみを信頼するべきかを指定します。
gateway.trust_all_server_cert_domains	false	ゲートウェイとサーバーの間で SSL 通信が行われるとき、サーバーの証明書がゲートウェイに提示されます。デフォルトでは、ゲートウェイはサーバーのホスト名がサーバーの証明書 CN と同じであるかどうかをチェックします。 この属性値が true に設定されている場合、ゲートウェイは受け取ったサーバーの証明書に対するドメインチェックを無効にします。

表 2-1 platform.conf ファイルのプロパティ (続き)

エン트리	デフォルト値	説明
gateway.virtualhost		ゲートウェイマシンに複数のホスト名が設定されている場合、このフィールドで別の名前およびアイデンティティプロバイダアドレスを指定できます。
gateway.virtualhost.defaultOrg=org		<p>ユーザーがログインするデフォルトの org を指定します。</p> <p>たとえば、仮想ホストフィールドのエントリが次のような場合を考えます。</p> <pre>gateway.virtualhost=test.com employee.test.com Managers.test.com</pre> <p>デフォルトの org エントリは、次のようになります。</p> <pre>test.com.defaultOrg = o=root,dc=test,dc=com employee.test.com.defaultOrg = o=employee,dc=test,dc=com Manager.test.com.defaultOrg = o=Manager,dc=test,dc=com</pre> <p>ユーザーは <code>https://manager.test.com</code> を使用して、<code>https://test.com/o=Manager,dc=test,dc=com</code> ではなくマネージャーの org にログインできます。</p> <p>注: virtualhost と defaultOrg は platform.conf ファイルでは大文字と小文字が区別されますが、URL で使用する場合は区別されません。</p>

表 2-1 platform.conf ファイルのプロパティ (続き)

エントリ	デフォルト値	説明
gateway. notification.url		ゲートウェイのホスト、プロトコル、ポートの組み合わせが通知 URL の作成に使用されます。これは Access Manager からセッション通知を受け取る際に使用されます。 notification URL が組織名と一致しないことを確認します。通知 URL が組織名と一致する場合、その組織に接続しようとするとログインページではなく空のページが表示されます。
gateway.retries		ゲートウェイが起動時に Portal Server にアクセスを試みる回数。

表 2-1 platform.conf ファイルのプロパティ (続き)

エントリ	デフォルト値	説明
gateway.debug	error	<p>ゲートウェイのデバッグレベルを設定します。デバッグログファイルの場所は、<i>debug_directory/files</i> です。デバッグファイルの場所は、<i>gateway.debug.dir</i> エントリに指定されます。</p> <p>次のデバッグレベルがあります。</p> <p>error: 重要なエラーのみがデバッグファイルにログとして記録される。このようなエラーが発生すると、通常はゲートウェイの機能が停止する。</p> <p>warning: 警告メッセージがログとして記録される。</p> <p>message: すべてのデバッグメッセージがログとして記録される。</p> <p>on: すべてのデバッグメッセージがコンソールに表示される。</p> <p>次のデバッグファイルがあります。</p> <p><i>srapGateway.gateway-profile-name</i> : ゲートウェイデバッグメッセージを格納する。</p> <p><i>Gateway_to_from_server.gateway-profile-name</i> : メッセージモードの場合、ゲートウェイと内部サーバーの間のすべての要求と応答のヘッダーがこのファイルに格納される。</p> <p>このファイルを生成するには、 /var/opt/SUNWps/debug ディレクトリの書き込み権限を変更します。</p> <p><i>Gateway_to_from_browser.gateway-profile-name</i> : メッセージモードの場合、ゲートウェイとクライアントブラウザの間のすべての要求と応答のヘッダーがこのファイルに格納される。</p> <p>このファイルを生成するには、 /var/opt/SUNWps/debug ディレクトリの書き込み権限を変更します。</p>

表 2-1 platform.conf ファイルのプロパティ (続き)

エントリ	デフォルト値	説明
gateway.debug.dir		すべてのデバッグファイルが生成されるディレクトリ。 このディレクトリは、gateway.user 内のユーザーがファイルの書き込みを行うための十分な権限を必要とします。
gateway.logdelimiter		現在は使用されていません。
gateway.external.ip		複数の IP アドレスを持つマルチホームゲートウェイマシンでは、外部 IP アドレスをここに指定する必要があります。この IP は Netlet が FTP を実行するために使用されます。
gateway.certdir		証明書データベースの場所を指定します。
gateway.allow.client.caching	true	クライアントのキャッシングを許可または拒否します。 許可する場合、クライアントのブラウザはスタティックページおよびイメージをキャッシュして (ネットワークトラフィックを低減することで) パフォーマンスを向上できます。 拒否する場合、キャッシュは行われずセキュリティは高まりますが、ネットワークの負荷が高くなるのでパフォーマンスは低下します。
gateway.userProfile.cacheSize		ゲートウェイでキャッシュされるユーザープロファイルのエントリ数。エントリ数がこの値を超えると、キャッシュをクリーンアップするために頻繁に再試行が行われます。
gateway.userProfile.cacheSleepTime		キャッシュクリーンアップのためのスリープ時間 (秒単位) を設定します。
gateway.userProfile.cacheCleanupTime		プロファイルエントリが削除されるまでの最大時間 (秒)。

表 2-1 platform.conf ファイルのプロパティ (続き)

エントリ	デフォルト値	説明
gateway.bindipaddress		マルチホームマシンで、ゲートウェイがサーバーソケットをバインドする IP アドレス。すべてのインタフェースを待機するようにゲートウェイを設定するには、IP アドレスを gateway.bindipaddress=0.0.0.0 に置き換えます。
gateway.sockretries	3	現在は使用されていません。
gateway.enable.accelerator	false	true に設定した場合、外部アクセラレータの使用が許可されます。
gateway.enable.customurl	false	true に設定した場合、管理者はゲートウェイがページをリライトするためのカスタム URL を指定できます。
gateway.httpurl		ゲートウェイがページをリライトするためのカスタム URL 用の HTTP 逆プロキシ URL。プロキシレットが有効の場合、このエントリを使用します。
gateway.httpsurl		ゲートウェイがページをリライトするためのカスタム URL 用の HTTPS 逆プロキシ URL。プロキシレットが有効の場合、このエントリを使用しないでください。
gateway.favicon		favicon.icon ファイルに対する要求をゲートウェイがリダイレクトする URL。 これは、Internet Explorer および Netscape 7.0 以降の「お気に入り」のアイコンとして使用されます。 何も指定しない場合、ゲートウェイはファイルが見つからないことを意味する 404 メッセージをブラウザに返します。
gateway.logging.password		ゲートウェイがアプリケーションセッションの作成に使用する「amService-srapGateway」ユーザーの LDAP パスワード。 暗号化された形式、プレーンテキストのいずれかを指定できます。
http.proxyHost		このプロキシホストが Portal Server へのアクセスに使用されます。

表 2-1 platform.conf ファイルのプロパティ (続き)

エントリ	デフォルト値	説明
http.proxyPort		Portal Server へのアクセスに使用されるホスト用のポート。
http.proxySet		プロキシホストが必要な場合は、このプロパティを true に設定します。 false に設定すると、http.proxyHost および http.proxyPort は無視されます。
portal.server.instance		このプロパティの値には、対応する /etc/opt/SUNWam/config/AMConfig-instance-name.properties ファイルを指定します。この値がデフォルトの場合は、AMConfig.properties をポイントします。
gateway.cdm.cacheSleepTime	60000	クライアント検出モジュールの応答が Access Manager からゲートウェイに送信したキャッシュのタイムアウト値。
gateway.cdm.cacheCleanupTime	300000	クライアント検出モジュールの応答が Access Manager からゲートウェイに送信したキャッシュのタイムアウト値。
netletproxy.port	10555	Netlet プロキシデーモンは、このポートで要求を待機します。
rewriterproxy.port	10555	リライタプロキシデーモンは、このポートで要求を待機します。
gateway.ignoreServerList	false	true に設定した場合、Access Manager サーバーの URL は AMConfig.properties ファイルで指定した値を使用して作成されます。Access Manager サーバーがロードバランサの背後にある場合、このプロパティを true に設定します。

ゲートウェイのインスタンスの作成

gwmultiinstance スクリプトを使用して、ゲートウェイのインスタンスを作成または削除します。このスクリプトは、ゲートウェイプロファイルが作成された後で実行します。

1. `root` としてログインし、次のディレクトリに移動します。

```
gateway-install-root/SUNWps/bin/
```

2. 次の複数インスタンススクリプトを実行します。

```
./gwmultiinstance
```

次のいずれかのインストールオプションを選択します。

- 1) Create a new gateway instance
- 2) Remove a gateway instance
- 3) Remove all gateway instances
- 4) Exit

1 を選択した場合は、次の質問に答えます。

```
What is the name of the new gateway instance?
```

```
What protocol will the new gateway instance use? [https]
```

```
What port will the new gateway instance listen on?
```

```
What is the fully qualified hostname of the portal server?
```

```
What port should be used to access the portal server?
```

```
What protocol should be used to access the portal server? [http]
```

```
What is the portal server deploy URI?
```

```
What is the organization DN? [dc=iportal,dc=com]
```

```
What is the Access Manager URI? [/amserver]
```

```
What is the Access Manager password encryption key?
```

```
Please provide the following information needed for creating a self-signed certificate:
```

```
What is the name of your organization?
```

```
What is the name of your division?
```

```
What is the name of your city or locality?
```

```
What is the name of your state or province?
```

What is the two-letter country code?

What is the password for the Certificate Database?Again?

What is the password for the logging user?Again?

Have you created the new gateway profile in the admin console? [y] /n

Start the gateway after installation? [y] /n

3. 新規ゲートウェイプロファイル名でゲートウェイの新規インスタンスを起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

gateway-profile-name は、ゲートウェイの新規インスタンスです。

ゲートウェイプロファイル以外に、AMConfig-*instance-name*.properties ファイルが /etc/opt/SUNWam/config ディレクトリに作成されます。

portal.server.instance プロパティが platform.conf ファイルに存在する場合は、対応する AMConfig-*instance-name*.properties ファイルがゲートウェイによって読み込まれます。portal.server.instance プロパティが platform.conf ファイルに存在しない場合は、デフォルトの AMConfig ファイル (AMConfig.properties) がゲートウェイによって読み込まれます。

マルチホームゲートウェイのインスタンスの作成

マルチホームゲートウェイのインスタンスを作成する場合、つまり 1 つの Portal Server 上に複数のゲートウェイを作成する場合は、platform.conf ファイルを次のように変更する必要があります。

```
gatewaybindipaddress= 0.0.0.0
```

同じ LDAP を使用するゲートウェイインスタンスの作成

最初のゲートウェイを作成したあとで、同じ LDAP を使用する複数のゲートウェイを作成する場合は、次の操作を行います。

`/etc/opt/SUNWam/config/` の `AMConfig-instance-name.properties` の次の領域を、最初にインストールしたゲートウェイインスタンスと一致するように変更します。

1. パスワードの暗号化と復号化に使用される鍵を、最初のゲートウェイに使用されている文字列に置き換えます。

```
am.encryption.pwd= string_key_specified_in_gateway-install
```

2. アプリケーション認証モジュールの共有シークレットである鍵を置き換えます。

```
com.ipplanet.am.service.secret= string_key_specified_in_gateway-install
```

3. `/etc/opt/SUNWam/config/ums` の `serverconfig.xml` の次の領域を、最初にインストールした Portal Server のインスタンスと一致するように変更します。

```
<DirDN> cn=puser,ou=DSAME Users,dc=sun,dc=net</DirDN>
```

```
<DirPassword>string_key_specified_in_gateway-install</DirPassword>
```

```
<DirDN>cn=dsameuser,ou=DSAME Users,dc=sun,dc=net</DirDN>
```

```
<DirPassword>string_key_specified_in_gateway-install</DirPassword>
```

4. `amserver` サービスを再起動します。

chroot 環境でのゲートウェイの実行

chroot 環境でセキュリティを高めるには、chroot ディレクトリのコンテンツを最小限に抑える必要があります。たとえば、chroot のディレクトリのファイルを修正できるプログラムが存在する場合、chroot はサーバーで chroot ツリーのファイルが攻撃者によって修正されるのを保護しません。CGI プログラムは bourne シェル、C シェル、Korn シェル、または Perl などのインタプリタ型言語では記述できませんが、バイナリにコンパイルする必要があるため、インタプリタが chroot ディレクトリツリーに存在する必要がありません。

注 chroot 環境では watchdog 機能はサポートされません。

▶ chroot をインストールするには

1. 端末ウィンドウで root として、次のファイルをネットワーク上のコンピュータ、バックアップテープ、フロッピーディスクなどの外部ソースにコピーします。

```
cp /etc/vfstab external-device
cp /etc/nsswitch.conf external-device
cp /etc/hosts external-device
```

2. 次の場所から mkchroot スクリプトを実行します。

```
portal-server-install-root/SUNWps/bin/chroot
```

注 mkchroot スクリプトの実行が開始すると、Ctrl-C でこのスクリプトを終了することはできません。

mkchroot スクリプトの実行中にエラーが発生した場合は、[56 ページの「mkchroot スクリプトの実行の失敗」](#)を参照してください。

別の root ディレクトリ (`new_root_directory`) が要求されます。スクリプトにより新しいディレクトリが作成されます。

次の例では、`new_root_directory` は `/safedir/chroot` です。

```
mkchroot version 6.0

Enter the full path name of the directory which will be the
chrooted tree:/safedir/chroot
Using /safedir/chroot as root.
Checking available disk space...done
/safedir/chroot is on a setuid mounted partition.
```

```

Creating filesystem structure...dev etc sbin usr var proc opt bin
lib tmp etc/lib usr/platform usr/bin usr/sbin usr/lib
usr/openwin/lib var/opt var/tmp dev/fd done
Creating devices...null tcp ticots ticlts ticotsord tty udp zero
conslog done
Copying/creating etc files...group passwd shadow hosts
resolv.conf netconfig nsswitch.conf
done
Copying binaries.....done
Copying libraries.....done
Copying zoneinfo (about 1 MB)..done
Copying locale info (about 5 MB).....done
Adding comments to /etc/nsswitch.conf ...done
Creating loopback mount for/safedir/chroot/usr/java1.2...done
Creating loopback mount for/safedir/chroot/proc...done
Creating loopback mount for/safedir/chroot/dev/random...done
Do you need /dev/fd (if you do not know what it means, press
return) [n]:
Updating /etc/vfstab...done
Creating a /safedir/chroot/etc/mnttab file, based on these
loopback mounts.
Copying SRAP related data ...
Using /safedir/chroot as root.
Creating filesystem structure.....done
mkchroot successfully done.

```

3. platform.conf ファイル内に記述されている Java ディレクトリを、次のコマンドを使用して chroot ディレクトリに手でマウントします。

```

mkdir -p /safedir/chroot/java-dir
mount -F lofs java-dir /safedir/chroot/java-dir

```

Solaris 9 の場合は次のコマンドを使用します。

```

mkdir -p /safedir/chroot/usr/lib/32
mount -F lofs /usr/lib/32 /safedir/chroot/usr/lib/32
mkdir -p /safedir/chroot/usr/lib/64
mount -F lofs /usr/lib/64 /safedir/chroot/usr/lib/64

```

システム起動時にこのディレクトリをマウントするには、/etc/vfstab ファイルに対応するエントリを追加します。

```

java-dir - /safedir/chroot/java-dir lofs - no -

```

Solaris 9 の場合は次を追加します。

```
/usr/lib/32 - /safedir/chroot/usr/lib/32 lofs - no -  
/usr/lib/64 - /safedir/chroot/usr/lib/64 lofs - no -
```

Linux の場合は次を追加します。

```
# mount red.iplanet.com:/misc/export /misc/local
```

各表記の意味は次のとおりです。

red.iplanet.com は NFS ファイルサーバーのホスト名です。

/misc/export は、red.iplanet.com がエクスポートしているファイルシステムです。

/misc/local は、ローカルマシン上でファイルシステムをマウントする場所です。

注：ローカルマシン上のマウントポイントディレクトリ (上の例では /misc/local) が存在する必要があります。

mount コマンドを実行した後 (および、クライアントが red.iplanet.com NFS サーバーからの適切なアクセス権を持つ場合)、クライアントユーザーはコマンド `ls /misc/local` を実行することによって、red.iplanet.com 上の /misc/export 内のファイルを一覧表示できます。

4. 次のコマンドを入力して、ゲートウェイを再起動します。

```
chroot /safedir/chroot ./gateway-install-root/SUNWps/bin/gateway start  
ゲートウェイを停止中 ... 完了。  
ゲートウェイを開始中 ...  
完了。
```

mkchroot スクリプトの実行の失敗

mkchroot スクリプトの実行中にエラーが発生した場合、スクリプトによりファイルは初期状態に復元されます。

次のサンプルでは、/safedir/chroot は chroot ディレクトリです。

次のエラーメッセージが表示される場合があります。

Not a Clean Exit

1. この場合、「[chroot をインストールするには](#)」の手順 1 で使用したバックアップファイルを元の場所にコピーし、次のコマンドを実行します。

```
umount /safedir/chroot/usr/java1.2  
umount /safedir/chroot/proc  
umount /safedir/chroot/dev/random
```

2. `/safedir/chroot` ディレクトリを削除します。

chroot 環境でのゲートウェイの再起動

ゲートウェイマシンを再起動した場合、次の手順を実行して chroot 環境でゲートウェイを再起動します。

▶ chroot 環境でゲートウェイを再起動するには

1. 「/」ディレクトリから実行中のゲートウェイを停止します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name stop
```

2. ゲートウェイを起動して、chroot ディレクトリから次のコマンドを実行します。

```
chroot /safedir/chroot ./portal-server-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

注 `/safedir/chroot/etc` ファイル (`passwd` や `hosts` など) は `/etc` ファイルのように管理する必要がありますが、chroot ツリーで実行中のプログラムが要求するホストとアカウント情報を追加するだけです。

たとえば、システムの IP アドレスを変更する場合は `/safedir/chroot/etc/hosts` も変更します。

ゲートウェイの起動と停止

デフォルトでは、ゲートウェイはユーザー `noaccess` として起動されます。

▶ ゲートウェイを起動するには

1. ゲートウェイをインストールし、必要なプロファイルを作成した後、次のコマンドを実行してゲートウェイを起動します。

```
gateway-install-root/SUNWps/bin/gateway -n default start
```

`default` はインストール時に作成されたデフォルトのゲートウェイプロファイルです。独自のプロファイルを後から作成し、その新しいプロファイルでゲートウェイを再起動することができます。[40 ページの「ゲートウェイプロファイルの作成」](#)を参照してください。

ゲートウェイのインスタンスが複数ある場合は、次のコマンドを使用します。

```
gateway-install-root/SUNWps/bin/gateway start
```

このコマンドにより、指定されたマシン上に設定されているすべてのゲートウェイインスタンスが起動します。

注 サーバー (ゲートウェイのインスタンスを設定したマシン) を再起動すると、ゲートウェイで設定されたすべてのインスタンスが再起動します。

 /etc/opt/SUNWps ディレクトリに古いプロファイルまたはバックアップ用のプロファイルが残っていないことを確認してください。

2. 指定されたポートでゲートウェイが稼働しているかどうかを確認する場合は、次のコマンドを実行します。

```
netstat -a | grep port-number
```

ゲートウェイのデフォルトのポートは、443 です。

▶ **ゲートウェイを停止するには**

1. ゲートウェイを停止するには、次のコマンドを実行します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name stop
```

ゲートウェイのインスタンスが複数ある場合は、次のコマンドを使用します。

```
gateway-install-root/SUNWps/bin/gateway stop
```

このコマンドにより、指定されたマシンで稼働するすべてのゲートウェイインスタンスが停止します。

2. ゲートウェイプロセスが稼働していないかどうかを確認する場合は、次のコマンドを実行します。

```
/usr/bin/ps -ef | grep entsys
```

ゲートウェイの再起動

通常はゲートウェイを再起動する必要はありません。再起動が必要なのは、次のいずれかに該当する場合だけです。

- 新規プロファイルを作成し、新しいプロファイルをゲートウェイに割り当てる必要がある。
- 既存のプロファイルの属性を修正し、変更を有効にする必要がある。
- OutOfMemory エラーなどによりゲートウェイがクラッシュする。
- ゲートウェイが異常停止し、要求に応答しない。

▶ 別のプロファイルでゲートウェイを再起動するには

ゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n new-gateway-profile-name start
```

▶ ゲートウェイを再起動するには

端末ウィンドウで root として接続し、次の操作を行います。

- 次の方法で watchdog プロセスを開始します。

```
gateway-install-root/SUNWps/bin/gateway watchdog on
```

crontab ユーティリティでエントリが作成され、watchdog プロセスが有効になります。watchdog は、特定のマシンおよびゲートウェイポートで実行されているすべてのゲートウェイインスタンスを監視し、停止しているゲートウェイを再起動します。

▶ ゲートウェイ watchdog を設定するには

watchdog がゲートウェイを監視する間隔を設定することができます。この間隔はデフォルトでは 60 秒に設定されています。これを変更する場合は、crontab ユーティリティで次の行を編集します。

```
0-59 * * * * gateway-install-root/SUNWps/bin/  
/var/opt/SUNWps/.gw. 5 > /dev/null 2>&1
```

crontab のエントリを設定する方法については、crontab のマニュアルページを参照してください。

仮想ホストの指定

仮想ホストとは、同じマシンの IP とホスト名をポイントする追加のホスト名のことで、たとえば、ホスト名 `a.b.c` がホスト IP アドレス `192.155.205.133` をポイントしている場合には、同じ IP アドレスをポイントする別のホスト名 `c.d.e` を追加できます。

▶ 仮想ホストを指定するには

1. `root` としてログインし、目的のゲートウェイインスタンスの `platform.conf` ファイルを編集します。

```
/etc/opt/SUNWps/platform.conf.gateway-profile-name
```

2. 次のエントリを追加します。

```
gateway.virtualhost=fully-qualified-gateway-host gateway-ip-address  
fully-qualified-reverse-proxyhost
```

```
gateway.enable.customurl=true (この値はデフォルトでは、false に設定されている)
```

3. ゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

これらの値を指定しない場合、ゲートウェイは通常どおりに動作します。

Access Manager へアクセスするプロキシの指定

ゲートウェイが、Portal Server に配備されている SRA コア (RemoteConfigServlet) にアクセスするために使用するプロキシホストを指定することができます。このプロキシは、Portal Server および Access Manager にアクセスするためにゲートウェイが使用します。

▶ プロキシを指定するには

1. コマンド行で、次のファイルを編集します。

```
/etc/opt/SUNWps/platform.conf.gateway-profile-name
```

2. 次のエントリを追加します。

```
http.proxyHost=proxy-host
```

```
http.proxyPort=proxy-port
```

```
http.proxySet=true
```

3. サーバーへ要求を行うために指定されたプロキシを使用するには、ゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

Web プロキシの使用

サードパーティー製の Web プロキシを使用して HTTP リソースにアクセスするよう
に、ゲートウェイを設定することができます。Web プロキシは、クライアントとイン
ターネットの間に設置されます。

Web プロキシの設定

ドメインおよびサブドメインごとに異なるプロキシを使用できます。これらのエン
トリから、特定のドメインの特定のサブドメインへのアクセスに使用するプロキシが
ゲートウェイに伝えられます。ゲートウェイで指定したプロキシ設定は次のように機
能します。

- ゲートウェイサービスの「ドメインとサブドメインのプロキシ」フィールドで、
必要なプロキシとドメインおよびサブドメインのリストを作成します。

ドメインとサブドメインのプロキシの設定については、[260 ページの「ドメイン
とサブドメインのプロキシのリストの作成」](#)を参照してください。
- 「プロキシを使用する」オプションを選択すると、次のような設定になります。
 - 指定されたホストに、「ドメインとサブドメインのプロキシ」フィールドで指定し
たプロキシが使用されます。
 - 「ドメインとサブドメインのプロキシ」リストで指定したドメインとサブドメ
イン内の、特定の URL に直接接続できるようにするには、「Web プロキシを使用しな
い URL」フィールドにその URL を指定します。
- 「プロキシを使用する」オプションを選択しない場合は、次のような設定になり
ます。
 - 「ドメインとサブドメインのプロキシ」フィールドで指定したドメインとサブド
メイン内の特定の URL にプロキシを使用するには、「Web プロキシを使用しない
URL」リストにその URL を指定します。

「プロキシの使用」オプションは無効になっていますが、「Web プロキシを使
用しない URL」リスト内の URL への接続にプロキシが使用されます。これ
らの URL のプロキシは、「ドメインとサブドメインのプロキシ」リストから
取得されます。

「プロキシを使用する」オプションの設定については、[258 ページの「Web プロキシ使
用の有効化」](#)を参照してください。

図 2-1 は、ゲートウェイサービスのプロキシ設定に基づいて Web プロキシ情報が解決
される手順を示しています。

図 2-1 Web プロキシの管理

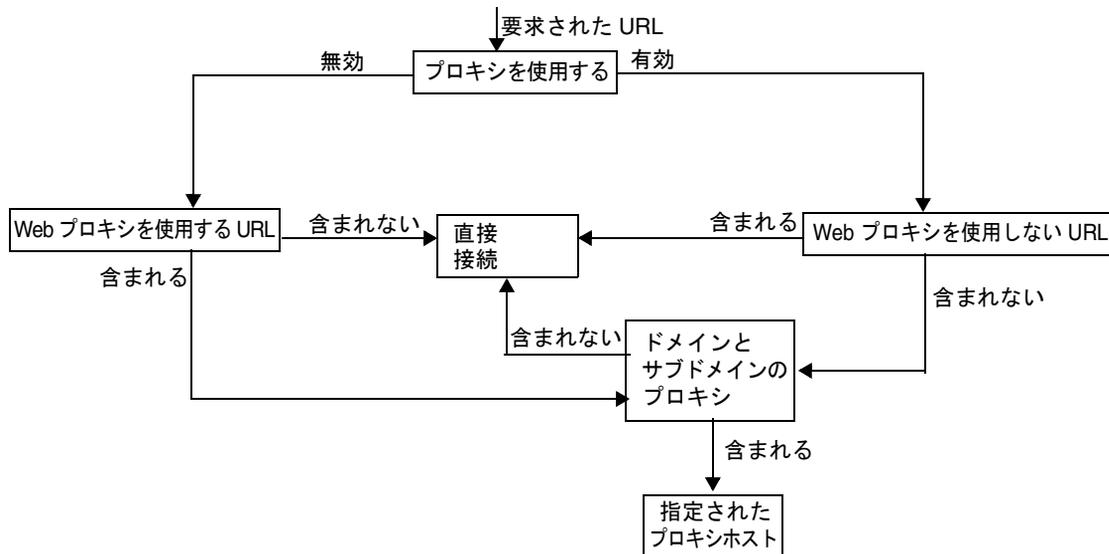


図 2-1 では、「プロキシを使用する」が選択され、「Web プロキシを使用しない URL」リストに要求された URL が含まれている場合、ゲートウェイは指定されたホストに直接接続します。

「プロキシを使用する」が選択され、「Web プロキシを使用しない URL」リストに要求された URL が含まれていない場合、ゲートウェイは指定されたプロキシを経由してホストに接続します。プロキシが指定されている場合は、「ドメインとサブドメインのプロキシ」リスト内でプロキシが検索されます。

「プロキシを使用する」が無効で、「Web プロキシを使用する URL」リストに要求された URL が含まれている場合、ゲートウェイは「ドメインとサブドメインのプロキシ」リストのプロキシ情報を使用して目的のホストに接続します。

「プロキシを使用する」が無効で、「Web プロキシを使用する URL」リストに要求された URL が含まれていない場合、ゲートウェイは指定されたホストに直接接続します。

上記のいずれの条件も満たさず、直接接続が不可能な場合は、ゲートウェイは接続不可を伝えるエラーを表示します。

注 標準のポータルデスクトップのブックマークチャネルを通じて URL にアクセスする場合、上記のいずれの条件にも合わない場合は、ゲートウェイはブラウザにリダイレクトを送信します。ブラウザは独自のプロキシ設定を使用して URL にアクセスします。

構文

```
domainname [web_proxy1:port1] | subdomain1 [web_proxy2:port2] | .....
```

例

```
sesta.com wp1:8080 | red wp2:8080 | yellow | * wp3:8080
```

* はすべてに一致するワイルドカードです。

各表記の意味は次のとおりです。

sesta.com はドメイン名、wp1 はポート 8080 にアクセスするプロキシです。

red はサブドメイン、wp2 はポート 8080 にアクセスするプロキシです。

yellow はサブドメインです。プロキシが指定されていないため、ドメインに指定されたプロキシ、つまりポート 8080 の wp1 が使用されます。

* は、他のすべてのサブドメインがポート 8080 で wp3 を使用する必要があることを表します。

注 デフォルトでは、ポートを指定しない場合ポート 8080 が使用されます。

Web プロキシ情報の処理

クライアントが特定の URL へのアクセスを試みると、URL のホスト名が「ドメインとサブドメインのプロキシ」リスト内のエントリと照合されます。指定されたホスト名で最も長いサフィックスに一致するエントリが選ばれます。たとえば、ホスト名 host1.sesta.com が要求されていると考えます。

- 「ドメインとサブドメインのプロキシ」リストで host1.sesta.com がスキャンされます。一致するエントリが見つかり、このエントリに指定されたプロキシがホストの接続に使用されます。
- 見つからなかった場合、リストで *.sesta.com がスキャンされます。エントリが見つかり、対応するプロキシが使用されます。
- 見つからなかった場合、リストで sesta.com がスキャンされます。エントリが見つかり、対応するプロキシが使用されます。
- 見つからなかった場合、リストで *.com がスキャンされます。エントリが見つかり、対応するプロキシが使用されます。
- 見つからなかった場合、リストで com がスキャンされます。エントリが見つかり、対応するプロキシが使用されます。
- 見つからなかった場合、リストで * がスキャンされます。エントリが見つかり、対応するプロキシが使用されます。
- 見つからなかった場合、直接接続が試みられます。

「ドメインとサブドメインのプロキシ」リストに次のようなエントリがあるとします。

```
com p1 | host1 p2 | host2 | * p3
sesta.com p4 | host5 p5 | * p6
florizon.com | host6
abc.sesta.com p8 | host7 p7 | host8 p8 | * p9
host6.florizon.com p10
host9.sesta.com p11
siroe.com | host12 p12 | host13 p13 | host14 | * p14
siroe.com | host15 p15 | host16 | * p16
* p17
```

ゲートウェイは、表 2-2 に示されるテーブルでこれらのエントリを内部的にマッピングします。

表 2-2 ドメインとサブドメインのプロキシリストのエントリのマッピング

番号	「ドメインとサブドメインのプロキシ」リストのエントリ	プロキシ	説明
1	com	p1	リストで指定されたプロキシ
2	host1.com	p2	リストで指定されたプロキシ
3	host2.com	p1	host2 に対してプロキシが指定されないため、ドメインのプロキシが使用されません。
4	*.com	p3	リストで指定されたプロキシ
5	sesta.com	p4	リストで指定されたプロキシ
6	host5.sesta.com	p5	リストで指定されたプロキシ
7	*.sesta.com	p6	リストで指定されたプロキシ
8	florizon.com	直接	詳細はエントリ 14 の説明を参照
9	host6.florizon.com	-	詳細はエントリ 14 の説明を参照
10	abc.sesta.com	p8	リストで指定されたプロキシ
11	host7.abc.sesta.com	p7	リストで指定されたプロキシ
12	host8.abc.sesta.com	p8	リストで指定されたプロキシ
13	*.abc.sesta.com	p9	リストで指定されたプロキシ abc.sesta.com ドメインの host7 と host8 以外のすべてのホストについては、p9 がプロキシとして使用されます。

表 2-2 ドメインとサブドメインのプロキシリストのエントリのマッピング (続き)

番号	「ドメインとサブドメインのプロキシ」リストのエントリ	プロキシ	説明
14	host6.florizon.com	p10	エントリ 9 と同じエントリ。エントリ 9 は直接接続を指定するのに対し、このエントリはプロキシ p10 の使用を指定します。このような 2 つのエントリがある場合、プロキシ情報のあるエントリが有効なエントリと見なされます。もう 1 つのエントリは無視されます。
15	host9.sesta.com	p11	リストで指定されたプロキシ
16	siroe.com	直接	siroe.com に対して指定されるプロキシがないため、直接接続が試みられません。
17	host12.siroe.com	p12	リストで指定されたプロキシ
18	host13.siroe.com	p13	リストで指定されたプロキシ
19	host14.siroe.com	直接	host14 に対して指定されるプロキシがないため、直接接続が試みられます。
20	*.siroe.com	p14	エントリ 23 の説明を参照
21	host15.siroe.com	p15	リストで指定されたプロキシ
22	host16.siroe.com	直接	host16 または siroe.com に対して指定されるプロキシがないため、直接接続が試みられます。
23	*.siroe.com	p16	エントリ 20 に類似していますが、指定されるプロキシが異なります。このような場合、ゲートウェイの正確な動作がわかりません。2 つのプロキシのいずれかが使用されます。
24	*	p17	要求された URL に一致するエントリが存在しない場合、プロキシとして p17 が使用されます。

注 「ドメインとサブドメインのプロキシ」リストでは、プロキシエントリを「|」記号で区切らずに、リストに個別に入力する方が簡単です。たとえば、次のように表記されるエントリがあるとします。

```
sesta.com p1 | red p2 | * p3
```

このエントリは次のように指定できます。

```
sesta.com p1
```

```
red.sesta.com p2
```

```
*.sesta.com p3
```

反復されたエントリやその他のあいまいなエントリを見つけやすくなります。

ドメインとサブドメインのプロキシリストに基づくリライト

リライトも、「ドメインとサブドメインのプロキシ」リストのエントリを使用します。リライトは、ドメインが「ドメインとサブドメインのプロキシ」リストのドメインに一致するすべての URL をリライトします。

警告 「ドメインとサブドメインのプロキシ」リストのエントリ * は、リライトの対象と見なされません。たとえば、表 2-2 の例では、エントリ 24 はリライトの対象になりません。

リライトについては、第 3 章「プロキシレットとリライト」を参照してください。

デフォルトのドメインとサブドメイン

URL の最終ホストが完全修飾名になっていない場合、完全修飾名に到達するためにデフォルトのドメインおよびサブドメインが使用されます。

管理コンソールの「デフォルトのドメイン」フィールドに、次のエントリが設定されていると仮定します。

```
red.sesta.com
```

注 「ドメインとサブドメインのプロキシ」リストには、対応するエントリが必要です。

上記の例では、sesta.com がデフォルトのドメイン、デフォルトのサブドメインは red です。

URL、host1 が要求された場合、これはデフォルトのドメインとサブドメインを使用して host1.red.sesta.com として解決されます。「ドメインとサブドメインのプロキシ」リストで host1.red.sesta.com が検索されます。

自動プロキシ設定の使用

「ドメインとサブドメインのプロキシ」リストの情報を無視するには、自動プロキシ設定機能を有効にします。この設定については、[262 ページの「自動プロキシ設定サポートの有効化」](#)を参照してください。

プロキシ自動設定 (PAC) ファイルを使用するときは、次の点に注意してください。

- Portal Server、ゲートウェイ、Netlet、およびプロキシレットは、Rhino ソフトウェアを使用して PAC ファイルをパースします。SUNWrhino パッケージは、JavaTM Enterprise System アクセサリ CD からインストールできます。
このパッケージに含まれている js.jar ファイルは、/usr/share/lib ディレクトリに存在している必要があります。このディレクトリは、ゲートウェイおよび Portal Server マシンの webserver/appserver クラスパスに追加してください。このクラスパスに見つからなかった場合、Portal Server、ゲートウェイ、Netlet、およびプロキシレットは PAC ファイルをパースできません。
- ゲートウェイマシンの \$JRE_HOME/lib/ext ディレクトリに js.jar が存在する必要があります。このファイルが存在しない場合、ゲートウェイは PAC ファイルをパースできません。
- ゲートウェイは起動時に、ゲートウェイプロファイルの「自動プロキシ設定ファイルの位置」フィールドに指定されている場所から PAC ファイルをフェッチします。場所の設定については、[263 ページの「自動プロキシ設定ファイルの場所の指定」](#)を参照してください。
- ゲートウェイは、URLConnection API を使用してこの場所にアクセスします。ゲートウェイにアクセスするようにプロキシを設定しなければならないときは、プロキシを次のように設定します。

- a. コマンド行で、次のファイルを編集します。

```
/etc/opt/SUNWps/platform.conf.gateway-profile-name
```

- b. 次のエントリを追加します。

```
http.proxyHost=web-proxy-hostname
```

```
http.proxyPort=web-proxy-port
```

```
http.proxySet=true
```

- c. 指定のプロキシを使用するために、ゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

- PAC ファイルの初期化に失敗した場合は、ゲートウェイは「ドメインとサブドメインのプロキシ」リストの情報を使用します。
- PAC ファイルから空白文字列または「NULL」が返される場合、ゲートウェイはそのホストがイントラネットに属していないと判断します。これは、「ドメインとサブドメインのプロキシ」に含まれないホストの扱いと似ています。

ゲートウェイにホストへの直接接続を使用させたい場合は、「DIRECT」を返します。68 ページの「[DIRECT または NULL のいずれかが返される例](#)」を参照してください。

- 複数のプロキシが指定されている場合、ゲートウェイは最初に返されるプロキシだけを使用します。ホストに指定されている複数のプロキシの間で、フェイルオーバーやロードバランスは行われません。
- ゲートウェイは SOCKS プロキシを無視して直接接続を試み、ホストがイントラネットの一部であると解釈します。
- イン트라ネットの一部に含まれないホストへのアクセスに使用するプロキシを指定するには、「STARPROXY」というプロキシタイプを使用します。これは、PAC 形式のファイル拡張子で、ゲートウェイプロファイルの「ドメインとサブドメインのプロキシ」セクションに指定される * proxyHost:port エントリと似ています。69 ページの「[STARPROXY が返される例](#)」を参照してください。

サンプル PAC ファイルの使用

次の例は、「ドメインとサブドメインのプロキシ」リストに含まれる URL と、それに対応する PAC ファイルを示しています。

DIRECT または NULL のいずれかが返される例

次の「ドメインとサブドメインのプロキシ」を使用する場合を考えます。

```
*intranet1.com proxy.intranet.com:8080
```

```
intranet2.com proxy.intranet1.com:8080
```

対応する PAC ファイルは次のようになります。

```
// Start of the PAC File
```

```
function FindProxyForURL(url, host) {
    if (dnsDomainIs(host, ".intranet1.com")) {
        return "DIRECT";
    }
}
```

```

        if (dnsDomainIs(host, ".intranet2.com")) {
            return "PROXY proxy.intranet1.com:8080";
        }
        return "NULL";
    }
}
//End of the PAC File

```

STARPROXY が返される例

次の「ドメインとサブドメインのプロキシ」を使用する場合は考えます。

```

intranet1.com
intranet2.com.proxy.intranet1.com:8080
internetproxy.intranet1.com:80

```

対応する PAC ファイルは次のようになります。

```

// Start of the PAC File
function FindProxyForURL(url, host) {
    if (dnsDomainIs(host, ".intranet1.com")) {
        return "DIRECT";
    }
    if (dnsDomainIs(host, ".intranet2.com")) {
        return "PROXY proxy.intranet1.com:8080;" +
            "PROXY proxy1.intranet1.com:8080";
    }
    return "STARPROXY internetproxy.intranet1.com:80";
}
//End of the PAC File

```

この場合、要求が `.intranet2.com` domain 内のホストに対するものであれば、ゲートウェイは `proxy.intranet1.com:8080` にアクセスします。 `proxy.intranet1.com:8080` がダウンしている場合、要求は失敗します。ゲートウェイは、フェイルオーバーを行わず、 `proxy1.intranet1.com:8080` へアクセスします。

Netlet プロキシの使用

Netlet パケットはゲートウェイで解読され、宛先サーバーに送られます。ただし、ゲートウェイはすべての Netlet 宛先ホストにアクセスする場合、非武装ゾーン (DMZ) とイントラネット間のファイアウォールを経由する必要があります。これにはファイアウォールで多くのポートを開かなければなりません。Netlet プロキシを使用することで、ファイアウォールで開かれるポートの数を最小化することができます。

Netlet プロキシは、ゲートウェイを経由してクライアントからの安全なトンネルをイントラネット内の Netlet プロキシまで拡張することで、ゲートウェイとイントラネット間のセキュリティを補強します。プロキシを使用すると、Netlet パケットが Netlet プロキシにより解読され、送信先に送られます。

Netlet プロキシは、次のような点で便利です。

- セキュリティのレイヤーを補強します。
- 配備サイズが大きな環境で、ゲートウェイから内部ファイアウォールに必要以上の IP アドレスおよびポートを使用しないようにします。
- ゲートウェイと Portal Server 間で開かれるポートの数を 1 つに制限します。このポート数はインストール時に設定できます。
- [図 2-2](#) の「Netlet プロキシをインストールした場合」に示すように、クライアントとゲートウェイ間の安全なチャネルを Portal Server まで延長します。Netlet プロキシはデータの暗号化によってセキュリティを改善しますが、システムリソースの使用を増やす場合があります。Netlet プロキシのインストールについては、『Sun Java Enterprise System インストールガイド』を参照してください。

次の処理を実行できます。

- Portal Server ノードまたは別のノードで Netlet プロキシのインストールを選択します。
- 複数の Netlet プロキシをインストールし、それらを管理コンソールで単一のゲートウェイに対して設定します。これはロードバランスに役立ちます。詳細については、[244 ページの「Netlet プロキシの有効化とリストの作成」](#)を参照してください。
- 単一のマシンで Netlet プロキシの複数のインスタンスを設定します。
- ゲートウェイの複数のインスタンスに対して、Netlet プロキシの単一のインストールを設定します。
- Web プロキシ経由のトンネル Netlet。この設定については、[263 ページの「Web プロキシを通じた Netlet トンネリングの有効化」](#)を参照してください。

図 2-2 は、Netlet プロキシをインストールした場合とインストールしない場合のゲートウェイと Portal Server の 3 つの実装例を示しています。クライアント、2 つのファイアウォール、2 つのファイアウォールの間にあるゲートウェイ、Portal Server、および Netlet 宛先サーバーから構成されます。

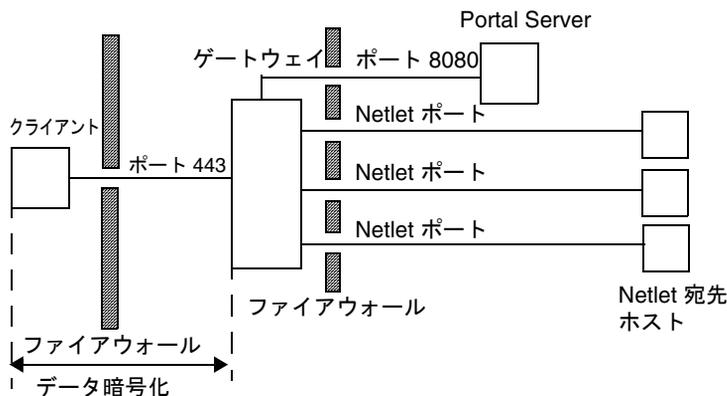
最初の例では、Netlet プロキシをインストールしていないゲートウェイと Portal Server を示しています。ここでは、クライアントからゲートウェイの間だけでデータの暗号化が行われます。Netlet 接続の要求があるたびに、2 番目のファイアウォールでポートが開かれます。

2 番目の例では、ゲートウェイと、Netlet プロキシがインストールされている Portal Server を示しています。この場合、データの暗号化はクライアントから Portal Server までのすべての区間に拡張されています。すべての Netlet が Netlet プロキシを通じてルーティングされているため、Netlet 要求に対して 2 番目のファイアウォールで開く必要があるのは 1 つのポートのみです。

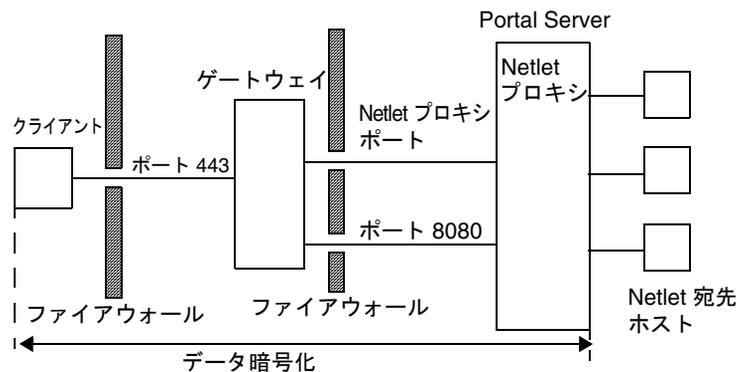
3 番目の例では、Netlet プロキシが別のノードにインストールされている Portal Server とゲートウェイを示しています。別のノードに Netlet プロキシをインストールすると、Portal Server ノードの負荷が減少します。ここでも、2 番目のファイアウォールで開く必要があるのは 2 つのポートのみです。1 つのポートは Portal Server への要求を処理し、もう 1 つのポートは Netlet の要求を Netlet プロキシサーバーにルーティングします。

図 2-2 Netlet プロキシの実装

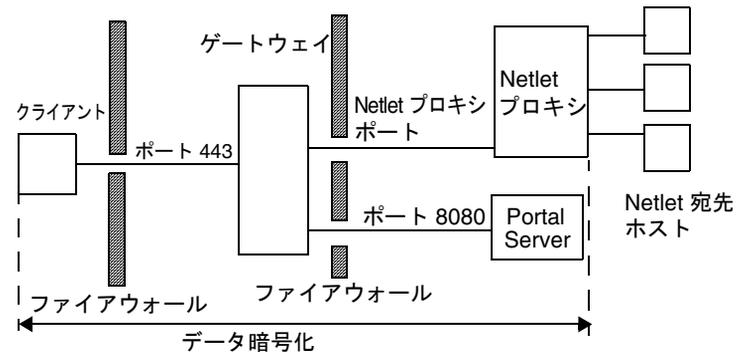
Netlet プロキシを設定しない場合



Netlet プロキシを Portal Server にインストールした場合



Netlet プロキシを別のノードにインストールした場合



Netlet プロキシのインスタンスの作成

Netlet プロキシの新しいインスタンスを Portal Server ノードまたは別のノードに作成するときは、nlpmultiinstance スクリプトを使用します。このスクリプトは、ゲートウェイプロファイルが作成された後で実行します。

1. root としてログインし、次のディレクトリに移動します。

```
netlet-install-dir/SUNWps/bin
```

2. 次の複数インスタンススクリプトを実行します。

```
./nlpmultiinstance
```

3. nlpmultiinstance スクリプトが表示する質問に答えます。
 - What is the name of the new netlet proxy instance?
 - このノードに同じ名前でインスタンスが設定されている場合は、この Netlet プロキシインスタンスで同じ設定を使用するかどうかの確認が求められます。
 - yes を指定した場合は、次の 2 つの質問に答えます。
 - What port will the new netlet proxy instance listen on?
 - Start the netlet proxy after installation?
 - no を指定した場合は、次の質問に答えます。
 - What protocol will the new netlet proxy instance use?
 - What port will the new netlet proxy instance listen on?
 - What is the name of your organization?
 - What is the name of your division?
 - What is the name of your city or locality?
 - What is the name of your state or province?
 - What is the two-letter country code?
 - What is the password for the certificate Database?
 - What is the password for the logging user?
 - Have you created the new gateway profile in the admin console?
 - If you answered yes, start the netlet proxy after installation?
4. 適切なゲートウェイプロファイル名で Netlet プロキシの新規インスタンスを起動します。

```
netlet-proxy-install-root/SUNWps/bin/netletd -n gateway-profile-name start
```

ここで *gateway-profile-name* は必要なゲートウェイインスタンスのプロファイル名です。

Netlet プロキシの有効化

Netlet プロキシを有効化するときは、Access Manager 管理コンソールの「SRA 設定」の下にある「ゲートウェイ」サービスを使用します。244 ページの「Netlet プロキシの有効化とリストの作成」を参照してください。

Netlet プロキシの再起動

プロキシが何らかの理由で強制終了した場合に再起動するように、Netlet プロキシを設定することができます。Netlet プロキシを監視し、Netlet プロキシが停止したときに再起動するように watchdog プロセスをスケジューリングできます。

Netlet プロキシは手動で再起動することもできます。

▶ Netlet プロキシを再起動するには

端末ウィンドウで root として接続し、次の操作を行います。

- 次の方法で watchdog プロセスを開始します。

```
netlet-proxy-install-root/SUNWps/bin/netletd watchdog on
```

crontab ユーティリティでエントリが作成され、watchdog プロセスが有効になります。watchdog は Netlet プロキシポートを監視し、ポートが停止した場合にプロキシを再起動します。

- 次の方法で、Netlet プロキシを手動で起動します。

```
netlet-proxy-install-root/SUNWps/bin/netletd -n gateway-profile-name start
```

ここで *gateway-profile-name* は必要なゲートウェイインスタンスのプロファイル名です。

▶ Netlet プロキシの watchdog を設定するには

watchdog が Netlet プロキシの状態を監視する間隔を設定することができます。この間隔はデフォルトでは 60 秒に設定されています。これを変更する場合は、crontab ユーティリティで次の行を編集します。

```
0-59 * * * * netlet-install-dir/bin/checkgw /var/opt/SUNWps/.gw 5 >
/dev/null 2>&1
```

リライタプロキシの使用

リライタプロキシは、イントラネット上にインストールされます。ゲートウェイは、コンテンツを直接取得せずにすべての要求をリライタプロキシに送信し、リライタプロキシはコンテンツをフェッチしてゲートウェイに返します。

リライタプロキシを使用する2つの利点を次に示します。

- ゲートウェイとサーバー間にファイアウォールが存在する場合、ファイアウォールが開放する必要があるのは2つのポートに対してのみです。1つはゲートウェイとリライタプロキシの間のポート、もう1つはゲートウェイと Portal Server の間のポートです。
- 送信先のサーバーが (HTTPS ではなく) HTTP プロトコルのみをサポートしている場合でも、ゲートウェイとイントラネットの間の HTTP トラフィックは安全です。

リライタプロキシを指定しない場合、いずれかのイントラネットコンピュータにアクセスしようとする、ゲートウェイコンポーネントによりイントラネットコンピュータに直接つながります。

リライタプロキシをロードバランサとして使用する場合は、リライタの `platform.conf.instance_name` がロードバランサ URL をポイントしている必要があります。また、ロードバランサのホストが Portal Servers リストに指定されている必要があります。

リライタプロキシの複数インスタンスをゲートウェイの各インスタンスに割り当てる場合 (Portal Server ノード上でなくてもかまわない) には、`platform.conf` ファイルで、リライタプロキシに対して1つのポートエントリを入力するのではなく、`host-name:port` の形式でリライタプロキシごとに詳細を入力します。

リライタプロキシのインスタンスの作成

リライタプロキシの新しいインスタンスを Portal Server ノードに作成するときは、`rwpmultiinstance` スクリプトを使用します。このスクリプトは、ゲートウェイプロファイルが作成された後で実行します。

1. `root` としてログインし、次のディレクトリに移動します。

```
rewriter-proxy-install-root/SUNWps/bin
```

2. 次の複数インスタンスのスクリプトを実行します。

```
./rwpmultiinstance
```

3. スクリプトが表示する質問に答えます。

- What is the name of the new rewriter proxy instance?
- このノードに同じ名前でもリライタプロキシインスタンスが設定されている場合は、このリライタプロキシインスタンスで同じ設定を使用するかどうかの確認が求められます。
- `yes` を指定した場合は、次の 2 つの質問に答えます。
 - What port will the new rewriter proxy instance listen on?
 - Start the rewriter proxy after installation?
- `no` を指定した場合は、次の質問に答えます。
 - What protocol will the new rewriter proxy instance use?
 - What port will the new rewriter proxy instance listen on?
 - What is the name of your organization?
 - What is the name of your division?
 - What is the name of your city or locality?
 - What is the name of your state or province?
 - What is the two-letter country code?
 - What is the password for the certificate Database?
 - What is the password for the logging user?
 - Have you created the new gateway profile in the admin console?
 - If you answered yes, start the rewriter proxy after installation?

4. 適切なゲートウェイプロファイル名でリライタプロキシの新規インスタンスを起動します。

```
rewriter-proxy-install-root/SUNWps/bin/rwproxyd -n gateway-profile-name start
```

ここで *gateway-profile-name* は必要なゲートウェイインスタンスのプロファイル名です。

リライタプロキシの有効化

リライタプロキシを有効化するときは、Access Manager 管理コンソールの「SRA 設定」の下にある「ゲートウェイ」サービスを使用します。242 ページの「リライタプロキシの有効化とリストの作成」を参照してください。

リライタプロキシの再起動

プロキシが何らかの理由で強制終了した場合に、リライタプロキシが再起動するように設定することができます。リライタプロキシを監視し、リライタプロキシが強制終了したときに再起動するように *watchdog* プロセスをスケジューリングできます。

リライタプロキシは手動で再起動することもできます。

▶ リライタプロキシを再起動するには

端末ウィンドウで *root* として接続し、次の操作を行います。

- 次の方法で *watchdog* プロセスを開始します。

```
rewriter-proxy-install-root/SUNWps/bin/rwproxd watchdog on
```

crontab ユーティリティでエントリが作成され、*watchdog* プロセスが有効になります。*watchdog* はポートを監視し、ポートが停止した場合にプロキシを再起動します。

- 手動で起動するには、次のとおり実行します。

```
rewriter-proxy-install-root/SUNWps/bin/rwproxd -n gateway-profile-name start
```

ここで *gateway-profile-name* は必要なゲートウェイインスタンスのプロファイル名です。

▶ リライタプロキシの *watchdog* を設定するには

watchdog がリライタプロキシの状態を監視する間隔を設定することができます。この間隔はデフォルトでは 60 秒に設定されています。これを変更する場合は、*crontab* ユーティリティで次の行を編集します。

```
0-59 * * * * rewriter-proxy-install-root/bin/checkgw /var/opt/SUNWps/.gw 5 > /dev/null 2>&1
```

ゲートウェイでの逆プロキシの使用

プロキシサーバーがインターネットのコンテンツをイントラネットに配信するのに対して、逆プロキシサーバーはイントラネットのコンテンツをインターネットに配信します。逆プロキシを配備するときに、インターネットコンテンツのロードバランスおよびキャッシングが行われるように設定できます。

ゲートウェイの前にサードパーティーの逆プロキシがある配備の場合、応答は、ゲートウェイの URL ではなく逆プロキシの URL でリライトされる必要があります。このためには、次のように設定します。

▶ 逆プロキシを有効化するには

1. root としてログインし、目的のゲートウェイインスタンスの `platform.conf` ファイルを編集します。

```
/etc/opt/SUNWps/platform.conf.gateway-profile-name
```

2. 次のエントリを追加します。

```
gateway.virtualhost=fully-qualified-gateway-host gateway-ip-address  
fully-qualified-reverse-proxyhost
```

```
gateway.enable.customurl=true (この値はデフォルトでは、false に設定されている)
```

```
gateway.httpurl=http reverse-proxy-URL
```

```
gateway.httpsurl=https reverse-proxy-URL
```

`gateway.httpurl` は、ゲートウェイプロファイルに HTTP ポートとしてリストされているポートで受信される要求への応答をリライトするために使用されます。

`gateway.httpsurl` は、ゲートウェイプロファイルに HTTPS ポートとしてリストされているポートで受信される要求への応答をリライトするために使用されます。

3. ゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

これらの値を指定しない場合、ゲートウェイは通常どおりに動作します。

クライアント情報の取得

ゲートウェイがいずれかの内部サーバーにクライアント要求を転送するときに、HTTP 要求に HTTP ヘッダーが追加されます。3つのヘッダーを使用して追加のクライアント情報を取得し、ゲートウェイの存在を検出することができます。

HTTP 要求ヘッダーを表示するには、platform.conf ファイル内のエントリを gateway.error=message に設定し、サーブレット API から request.getHeader() を使用します。次の表は、HTTP ヘッダー内の情報を示しています。

表 2-3 HTTP ヘッダー内の情報

ヘッダー	構文	説明
PS-GW-PDC	X-PS-GW- PDC: true/false	ゲートウェイで PDC が有効であるかどうかを示します。
PS-Netlet	X-PS-Netlet:enable d=true/false	<p>ゲートウェイで Netlet が有効化されているか、それとも無効化されているかを示します。</p> <p>Netlet が有効化されている場合は、暗号化オプションが生成され、ゲートウェイが HTTPS モード (encryption=ssl) または HTTP モード (encryption=plain) のどちらで実行されているかが示されます。</p> <p>例</p> <p>PS-Netlet: enabled=false</p> <p>Netlet は無効化されています。</p> <p>PS-Netlet: enabled=true; encryption=ssl</p> <p>Netlet は有効で、ゲートウェイは SSL モードで稼動しています。</p> <p>Netlet が有効でない場合は、encryption=ssl/plain は生成されません。</p>
PS-GW-URL	X-PS-GW-URL: http(s)://gateway URL(:port)	<p>クライアントが接続している URL を示します。</p> <p>ポートが標準ポートでない場合 (つまり、80/443 以外のポートでゲートウェイが HTTP/HTTPS モードで稼動している場合) は、「:port」が追加されます。</p>

表 2-3 HTTP ヘッダー内の情報 (続き)

ヘッダー	構文	説明
PS-GW-Rewriting-URL	X-PS-GW-URL: http(s)://gateway URL(:port)/[SessionInfo]	<p>ゲートウェイがすべてのページをリライトする URL を示します。</p> <ol style="list-style-type: none"> 1. ブラウザが Cookie をサポートする場合、このヘッダーの値は PS-GW-URL ヘッダーと同じです。 2. ブラウザが Cookie をサポートしない場合は、次のようになります。 <ul style="list-style-type: none"> • 宛先ホストが「ユーザーセッション Cookie を転送する URL」フィールドに含まれる場合は、ゲートウェイがページをリライトする実際の URL (コード化された セッション ID 情報が含まれる) • 宛先ホストが「ユーザーセッション Cookie を転送する URL」フィールドに含まれない場合は、SessionInfo 文字列は「\$SessionID」となる <p>注 : 認証ページからの応答のように、応答の過程でユーザーの Access Manager のセッション ID が変更された場合、ページは、それまでヘッダーに指定されていた値ではなくその値で書き換えられます。</p> <p>例</p> <ul style="list-style-type: none"> • ブラウザが Cookie をサポートする場合 <pre>PS-GW-Rewriting-URL: https://siroe.india.sun.com:10443/</pre> • ブラウザが Cookie をサポートせず、エンドサーバーが「ユーザーセッション Cookie を転送する URL」フィールドに含まれる場合 <pre>PS-GW-Rewriting-URL: https://siroe.india.sun.com:10443/SessIDValCustomEncodedValue/</pre> • ブラウザが Cookie をサポートしないが、エンドサーバーが「ユーザーセッション Cookie を転送する URL」フィールドに含まれない場合 <pre>PS-GW-Rewriting-URL: https://siroe.india.sun.com:10443/\$SessionID</pre>

表 2-3 HTTP ヘッダー内の情報 (続き)

ヘッダー	構文	説明
PS-GW-ClientIP	X-PS-GW-ClientIP: IP	ゲートウェイが receivedSocket.getInetAddress().getHostAddress() から取得した IP。 クライアントがゲートウェイに直接接続する場合、 これによって IP が特定されます。

認証連鎖の使用

認証連鎖することにより、通常の認証メカニズムを超えた高いレベルのセキュリティがもたらされます。ユーザーを複数の認証メカニズムで認証することができます。

ここでは、PDC (Personal Digital Certificate) 認証によってゲートウェイで認証連鎖を有効化する手順だけを説明します。PDC 認証を使用しない場合のゲートウェイでの認証連鎖については、『Access Manager 管理ガイド』を参照してください。

たとえば、PDC と Radius 認証モジュールを連鎖させると、ユーザーは標準のポータルデスクトップにアクセスするために 3 つのモジュールすべてについて認証が必要になります。

注 PDC が有効になっていると、PDC が常に最初の認証モジュールとしてユーザーに提示されます。

▶ 既存の PDC インスタンスに認証モジュールを追加するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 適切な組織を選択します。
3. 「表示」 ドロップダウンメニューから「サービス」を選択します。
左の区画にサービスが表示されます。
4. 「認証設定」の隣の矢印をクリックします。
「サービスインスタンスリスト」が表示されます。
5. gatewaypdc をクリックします。
「gatewaypdc プロパティを表示」ページが表示されます。
6. 「認証設定」の「編集」のリンクをクリックします。
「モジュールの追加」が表示されます。

7. 「モジュール名」を選択し、「適用基準」を「必修」に設定します。オプションは空白のまま残せます。
8. 「了解」をクリックします。
9. 1つまたは複数のモジュールを追加したら、「保存」をクリックします。
10. 「gatewaypc プロパティの表示」ページをクリックします。
11. 変更を有効にするために、ゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name  
start
```

ワイルドカード証明書の使用

ワイルドカード証明書は、ホストの完全修飾 DNS 名にワイルドカード文字を含む単一の証明書を受け付けます。

これによって、同じドメイン内で証明書が複数のホストを保証することが可能になります。たとえば、*.domain.com の証明書は abc.domain.com と abc1.domain.com に使用できます。実際には、この証明書は domain.com ドメイン内のすべてのホストに有効です。

ブラウザキャッシングの無効化

ゲートウェイコンポーネントは Web ブラウザのみを使用して任意の場所からバックエンド企業データへの安全なアクセスを提供します。そのため、クライアントが情報をローカルにキャッシュする必要がない場合があります。

ゲートウェイを通じてリダイレクトされるページのキャッシングを無効にするには、そのゲートウェイの platform.conf ファイルの属性を修正します。

このオプションを無効にすると、ゲートウェイのパフォーマンスに影響する場合があります。標準のポータルデスクトップが再表示されるたびに、ブラウザがすでにキャッシュしているイメージを含めページが参照するすべてのデータをゲートウェイで取り出す必要があるためです。ただし、この機能を有効にしても、リモートアクセスされた安全なコンテンツの足跡は、クライアントサイトでキャッシュとして残りません。これがパフォーマンスへの影響よりも重要な意味を持つのは、企業 IT の制御下でないインターネットカフェやその類のリモートロケーションから企業ネットワークにアクセスしている場合です。

▶ ブラウザキャッシングを無効にする手順

1. root としてログインし、目的のゲートウェイインスタンスの platform.conf ファイルを編集します。

```
/etc/opt/SUNWps/platform.conf.gateway-profile-name
```

2. 次の行を編集します。

```
gateway.allow.client.caching=true
```

この値はデフォルトでは、true に設定されています。この値を false に変更するとクライアントサイドでのブラウザキャッシングが無効になります。

3. ゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

ゲートウェイサービスのユーザーインターフェースのカスタマイズ

ここでは、編集可能な各種のプロパティファイルについて説明します。

srapGateway.properties ファイル

このファイルは、次の目的のために編集できます。

- ゲートウェイの実行時に表示されるエラーメッセージをカスタマイズします。
 - HTML-CharSets=ISO-8859-1 は、このファイルの作成に使用された文字セットを示しています。
 - 中カッコで囲まれた番号 ({0} など) は、実行時に表示される値です。この番号に対応するラベルを変更できます。また、必要に応じてラベルを並べ替えることができます。番号とメッセージは関連付けられるため、表示されるメッセージにラベルが対応していることを確認してください。
- ログ情報をカスタマイズします。

デフォルトでは、srapGateway.properties ファイルは

portal-server-install-root/SUNWps/locale ディレクトリ内にあります。ゲートウェイマシンに表示されるすべてのメッセージ (ゲートウェイ関連のメッセージ) は、メッセージの言語に関わりなく、このファイルに格納されます。

クライアントの標準のポータルデスクトップに表示されるメッセージの言語を変更する必要がある場合、このファイルを

portal-server-install-root/SUNWps/locale_en_US などの各ロケールのディレクトリにコピーします。

srapgwadminmsg.properties ファイル

このファイルは、次の目的のために編集できます。

- 管理コンソールのゲートウェイサービスのボタンとして表示されるラベルをカスタマイズします。
- ゲートウェイを設定しているときに表示される状況メッセージとエラーメッセージをカスタマイズします。

連携管理の使用

連携管理により、ユーザーが1つのネットワーク ID を持つように、ユーザーはユーザーのローカル ID を収集できます。連携管理ではネットワーク ID を使用して、ユーザーによる1つのサービスプロバイダサイトへのログインを許可し、ID を再認証することなく、他のサービスプロバイダサイトへのアクセスを許可します。これをシングルサインオンと呼びます。

Portal Server では、連携管理をオープンモードとセキュアモードに設定できます。連携管理をオープンモードに設定する方法については、『Portal Server 管理ガイド』を参照してください。Secure Remote Access を使用して連携管理を設定する前に、これがオープンモードで機能することを確認します。ユーザーが同じブラウザで連携管理をオープンモードとセキュアモードの両方で使用できるようにするには、ブラウザから Cookie とキャッシュをクリアする必要があります。

連携管理の詳細については、『Access Manager Federation Management Guide』を参照してください。

連携管理の例

ユーザーは、最初のサービスプロバイダに対して認証を行います。サービスプロバイダは、Web ベースのサービスを提供する営利、または非営利の組織です。この広範な分類には、インターネットポータル、小売、運輸、金融、エンターテインメント、図書館、大学、政府などの機関が含まれます。

サービスプロバイダは、Cookie を使用してユーザーのセッション情報をクライアントブラウザに格納します。また、Cookie にはユーザーの ID プロバイダも含まれます。

ID プロバイダは、認証サービスの提供に特化したサービスプロバイダです。認証の管理サービスとして、識別情報を維持、管理します。ID プロバイダが行う認証は、そのプロバイダと関連するすべてのサービスプロバイダで尊重されます。

ユーザーが、ID プロバイダと関連しないサービスにアクセスしようとする、ID プロバイダはそのサービスプロバイダに Cookie を転送します。次に、このサービスプロバイダは、Cookie 内で呼び出される ID プロバイダにアクセスします。

ただし、異なる DNS ドメインの間で Cookie を読み取ることはできません。このため、サービスプロバイダを適切な ID プロバイダにリダイレクトし、そのユーザーのシングルサインオンを実現するために、共通ドメイン Cookie サービスが使用されません。

連携管理リソースの設定

連携リソース (サービスプロバイダ、ID プロバイダ、共通ドメイン Cookie サービス (Common Domain Cookie Service、CDCS)) は、それぞれが常駐するゲートウェイプロファイルベースで設定されます。ここでは、次の 3 つの例の設定方法について説明します。

1. すべてのリソースが企業イントラネット上に存在する場合
2. すべてのリソースが企業イントラネット上に存在しない場合、または ID プロバイダがインターネット上に存在する場合
3. すべてのリソースが企業イントラネット上に存在しない場合、または、サービスプロバイダがインターネット上のサードパーティーで、ID プロバイダがゲートウェイによって保護されている場合

設定 1

この設定では、サービスプロバイダ、ID プロバイダ、共通ドメイン Cookie サービス (CDCS) が同一の企業イントラネットに配備され、ID プロバイダはインターネット DNS (Domain Name Server) に公開されていません。CDCS の使用はオプションです。

この設定では、ゲートウェイは Portal Server であるサービスプロバイダをポイントします。この設定は、Portal Server の複数のインスタンスで有効です。

1. Access Manager 管理コンソールに管理者としてログインします。
2. 管理コンソールの「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「コア」タブをクリックします。

6. 「Cookie 管理を有効」 チェックボックスにチェックマークを付けて、Cookie 管理を有効化します。
7. 「Portal Server」 フィールドまでスクロールし、「非認証 URL」 リストに含まれる /amserver や /portal/dt などの相対 URL を使用できるように Portal Server 名を入力します。

例

```
http://idp-host:port/amserver/js
http://idp-host:port/amserver/UI/Login
http://idp-host:port/amserver/css
http://idp-host:port/amserver/SingleSignOnService
http://idp-host:port/amserver/UI/blank
http://idp-host:port/amserver/postLogin
http://idp-host:port/amserver/login_images
```

8. 「Portal Server」 フィールドまでスクロールし、Portal Server 名を入力します。たとえば、/amserver と入力します。
9. 「保存」 をクリックします。
10. 「セキュリティ」 タブをクリックします。
11. 「非認証 URL」 リストまでスクロールし、連携リソースを追加します。

例

```
/amserver/config/federation
/amserver/IntersiteTransferService
/amserver/AssertionConsumerservice
/amserver/fed_images
/amserver/preLogin
/portal/dt
```

12. 「追加」 をクリックします。
13. 「保存」 をクリックします。
14. 「非認証 URL」 リストに含まれる URL への到達にプロキシが必要な場合は、「プロキシ」 タブをクリックします。
15. 「ドメインとサブドメインのプロキシ」 フィールドまでスクロールし、適切な Web プロキシを入力します。
16. 「追加」 をクリックします。

17. 「保存」をクリックします。
18. 端末ウィンドウから、次のコマンドを指定してゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

設定 2

この設定では、ID プロバイダと共通ドメイン Cookie プロバイダ (CDCP) は企業イントラネットに配備されていません。または、ID プロバイダがインターネット上のサードパーティープロバイダとして存在します。

この設定では、ゲートウェイは Portal Server であるサービスプロバイダをポイントします。この設定は、Portal Server の複数のインスタンスで有効です。

1. Access Manager 管理コンソールに管理者としてログインします。
2. 管理コンソールの「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「コア」タブをクリックします。
6. 「Cookie 管理を有効」チェックボックスにチェックマークを付けて、Cookie 管理を有効化します。
7. 「Portal Server」フィールドをスクロールし、「非認証 URL」リストに含まれる /amserver や /portal/dt などの相対 URL を使用できるようにサービスプロバイダの Portal Server 名を入力します。

```
http://idp-host:port/amserver/js
```

```
http://idp-host:port/amserver/UI/Login
```

```
http://idp-host:port/amserver/css
```

```
http://idp-host:port/amserver/SingleSignOnService
```

```
http://idp-host:port/amserver/UI/blank
```

```
http://idp-host:port/amserver/postLogin
```

```
http://idp-host:port/amserver/login_images
```

8. 「保存」をクリックします。
9. 「セキュリティ」タブをクリックします。
10. 「非認証 URL」リストまでスクロールし、連携リソースを追加します。

例

```
/amserver/config/federation
/amserver/IntersiteTransferService
/amserver/AssertionConsumerservice
/amserver/fed_images
/amserver/preLogin
/portal/dt
```

11. 「追加」 をクリックします。
12. 「保存」 をクリックします。
13. 「非認証 URL」 リストに含まれる URL への到達にプロキシが必要な場合は、「プロキシ」 タブをクリックします。
14. 「ドメインとサブドメインのプロキシ」 フィールドまでスクロールし、適切な Web プロキシを入力します。
15. 「追加」 をクリックします。
16. 「保存」 をクリックします。
17. 端末ウィンドウから、次のコマンドを指定してゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

設定 3

この設定では、ID プロバイダと共通ドメイン Cookie プロバイダ (CDCP) は企業イントラネットに配備されていません。または、サービスプロバイダがインターネット上のサードパーティープロバイダとして存在し、ID プロバイダはゲートウェイによって保護されています。

この設定では、ゲートウェイは Portal Server である ID プロバイダをポイントします。

この設定は、Portal Server の複数のインスタンスで有効です。インターネット上でこのような設定が行われることはほとんどありませんが、一部の企業ネットワークではイントラネット内でこのような設定を行なっています。この設定では、ID プロバイダはファイアウォールによって保護されたサブネットに常駐し、サービスプロバイダには企業ネットワーク内から直接アクセスできます。

1. Access Manager 管理コンソールに管理者としてログインします。
2. 管理コンソールの「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。

「ゲートウェイプロファイルを編集」ページが表示されます。

5. 「コア」タブをクリックします。
6. 「Cookie 管理を有効」チェックボックスにチェックマークを付けて、Cookie 管理を有効化します。
7. 「Portal Server」フィールドをスクロールし、「非認証 URL」リストに含まれる /amserver や /portal/dt などの相対 URL を使用できるように ID プロバイダの Portal Server 名を入力します。

```
http://idp-host:port/amserver/js  
http://idp-host:port/amserver/UI/Login  
http://idp-host:port/amserver/css  
http://idp-host:port/amserver/SingleSignOnService  
http://idp-host:port/amserver/UI/blank  
http://idp-host:port/amserver/postLogin  
http://idp-host:port/amserver/login_images
```
8. 「保存」をクリックします。
9. 「セキュリティ」タブをクリックします。
10. 「非認証 URL」リストまでスクロールし、連携リソースを追加します。

例

```
/amserver/config/federation  
/amserver/IntersiteTransferService  
/amserver/AssertionConsumerservice  
/amserver/fed_images  
/amserver/preLogin  
/portal/dt
```

11. 「追加」をクリックします。
12. 「保存」をクリックします。
13. 「非認証 URL」リストに含まれる URL への到達にプロキシが必要な場合は、「プロキシ」タブをクリックします。
14. 「ドメインとサブドメインのプロキシ」フィールドまでスクロールし、適切な Web プロキシを入力します。
15. 「追加」をクリックします。
16. 「保存」をクリックします。

17. 端末ウィンドウから、次のコマンドを指定してゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

プロキシレットとリライタ

この章では、プロキシレットとリライタについて説明します。これらのコンポーネントを使用して、イントラネットの Web ページにゲートウェイ経由でアクセスできます。この機能は、さまざまな方法で実現します。たとえば、Web ページは、プロキシレットではなくリライタがパースします。

プロキシレットについては、以下のトピックで説明します。

- [プロキシレットの概要](#)

リライタについては、以下のトピックで説明します。

- [リライタの概要](#)
- [文字セットのエンコーディング](#)
- [リライタの使用例](#)
- [ルールセットの記述](#)
- [パブリックインタフェース \(ルールセット DTD\)](#)
- [ゲートウェイサービスのリライタの設定](#)
- [デバッグログを使用した障害追跡](#)
- [パブリックインタフェース \(ルールセット DTD\)](#)
- [サンプルの操作](#)
- [ケーススタディ](#)
- [6.x と 3.0 のルールセットのマッピング](#)

プロキシレットの概要

プロキシレットとは、それ自体でクライアントマシンのプロキシサーバーを設定する Java アプレットです。プロキシレットは、プロキシ設定がローカルのプロキシサーバーまたはプロキシレットをポイントするように、クライアントマシンのプロキシ自動設定 (PAC) ファイルを読み取り、変更します。

プロキシレットはゲートウェイからのトランスポートモードを継承します。ゲートウェイが SSL に基づいて動作するように設定されている場合には、クライアントマシンとゲートウェイ間のチャンネルのセキュリティが確保されます。暗号化する場合、プロキシレットは、クライアントの JVM が 1.4 以降の場合または必要な jar ファイルがクライアントマシン上にある場合に JSSE API を使用します。それ以外の場合には、KSSL API が使用されます。

ゲートウェイにリダイレクトされる URL のドメインとサブドメインは、ゲートウェイプロファイルに指定されています。ゲートウェイプロファイルに指定されていないドメインが URL に含まれる場合、その要求はインターネットにリダイレクトされます。特定の URL ドメインがゲートウェイプロファイルに指定されている場合には、クライアントのプロキシ設定はゲートウェイをポイントするようにリセットされます。

ゲートウェイで PDC (Personal Digital Certificate) が有効の場合、プロキシレットはクライアント側の認証をサポートします。PDC が有効かどうかを確認する方法については、[79 ページの「クライアント情報の取得」](#)を参照してください。

プロキシレットは、クライアントの IP アドレスとポートが指定されている Access Manager 管理コンソールから有効にします。プロキシレットが有効になると、クライアントマシンが次の点を満たしているかどうかを確認されます。

- ブラウザのアクセス権が正しいかどうか
- JVM バージョン 2 であるかどうか (すべてのブラウザに対して)
- ブラウザが Netscape 7.0/7.1、Mozilla 1.4/1.5、Internet Explorer 5.5 以降かどうか
- マシンまたはデバイスがサーバーアプリケーションを実行できるかどうか

これらの要件を満たしている場合には、小さなアプレットがダウンロードされ、クライアントマシン上で起動されます。クライアントの JRE が 1.3.1 以降でない場合には、プロキシレットと一緒に JRE が自動的にダウンロードされます。

プロキシレットが使用される場合、プロキシレットはプロキシ設定を PAC (Proxy Auto Configuration) ファイルまたはプロキシ設定リストから取得します。

プロキシレットを使用する利点

リライタと異なり、プロキシレットはインストール後の変更をほとんど、またはまったく必要としません。Microsoft Exchange Server などのサードパーティソフトウェアとの統合も簡単に行うことができます。プロキシレットは Web コンテンツを扱わないので、ゲートウェイのパフォーマンスも向上します。プロキシレットはコンテンツまたはデータを変更しないので、ユーザーは tar および gzip ファイルなど任意のコンテンツをダウンロードできます。

プロキシレットの設定

プロキシレットを有効にして設定する方法については、[319 ページの第 12 章「プロキシレットの設定」](#)を参照してください。

注 プロキシレットを実行する適切な Java 仮想マシン (JVM) がない場合、ブラウザは sun.com サイトに接続して Java Runtime Environment (JRE) をダウンロードします。ユーザーのブラウザ設定に正しい値が設定されていない場合、またはユーザーがインターネットにアクセスしないで直接プロキシ設定を使用している場合、プロキシレットはダウンロードできません。

リライタの概要

SRA のリライタコンポーネントを使用すると、エンドユーザーは Web ページの URI (Uniform Resource Identifier) リファレンスをゲートウェイをポイントするように変更することによって、イントラネットをブラウズすることができます。URI は、登録されているネームスペースにネームをカプセル化し、それにネームスペースのラベルを付ける方法を定義します。最も一般的な URI は URL (Uniform Resource Locator) です。リライタは HTTP または HTTPS だけをサポートし、このサポートは、プロトコルでの大文字の使用に影響されません。リライタは、相対 URL の一部として使用される場合にだけバックスラッシュをサポートします。

次に例を示します。

`http://abc.sesta.com¥index.html` はリライトされます。

次の URL はリライトされません。

`http:¥¥abc.sesta.com`

`http:./abc.com`

文字セットのエンコーディング

HTTP の規格では、HTTP ヘッダーまたは HTML メタタグに Web ページの文字セットを指定する必要があります。ただし、この情報が指定されていないこともあります。文字セットがわからない場合には、データのエンコーディングが設定されず、作成者が意図したようにデータが表示されません。

文字セットを検出するために、サードパーティ製品を使用することもできます。この製品を有効にするには、Java™ Enterprise System アクセサリ CD から SUNWjchdt パッケージをインストールします。この製品は、インストールするとリライタによって検出され、必要に応じて使用されます。

注 この製品を使用すると、パフォーマンスが低下することがあるため、必要な場合にだけインストールしてください。インストール、設定、および使用方法については、jcharset_readme.txt を参照してください。

リライタの使用例

ユーザーがゲートウェイを通じてイントラネット Web ページにアクセスしようとするときに、Web ページはリライタによって使用可能となります。リライタは、次のコンポーネントによって使用されます。

- URL スクレイパー
- ゲートウェイ

URL スクレイパー

URL スクレイパープロバイダは、設定されている URI からコンテンツを取得し、それをブラウザに送信する前にすべての相対 URI を絶対 URI に展開します。

たとえば、コンテンツを持つ次のサイトにユーザーがアクセスしようとするとして、

```
<a href=" ../mypage.html ">
```

リライタはこれを次のように変換します。

```
<a href="http://yahoo.com/mypage.html">
```

ここで、http://yahoo.com/test/ はページのベース URL です。

URL スクレイパープロバイダの詳細については、『Portal Server 管理ガイド』を参照してください。

ゲートウェイ

ゲートウェイは、インターネットポータルからコンテンツを取得し、そのコンテンツをブラウザに送信する前に、既存の URI の前にゲートウェイ URI プレフィックスを追加します。これにより、そのブラウザからの以後の URI 要求はゲートウェイに向けられます。

たとえば、インターネット上のマシンにある HTML ページの次のコンテンツにユーザーがアクセスするとします。

```
<a href="http://mymachine.intranet.com/mypage.html">
```

リライタは、次のようにゲートウェイを参照するプレフィックスを URL に追加します。

```
<a href="https://gateway.company.com/http://mymachine.intranet.com/mypage.html">
```

ユーザーがこのアンカーに関連するリンクをクリックすると、ブラウザはゲートウェイにアクセスします。ゲートウェイは、mymachine.intranet.com から mypage.html のコンテンツをフェッチします。

ゲートウェイはいくつかのルールを使用して、フェッチされた Web ページのリライトする要素を判断します。

ルールセットの記述

ルールの定義は、「サービス設定」タブの「Portal Server 設定」セクションで行います。

ルールセットの定義については、『Portal Server 管理ガイド』を参照してください。新しいルールセットを作成したら、必要なルールを定義する必要があります。

ここでは、次の項目について説明します。

- [パブリックインタフェース \(ルールセット DTD\)](#)
- [XML DTD の例](#)
- [ルールの記述手順](#)
- [ルールセットのガイドライン](#)
- [ルールセットのルート要素の定義](#)
- [再帰機能の使用](#)
- [HTML コンテンツのルール](#)
- [JavaScript コンテンツのルール](#)

- [XML コンテンツのルール](#)
- [カスケードスタイルシートのルール](#)
- [WML のルール](#)

パブリックインタフェース (ルールセット DTD)

ルールセット DTD の例を示します。

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!--
```

The following constraints are not represented in DTD, but taken care programatically

1. In a Rule, All Mandatory attributes cannot be "*".
2. Only one instance of the below elements is allowed, but in any order.

1)HTMLRules

2)JSRules

3)XMLRules

3. ID should always be in lower case.

```
-->
```

```
<!ENTITY % eURL &#180;URL&#180;>
```

```
<!ENTITY % eEXPRESSION &#180;EXPRESSION&#180;>
```

```
<!ENTITY % eDHTML &#180;DHTML&#180;>
```

```
<!ENTITY % eDJS &#180;DJS&#180;>
```

```
<!ENTITY % eSYSTEM &#180;SYSTEM&#180;>
```

```
<!ENTITY % ruleSetElements &#180;(HTMLRules | JSRules | XMLRules)?&#180;>
```

```
<!ENTITY % htmlElements &#180;(Form | Applet | Attribute)*&#180;>
```

```
<!ENTITY % jsElements &#180;(Variable | Function)*&#180;>
```

```
<!ENTITY % xmlElements &#180;(Attribute | TagText)*&#180;>
```

```
<!ELEMENT RuleSet
(%ruleSetElements;,%ruleSetElements;,%ruleSetElements;)>
<!ATTLIST RuleSet
  id ID #REQUIRED
  extends CDATA "none"
>

<!-- Rules for identifying rules in HTML content -->
<!ELEMENT HTMLRules (%htmlElements;)>
<!ELEMENT Form EMPTY>
<!ATTLIST Form
  name CDATA #REQUIRED
  field CDATA #REQUIRED
  valuePatterns CDATA ""
  source CDATA "*"
>

<!ELEMENT Applet EMPTY>
<!ATTLIST Applet
  code CDATA #REQUIRED
  param CDATA "*"
  valuePatterns CDATA ""
  source CDATA "*"
>

<!-- Rules for identifying rules in JS content -->
<!ELEMENT JSRules (%jsElements;)>
<!ELEMENT Variable EMPTY>
<!ATTLIST Variable
  name CDATA #REQUIRED
```

```

    type (%eURL; | %eEXPRESSION; | %eDHTML; | %eDJS; | %eSYSTEM;)
"EXPRESSION"

    source CDATA "*"
>

<!ELEMENT Function EMPTY>
<!ATTLIST Function
    name CDATA #REQUIRED
    paramPatterns CDATA #REQUIRED
    type (%eURL; | %eEXPRESSION; | %eDHTML; | %eDJS;) "EXPRESSION"
    source CDATA "*"
>

<!-- Rules for identifying rules in XML content -->
<!ELEMENT XMLRules (%xmlElements;)>
<!ELEMENT TagText EMPTY>
<!ATTLIST TagText
    tag CDATA #REQUIRED
    attributePatterns CDATA ""
    source CDATA "*"
>

<!ELEMENT Attribute EMPTY>
<!ATTLIST Attribute
    name CDATA #REQUIRED
    tag CDATA "*"
    valuePatterns CDATA ""
    type (%eURL; | %eDHTML; | %eDJS; ) "URL"
    source CDATA "*"
>

```

注 ルールの値の一部としてアスタリスク (*) を使用できます。ただし、すべての必須属性では、値に * だけを指定することはできません。このようなルールは無視されますが、メッセージは `RuleSetInfo` ログファイルに記録されます。このログファイルについては、[138 ページの「デバッグファイル名」](#) を参照してください。

XML DTD の例

ここでは、ルールセットの例を示します。リライタがこれらのルールをどのように解釈するかについては、[172 ページの「ケーススタディ」](#) を参照してください。

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
Rules for integrating a mail client with the gateway.
-->
<!DOCTYPE RuleSet SYSTEM "jar://rewriter.jar/resources/RuleSet.dtd">
<RuleSet type="GROUPED" id="owa">
<HTMLRules>
  <Attribute name="action" />
  <Attribute name="background" />
  <Attribute name="codebase" />
  <Attribute name="href" />
  <Attribute name="src" />
  <Attribute name="lowsrc" />
  <Attribute name="imagePath" />
  <Attribute name="viewClass" />
  <Attribute name="emptyURL" />
  <Attribute name="draftsURL" />
  <Attribute name="folderURL" />
  <Attribute name="prevMonthImage" />
  <Attribute name="nextMonthImage" />
  <Attribute name="style" />
  <Attribute name="content" tag="meta" />
```

```
</HTMLRules>
<JSRules>
<!-- Rules for Rewriting JavaScript variables in URLs -->
  <Variable name="URL"> _fr.location </Variable>
  <Variable name="URL"> g_szUserBase </Variable>
  <Variable name="URL"> g_szPublicFolderUrl </Variable>
  <Variable name="URL"> g_szExWebDir </Variable>
  <Variable name="URL"> g_szViewClassURL </Variable>
  <Variable name="URL"> g_szVirtualRoot </Variable>
  <Variable name="URL"> g_szBaseURL </Variable>
  <Variable name="URL"> g_szURL </Variable>
  <Function name="EXPRESSION" name="NavigateTo" paramPatterns="y"/>
</JSRules>
<XMLRules>
  <Attribute name="xmlns"/>
  <Attribute name="href" tag="a"/>
  <TagText tag="baseroot" />
  <TagText tag="prop2" />
  <TagText tag="prop1" />
  <TagText tag="img" />
  <TagText tag="xsl:attribute"
  attributePatterns="name=src" />
</XMLRules>
</RuleSet>
```

ルールの記述手順

次に、ルールを記述するための一般的な手順を示します。

- コンテンツのリライトが必要な HTML ページを含むディレクトリを特定します。
- これらのディレクトリで、リライトが必要なページを特定します。
- 各ページでリライトが必要な URL を特定します。「http」および「/」を検索すると、ほとんどの URL を簡単に見つけることができます。

- URL のコンテンツタイプ (HTML、JavaScript、または XML) を識別します。
- これらの各 URL のリライトに必要なルールを記述するには、Access Manager 管理コンソールの「Portal Server 設定」の「リライタ」で必要なルールセットを編集します。
- これらのルールを結合し、そのドメインのルールセットにまとめます。

ルールセットのガイドライン

次の点に注意してください。

- 特定のホストの優先順位は、URI の最長一致に基づいて決定されます。次のルールセットを例に示します。

```
mail1.central.abc.com|iplanet_mail_ruleset
*.sfbay.abc.com|sfbay_ruleset
*.abc.com|generic_ruleset
```

最長一致を含む `sfbay_ruleset` が使用されます。

- ルールセットのルールは、ルールが特定の文と一致するまでページの各文に順に適用されます。

ルールを記述する場合、ルールの順序に注意してください。ルールはルールセットに現れる順番で、ページ内の文に適用されます。特定のルール、および「*」を含む一般的なルールを適用する場合は、特定のルールを最初に定義し、次に一般的なルールを定義してください。この方法で定義しないと、特定のルールを適用する前に、一般的なルールがすべての文に適用されてしまいます。

- すべてのルールは `<RuleSet>`、`</RuleSet>` タグで囲む必要があります。
- ルールセットの `<HTMLRules>` `</HTMLRules>` セクションに、HTML コンテンツのリライトに必要なすべてのルールを指定します。
- ルールセットの `<JSRules>` `</JSRules>` セクションに、JavaScript コンテンツのリライトに必要なすべてのルールを指定します。
- ルールセットの `<XMLRules>` `</XMLRules>` セクションに、XML コンテンツのリライトに必要なすべてのルールを指定します。
- イン트라ネットページで、リライトの必要のある URL を特定し、ルールセットの適切なセクション (HTML、JSRules、または XMLRules) に必要なルールを指定します。
- 必要なドメインにルールセットを割り当てます。詳細については、[274 ページの「URI とルールセットのマッピングリストの作成」](#)を参照してください。
- ゲートウェイを再起動して変更を適用します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

ルールセットのルート要素の定義

ルールセットのルート要素には、次の2つの属性があります。

- **RuleSetName**:たとえば、`default_ruleset` があります。この名前は、URI マッピングのためにルールセットで参照されます。
- **Extends**:ルールセットの継承機能を参照する属性。この値は、ルールセットの取得元となるルールセットをポイントします。

新しい独立したルールセットがその他のルールセットに依存しないことを指定するには、`none` という値を指定します。ルールセットが別のルールセットに依存することを指定するには、`RuleSetName` を指定します。

再帰機能の使用

リライタは、再帰機能を使用して、一致する文字列パターンの最後まで同じパターンを検索します。

たとえば、リライタが次の文字列をパースする場合を考えます。

```
<a href="src=abc.jpg,src=bcd.jpg,src=xyz.jpg">
```

次のルールがあるとします。

```
<Attribute name="href" valuePatterns="*src=*" />
```

このルールは、最初に見つかったパターンだけを次のようにリライトします。

```
<a href="src=http://jane.sun.com/abc.jpg">
```

一方、次のように再帰オプションを使用した場合を考えます。

```
<Attribute name="href" valuePatterns="REC:*src=*" />;
```

リライタは再帰機能を使用して、一致する文字列パターンの最後まで同じパターンを検索します。この出力は次のようになります。

```
<a href="src=http://jane.sun.com/abc.jpg,src=http://jane.sun.com/bcd.jpg,src=http://jane.sun.com/xyz.jpg">
```

言語ベースのルールの定義 (ルールの定義)

ルールは、次の言語に基づきます。

- HTML
- JavaScript
- XML

HTML コンテンツのルール

Web ページの HTML コンテンツは、さらに属性、フォーム、アプレットに分類されます。これに従って、HTML コンテンツのルールは次のように分類されます。

- [HTML コンテンツの属性ルール](#)
- [HTML コンテンツのフォームルール](#)
- [HTML コンテンツのアプレットルール](#)

HTML コンテンツの属性ルール

このルールは値をリライトする必要のあるタグの属性を特定します。属性値には、簡易 URL、JavaScript、DHTML コンテンツがあります。

例

- 画像の場所を示す「img」タグの src 属性 (簡易 URL)
- リンクのクリックを処理する href 属性の onClick 属性 (DJS)

この節は、次の項目から構成されています。

- [属性ルールの構文](#)
- [属性ルールの例](#)
- [DJS 属性の例](#)

属性ルールの構文

```
<Attribute name="attributeName" [tag="*" valuePatterns="" source="*" type="URL|DHTML|DJS"] />
```

各表記の意味は次のとおりです。

attributeName は属性名です (必須)。

tag は、この属性が属するタグです (省略可能、デフォルトは任意のタグを意味する *)。

valuePatterns については、108 ページの「ルールでのパターンマッチングの使用」を参照してください。

source は、この属性が定義されているページの URI を指定します (省略可能、デフォルトは任意のページを意味する *)。

type は関数のタイプを指定します (省略可能)。これには次の値があります。

URL : 簡易 URL (デフォルト値)

DHTML : DHTML コンテンツ。この種類のコンテンツは、標準の HTML コンテンツに見られ、Microsoft の HTC 形式のファイルで使用されます。

DJS : JavaScript コンテンツ。onClick や onMouseover など、すべての HTML イベントハンドラには、HTML 属性に JavaScript が組み込まれています。

属性ルールの例

ページのベース URL が次の URL であると仮定します。

```
http://mymachine.intranet.com/mypage.html
```

ページコンテンツ

```
<a href="http://mymachine.intranet.com/mypage.html">
```

ルール

```
<Attribute name="href"/>
```

または

```
<Attribute name="href" tag="a"/>
```

出力

```
<a href=gateway-URL/http://mymachine.intranet.com/myhome.html>
```

説明

リライトされる URL はすでに絶対 URL であるため、ゲートウェイ URL だけがこの URL にプレフィックスとして追加されます。

DJS 属性の例

ページのベース URL が次の URL であると仮定します。

```
http://abc.sesta.com/focus.html
```

ページコンテンツ

```
<Form>
```

```
<input TYPE=TEXT SIZE=20 value=focus  
onClick="Check('/focus.html','focus');return;">
```

```
</Form>
```

ルール

```
<Attribute name="onClick" type="DJS"/>
```

```
<Function type="URL" name="Check" paramPatterns="y, "/>
```

出力

```
<Form>
```

```
<INPUT TYPE=TEXT SIZE=20 value=focus
onClick="Check('gateway-URL/http://abc.sesta.com/focus.html','focus')
;return; ">
```

```
</Form>
```

説明

指定されたページコンテンツをリライトするには、2つのルールが必要です。最初のルールは onClick JavaScript トークンを特定します。2番目のルールは、リライトが必要な check 関数のパラメータを特定します。この場合、paramPatterns に値 y が指定されているため、最初のパラメータだけがリライトされます。

ゲートウェイ URL と JavaScript トークンが表示されるベース URL が、必要なパラメータの前に指定されます。

HTML コンテンツのフォームルール

ユーザーが参照する HTML ページにはフォームが含まれていることがあります。一部のフォーム要素は、値として URL をとることがあります。

この節は、次の項目から構成されています。

- [フォームルールの構文](#)
- [フォームルールの例](#)

フォームルールの構文

```
<Form name="form1" field="visit" [valuePatterns="" source="*"]/>
```

各表記の意味は次のとおりです。

name はフォーム名です (必須)。

field は値をリライトする必要があるフォームのフィールドです (必須)。

valuePatterns については、[108 ページの「ルールでのパターンマッチングの使用」](#)を参照してください。

source は、このフォーム定義が存在するページの URL です (省略可能、デフォルトは任意のページを意味する *)。

フォームルールの例

ページのベース URL が次の URL であると仮定します。

```
http://test.siroe.com/testcases/html/form.html
```

ページコンテンツ

ページ URI が form.html で、サーバーの root ディレクトリに格納されていると仮定します。

```
<form name=form1 method=POST
action="http://test.siroe.com/testcases/html/form.html">
<input type=hidden name=abc1 value="0|1234|/test.html">
</form>
```

form1 の一部である abc1 という隠しフィールドの値に含まれる /test.html をリライトするとします。この場合、次のルールが必要です。

ルール

```
<Form source="*/form.html" name="form1" field="abc1"
valuePatterns="0|1234|"/>
<Attribute name="action"/>
```

出力

```
<FORM name="form1" method="POST"
action="gateway-URL/http://test.siroe.com/testcases/html/form.html">
<input type=hidden name=abc1
value="0|1234|gateway-URL/http://test.siroe.com/test.html">
</FORM>
```

説明

action タグは定義済みのいくつかの HTML 属性ルールを使用してリライトされます。

入力タグ属性値の value は、出力に示されるようにリライトされます。指定された valuePatterns が検索され、一致した valuePatterns に続くすべてのコンテンツは、先頭にゲートウェイ URL とページのベース URL を追加する方法でリライトされます。[108 ページの「ルールでのパターンマッチングの使用」](#)を参照してください。

HTML コンテンツのアプレットルール

単一の Web ページに複数のアプレットが含まれていたり、各アプレットに多くのパラメータが指定されていることがあります。リライトは、ルールに指定されている値とアプレットの HTML 定義を一致させ、アプレットのパラメータ定義の一部として含まれる URL の値を変更します。この置換はサーバーで実行され、ユーザーが特定の Web ページを参照しているときには行われません。このルールは、HTML コンテンツのアプレットタグとオブジェクトタグの両方のパラメータを識別し、それをリライトします。

この節は、次の項目から構成されています。

- [アプレットルールの構文](#)
- [アプレットルールの例](#)

アプレットルールの構文

```
<Applet code="ApplicationClassName/ObjectID" param="parametername"
[valuePatterns=" " source="*"] />
```

各表記の意味は次のとおりです。

code はアプレットクラスまたはオブジェクトクラスの名前です (必須)。

param は値をリライトする必要があるパラメータの名前です (必須)。

valuePatterns については、[108 ページの「ルールでのパターンマッチングの使用」](#)を参照してください。

source は、アプレット定義が存在するページの URL です (省略可能、デフォルトは任意のページを意味する *)。

アプレットルールの例

ページのベース URL が次の URL であると仮定します。

```
http://abc.siroe.com/casestudy/test/HTML/applet/rule1.html
```

ページコンテンツ

```
<applet codebase="appletcode" code="RewriteURLinApplet.class"
archive="/test.jar">
```

```
<param name=Test1 value="/index.html">
```

```
</applet>
```

ルール

```
<Applet source="*/rule1.html" code="RewriteURLin*.class"
param="Test*" />
```

出力

```
<APPLET
codebase="gateway-URL/http://abc.siroe.com/casestudy/test/HTML/applet
/appletcode" code="RewriteURLinApplet.class" archive="/test.jar">

<param name="Test1"
value="gateway-URL/http://abc.siroe.com/index.html">

</APPLET>
```

説明

default_gateway_ruleset に <Attribute name="codebase"/> が定義されているため、codebase attribute はリライトされます。

名前が Test で始まるすべてのパラメータがリライトされます。アプレットコードが表示されるページのベース URL、およびゲートウェイ URL が、param タグの value 属性の値の前に追加されます。

ルールでのパターンマッチングの使用

valuePatterns フィールドを使用してパターンマッチングを実行し、リライトが必要な文の特定部分を識別することができます。

ルールの一部として valuePatterns を指定すると、一致したパターンに続くすべてのコンテンツがリライトされます。

次のフォーム例のルールを考えます。

```
<Form source="*/source.html" name="form1" field="visit"
[valuePatterns="0|1234|"]/>
```

各表記の意味は次のとおりです。

source は、フォームが表示される HTML ページの URL です。

name はフォーム名です。

field は値をリライトする必要があるフォームのフィールドです。

valuePatterns はリライトが必要な部分文字列を示します。valuePatterns の後に表示されるすべてのコンテンツはリライトされます (省略可能、デフォルトは値全体のリライトが必要であることを示す")。)

valuePatterns への特殊文字の指定

¥ (円記号) でエスケープすることにより、特殊文字を指定できます。

例

```
<Form source="*/source.html" name="form1" field="visit"
[valuePatterns="0|1234|¥;original text|changed text"]/>
```

valuePatterns でのワイルドカードの使用

アスタリスク (*) を使用して、リライトのパターンマッチングを実行できます。

valuePatterns フィールドに * だけを指定することはできません。* はあらゆる項目との一致を示すため、valuePattern に続くコンテンツがなくなり、リライタがリライトするコンテンツもなくなります。* は *abc のように、ほかの文字列と組み合わせて使用できます。この場合、*abc に続くすべてのコンテンツがリライトされます。

注 アスタリスク (*) はルールの中のフィールドでも、ワイルドカードとして使用できます。ただし、ルールのすべてのフィールドに * を使用することはできません。すべてのフィールドに * が含まれている場合、ルールは無視されます。エラーメッセージは表示されません。

* や ** は、セミコロンやカンマなどの区切り文字と一緒に使用できます。区切り文字は、元の文に含まれる複数のフィールドを区切ります。1 文字のワイルドカード (*) はリライトされないフィールドと一致し、2 文字のワイルドカード (**) はリライトが必要なフィールドと一致します。

表 3-1 は、* ワイルドカードの使用例を示しています。

表 3-1 * ワイルドカードの使用例

URL	valuePatterns	説明
url1, url2, url3, url4	valuePatterns = "**, *, **, *"	この場合、** がリライトされる部分を表すため、url1 と url3 がリライトされます。
XYZABCh <code>http://host1.sesta.com/dir1.html</code>	valuePatterns = "*ABC"	この場合、 <code>http://host1.sesta.com/dir1.html</code> の部分だけがリライトされます。*ABC の後のすべてをリライトする必要があります。
"0 dir1 dir2 dir3 dir4 test url1	valuePatterns = "* * ** * ** * "	この場合、dir2、dir4、および url1 がリライトされます。リライトが必要な最後のフィールドは、** を使用して指定する必要はありません。

JavaScript コンテンツのルール

JavaScript はさまざまな場所に URL を含んでいます。リライタは JavaScript を直接パースできないため、URL 部分を特定できません。JavaScript プロセッサで URL を識別、解釈できるようにするために、特別なルールセットを記述する必要があります。

URL を含む JavaScript 要素は次のように分類されます。

- [変数](#)
- [関数の引数](#)

変数

汎用構文

```
<Variable name="variableName"  
[type="URL|EXPRESSION|DHTML|DJS|SYSTEM" source="*"]>
```

JavaScript の変数は、その値の種類に応じてさらに次の 5 つのカテゴリに分類されます。

- [URL 変数](#)
- [EXPRESSION 変数](#)
- [DHTML \(ダイナミック HTML\) 変数](#)
- [DJS \(ダイナミック JavaScript\) 変数](#)
- [SYSTEM 変数](#)

URL 変数

この変数の値は、URL として扱うことができる単純文字列です。

この節は、次の項目から構成されています。

- [URL 変数の構文](#)
- [URL 変数の例](#)

URL 変数の構文

```
<Variable name="variableName" type="URL" [source="*"]>
```

各表記の意味は次のとおりです。

`variableName` は変数名です。`variablename` の値がリライトされます (必須)。

`type` は URL 変数です (必須、値は URL でなければならない)。

`source` は、この JavaScript 変数が含まれるページの URI です (省略可能、デフォルトは任意のページを意味する *)。

URL 変数の例

ベース URL が次の URL であると仮定します。

```
http://abc.siroe.com/tmp/page.html
```

ページコンテンツ

```

<script LANGUAGE="Javascript">
<!--
//URL Variables
var imgsrc1="/tmp/tmp.jpg";
var imgsrc2="http://srap.sesta.com/tmp/tmp.jpg";
var imgsrc3=imgsrc2;
//-->
</SCRIPT>

```

ルール

```
<Variable name="imgsrc*" type="URL"/>
```

出力

```

<script LANGUAGE="Javascript">
<!--
//URL Variables
var imgsrc="gateway-URL/http://abc.siroe.com/tmp/tmp.jpg";
var imgsrc="gateway-URL/http://srap.sesta.com/tmp/tmp.jpg";
var imgsrc2=imgsrc1;
//-->
</SCRIPT>

```

説明

タイプが URL で、名前が `imgsrc` から始まるすべての変数がリライトされます。出力の最初の行では、ゲートウェイ URL と変数が表示されるページのベース URL が先頭に指定されます。2 行目にはすでに絶対パスが指定されているため、ゲートウェイ URL だけがプレフィックスとして追加されます。3 番目の変数 `imgsrc2` は、値が文字列ではなく別の JavaScript 値であるためリライトされません。

EXPRESSION 変数

EXPRESSION 変数の右側には式が指定されます。この式の結果は URL です。リライタは、このような式をサーバーで評価できないため、HTML ページに JavaScript 関数 (psSRAPRewriter_convert_expression) を追加します。この関数はパラメータとして式をとり、クライアントブラウザで要求される URL に対して式を評価します。

文に含まれる URL が単一の URL であるか EXPRESSION URL であるかが明らかでないときは、どちらの場合にも適用できる EXPRESSION ルールを使用してください。

この節は、次の項目から構成されています。

- [EXPRESSION 変数の構文](#)
- [EXPRESSION 変数の例](#)

EXPRESSION 変数の構文

```
<Variable name="variableName" [type="EXPRESSION" source="*"]/>
```

各表記の意味は次のとおりです。

variableName は、値として式を持つ JavaScript 変数の名前です (必須)。

type は JavaScript 変数のタイプです (省略可能、デフォルト値は EXPRESSION)。

source はページの URI です (省略可能、デフォルトは任意のソースを意味する *)。

EXPRESSION 変数の例

ページのベース URL が次の URL であると仮定します。

```
http://abc.siroe.com/dir1/dir2/page.html
```

ページコンテンツ

```
<script LANGUAGE="Javascript">
<!--
//EXPRESSION 変数
var expvar= getURIPreFix() + ".././images/graphics"+".gif";
document.write("<A HREF="+expvar+">Link to XYZ content</A><P>")
var expvar=".././images/graphics"+".gif";
//-->
</SCRIPT>

ルール

<Variable name="expvar" type="EXPRESSION"/>
```

または

```
<Variable name="expvar"/>
```

出力

```
var expvar=psSRAPRewriter_convert_expression(getURIPreFix() +
"../../images/graphics"+" .gif");
document.write("<a href="+expvar+">>Link to XYZ content</A><P>")
var expvar="gateway-URL/http://abc.siroe.com/images/graphics"+" .gif";
```

説明

関数 `psSRAPRewriter_convert_expression` が、式変数 `expvar` の最初の行の右側の部分に先行して指定されます。この関数は、実行時に式を処理し、コンテンツをリライトします。3 行目では、値が簡易 URL にリライトされます。

DHTML (ダイナミック HTML) 変数

これは HTML コンテンツを含む JavaScript 変数です。

この節は、次の項目から構成されています。

- [DHTML 変数の構文](#)
- [DHTML 変数の例](#)

DHTML 変数の構文

```
<Variable name="variableName" type="DHTML" [source="*"]/>
```

各表記の意味は次のとおりです。

`variableName` は DHTML コンテンツを持つ JavaScript 変数の名前です (必須)。

`type` は変数のタイプです (必須、値は DHTML である必要がある)。

`source` はページの URL です (省略可能、デフォルトは任意のページを意味する *)。

DHTML 変数の例

ページのベース URL が次の URL であると仮定します。

```
http://abc.sesta.com/graphics/set1/graphics/jsscript/JSVAR/page.html
```

ページコンテンツ

```
<script LANGUAGE="Javascript">
```

```
<!--
```

```
//DHTML Var
```

```
var dhtmlVar="<a href=../../images/test.html>"
```

```
var dhtmlVar="<a href=/images/test.html>"
```

```
var dhtmlVar="<a href=images/test.html>"
//-->
</SCRIPT>
ルール
<Variable name="dhtmlVar" type="DHTML"/>
<Attribute name="href"/>
または
<Attribute name="href" tag="a"/>
出力
<script LANGUAGE="Javascript">
<!--
//DHTML Var
var dhtmlVar="<a
href=gateway-URL/http://abc.sesta.com/graphics/set1/graphics/images/t
est.html>"
var dhtmlVar="<a
href=gateway-URL/http://abc.sesta.com/images/test.html>"
var dhtmlVar="<a
href=gateway-URL/http://abc.sesta.com/graphics/set1/graphics/jscript/
JSVAR/images/test.html>"
//-->
</SCRIPT>
```

説明

JavaScript パーサーは dhtmlVar の値を HTML コンテンツとして読み取り、HTML パーサー経由でそのコンテンツを送信します。HTML パーサーは HTML ルールを適用するため、href 属性ルールとの一致によって URL がリライトされます。

DJS (ダイナミック JavaScript) 変数

これは JavaScript コンテンツを含む JavaScript 変数です。

この節は、次の項目から構成されています。

- [DJS 変数の構文](#)
- [DJS 変数の例](#)

DJS 変数の構文

```
<Variable name="variableName" type="DJS" [source="*"] />
```

各表記の意味は次のとおりです。

variable は JavaScript を値として持つ JavaScript 変数の名前です。

DJS 変数の例

ページのベース URL が次の URL であると仮定します。

```
http://abc.sesta.com/dir1/dir2/dir3/jscript/dir4/page.html
```

ページコンテンツ

```
//DJS Var
```

```
var dJSVar="var dJSimgsrc='/tmp/tmp.jpg';"
```

```
var dJSVar="var dJSimgsrc='../tmp/tmp.jpg';"
```

```
var dJSVar="var dJSimgsrc='http://abc.sesta.com/tmp/tmp.jpg';"
```

ルール

```
<Variable name="DJS">dJSVar/>
```

```
<Variable name="URL">dJSimgsrc/>
```

出力

```
//DJS Var - need 2 rules
```

```
var dJSVar="var
```

```
dJSimgsrc='gateway-URL/http://abc.sesta.com/tmp/tmp.jpg';"
```

```
var dJSVar="var
```

```
dJSimgsrc='gateway-URL/http://abc.sesta.com/dir1/dir2/dir3/jscript/tmp/
tmp.jpg';"
```

```
var dJSVar="var
```

```
dJSimgsrc='gateway-URL/http://abc.sesta.com/tmp/tmp.jpg';"
```

説明

ここでは、2つのルールが必要です。最初のルールは動的 JavaScript 変数 dJSVar を検索します。この変数の値は、同じくタイプが URL の JavaScript になります。次に 2 番目のルールが適用され、この JavaScript 変数の値がリライトされます。

SYSTEM 変数

これらは、使用によって宣言されない変数であり、サポートは限定されます。これらの変数は JavaScript 標準の一部として利用可能です。たとえば、

window.location.pathname などがあります。

この節は、次の項目から構成されています。

- [SYSTEM 変数の構文](#)
- [SYSTEM 変数の例](#)

SYSTEM 変数の構文

```
<Variable name="variableName" type="SYSTEM" [source="*"]/>
```

各表記の意味は次のとおりです。

variableName は JavaScript のシステム変数です (必須)。値は document.URL、document.domain、location、document.location、location.pathname、location.href、location.protocol、location.hostname、location.host、location.port のいずれかのパターンと一致する必要があります。これは、すべて generic_ruleset に含まれます。これらのシステム var ルールを変更しないでください。

type には、システムタイプの値を指定します (必須、値は DJS)。

source はこのページの URI です (省略可能、デフォルトは任意のページを意味する *)。

SYSTEM 変数の例

ページのベース URL が次の URL であると仮定します。

```
http://abc.siroe.com/dir1/page.html
```

ページコンテンツ

```
<script LANGUAGE="Javascript">
```

```
<!--
```

```
//SYSTEM 変数
```

```
alert (window.location.pathname);
```

```
//-->
```

```
</SCRIPT>
```

ルール

```
<Variable name="window.location.pathname" type="SYSTEM"/>
```

出力

```
</SCRIPT>
```

```
<SCRIPT LANGUAGE="Javascript">
```

```
<!--
```

```
//SYSTEM 変数
```

```
alert (psSRAPRewriter_convert_pathname(window.location.pathname));
```

```
//-->
</SCRIPT>
```

説明

リライトは、ルールと一致するシステム変数を検索し、プレフィックスとして `psSRAPRewriter_convert_system` 関数を追加します。この関数は、実行時にシステム変数を処理し、処理後の URL をリライトします。

関数の引数

値のリライトが必要な関数パラメータは、次の 4 つのカテゴリに分類されます。

- URL パラメータ
- EXPRESSION パラメータ
- DHTML パラメータ
- DJS パラメータ

汎用構文

```
<Function name="functionName" paramPatterns="y,y,"
[type="URL|EXPRESSION|DHTML|DJS" source="*"] />
```

各表記の意味は次のとおりです。

`name` は JavaScript 関数の名前です (必須)。

`paramPatterns` は、リライトが必要なパラメータを指定します (必須)。

`y` によって指定される位置は、リライトが必要なパラメータを示します。たとえば、構文の最初のパラメータはリライトするが、2 番目のパラメータはリライトしない、という指定が可能です。

`type` はこのパラメータが必要とする値の種類を指定します (省略可能、デフォルトは EXPRESSION タイプ)。

`source` はページのソース URI です (省略可能、デフォルトは任意のページを意味する *)。

URL パラメータ

関数は、このパラメータを文字列としてとり、この文字列は URL として扱うことができます。

この節は、次の項目から構成されています。

- URL パラメータの構文
- URL パラメータの例

URL パラメータの構文

```
<Function name="functionName" paramPatterns="y,," type="URL"  
[source="*"]/>
```

各表記の意味は次のとおりです。

name は、パラメータのタイプが URL である関数の名前です (必須)。

paramPatterns は、リライトが必要なパラメータを指定します (必須)。

y によって指定される位置は、リライトが必要なパラメータを示します。たとえば、構文の最初のパラメータはリライトするが、2 番目のパラメータはリライトしない、という指定が可能です。

type は関数のタイプです (必須、値は URL である必要がある)。

source は、この関数の呼び出しが含まれるページの URL です (省略可能、デフォルトは任意の URL を意味する *)。

URL パラメータの例

ページのベース URL が次の URL であると仮定します。

```
http://abc.sesta.com/test/rewriter/test1/jscript/test2/page.html
```

ページコンテンツ

```
<script language="JavaScript">
```

```
<!--
```

```
function test(one,two,three) {
```

```
alert(one + "##" + two + "##" +three);
```

```
}
```

```
test("/test.html","../test.html","123");
```

```
window.open("/index.html","gen",width=500,height=500);
```

```
//-->
```

```
</SCRIPT>
```

ルール

```
<Function name="URL" name="test" paramPatterns="y,y,"/>
```

```
<Function name="URL" name="window.open" paramPatterns="y,,,"/>
```

出力

```
<SCRIPT language="JavaScript">
```

```
<!--
```

```
function test(one,two,three) {
alert(one + "##" + two + "##" +three);
}
test ("gateway-URL/http://abc.sesta.com/test.html", "gateway-URL/http://abc.sesta.com/test/rewriter/test1/jscript/test.html", "123");
window.open ("gateway-URL/http://abc.sesta.com/index.html", "gen", width=500,height=500);
//-->
</SCRIPT>
```

説明

最初のルールは、関数 `test` の最初の 2 つのパラメータをリライトする必要があることを示します。したがって、`test` 関数の最初の 2 つのパラメータがリライトされます。2 番目のルールは、`window.open` 関数の最初のパラメータをリライトする必要があることを示します。`window.open` 関数内の URL の先頭に、ゲートウェイ URL と、関数パラメータが含まれるページのベース URL が追加されます。

EXPRESSION パラメータ

このパラメータは、値として式をとり、この式の評価結果が URL となります。

この節は、次の項目から構成されています。

- [EXPRESSION パラメータの構文](#)
- [EXPRESSION パラメータの例](#)

EXPRESSION パラメータの構文

```
<Function name="functionName" paramPatterns="y" [type="EXPRESSION" source="*"]/>
```

各表記の意味は次のとおりです。

`name` は関数名です (必須)。

`paramPatterns` は、リライトが必要なパラメータを指定します (必須)。

`y` によって指定される位置は、リライトが必要な関数パラメータを示します。上の構文では、最初のパラメータだけがリライトされます。

`type` は、式の値のタイプを指定します (省略可能)。

`source` は、この関数を呼び出すページの URI です。

EXPRESSION パラメータの例

ページのベース URL が次の URL であると仮定します。

```
http://abc.sesta.com/dir1/dir2/page.html
ページコンテンツ
<script language="JavaScript">
<!--
function jstest2() {
return ".html";
}
function jstest1(one) {
return one;
}
var dir="/images/test"
var test1=jstest1(dir+"/test"+jstest2());
document.write("<a HREF="+test1+">TEST</a>");
alert(test1);
//-->
</SCRIPT>
ルール
<Function type="EXPRESSION" name="jstest1" paramPatterns="y"/>
または
<Function name="jstest1" paramPatterns="y"/>
出力
<script language="JavaScript">
<!--
function jstest2() {
return ".html";
}
function jstest1(one) {
return one;
}
var dir="/images/test"
```

```

var
test1=jstest1(psSRAPRewriter_convert_expression(dir+"/test"+jstest2
()));
document.write("<a HREF="+test1+">TEST</a>");
alert(test1);
//-->
</SCRIPT>

```

説明

このルールは、これが **EXPRESSION** 関数のパラメータであると見なすことによって、`jstest1` 関数の最初のパラメータをリライトする必要があることを示します。ページコンテンツの例では、最初のパラメータは実行時にだけ評価される式です。リライタはこの式の先頭に `psSRAPRewriter_convert_expression` 関数を追加します。式が評価され、`psSRAPRewriter_convert_expression` 関数は実行時に出力をリライトします。

注 上の例では、JavaScript 変数ルールの一部として、変数 `test1` は必要ありません。リライトは、`jstest1` の関数ルールによって行われます。

DHTML パラメータ

これは、値が HTML の関数パラメータです。

HTML ページを動的に生成する `document.write()` などのネイティブ JavaScript メソッドは、このカテゴリに分類されます。

この節は、次の項目から構成されています。

- [DHTML パラメータの構文](#)
- [DHTML パラメータの例](#)

DHTML パラメータの構文

```
<Function name="functionName" paramPatterns="y" type="DHTML"
[source="*"] />
```

各表記の意味は次のとおりです。

`name` は関数名です。

`paramPatterns` は、リライトが必要なパラメータを指定します (必須)。

`y` によって指定される位置は、リライトが必要な関数パラメータを示します。上の構文では、最初のパラメータだけがリライトされます。

DHTML パラメータの例

ページのベース URL が次の URL であると仮定します。

```
http://xyz.siroe.com/test/rewriter/test1/jscript/JSFUNC/page.html
```

ページコンテンツ

```
<script>
<!--
document.write('<a href="/index.html">write</a><BR>')
document.writeln('<a href="index.html">writeln</a><BR>')
document.write("http://abc.sesta.com/index.html<BR>")
document.writeln("http://abc.sesta.com/index.html<BR>")
//-->
</SCRIPT>
```

ルール

```
<Function name="DHTML" name="document.write" paramPatterns="y"/>
<Function name="DHTML" name="document.writeln" paramPatterns="y"/>
<Attribute name="href"/>
```

出力

```
<SCRIPT>
<!--
document.write('<a
href="gateway-URL/http://xyz.siroe.com/index.html">write</a><BR>')
document.writeln('<a
href="gateway-URL/http://xyz.siroe.com/test/rewriter/test1/jscript/JSFU
NC/index.html">writeln</a><BR>')
document.write("http://abc.sesta.com/index.html<BR>")
document.writeln("http://abc.sesta.com/index.html<BR>")
//-->
</SCRIPT>
```

説明

最初のルールは、関数 `document.write` の最初のパラメータをリライトする必要があることを示します。2 番目のルールは、関数 `document.writeln` の最初のパラメータをリライトする必要があることを指定します。3 番目のルールは、名前に `href` を含むすべての属性をリライトする必要があることを指定する簡単な HTML ルールです。この例では、DHTML パラメータルールは関数内のリライトの必要があるパラメータを特定します。この場合、HTML 属性ルールが適用され、特定されたパラメータが実際にリライトされます。

DJS パラメータ

これは、値が JavaScript の関数パラメータです。

この節は、次の節から構成されています。

- [DJS パラメータの構文](#)
- [DJS パラメータの例](#)

DJS パラメータの構文

```
<Function name="functionName" paramPatterns="y" type="DJS"
[source="*"]/>
```

各表記の意味は次のとおりです。

`name` は、1 つのパラメータが DJS である関数の名前です (必須)。

`paramPatterns` は、上の関数のどのパラメータが DJS であるかを指定します (必須)。

`y` によって指定される位置は、リライトが必要な関数パラメータを示します。上の構文では、最初のパラメータだけがリライトされます。

`type` は DJS です (必須)。

`source` はページの URI です (省略可能、デフォルトは任意の URI を意味する *)。

DJS パラメータの例

ページのベース URL が次の URL であると仮定します。

```
http://abc.sesta.com/page.html
```

ページコンテンツ

```
<script>
```

```
menu.addItem(new NavBarMenuItem("All Available
Information","JavaScript:top.location='http://abc.sesta.com'"));
```

```
</script>
```

ルール

```
<Function name="DJS" name="NavBarMenuItem" paramPatterns="y"/>
```

```
<Variable name="URL">top.location</Variable>
```

出力

```
<script>
```

```
menu.addItem(new NavBarMenuItem("All Available  
Information", "JavaScript:top.location='gateway-URL/http://abc.sesta.com  
'"));
```

```
</script>
```

説明

最初のルールは、JavaScript を含む関数 `NavBarMenuItem` の 2 番目のパラメータをリライトする必要があることを指定します。JavaScript 内で、変数 `top.location` もリライトする必要があります。この変数は 2 番目のルールを使用してリライトされます。

XML コンテンツのルール

Web ページには、URL を含む XML コンテンツが含まれていることがあります。リライトが必要な XML コンテンツは、2 つのカテゴリに分類されます。

- [タグテキスト](#) (タグの PCDATA または CDATA と同様)
- [属性](#)

タグテキスト

このルールは、タグ要素の PCDATA または CDATA をリライトするためのものです。

この節は、次の項目から構成されています。

- [タグテキストの構文](#)
- [タグテキストの例](#)

タグテキストの構文

```
<TagText tag="tagName" [attributePatterns="attribute_patterns_for_  
this_tag" source="*"]/>
```

各表記の意味は次のとおりです。

`tag` はタグ名です。

`attributePatterns` はこのタグの属性と属性値パターンです (省略可能、省略した場合はこのタグは属性を一切持たない)。

`source` はこの XML ファイルの URI です (省略可能、デフォルトは任意の XML ページを意味する *)。

タグテキストの例

ページのベース URL が次の URL であると仮定します。

```
http://abc.sesta.com/test/rewriter/test1/xml/page.html
```

ページコンテンツ

```
<xml>
<Attribute name="src">test.html</attribute>
<attribute>abc.html</attribute>
</xml>
```

ルール

```
<TagText tag="attribute" attributePatterns="name=src"/>
```

出力

```
<xml>
<Attribute
name="src">gateway-URL/http://abc.sesta.com/test/rewriter/test1/xml/tes
t.html</attribute>
<attribute>abc.html</attribute>
</xml>
```

説明

ページコンテンツの最初の行には**属性の例**が含まれます。ページコンテンツの2行目には、名前が **name** で値が **src** の属性が含まれず、リライトは行われません。これをリライトするには、`<TagText tag="attribute"/>` も必要です。

属性

XML 属性のルールは、HTML の属性ルールに似ています。118 ページの「[HTML コンテンツの属性ルール](#)」を参照してください。違いは、XML の属性ルールでは大文字と小文字が区別され、HTML の属性ルールでは区別されないことです。これは、XML では大文字と小文字が区別され、HTML では区別されないためです。

リライタは、属性名に基づいて属性値を変換します。

この節は、次の項目から構成されています。

- [属性の構文](#)
- [属性の例](#)

属性の構文

```
<Attribute name="attributeName" [tag="*" type="URL"
valuePatterns="*" source="*"]/>
```

各表記の意味は次のとおりです。

attributeName は属性名です (必須)。

tag は、この属性が含まれるタグの名前です (省略可能、デフォルトは任意のタグを意味する *)。

valuePatterns については、[108 ページ](#)の「ルールでのパターンマッチングの使用」を参照してください。

source は XML ページの URI です (省略可能、デフォルトは任意の XML ページを意味する *)。

属性の例

ページのベース URL が次の URL であると仮定します。

```
http://abc.sesta.com/test/rewriter/test1/xml/page.html
```

ページコンテンツ

```
<xml>
<baseroot href="/root.html"/>
<img href="image.html"/>
<string href="1234|substring.html"/>
<check href="1234|string.html"/>
</xml>
```

ルール

```
<Attribute name="href"tag="check" valuePatterns="1234|"/>
```

出力

```
<xml>
<baseroot href="/root.html"/>
<img href="image.html"/>
<string href="1234|substring.html"/>
<check
href="1234|gateway-URL/http://abc.sesta.com/test/rewriter/test1/xml/s
tring.html"/>
</xml>
```

説明

上記の例では、4行目だけがルールに指定されたすべての条件と一致するため、リライトされます。108ページの「ルールでのパターンマッチングの使用」を参照してください。

カスケードスタイルシートのルール

HTML ページのカスケードスタイルシート (CSS2 も含まれる) も変換されます。この変換のために定義されるルールはありません。これは、URL が CSS の `url()` 関数とインポート構文にだけ表示されるためです。

WML のルール

WML は HTML に似ているため、WML コンテンツには HTML ルールが適用されます。WML コンテンツの汎用ルールセットを使用してください。103ページの「HTML コンテンツのルール」を参照してください。

再帰機能の使用

リライタは、再帰機能を使用して、一致する文字列パターンの最後まで同じパターンを検索します。

たとえば、リライタが次の文字列をパースする場合は考えます。

```
<a href="src=abc.jpg,src=bcd.jpg,src=xyz.jpg">
```

次のルールがあるとします。

```
<Attribute name="href" valuePatterns="*src="*/>
```

このルールは、最初に見つかったパターンだけを次のようにリライトします。

```
<a href="src=http://jane.sun.com/abc.jpg">
```

一方、次のように再帰オプションを使用した場合は考えます。

```
<Attribute name="href" valuePatterns="REC:*src="*/>;
```

リライタは再帰機能を使用して、一致する文字列パターンの最後まで同じパターンを検索します。この出力は次のようになります。

```
<a href="src=http://jane.sun.com/abc.jpg,src=http://jane.sun.com/bcd.jpg,src=http://jane.sun.com/xyz.jpg">
```

ゲートウェイサービスのリライトの設定

「リライト」タブでゲートウェイサービスを使用することで、次の基本タスクと高度なタスクを実行できます。

- 基本タスク
 - すべての URI のリライトの有効化
 - URI とルールセットのマッピングリストの作成
 - パースする MIME タイプリストの作成
 - リライトしない URI のリストの作成
 - デフォルトドメインの指定
- 高度なタスク
 - MIME 推測の有効化
 - パースする URI マッピングリストの作成
 - マスキングの有効化
 - マスキングのためのシード文字列の指定
 - マスクしない URI のリストの作成
 - ゲートウェイプロトコルと元の URI プロトコルの同一化

基本タスク

すべての URI のリライトの有効化

ゲートウェイサービスで「すべての URL のリライトを有効」オプションを有効にすると、「ドメインとサブドメインのプロキシ」リストのエントリをチェックせずに、リライトはすべての URL をリライトします。「ドメインとサブドメインのプロキシ」リストのエントリは無視されます。

▶ ゲートウェイによるすべての URL のリライトを有効にするには

1. Sun Java System Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。

5. 「リライト」タブをクリックします。
6. 「すべての URL のリライトを有効」チェックボックスにチェックマークを付け、ゲートウェイによるすべての URL のリライトを有効にします。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

URI とルールセットのマッピングリストの作成

ルールセットは、Access Manager 管理コンソールの「Portal Server 設定」の下のリライトサービスに作成されます。詳細については、『Portal Server 管理ガイド』を参照してください。

ルールセットを作成したら、「URI をルールセットにマップ」フィールドを使用してドメインとルールセットを関連付けます。デフォルトでは、「URI をルールセットにマップ」リストに次の 2 つのエントリが追加されます。

- `*://*.Sun.COM/portal/*|default_gateway_ruleset`

この `sun.com` はポータルインストールドメインで、`/portal` はポータルのインストールコンテキストです。

- `*|generic_ruleset`

これは、ドメインが `sun.com` のポータルディレクトリのすべてのページに `default_gateway_ruleset` を適用することを指定しています。他のすべてのページには、汎用ルールセットが適用されます。`default_gateway_ruleset` と `generic_ruleset` は、すでにパッケージ化されているルールセットです。

注 標準のポータルデスクトップに表示されるすべてのコンテンツには、それがどこからフェッチされたかに関係なく `default_gateway_ruleset` のルールセットが適用されます。

たとえば、URL `yahoo.com` のコンテンツを集めるように標準のポータルデスクトップを設定すると仮定します。Portal Server は `sesta.com` 内にあります。フェッチされたコンテンツに `sesta.com` のルールセットが適用されます。

注 ルールセットを指定するドメインは、「ドメインとサブドメインのプロキシ」リストに含まれている必要があります。

構文内でのワイルドカードの使用

ルールセット内でアスタリスクを使用して、完全修飾 URI または部分 URI をマッピングできます。

たとえば、次のように指定することで、index.html ページに java_index_page_ruleset を適用できます。

```
www.sun.com/java/index.html/java_index_page_ruleset
```

または、次のように指定することで、java ディレクトリのすべてのページを java_directory_ruleset に適用できます。

```
www.sun.com/java/* /java_directory_ruleset
```

▶ URI をルールセットにマッピングするには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「リライト」タブをクリックします。
6. 「URI をルールセットにマップ」フィールドまでスクロールします。
7. 「URI をルールセットにマップ」フィールドに適切なドメイン名またはホスト名とルールセットを入力し、「追加」をクリックします。

「URI をルールセットにマップ」リストにエントリが追加されます。

ドメインまたはホスト名とルールセットは次の形式で指定します。

ドメイン名 | ルールセット名

例

```
eng.sesta.com|default
```

パースする MIME タイプリストの作成

リライトには、コンテンツタイプ (HTML、JAVASCRIPT、CSS、および XML) に基づいて Web ページをパースするための 4 種類のパーサーがあります。デフォルトでは、これらのパーサーには一般的な MIME タイプが関連付けられています。新しい MIME タイプとこれらのパーサーの関連付けは、ゲートウェイサービスの「パーサーを MIME タイプにマップ」フィールドで行います。これにより、リライト機能を他の MIME タイプに拡張できます。

複数のエントリは、セミコロン (;) またはカンマ (,) で区切ります。

例

```
HTML=text/html;text/htm;text/x-component;text/wml;text/vnl/wap.wml
```

これは、これらの MIME が HTML リライターに送られ、URL のリライトに HTML ルールを適用することを指定しています。

ヒント MIME マッピングリストから不要なパーサーを削除すると、処理速度が向上します。たとえば、特定のイントラネットのコンテンツに JavaScript が含まれないことが確実な場合は、MIME マッピングリストから JavaScript エントリを削除できます。

▶ MIME のマッピングを指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 編集する属性が含まれるゲートウェイプロファイルをクリックします。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「リライト」タブをクリックします。
6. 「パーサーを MIME タイプにマップ」フィールドまでスクロールし、編集ボックスに必要な MIME タイプを追加します。複数のエントリを区切るときは、セミコロンまたはカンマを使用します。

エントリは `HTML=text/html;text/htm` の形式で指定します。

7. 「追加」をクリックし、必要なエントリをリストに追加します。
8. 「保存」をクリックし、変更内容を記録します。
9. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

リライトしない URI のリストの作成

▶ リライトしない URI を指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。

「ゲートウェイ」ページが表示されます。

4. 編集する属性が含まれるゲートウェイプロファイルをクリックします。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「リライト」タブをクリックし、「基本」サブセクションを表示します。
6. 「リライトしない URI」フィールドまでスクロールし、編集ボックスに URI を追加します。

注：このリストに #* を追加することで、href ルールがルールセットの一部である場合でも URI をリライトできます。

7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

デフォルトドメインの指定

デフォルトのドメインとサブドメインは、URL にホスト名だけが含まれ、ドメインとサブドメインが指定されていない場合に便利です。この場合、ゲートウェイはホスト名がデフォルトのドメインとサブドメイン内にあると仮定し、そのように処理を進めます。

たとえば、URL のホスト名が host1、デフォルトのドメインとサブドメインが red.sesta.com のように指定されている場合、ホスト名は host1.red.sesta.com として解決されます。

▶ デフォルトドメインを指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブをクリックします。
3. 「SRA 設定」の「ゲートウェイ」の隣の右矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「デフォルトのドメイン」フィールドまでスクロールし、必要なデフォルト値を subdomain.domain の形式で入力します。
6. 「保存」をクリックし、変更内容を記録します。
7. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

高度なタスク

MIME 推測の有効化

リライトは、パーサーの選択にページの MIME タイプを使用します。WebLogic や Oracle などの一部の Web サーバーは MIME タイプを送信しません。これに対応するには、「パーサーと URI のマッピング」リストボックスにデータを追加して、MIME 推測機能を有効にします。

▶ MIME 推測を有効にするには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 編集する属性が含まれるゲートウェイプロファイルをクリックします。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「リライト」タブをクリックし、詳細サブセッションを表示します。
6. 「MIME 推測を有効」チェックボックスにチェックマークを付け、MIME 推測を有効にします。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

パースする URI マッピングリストの作成

MIME 推測機能が有効で、サーバーが MIME タイプを送信しない場合は、このリストを使用してパースする URI をマッピングします。

複数の URI はセミコロンで区切られます。

たとえば、HTML=* .html; *.htm; *Servlet のように指定します。

この例の設定では、HTML リライトは拡張子が html、htm、Servlet のすべてのページのコンテンツをリライトします。

▶ パーサーを URI にマッピングするには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。

3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 編集する属性が含まれるゲートウェイプロファイルをクリックします。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「リライト」タブをクリックし、「基本」サブセクションを表示します。
6. 「パーサーを MIME タイプにマップ」フィールドまでスクロールし、編集ボックスにデータを追加します。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

マスキングの有効化

マスキングを有効にすることで、リライトはページのイントラネット URL が判読されないように URI をリライトします。

▶ マスキングを有効にするには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 編集する属性が含まれるゲートウェイプロファイルをクリックします。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「リライト」タブをクリックし、「基本」サブセクションを表示します。
6. 「マスキングを有効」チェックボックスにチェックマークを付け、マスキングを有効にします。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

マスキングのためのシード文字列の指定

URI のマスキングには、シード文字列が使用されます。シード文字列は、マスキングアルゴリズムによって生成されるランダムな文字列です。

注 マスクされた URI をブックマークしても、このシード文字列が変更されたり、ゲートウェイが再起動された場合は機能しなくなります。

▶ マスキングのためのシード文字列を指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 編集する属性が含まれるゲートウェイプロファイルをクリックします。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「リライト」タブをクリックし、「基本」サブセクションを表示します。
6. 「マスキングのシード文字列」フィールドまでスクロールし、編集ボックスに文字列を追加します。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

マスクしない URI のリストの作成

アプレットなど一部のアプリケーションはインターネット URI を必要とし、マスクすることができません。これらのアプリケーションを指定するには、リストボックスに URI を追加します。

たとえば、次のように追加します。

```
*/Applet/Param*
```

リストボックスに追加した URL は、コンテンツの URI

`http://abc.com/Applet/Param1.html` がルールセット内のルールと一致する場合にマスクされません。

▶ マスクしない URI のリストを作成するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。

3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 編集する属性が含まれるゲートウェイプロファイルをクリックします。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「リライト」タブをクリックし、「基本」サブセクションを表示します。
6. 「マスクしない URI」フィールドまでスクロールし、編集ボックスに URI を追加します。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

ゲートウェイプロトコルと元の URI プロトコルの同一化

ゲートウェイが HTTP と HTTPS の両方のモードで稼動する場合、HTML コンテンツ内で参照されるリソースへのアクセスに同じプロトコルを使用するようにリライトを設定できます。

たとえば、元の URL が `http://intranet.com/Public.html` であれば、HTTP ゲートウェイが追加されます。元の URL が `https://intranet.com/Public.html` であれば、HTTPS ゲートウェイが追加されます。

注 これは、スタティックな URI だけに適用され、JavaScript によって生成されるダイナミック URI には適用されません。

▶ ゲートウェイプロトコルと元の URI プロトコルを同一化するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 編集する属性が含まれるゲートウェイプロファイルをクリックします。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「リライト」タブをクリックし、「基本」サブセクションを表示します。
6. 「ゲートウェイプロトコルを元の URI プロトコルと同じにする」チェックボックスにチェックマークを付けます。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

デバッグログを使用した障害追跡

リライタに関する問題の原因を特定するには、デバッグログを有効にする必要があります。

デバッグメッセージは、次のように分類されます。

- **Error**: リライタが修復できないエラー。
- **Warning**: リライタの動作に重大な影響を及ぼさない警告。リライタはこのようなエラーを修復できますが、動作不良が生じる可能性もあります。一部の警告メッセージは情報提供用です。たとえば、警告メッセージとして「**Not rewriting image content**」がログに記録されたとします。リライタは画像をリライトするという動作を想定していないので、これは問題ありません。
- **Message**: リライタが提供する最上位レベルの情報。

リライタのデバッグレベルの設定

▶ リライタのデバッグレベルを設定するには

1. ゲートウェイマシンに root としてログインし、次のファイルを編集します。

```
gateway-install-root/SUNWam/config/AMConfig-instance-name.properties
```

2. デバッグレベルを設定します。

```
com.ipplanet.services.debug.level=
```

次のデバッグレベルがあります。

error: 重要なエラーだけがログとしてデバッグファイルに記録されます。このようなエラーが発生すると、通常、リライタは機能を停止します。

warning: 警告メッセージがログに記録されます。

message: すべてのデバッグメッセージがログに記録されます。

off: デバッグメッセージはログに記録されません。

3. `AMConfig-instance-name.properties` ファイルの次のプロパティに、デバッグファイルのディレクトリを指定します。

```
com.ipplanet.services.debug.directory=/var/opt/SUNWam/debug
```

この `/var/opt/SUNWam/debug` は、デフォルトのデバッグディレクトリです。

4. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

デバッグファイル名

デバッグレベルを「message」に設定すると、複数のファイルが生成されます。表 3-2 はリライタのデバッグファイルとその内容を示しています。

表 3-2 リライタのデバッグファイル

ファイル名	説明
RuleSetInfo	リライトに使用されたすべてのルールは、このファイルに記録されます。
Original Pages	ページの URI、解決された URI (ページ URI と異なる場合)、コンテンツの MIME、ページに適用されたルールセット、パーサー MIME、および元のコンテンツが記録されます。 このファイルには、パースに関連する具体的な error/warning/message も記録されます。 message モードではすべての内容が記録され、warning モードと error モードではリライト時に発生した例外だけが記録されます。
Rewritten Pages	ページの URI、解決された URI (ページ URI と異なる場合)、コンテンツの MIME、ページに適用されたルールセット、パーサー MIME、およびリライトされたコンテンツが記録されます。 この情報は、デバッグモードを message に設定した場合にだけ記録されます。
Unaffected Pages	このファイルには、変更されなかったページのリストが含まれます。
URIInfo Pages	検出され、変換された URL が記録されます。コンテンツが元のデータと同じ状態で残されたすべてのページの詳細が記録されます。 記録される詳細情報は ページの URI、MIME、符号化データ、リライト時に適用されたルールセットの ID、およびパーサー MIME です。

これらのファイルのほかに、リライタはこれらのファイルに記録されないデバッグメッセージを記録するファイルを生成します。このファイルの名前は 2 つの部分から構成されます。最初の部分は pwRewriter または psSRARewriter で、2 番目の部分は portal または gateway-profile-name を使用した拡張子です。

デバッグファイルは、ポータルまたはゲートウェイに表示されます。これらのファイルは、AMConfig-instance-name.properties ファイルに指定されているディレクトリに格納されます。

リライタコンポーネントは、デバッグ用に次のファイルを生成します。

prefix_RuleSetInfo.extension

prefix_OriginalPages.extension

prefix_RewrittenPages.extension

prefix_UnaffectedPages.extension

prefix_URIInfo.extension

各表記の意味は次のとおりです。

prefix は、URL スクレイパーを使用した場合は `psRewriter`、ゲートウェイを使用した場合は `psSRAPRewriter` です。

extension は、URL スクレイパーを使用した場合は `portal`、ゲートウェイを使用した場合は `gateway-profile-name` です。

たとえば、ページの変換にゲートウェイ上のリライトとデフォルトのゲートウェイプロファイルを使用した場合は、次のデバッグファイルが生成されます。

`psSRAPRewriter_RuleSetInfo.default`

`psSRAPRewriter_OriginalPages.default`

`psSRAPRewriter_RewrittenPages.default`

`psSRAPRewriter_UnaffectedPages.default`

`psSRAPRewriter_URIInfo.default`

`psSRAPRewriter.default`

サンプルの操作

ここで説明する内容は次のとおりです。

- リライトが必要なコンテンツを含む簡単な HTML ページ
- コンテンツのリライトに必要なルール
- リライトされた HTML ページ

これらのサンプルページは、*portal-server-URL/rewriter* ディレクトリ内にあります。ルールを適用する前にページの内容を参照し、その後、リライトされてゲートウェイを通じて出力されたファイルを参照することで、ルールがどのように機能しているかを理解することができます。一部のサンプルでは、ルールはすでに `default_gateway_ruleset` の一部として含まれています。一部のサンプルでは、ルールを `default_gateway_ruleset` に含めなければならない場合があります。これについては、該当箇所でも説明します。

注 太字で表示されている文は、リライトされたことを示します。

次のサンプルが用意されています。

- HTML
 - [HTML 属性のサンプル](#)
 - [HTML フォームのサンプル](#)
 - [HTML アプレットのサンプル](#)
- JavaScript
 - 変数
 - [JavaScript URL 変数のサンプル](#)
 - [JavaScript コンテンツのサンプル](#)
 - [JavaScript DHTML 変数のサンプル](#)
 - [JavaScript DJS 変数のサンプル](#)
 - [JavaScript SYSTEM 変数のサンプル](#)
 - 関数
 - [JavaScript URL 関数のサンプル](#)
 - [JavaScript EXPRESSION 関数のサンプル](#)
 - [JavaScript DHTML 関数のサンプル](#)
 - [JavaScript DJS 関数のサンプル](#)

- XML
 - [XML 属性のサンプル](#)

HTML コンテンツのサンプル

HTML 属性のサンプル

▶ **HTML 属性のサンプルを使用するには**

1. このサンプルには次の場所からアクセスできます。

`portal-server-URL/rewriter/HTML/attrib/attribute.html`

2. ゲートウェイサービスの「ドメインとサブドメインのプロキシ」リストに `abc.sesta.com` と `host1.siroe.com` が定義されていることを確認してください。
これが定義されていないと、直接の接続が想定され、ゲートウェイ URL がプレフィックスとして追加されません。

このサンプルに指定されているルールはすでに `default_gateway_ruleset` に定義されているので、追加の必要はありません。

リライト前の HTML

```
<html>
Rewriting starts
<head>
<title>TEST PAGE () </title>
</head>
ID-htmlattr.1
<br><br>
1. a href <a
href="http://abc.sesta.com/images/logo.gif">http://..
```

```
5. href <a href="../../../images/logo.gif">../../../images/</a> <br><br>
Rewriting ends
</html>
```

ルール

```
<Attribute name="href"/>
```

リライト後の HTML

```
<html>
Rewriting starts
<head>
<title>TEST PAGE () </title>
</head>
ID-htmlattr.1
<br><br>
```

**1. a href http://...
**

default_gateway_ruleset に <Attrib name="href"/> ルールがすでに定義されているので、この URL はリライトされます。URL はすでに絶対 URL であるため、ゲートウェイ URL だけがプレフィックスとして追加されます。ゲートウェイサービスの「ドメインとサブドメインのプロキシ」リストに abc.sesta.com が定義されていることを確認してください。これが定義されていないと、直接接続が想定されるため、ゲートウェイ URL がプレフィックスとして追加されません。

2. href https://...

// この場合も、ゲートウェイサービスの「ドメインとサブドメインのプロキシ」リストに host1.siroe.com が定義されていることを確認してください。これが定義されていないと、直接接続が想定されるため、ゲートウェイ URL がプレフィックスとして追加されません。

```
<br><br>
```

3. href ../images/

// 相対パスが指定されているため、必要なサブディレクトリの後にゲートウェイ URL と portal-server-URL がプレフィックスとして追加されます。用意されたサンプル構造で、HTML ディレクトリの下に images という名前のディレクトリが指定されないため、このリンクは機能しません。

```
<br><br>
```

```
4 href <a
```

```
href="gateway-URL/portal-server-URL/rewriter/HTML/attrib/images/logo.gif">images/..</a> <br><br>
```

// 相対パスが指定されているため、必要なサブディレクトリの後にゲートウェイ URL と Portal Server URL がプレフィックスとして追加されます。

```
5. href <a
```

```
href="gateway-URL/portal-server-URL/rewriter/images/logo.gif">../.. /images/</a> <br><br>
```

// 相対パスが指定されているため、必要なサブディレクトリの後にゲートウェイ URL と Portal Server URL がプレフィックスとして追加されます。用意されたサンプル構造で、Rewriter ディレクトリの下に images という名前のディレクトリが指定されないため、このリンクは機能しません。

```
Rewriting ends
```

```
</html>
```

HTML ダイナミック JavaScript トークンのサンプル

▶ HTML JavaScript トークンのサンプルを使用するには

1. このサンプルには次の場所からアクセスできます。

```
portal-server-URL/rewriter/HTML/jstokens/JStokens.html
```

2. このサンプルで指定されているルールを、default_gateway_ruleset の「JavaScript ソースをリライトするためのルール (Rules for Rewriting JavaScript Source)」セクションに追加します。
3. Access Manager 管理コンソールの「Portal Server 設定」のリライターサービスで default_gateway_ruleset を編集します。
4. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

リライト前の HTML

```
<html>
```

```
<head>
```

```
Rewriting starts
```

```
<script language="javascript">
function Check(test,ind){
if (ind == 'blur')
{alert("testing onBlur")}
if (ind == 'focus')
{alert("testing onFocus")}
}
</SCRIPT>
</head>
<body>
<form>
<input TYPE=TEXT SIZE=20 value=blur
onAbort="Check('/indexblur.html','blur');return;">
<input TYPE=TEXT SIZE=20 value=blur
onBlur="Check('/indexblur.html','blur');return;">
<input TYPE=TEXT SIZE=20 value=focus
onFocus="Check('/focus.html','focus');return;">
<input TYPE=TEXT SIZE=20 value=focus
onChange="Check('/focus.html','focus');return;">
<input TYPE=TEXT SIZE=20 value=focus
onClick="Check('/focus.html','blur');return;">
<br><br>
</form>
</body>
Rewriting ends
</html>
```

ルール

```
<Attribute name="onClick" type="DJS"/>
<Function type="URL" name="Check" paramPatterns="y"/>
```

注 <Function name="URL" name="Check" paramPatterns="y"/> は JavaScript 関数ルールです。JavaScript 関数のサンプルで詳しく説明します。

リライト後の HTML

```
<html>

<head>

Rewriting starts

<script language="javascript">
function Check(test,ind){
if (ind == 'blur')
{alert("testing onBlur")}
if (ind == 'focus')
{alert("testing onFocus")}
}
}
</SCRIPT>

</head>

<body>

<form>

<input TYPE=TEXT SIZE=20 value=blur onAbort="Check('gateway
URL/portal-server-URL/indexblur.html', 'blur');return;">

<input TYPE=TEXT SIZE=20 value=blur onBlur="Check('gateway
URL/portal-server-URL/indexblur.html', 'blur');return;">

<input TYPE=TEXT SIZE=20 value=focus onFocus="Check('gateway
URL/portal-server-URL/focus.html', 'focus');return;">

<input TYPE=TEXT SIZE=20 value=focus onChange="Check('gateway
URL/portal-server-URL/focus.html', 'focus');return;">

<input TYPE=TEXT SIZE=20 value=focus onClick="Check('gateway
URL/portal-server-URL/focus.html', 'blur');return;">

// このサンプルではすべての文がリライトされます。それぞれ、ゲートウェイと
Portal Server の URL が先頭に追加されます。これは、default_gateway_ruleset ファ
イルに onAbort、onBlur、onFocus、onChange、および onClick のルールが定義されて
いるためです。リライトは JavaScript トークンを検出し、後の処理のために
JavaScript 関数ルールに渡します。サンプルの 2 番目のルールは、リライトするパラ
メータをリライトに伝えます。

</body>

<br>

Rewriting ends
```

</html>

HTML フォームのサンプル

▶ フォームのサンプルを使用するには

1. 次の場所にあるサンプルフォームにアクセスします。

`portal-server-URL/rewriter/HTML/forms/formrule.html`

2. ゲートウェイサービスの「ドメインとサブドメインのプロキシ」リストに `abc.sesta.com` が定義されていることを確認してください。

これが定義されていないと、直接の接続が想定され、ゲートウェイ URL がプレフィックスとして追加されません。

3. このサンプルで指定されているルールを、`default_gateway_ruleset` の「HTML 属性をリライトするためのルール (Rules for Rewriting HTML Attributes)」セクションに追加します。

4. Access Manager 管理コンソールの「Portal Server 設定」のリライターサービスで `default_gateway_ruleset` を編集します。

5. 端末ウィンドウからゲートウェイを再起動します。

`gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start`

リライト前の HTML ページ

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
</head>
<body>
RW_START
<p>
<form name="form1" method="Post"
action="http://abc.sesta.com/casestudy/html/form.html">
<input type="hidden" name="name1" value="0|1234|/test.html">
<input type="hidden" name="name3" value="../../html/test.html">
<form name="form2" method="Post"
action="http://abc.sesta.com/testcases/html/form.html"><br>
<input type="hidden" name="name1"
value="0|1234|../../html/test.html"></form>
```

```
RW_END </p>
</body>
</html>
```

ルール

```
<Form source="*" name="form1" field="name1"
valuePatterns="0|1234|"/>
```

リライト後の HTML ページ

```
<HTML>
<HEAD>
RW_START
</HEAD>
<BODY>
<P>
<FORM name=form1 method=POST
action="gateway-URL/http://abc.sesta.com/casestudy/html/form.html"
// default_gateway_ruleset 内の HTML ルールの一部として <Attribute
name="action"/> が定義されているため、この URL はリライトされます。この URL
はすでに絶対 URL であるため、ゲートウェイ URL だけをプレフィックスとして追加
する必要があります。ゲートウェイサービスの「ドメインとサブドメインのプロキシ」
リストに abc.sesta.com が定義されていることを確認してください。これが定義され
ていないと、直接接続が想定されるため、ゲートウェイ URL がプレフィックスとして
追加されません。
<input type=hidden name=name1 value="0|1234|gateway
URL/portal-server-URL/test.html">
// ここではフォーム名は form1、フィールド名は name1 です。これはルールに指定さ
れたフォーム名とフィールド名に一致します。ルールはこの文の value に一致する
valuePatterns を 0|1234| と宣言します。したがって、valuePattern の後の URL が
リライトされます。Portal Server の URL とゲートウェイの URL が先頭に追加されま
す。valuePatterns の詳細については、108 ページの「ルールでのパターンマッチン
グの使用」を参照してください。
<input type=hidden name=name3 value="../../html/test.html">
name はルールに指定される field 名と一致しないため、この URL はリライトされま
せん。
</FORM>
```

```
<FORM name=form2 method=POST
action="gateway-URL/http://abc.sesta.com/casestudy/html/form.html">
<BR>
```

// <Attribute name="action"/> はデフォルトルールセットの HTML ルールの一部として定義されているため、この URL はリライトされます。この URL はすでに絶対 URL であるため、ゲートウェイ URL だけをプレフィックスとして追加する必要があります。

```
<input type=hidden name=name1 value="0|1234|../../html/test.html">
// フォーム名がルールに指定される名前と一致しないため、この URL はリライトされません。
</FORM>
</BODY>
RW_END
</HTML>
```

HTML アプレットのサンプル

▶ アプレットのサンプルを使用するには

1. アプレットの class ファイルを入手します。RewriteURLinApplet.class ファイルは、次の場所にあります。

```
portal-server-URL/rewriter/HTML/applet/appletcode
```

アプレットコードを参照するページのベース URL は次のとおりです。

```
portal-server-URL/rewriter/HTML/applet/rule1.html
```

2. このサンプルで指定されているルールを、default_gateway_ruleset の「HTML 属性をリライトするためのルール (Rules for Rewriting HTML Attributes)」セクションに追加します。
3. Access Manager 管理コンソールの「Portal Server 設定」のリライターサービスで default_gateway_ruleset を編集します。
4. ゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

リライト前の HTML

```
<html>
```

```
Rewriting starts
```

```
<br>
```

```

<applet codebase=appletcode code=RewriteURLinApplet.class
archive=/test>

<param name=Test1 value="/index.html">
<param name=Test2 value="../index.html">
<param name=Test3 value="../../index.html">

</applet>

Rewriting ends

</html>

```

ルール

```

<Applet source="*/rule1.html" code="RewriteURLinApplet.class"
param="Test*" />

```

リライト後の HTML

```

<HTML>

```

```

Rewriting starts

```

```

<BR>

```

```

<APPLET
codebase=gateway-URL/portal-server-URL/rewriter/HTML/applet/appletc
ode=RewriteURLinApplet.class archive=/test>

```

<Attrib name="codebase"/> ルールがすでに default_gateway_ruleset ファイルの一部として存在するため、この URL はリライトされます。ゲートウェイと Portal Server の URL が appletcode ディレクトリのパスの前にプレフィックスとして追加されます。

```

<param name=Test1 value="gateway-URL/portal-server-URL/index.html">

```

// ページのベース URL が rule1.html で、パラメータ名がルールに指定されたパラメータ Test* と一致するため、この URL はリライトされます。index.html は root レベルに指定されているため、ゲートウェイと Portal Server の URL がプレフィックスとして直接追加されます。

```

<param name=Test2
value="gateway-URL/portal-server-URL/rewriter/HTML/index.html">

```

// ページのベース URL が rule1.html で、パラメータ名がルールに指定されたパラメータ Test* と一致するため、この URL はリライトされます。必要に応じて、パスがプレフィックスとして追加されます。

```

<param name=Test3
value="gateway-URL/portal-server-URL/rewriter/index.html">

```

// ページのベース URL が rule1.html で、パラメータ名がルールに指定されたパラメータ Test* と一致するため、この URL はリライトされます。必要に応じて、パスがプレフィックスとして追加されます。

```
</APPLET>
Rewriting ends
</HTML>
```

JavaScript コンテンツのサンプル

JavaScript URL 変数のサンプル

▶ **JavaScript の URL 変数のサンプルを使用するには**

1. このサンプルには次の場所からアクセスできます。

```
portal-server-URL/rewriter/JavaScript/variables/url/js_urls.html
```

2. ゲートウェイサービスの「ドメインとサブドメインのプロキシ」リストに `abc.sesta.com` が定義されていることを確認してください。

これが定義されていないと、直接の接続が想定され、ゲートウェイ URL がプレフィックスとして追加されません。

3. このサンプルで指定されているルールを、`default_gateway_ruleset` の「JavaScript ソースをリライトするためのルール (Rules for Rewriting JavaScript Source)」セクションに追加します。
4. Access Manager 管理コンソールの「Portal Server 設定」のリライタサービスで `default_gateway_ruleset` を編集します。
5. ルールを追加した場合は、次のコマンドでゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

リライト前の HTML ページ

```
<html>
Rewriting starts
<head>
<title>JavaScript Variable test page</title>
</head>
<body>
<script LANGUAGE="Javascript">
```

```
<!--
//URL Variables
var imgsrc="/tmp/tmp.jpg";
var imgsrc="./tmp/tmp.jpg";
var imgsrc="../tmp/tmp.jpg";
var imgsrc="../../tmp/tmp.jpg";
var imgsrc="http://abc.sesta.com/tmp/tmp.jpg";
var imgsrc="../../../tmp/tmp.jpg";
var imgsrc="tmp/tmp.jpg";
//-->
</SCRIPT>
<br>
Testing JavaScript variables!
<br>

<br>
Image
</body>
<br>
Rewriting ends
</html>
```

ルール

```
<Variable name="imgsrc" type="URL"/>
```

リライト後の HTML ページ

```
<html>
Rewriting starts
<head>
<title>JavaScript Variable test page</title>
</head>
<body>
```

```

<script LANGUAGE="Javascript">
<!--
//URL Variables
var imgsrc="gateway-URL/portal-server-URL/tmp/tmp.jpg";
var
imgsrc="gateway-URL/portal-server-URL/rewriter/Javascript/variables
/url/tmp/tmp.jpg";
var
imgsrc="gateway-URL/portal-server-URL/rewriter/Javascript/variables
/tmp/tmp.jpg";
var
imgsrc="gateway-URL/portal-server-URL/rewriter/Javascript/tmp/tmp.j
pg";
var imgsrc="gateway-URL/http://abc.sesta.com/tmp/tmp.jpg";
var imgsrc="gateway-URL/portal-server-URL/rewriter/tmp/tmp.jpg";
var
imgsrc="gateway-URL/portal-server-URL/rewriter/Javascript/variables
/url/tmp/tmp.jpg";
// 上記のすべての URL は、タイプが URL で、ルールで指定された imgsrc という名
前を持つ JavaScript 変数です。したがってこれらの URL の先頭に、ゲートウェイと
Portal Server の URL がプレフィックスとして追加されます。必要に応じて、Portal
Server URL の後にパスが追加されます。
//-->
</SCRIPT>
<br>
Testing JavaScript variables!
<br>

default_gateway_ruleset に <Attribute name="src"/> ルールが定義されているの
で、この行はリライトされます。
<br>
Image
</body>

```

```
<br>
Rewriting ends
</html>
```

JavaScript EXPRESSION 変数のサンプル

▶ JavaScript の EXPRESSION 変数のサンプルを使用するには

1. このサンプルには次の場所からアクセスできます。
`portal-server-URL/rewriter/JavaScript/variables/expr/expr.html`
2. このサンプルで指定されているルールを、`default_gateway_ruleset` の「JavaScript ソースをリライトするためのルール (Rules for Rewriting JavaScript Source)」セクションに追加します (まだ追加していない場合)。
3. Access Manager 管理コンソールの「Portal Server 設定」のリライターサービスで `default_gateway_ruleset` を編集します。
4. ルールを追加した場合は、次のコマンドでゲートウェイを再起動します。
`gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start`

リライト前の HTML ページ

```
<html>
<head>
<title>JavaScript EXPRESSION Variables Test Page</title>
</head>
<body>
<script LANGUAGE="Javascript">
<!--
//EXPRESSION 変数
var expvar1="images";
var expvar2="/logo.gif";
var expvar = expvar1 + expvar2;
document.write("<A HREF="+expvar+">EXPRESSION</A><P>")
var expvar="/images/logo"+".gif";
document.write("<A HREF="+expvar+">EXPRESSION</A><P>")
//-->
```

```
</SCRIPT>
Testing JavaScript EXPRESSION variables
</body>
</html>
```

ルール

```
<Variable type="EXPRESSION" name="expvar"/>
```

リライト後の HTML ページ

```
<html>
<head>
<title>JavaScript EXPRESSION Variables Test Page</title>
</head>
<body>
<SCRIPT>
// リライタは、ラッパー関数 psSRAPRewriter_convert_expression をここに追加
// します。
</SCRIPT>
<script LANGUAGE="Javascript">
<!--
//EXPRESSION 変数
var expvar1="images";
var expvar2="/logo.gif";
var expvar =psSRAPRewriter_convert_expression( expvar1 + expvar2);
// リライタはこの文の右側を JavaScript EXPRESSION 変数として認識します。リラ
// イタはサーバー側でこの式の値を解決することができません。したがって
psSRAPRewriter_convert_expression 関数が式の前に追加されます。式はクライア
// ント側で評価され、必要に応じてリライトされます。
document.write("<A HREF="+expvar+">EXPRESSION</A><P>")
// 前の文のリライト後の値 expvar は、この式の値に到達するために使用されます。
// 結果は有効な URL ( サンプルのこの位置にグラフィックが配置される ) であるため、
// リンクが機能します。
var expvar="gateway URL/portal-server-URL/images/logo"+" .gif";
```

// リライタは `expvar` の右側を文字列式として認識します。これはサーバー側で解決できるため、直接リライトされます。

```
document.write("<A HREF="+expvar+">EXPRESSION</A><P>")
```

// 前の文のリライト後の値 `expvar` は、この式の値に到達するために使用されます。結果が有効な URL ではない (最終的な位置にグラフィックが配置されない) ため、リンクは機能しません。

```
//-->
```

```
</SCRIPT>
```

```
Testing JavaScript EXPRESSION variables
```

```
</body>
```

```
</html>
```

JavaScript DHTML 変数のサンプル

▶ JavaScript の DHTML 変数のサンプルを使用するには

1. このサンプルには次の場所からアクセスできます。

```
portal-server-URL/rewriter/JavaScript/variables/dhtml/dhtml.html
```

2. ゲートウェイサービスの「ドメインとサブドメインのプロキシ」リストに `abc.sesta.com` が定義されていることを確認してください。これが定義されていないと、直接の接続が想定され、ゲートウェイ URL がプレフィックスとして追加されません。
3. このサンプルで指定されているルールを、`default_gateway_ruleset` の「JavaScript ソースをリライトするためのルール (Rules for Rewriting JavaScript Source)」セクションに追加します (まだ追加していない場合)。Access Manager 管理コンソールの「Portal Server 設定」のリライタサービスで `default_gateway_ruleset` を編集します。
4. ルールを追加した場合は、次のコマンドでゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

リライト前の HTML ページ

```
<html>
```

```
<head>
```

```
<title>JavaScript DHTML Variable Test Page</title>
```

```
</head>
```

```
<body>
```

```
<script LANGUAGE="Javascript">
<!--
//DHTML Var
var dhtmlVar="<a href=../../images/test.html>"
var dhtmlVar="<a href=../images/test.html>"
var dhtmlVar="<a href=/images/test.html>"
var dhtmlVar="<a href=images/test.html>"
var dhtmlVar="<a href=http://abc.sesta.com/images/test.html>"
var dhtmlVar="<img src=http://abc.sesta.com/images/test.html>"
//-->
</SCRIPT>
<br><br>
Testing DHTML Variables
<br><br>
IMAGE
</body>
</html>
```

ルール

```
<Variable name="DHTML">dhtmlVar</Variable>
```

リライト後の HTML ページ

```
<html>
<head>
<title>JavaScript DHTML Variable Test Page</title>
</head>
<body>
<script LANGUAGE="Javascript">
<!--
//DHTML Var
var dhtmlVar="<a
href=gateway-URL/portal-server-URL/rewriter/JavaScript/images/test.
html>"
```

// JavaScript DHTML ルールは dhtmlVar の右側を動的 HTML コンテンツとして識別します。このため、default_gateway_ruleset ファイル内の HTML ルールが適用されます。動的 HTML には href 属性が含まれています。

default_gateway_ruleset には、<Attribute name="href"/> ルールが定義されています。したがって、href 属性の値がリライトされます。ただし、URL は絶対 URL ではありません。このため、相対 URL はページのベース URL、および必要なサブディレクトリに置き換えられます。次に、ゲートウェイ URL が URL のプレフィックスとして追加され、最終的なリライト出力となります。

```
var dhtmlVar="<a
href=gateway-URL/portal-server-URL/./images/test.html>"
```

// ページのベース URL が追加され、またゲートウェイ URL がプレフィックスとして追加されているため、最終的な URL は機能しません。これは最初の URL ../images/test.html が正確ではないためです。

```
var dhtmlVar="<a
href=gateway-URL/portal-server-URL/images/test.html>"
```

// ここでも、JavaScript DHTML ルールは右側を動的 HTML コンテンツとして識別し、それを HTML ルールに渡します。default_gateway_ruleset の HTML ルール <Attribute name="href"/> が適用され、文は次のようにリライトされます。ゲートウェイの URL と Portal Server の URL が先頭に追加されます。

```
var dhtmlVar="<a href=gateway
URL/portal-server-URL/rewriter/JavaScript/variables/dhtml/images/test.html>"
```

```
var dhtmlVar="<a href=gateway
URL/http://abc.sesta.com/images/test.html>"
```

```
var dhtmlVar="<img
src=gateway-URL/http://abc.sesta.com/images/test.html>"
```

// JavaScript DHTML ルールは右側の動的 HTML コンテンツを識別し、文を HTML ルールに渡します。default_gateway_ruleset 内の <Attribute name="src"/> ルールが適用されます。URL はすでに絶対 URL であるため、ゲートウェイ URL だけをプレフィックスとして追加する必要があります。ゲートウェイサービスの「ドメインとサブドメインのプロキシ」リストに abc.sesta.com が定義され、この URL がリライトされることを確認してください。

```
//-->
```

```
</SCRIPT>
```

```
<br><br>
```

```
Testing DHTML Variables
```

```
<br><br>
```

```

```

default_gateway_ruleset に <Attribute name="src"/> ルールが定義されているので、この行はリライトされます。

```
<br><br>
Image
</body>
</html>
```

JavaScript DJS 変数のサンプル

▶ JavaScript の DJS 変数のサンプルを使用するには

1. このサンプルには次の場所からアクセスできます。

```
portal-server-URL/rewriter/JavaScript/variables/djs/djs.html
```

2. ゲートウェイサービスの「ドメインとサブドメインのプロキシ」リストに abc.sesta.com が定義されていることを確認してください。これが定義されていないと、直接の接続が想定され、ゲートウェイ URL がプレフィックスとして追加されません。
3. このサンプルで指定される 2 つのルールを、default_gateway_ruleset の「JavaScript ソースをリライトするためのルール (Rules for Rewriting JavaScript Source)」セクションに追加します (まだ追加していない場合)。Access Manager 管理コンソールの「Portal Server 設定」のリライターサービスで default_gateway_ruleset を編集します。

4. ゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

リライト前の HTML ページ

```
<html>
<head>
<title>Dynamic JavaScript Variable Test Page</title>
</head>
<body>
<script LANGUAGE="Javascript">
<!--
var dJSVar="var dJSimgsrc='/tmp/tmp/jpg';"
```

```

var dJSVar="var dJSimgsrc='../ ../../tmp/tmp/jpg';"
var dJSVar="var dJSimgsrc='http://abc.sesta.com/tmp/tmp/jpg';"
//-->
</SCRIPT>
<br>
Testing Dynamic JavaScript Variables
<br>

<br>
Image
</body>
</html>

```

ルール

```

<Variable name="dJSVar" type="DJS"/>
<Variable name="dJSimgsrc" type="URL"/>

```

リライト後のHTML ページ

```

<html>
<head>
<title>Dynamic JavaScript Variable Test Page</title>
</head>
<body>
<script LANGUAGE="Javascript">
<!--
var dJSVar="var
dJSimgsrc='gateway-URL/portal-server-URL/tmp/tmp/jpg';"
var dJSVar="var
dJSimgsrc='gateway-URL/portal-server-URL/rewriter/tmp/tmp/jpg';"
var dJSVar="var
dJSimgsrc='gateway-URL/http://abc.sesta.com/tmp/tmp/jpg';"

```

// 上のすべての文は、ゲートウェイ URL と Portal Server URL でリライトされます。必要に応じて適切なパスがプレフィックスとして追加されます。最初のルールは、dJSVar の右側を動的 JavaScript 変数として識別します。これは 2 番目のルールに渡され、2 番目のルールは dJSimgsrc の右側をタイプ URL の JavaScript 変数として識別します。これにより、文は次のようにリライトされます。

```
//-->
</SCRIPT>
<br>
Testing Dynamic JavaScript Variables
<br>

default_gateway_ruleset に <Attribute name="src"/> ルールが定義されているの
で、この行はリライトされます。
<br>
Image
</body>
</html>
```

JavaScript SYSTEM 変数のサンプル

▶ JavaScript の SYSTEM 変数のサンプルを使用するには

1. このサンプルには次の場所からアクセスできます。
2. このサンプルで指定されているルールを、default_gateway_ruleset の「JavaScript ソースをリライトするためのルール (Rules for Rewriting JavaScript Source)」セクションに追加します (まだ追加していない場合)。
3. Access Manager 管理コンソールの「Portal Server 設定」のリライタサービスで default_gateway_ruleset を編集します。
4. ゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

リライト前の HTML ページ

```
<html>
<head>
```

```

<title>JavaScript SYSTEM Variables Test Page</title>
</head>
<body>
<script LANGUAGE="Javascript">
<!--
//SYSTEM 変数
alert(window.location.pathname);
//document.write("<A
HREF="+window.location.pathname+">SYSTEM</A><P>")
//-->
</SCRIPT>

Testing JavaScript SYSTEM Variables
<br>
This page displays the path where the current page is located when
loaded.
</body>
</html>

```

ルール

```
<Variable name="window.location.pathname" type="SYSTEM"/>
```

リライト後のHTML

```

<html>
<head>
<title>JavaScript SYSTEM Variables Test Page</title>
</head>
<body>
<SCRIPT>
convertssystem function definition...
</SCRIPT>
<script LANGUAGE="Javascript">
<!--
//SYSTEM 変数

```

```
alert(psSRAPRewriter_convert_system(window.location,
window.location.pathname, "window.location"));
```

// リライタは window.location.pathname を JavaScript の SYSTEM 変数として識別します。この変数の値はサーバー側で決定することができません。このため、リライタはこの変数の前に psSRAPRewriter_convert_pathname 関数を追加します。このラッパー関数は、クライアント側で変数の値を判断し、必要に応じてリライトします。

```
//-->
```

```
</SCRIPT>
```

```
Testing JavaScript SYSTEM Variables
```

```
<br>
```

```
This page displays the path where the current page is located when
loaded.
```

```
</body>
```

```
</html>
```

JavaScript URL 関数のサンプル

▶ JavaScript の URL 関数のサンプルを使用するには

1. このサンプルには次の場所からアクセスできます。

```
portal-server-URL/rewriter/JavaScript/functions/url/url.html
```

2. このサンプルで指定されているルールを、default_gateway_ruleset の「JavaScript ソースをリライトするためのルール (Rules for Rewriting JavaScript Source)」セクションに追加します (まだ追加していない場合)。Access Manager 管理コンソールの「Portal Server 設定」のリライタサービスで default_gateway_ruleset を編集します。

3. ゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

リライト前の HTML ページ

```
<html>
```

```
<body>
```

```
JavaScript URL Function Test Page
```

```
<br>
```

```
<script language="JavaScript">
```

```
<!--
```

```

function test(one,two,three)
{
alert(one + "##" + two + "##" +three);
}
test("/test.html","../test.html","123");
window.open("/index.html","gen",width=500,height=500);
//-->
</SCRIPT>
</body>
</html>

```

ルール

```

<Function type="URL" name="test" paramPatterns="y,y"/>
<Function type="URL" name="window.open" paramPatterns="y"/>

```

リライト後の HTML ページ

```

<html>
<body>
JavaScript URL Function Test Page
<br>
<script language="JavaScript">
<!--
function test(one,two,three)
{
alert(one + "##" + two + "##" +three);
}
test("/test.html","../test.html","123");
window.open("gateway-URL/portal-server-URL/index.html","gen",width=
500,height=500);
//-->
</SCRIPT>
</body>
</html>

```

JavaScript EXPRESSION 関数のサンプル

▶ JavaScript の EXPRESS 関数のサンプルを使用するには

1. このサンプルには次の場所からアクセスできます。

```
portal-server-URL/rewriter/JavaScript/functions/expr/expr.html
```

2. このサンプルで指定されているルールを、default_gateway_ruleset の「JavaScript ソースをリライトするためのルール (Rules for Rewriting JavaScript Source)」セクションに追加します (まだ追加していない場合)。
3. Access Manager 管理コンソールの「Portal Server 設定」のリライターサービスで default_gateway_ruleset を編集します。
4. ゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

リライト前の HTML ページ

```
<html>
<body>
JavaScript EXPRESSION Function Test Page
<br><br><br>
<script language="JavaScript">
<!--
function jstest2()
{
return ".html";
}
function jstest1(one)
{
return one;
}
var dir="/images/test"
var test1=jstest1(dir+"/test"+jstest2());
document.write("<a HREF="+test1+">Test</a>");
alert(test1);
//-->
```

```
</SCRIPT>
```

```
</body>
```

```
</html>
```

ルール

```
<Function type="EXPRESSION" name="jstest1" paramPatterns="y"/>
```

リライト後の HTML ページ

```
<html>
```

```
<body>
```

```
JavaScript EXPRESSION Function Test Page
```

```
<br><br><br>
```

```
<script>
```

```
<!--
```

```
// ここには、psSRAPRewriter_convert_expression を含むさまざまな関数が表示  
// されます。
```

```
//-->
```

```
</SCRIPT>
```

```
<script language="JavaScript">
```

```
<!--
```

```
function jstest2()
```

```
{
```

```
return ".html";
```

```
}
```

```
function jstest1(one)
```

```
{
```

```
return one;
```

```
}
```

```
var dir="/images/test"
```

```
var
```

```
test1=jstest1(psSRAPRewriter_convert_expression(dir+"/test"+jstest2  
( )));
```

// このルールは、関数 `jstest1` のタイプ `EXPRESSION` の最初のパラメータをリライトする必要があることを指定します。この式の値は `/test/images/test.html` です。この値の前に、Portal Server URL とゲートウェイ URL がプレフィックスとして追加されます。

```
document.write("<a HREF="+test1+">Test</a>");
alert(test1);
//-->
</SCRIPT>
</body>
</html>
```

JavaScript DHTML 関数のサンプル

▶ JavaScript の DHTML 関数のサンプルを使用するには

1. このサンプルには次の場所からアクセスできます。
`portal-server-URL/rewriter/JavaScript/functions/dhtml/dhtml.html`
2. このサンプルで指定されているルールを、`default_gateway_ruleset` の「JavaScript ソースをリライトするためのルール (Rules for Rewriting JavaScript Source)」セクションに追加します (まだ追加していない場合)。
3. Access Manager 管理コンソールの「Portal Server 設定」のリライターサービスで `default_gateway_ruleset` を編集します。
4. ゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

リライト前の HTML ページ

```
<html>
<head>
Testing JavaScript DHTML Functions
<br>
<br>
<script>
<!--
document.write('<a href="/index.html">write</a><BR>')
document.writeln('<a href="index.html">writeln</a><BR>')
```

```

document.write("http://abc.sesta.com/index.html<BR>")
document.writeln("http://abc.sesta.com/index.html<BR>")

//-->
</SCRIPT>
</head>
<body BGCOLOR=white>
<br><br>
Testing document.write and document.writeln
</body>
</html>

```

ルール

```

<Function type="DHTML" name=" document.write" paramPatterns="y"/>
<Function type="DHTML" name=" document.writeln" paramPatterns="y"/>

```

リライト後の HTML ページ

```

<html>
<head>
Testing JavaScript DHTML Functions
<br>
<br>
<script>
<!--
document.write('<a
href="gateway-URL/portal-server-URL/index.html">write</a><BR>')

// 最初のルールは、DHTML JavaScript 関数 document.write の最初のパラメータを
リライトする必要があることを示します。リライトは、最初のパラメータが単純な
HTML 文であることを識別します。default_gateway_ruleset の HTML ルールのセ
クションには <Attribute name="href" /> ルールが定義されており、リライトが必要な
文はこのルールによって決定されます。

document.writeln('<a
href="gateway-URL/portal-server-URL/rewriter/JavaScript/functions/d
html/index.html">writeln</a><BR>')

```

// 2 番目のルールは、DHTML JavaScript 関数 `document.writeln` の最初のパラメータをリライトする必要があることを示します。リライタは、最初のパラメータが単純な HTML 文であることを識別します。`default_gateway_ruleset` の HTML ルールのセクションには `<Attribute name="href" />` ルールが定義されており、リライトが必要な文はこのルールによって決定されます。

```
document.write("http://abc.sesta.com/index.html<BR>")
document.writeln("http://abc.sesta.com/index.html<BR>")
```

// DHTML ルールは関数 `document.write` と `document.writeln` を検出しますが、上の文はリライトされません。これは最初のパラメータが HTML ではないためです。パラメータは任意の文字列となり、リライタはこれをどのようにリライトするかを指示されていません。

```
//-->
</SCRIPT>
</head>
<body BGCOLOR=white>
<br><br>
Testing document.write and document.writeln
</body>
</html>
```

JavaScript DJS 関数のサンプル

▶ JavaScript の DJS 関数のサンプルを使用するには

1. このサンプルには次の場所からアクセスできます。

```
portal-server-URL/rewriter/JavaScript/functions/djs/djs.html
```

2. ゲートウェイサービスの「ドメインとサブドメインのプロキシ」リストに `abc.sesta.com` が定義されていることを確認してください。

これが定義されていないと、直接の接続が想定され、ゲートウェイ URL がプレフィックスとして追加されません。

3. このサンプルで指定されているルールを、`default_gateway_ruleset` の「JavaScript ソースをリライトするためのルール (Rules for Rewriting JavaScript Source)」セクションに追加します (まだ追加していない場合)。Access Manager 管理コンソールの「Portal Server 設定」のリライタサービスで `default_gateway_ruleset` を編集します。
4. ゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

リライト前の HTML ページ

```
<html>
Test for JavaScript DJS Functions
<br>
<script>
menu.addItem(new NavBarMenuItem("All Available
Information", "JavaScript:top.location='http://abc.sesta.com'"));
//menu.addItem(new NavBarMenuItem("All Available
Information", "http://abc.sesta.com"));
</script>
</html>
```

ルール

```
<Function type="DJS" name="NavBarMenuItem" paramPatterns="y"/>
<Variable type="URL" name="top.location"/>
```

リライト後の HTML ページ

```
<html>
Testing JavaScript DJS Functions
<br>
<script>
menu.addItem(new NavBarMenuItem("All Available
Information", "javaScript:top.location='gateway-URL/http://abc.sesta
.com'"));
```

// abc.sesta.com はゲートウェイサービスの「ドメインとサブドメインのプロキシ」リストのエントリです。したがって、リライタはこの URL をリライトする必要があります。ただし、これは絶対 URL であるため、Portal Server の URL をプレフィックスとして追加する必要はありません。DJS ルールは、DJS 関数 `NavBarMenuItem` の 2 番目のパラメータをリライトする必要があることを指定します。ただし、2 番目のパラメータは同じく JavaScript 変数です。2 番目のルールは、この変数の値をリライトする場合に必要となります。2 番目のルールは、JavaScript 変数 `top.location` の値をリライトする必要があることを指定します。これらのすべての条件に適合するため、URL がリライトされます。

```
//menu.addItem(new NavBarMenuItem("All Available
Information", "http://abc.sesta.com"));
```

// DJS ルールは、関数 `NavBarMenuItem` の 2 番目のパラメータをリライトする必要がありますが、この文はリライトされません。これはリライタが 2 番目のパラメータを `HTML` と認識しないためです。

```
</script>
</html>
```

XML 属性のサンプル

▶ XML 属性のサンプルを使用するには

1. このサンプルには次の場所からアクセスできます。

```
portal-server-URL/rewriter/XML/attrib.html
```

2. このサンプルで指定されているルールを、`default_gateway_ruleset` の「XML ソースをリライトするためのルール (Rules for Rewriting XML Source)」セクションに追加します (まだ追加していない場合)。
3. Access Manager 管理コンソールの「Portal Server 設定」のリライタサービスで `default_gateway_ruleset` を編集します。
4. ゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

リライト前の XML

```
<html>
RW_START
<body>
<xml>
<baseroot href="/root.html"/>
</xml>
<xml>
<img href="image.html"/>
</xml>
<xml>
<string href="1234|substring.html"/>
</xml>
<xml>
```

```
<check href="1234|string.html"/>
</xml>
</body>
RW_END
</html>
```

ルール

```
<Attribute name="href" tag="check" valuePatterns="1234|"/>
```

リライト後のHTML

```
<html>
Rewriting starts
<br>
<br>
<body>
<xml><baseroot href="/root.html"/></xml>
<xml><img href="image.html"/></xml>
<xml><string href="1234|substring.html"/></xml>
<xml><check
href="1234|gateway-URL/portal-server-URL/rewriter/XML/string.html"/
></xml>
// この文はルールで指定された条件と一致するため、リライトされます。Attribute
name は href、tag は check、valuePatterns は 1234 です。valuePatterns よりも後の
文字列はリライトされず。valuePatterns の詳細については、108 ページの「ルールでのパターンマッチングの使用」を参照してください。
</body>
Rewriting ends
</html>
```

ケーススタディ

ここでは、メールクライアントのソース HTML ページの例について説明します。このケーススタディでは、考えられるすべての例やルールについて説明することはできません。これはあくまでも、イントラネットページにルールを適用するために使用するルールセットの例です。

前提条件

このケーススタディは、次のような前提で行います。

- メールクライアントのベース URL は、abc.siroe.com とします。
- ゲートウェイの URL は gateway.sesta.com とします。
- ゲートウェイサービスの「ドメインとサブドメインのプロキシ」リストでエントリを関連付けます。

ページ例 1

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<!-- saved from
url=(0053)http://abc.siroe.com/mailclient/destin/?Cmd=navbar -->
<HTML XMLNS:WM><HEAD>
<META http-equiv=Content-Type content="text/html; CHARSET=utf-8">
<META http-equiv=Pragma content=no-cache>
<META http-equiv=Expires content=0><!--Copyright (c) 2000 Microsoft
Corporation. All rights reserved.--><!--CURRENT FILE== "IE5" "WIN32"
navbar -->
<STYLE>WM¥:DROPMENU {
BEHAVIOR: url(http://abc.siroe.com/mailweb/controls/dropmenu.htc)
}
</STYLE>
<LINK href="destin_files/navbar.css" type=text/css rel=stylesheet>
<SCRIPT language=javascript>
var g_szUserBase= "http://abc.siroe.com/mailclient/destin+"/";
var g_szFolder= ".";
var g_szVirtualRoot= "http://abc.siroe.com/mailweb";
var g_szImagePath= g_szVirtualRoot + "/img/";
```

```

</SCRIPT>
<SCRIPT src="/destin_files/navbar.js"></SCRIPT>
<META content="MSHTML 6.00.2600.0" name=GENERATOR></HEAD>
<BODY oncontextmenu=return(event.ctrlKey);
onselectstart=return(false);
id=outbar_mainbody style="BACKGROUND-COLOR:appworkspace"
leftMargin=0
topMargin=0 scroll=no>
<TABLE class=nbTableMain id=nbTableMain style="HEIGHT:100%"
cellSpacing=0
cols=1 cellPadding=0 rows="2">
<TBODY>
<TR>
<TD class=treeBrand>
<DIV class=treeOFLOW><IMG
style="PADDING-RIGHT:0px; PADDING-LEFT:0px; PADDING-BOTTOM:0px;
PADDING-TOP:0px"
src="/destin_files/logo-ie5.gif" border=0</DIV></TD></TR>
<TR height="100%">
<TD>
<TABLE class=nbTable cellSpacing=0 cols=1 cellPadding=0 rows="4">
<TBODY>
<TR>
<TD class=nbFlybar id=show_navbar onkeydown=flybar_keydown()
onclick=ToggleTab(this.id) tabIndex=0 noWrap>
<DIV class=treeOFLOW>Shortcuts</DIV></TD></TR>
<TR style="HEIGHT: 100%">
<TD id=idOutbarpane style="TEXT-ALIGN:center" vAlign=top><A
id=inbox
href="http://abc.siroe.com/mailclient/destin/Inbox/?Cmd=contents&am
p;Page=1"
target=viewer alt="Go to inbox"><IMG class=nbImage alt="Go to inbox"
src="destin_files/navbar-inbox.gif"></A>

```

```

<DIV class=nbLabel>Inbox</DIV><BR><A id=calendar
href="http://abc.siroe.com/mailclient/destin/Calendar/?Cmd=contents"
target=viewer alt="Go to calendar"><IMG class=nbImage
alt="Go to calendar" src="destin_files/navbar-calendar.gif"></A>
<DIV class=nbLabel>Calendar</DIV><BR><A id=contacts
href="http://abc.siroe.com/mailclient/destin/Contacts/?Cmd=contents"
target=viewer alt="Go to contacts"><IMG class=nbImage
alt="Go to contacts" src="destin_files/navbar-contacts.gif"></A>
<DIV class=nbLabel>Contacts</DIV><BR><A id=options
href="http://abc.siroe.com/mailclient/destin/?Cmd=options"
target=viewer alt="Go to options"><IMG class=nbImage
alt="Go to options" src="destin_files/navbar-options.gif"></A>
<DIV class=nbLabel>Options</DIV></TD></TR>
<TR style="HEIGHT: 1.5em">
<TD class=nbFlybar id=show_folders onkeydown=flybar_keydown()
onclick=ToggleTab(this.id) tabIndex=0 noWrap>
<DIV class=treeOFLOW>Folders</DIV></TD></TR>
<TR>
<TD class=nbTreeProgress id=treeProgress style="DISPLAY:none"
vAlign=top noWrap><SPAN id=idLoading
style="OVERFLOW:hidden">Loading...</SPAN>
</TD></TR></TBODY></TABLE></TD></TR></TBODY></TABLE>
</BODY></HTML>

```

説明

表 3-3 は、サンプルルールセットとケーススタディの間のマッピングを示しています。

表 3-3 サンプルルールセットとケーススタディのマッピング

ページコンテンツ	適用されるルール	リライタの出力	説明
<pre>var g_szVirtualRoot="http://abc.siroe.com/mailweb";</pre>	<pre><Variable name="URL"> g_szVirtualRoot </Variable></pre>	<pre>var g_szVirtualRoot= "http://gateway.sesta.com/http://abc.siroe.com/mailweb";</pre>	<p>g_szVirtualRoot は単一の URL を値に持つ変数です。</p> <p>このルールは、タイプ URL の変数 g_szVirtualRoot を検索するようにリライタに指示します。このような変数が Web ページに存在する場合、リライタはこれを絶対 URL に変換し、ゲートウェイ URL をプレフィックスとして追加します。</p>
<pre>src="/destin_files/logo-ie5.gif"</pre>	<pre><Attribute name="src" /></pre>	<pre>src="http://gateway.sesta.com/http://abc.siroe.com/destin_files/logo-ie5.gif"</pre>	<p>src は属性名であり、タグまたは valuePattern は付加されません。</p> <p>このルールは、src という名前の属性をすべて検索し、その属性の値をリライトするようにリライタに指示します。</p>

表 3-3 サンプルルールセットとケーススタディのマッピング (続き)

ページコンテンツ	適用されるルール	リライトの出力	説明
href="http://abc.siroe.com/mailclient/destination/Inbox/?Cmd=contents&Page=1"	<Attribute name="href" />	href="http://gateway.sesta.com/http://abc.siroe.com/mailclient/destination/Inbox/?Cmd=contents&Page=1"	href は属性名であり、タグまたは valuePattern は付加されません。 このルールは、href という名前の属性をすべて検索し、その属性の値をリライトするようにリライトに指示します。

注 ルールセットを適用する順序は、ホスト名 - サブドメイン - ドメインの順です。

たとえば、「ドメインベースのルールセット」リストに次のエントリを指定していると仮定します。

sesta.com|ruleset1

eng.sesta.com|ruleset2

host1.eng.sesta.com|ruleset3

ruleset3 は host1 のすべてのページに適用されます。

ruleset2 は、host1 から取得されたページを除く eng のすべてのページに適用されます。

ruleset1 は、eng サブドメインおよび host1 から取得されたページを除く、sesta.com ドメインのすべてのページに適用されます。

5. 「保存」をクリックし、変更内容を記録します。
6. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

Outlook Web Access 用のルールセット

SRA ソフトウェアでは、Sun Java System Web Server および IBM アプリケーションサーバー上で、Outlook Web Access (OWA) から MS Exchange 2000 SP3 インストールおよび MS Exchange 2003 にアクセスする機能がサポートされます。

▶ OWA のルールセットを設定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルをクリックします。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「URI をルールセットにマップ」フィールドで、Exchange 2000 がインストールされているサーバー名を入力し、それに続けて Exchange 2000 Service Pack 4 OWA ルールセットを入力します。

例

`exchange.domain.com|exchange_2000sp3_owa_ruleset`

6.x と 3.0 のルールセットのマッピング

次の表は、SRA のリライトルールと従来のリリースの Portal Server 製品とのマッピングを示しています。

表 3-4 SP3 のルールのマッピング

リライト 6.0 の DTD 要素	リライト 3.0 リストボックス名
HTML コンテンツのルール	
Attribute: URL	HTML 属性のリライト
Attribute: DJS	JavaScript を含む HTML 属性のリライト
Form	フォーム入力タグリストのリライト
Applet	アプレット / オブジェクトパラメータ値リストのリライト
JavaScript コンテンツのルール	
Variable: URL	URL タイプの JavaScript 変数のリライト
Variable: EXPRESSION	JavaScript 変数関数のリライト
Variable: DHTML	HTML タイプの JavaScript 変数のリライト
Variable: DJS	JavaScript タイプの JavaScript 変数のリライト
Variable: SYSTEM	JavaScript システム変数のリライト
Function: URL	JavaScript 関数パラメータのリライト

表 3-4 SP3 のルールのマッピング (続き)

リライタ 6.0 の DTD 要素	リライタ 3.0 リストボックス名
Function: EXPRESSION	JavaScript 関数パラメータ関数のリライト
Function: DHTML	HTML タイプの JavaScript 関数パラメータのリライト
Function: DJS	JavaScript タイプの JavaScript 関数パラメータのリライト
XML コンテンツのルール	
Attribute: URL	XML ドキュメントの属性値のリライト
TagText	XML ドキュメントのテキストデータのリライト
CSS コンテンツのルール	
ルールは不要です。デフォルトでは、すべての URL が変換されます。	
WML コンテンツのルール	
ルールは定義されていません。WML は HTML として処理され、HTML ルールが適用されます。	
WMLScript コンテンツのルール	
WML スクリプトはサポートされていません。	

NetFile

この章では、NetFile とその操作について説明します。NetFile の設定については、[287 ページの第 10 章「NetFile の設定」](#)を参照してください。

この章で説明する内容は次のとおりです。

- [NetFile の概要](#)
- [サポートされるファイルアクセスプロトコル](#)
- [NetFile のデバッグの有効化](#)
- [NetFile のロギングの有効化](#)

NetFile の概要

NetFile はリモートファイルシステムとリモートディレクトリへのアクセスと操作を可能にする、ファイルマネージャアプリケーションです。

SRA の NetFile コンポーネントは、Java1 および Java2 アプレットとして使用できます。ブラウザに Java2 プラグインをインストールしていない場合は、Java1 アプレットを使用できます。Java2 アプレットのインタフェースは改善され、より使いやすくなっています。

NetFile の主な機能は次のとおりです。

- 共有ファイルやフォルダの追加または削除
- ファイルのアップロードとダウンロード
- ファイルとフォルダの検索
- GZIP と ZIP によるファイル圧縮
- NetFile 環境内でのメール機能
- 現在の NetFile セッション情報の保存

- ファイルのドラッグ&ドロップ

NetFile の設定については、第 10 章「NetFile の設定」を参照してください。

サポートされるファイルアクセスプロトコル

NetFile では FTP、NFS、および jCIFS (Microsoft Windows) の各プロトコルを使用してリモートシステムにアクセスできます。NetFile には次のファイルアクセスプロトコル機能が含まれています。

- ユーザーが AUTODETECT を指定してシステムを追加すると、NetFile は次の順に使用プロトコルを自動的に検出します。
 - ポート 21 で FTP サーバーのホストをチェックします。FTP 応答に文字列「NetWare」が含まれていれば、NETWARE ホストと見なされます。
 - ポート 2049 で NFS サーバーのホストをチェックします。
 - ポート 139 で Microsoft Windows のホストをチェックします。
 - 上のすべてに該当しない場合、ホストタイプの判別が不可能であるというメッセージが表示されます。

要求されるホストとの接続には、最初に検出されるファイルシステムのタイプが使用されます。ホストの検出順序は、Access Manager の管理コンソールで変更できます。

注 サーバーが標準以外のポートで稼動していると、接続に失敗します。

- NetFile では、使用するファイルサーバーおよびプロトコルをユーザーが選択できます。

次に、それぞれのプロトコルについて、サポートされるプラットフォームを示します。

表 4-1 ファイルシステムとサポートされるプロトコル

ファイルシステム/プロトコル	プラットフォーム
FTP	Novell Netware の Novell FTP 5.1 サーバー
	Microsoft Windows 4.0 の MS FTP サーバー 4.0
	Microsoft Windows 2000 の MS FTP サーバー 5.0
	Solaris FTP サーバー
	WU_FTP 2.6.1
	ProFTPD 1.2.8
NFS	vsFTPd 1.2.0
	Solaris 2.6 以降
jCIFS	Microsoft Windows 95/98/NT/2000/ME/XP

注 NetFile を使用して ProFTPD サーバーにファイルをアップロードするには、ProFTPD サーバーが稼動するホストの `proftpd.conf` ファイルで「AllowStoreRestart」を「on」に設定する必要があります。

注 Novel 1 Netware は FTP サーバーを通じてのみサポートされ、ネイティブアクセスを通じてはサポートされません。

注 Microsoft Windows (SMB/CIFS) ファイルシステムにアクセスするには、Portal Server 上に jCIFS がインストールされている必要があります。jCIFS は、CIFS/SMB ネットワーキングプロトコルを実装するオープンソースのクライアントライブラリです。

6. NetFile サービスに基づいて NetFile ポリシーを作成し、NetFile へのアクセスを必要とする組織とロールに NetFile ポリシーを割り当てます。
7. NetFile へのアクセスを必要とする各ユーザーに NetFile を割り当てます。
ポリシーとサービスの作成と割り当ての詳細については、『Access Manager 管理ガイド』を参照してください。

NetFile のデバッグの有効化

デバッグ情報の場所は、Portal Server ノードの `AMConfig-instance-name.properties` ファイルに設定されている `com.ipplanet.services.debug.directory` 属性の値によって異なります。

たとえば、`com.ipplanet.services.debug.directory` 属性に次の値が設定されているとします。

```
/var/opt/SUNWam/debug/
```

この場合、NetFile のデバッグ情報は `/var/opt/SUNWam/debug` ディレクトリの `srapNetFile` ファイルから取得できます。

詳細については、『Access Manager 管理ガイド』を参照してください。

NetFile のロギングの有効化

NetFile のロギングを有効にするには、Access Manager ロギングサービスを使用するログの場所を指定します。ログファイルの名前は `srapNetFile` で、このファイルのデフォルトの位置は `/var/opt/SUNWam/logs` ディレクトリです。

Netlet

この章では、ユーザーのリモートデスクトップとイントラネット上のアプリケーションを実行しているサーバーとの間で、Netlet を使用してアプリケーションを安全に実行する方法について説明します。Netlet の設定については、[305 ページの第 11 章「Netlet の設定」](#)を参照してください。

この章で説明する内容は次のとおりです。

- [Netlet の概要](#)
- [リモートホストからのアプレットのダウンロード](#)
- [Netlet ルールの定義](#)
- [Netlet ルールの例](#)
- [Netlet ログインの有効化](#)
- [デバッグログインの有効化](#)
- [Sun Ray 環境での Netlet の実行](#)

Netlet の概要

Sun Java™ System Portal Server のユーザーが、一般的なアプリケーションや企業専用のアプリケーションをリモートデスクトップで安全に実行できると便利な場合があります。プラットフォームに Netlet を設定すると、このようなアプリケーションに安全にアクセスできるようになります。

Netlet を使用することで、インターネットなどのセキュリティの弱いネットワークで一般的な TCP/IP サービスを安全に実行できます。TCP/IP アプリケーション (Telnet や SMTP など)、HTTP アプリケーション、同じポートを使用するすべてのアプリケーションを実行できます。

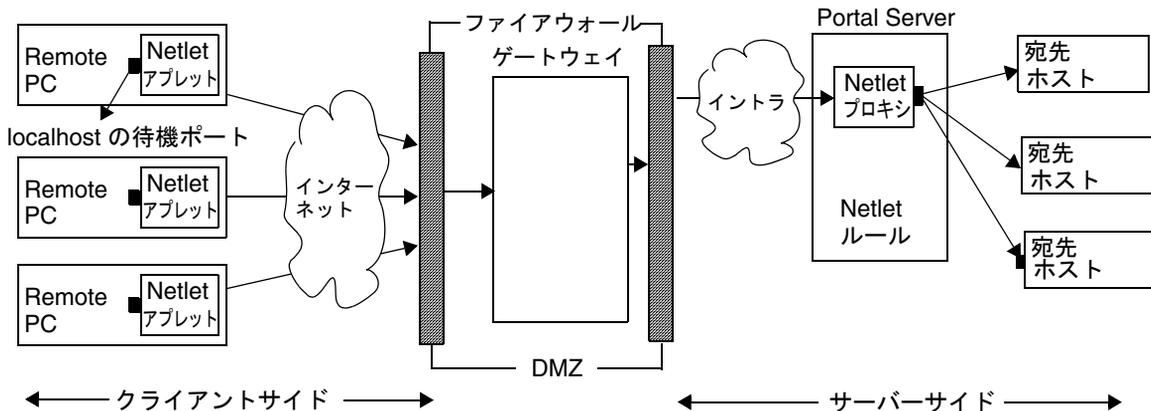
アプリケーションが TCP/IP ベースであるか同じポートを使用する場合、Netlet を介してアプリケーションを実行できます。

注 ダイナミックポートは、FTP を使用する場合にだけサポートされます。
 Microsoft Exchange を使用する場合は、OWA (Outlook Web Access) を使
 用します。

Netlet のコンポーネント

図 5-1 は、Netlet で使用される各種コンポーネントを示しています。

図 5-1 Netlet のコンポーネント



localhost の待機ポート

これは Netlet アプレットが待機するクライアントマシン上のポートです。クライアントマシンは localhost です。

Netlet アプレット

Netlet アプレットは、リモートクライアントマシンと、Telnet、Graphon、Citrix などのイントラネットアプリケーションの間で、暗号化された TCP/IP トンネルの設定を担当します。アプレットはパケットを暗号化してゲートウェイに送信し、ゲートウェイからの応答パケットを解読してローカルアプリケーションに送信します。

スタティックルールの場合、Netlet アプレットは、ユーザーがポータルにログインすると自動的にダウンロードされます。ダイナミックルールの場合、ダイナミックルールに対応するリンクをユーザーがクリックしたときにアプレットがダウンロードされます。スタティックルールとダイナミックルールについては、[192 ページの「ルールのタイプ」](#)を参照してください。

Sun Ray 環境での Netlet の実行については、[206 ページの「Sun Ray 環境での Netlet の実行」](#)を参照してください。

Netlet ルール

Netlet ルールでは、クライアントマシンで実行する必要があるアプリケーションが、対応する宛先ホストにマッピングされます。つまり Netlet は、Netlet ルールに定義されたポートに送信されたパケットに対してだけ動作します。これにより、セキュリティが向上します。

管理者は Netlet の機能に対して特定のルールを設定する必要があります。これらのルールによって、使用される暗号化方式や、呼び出す URL、ダウンロードするアプレット、宛先ポート、宛先ホストなどの詳細が指定されます。クライアントマシン上のユーザーが Netlet を通じて要求を行う場合、これらのルールに基づいて接続の確立方法が速やかに決定されます。詳細については、[187 ページの「Netlet ルールの定義」](#)を参照してください。

Netlet プロバイダ

これは Netlet の UI コンポーネントです。プロバイダを使用することで、Portal Server のデスクトップから必要なアプリケーションを設定できます。プロバイダにリンクが作成され、ユーザーはこのリンクをクリックして必要なアプリケーションを実行します。また、デスクトップ Netlet プロバイダで、ダイナミックルールの宛先ホストを指定できます。[187 ページの「Netlet ルールの定義」](#)を参照してください。

Netlet プロキシ (オプション)

ゲートウェイは、リモートクライアントマシンとゲートウェイ間の安全なトンネルを保証します。Netlet プロキシの使用は任意です。インストール時にこのプロキシをインストールしない選択も可能です。Netlet プロキシについては、[70 ページの「Netlet プロキシの使用」](#)を参照してください。

Netlet の使用例

Netlet 使用時には、次の一連のイベントが行われます。

1. リモートユーザーが **Portal Server** デスクトップにログインします。
2. ユーザー、ロール、または組織にスタティック Netlet ルールが定義されている場合は、リモートクライアントに Netlet アプレットが自動的にダウンロードされます。

ユーザー、ロール、または組織にダイナミックルールが定義されている場合は、Netlet プロバイダに必要なアプリケーションを手動で設定する必要があります。Netlet アプレットは、ユーザーが Netlet プロバイダのアプリケーションリンクをクリックしたときにダウンロードされます。スタティックルールとダイナミックルールについては、[187 ページの「Netlet ルールの定義」](#)を参照してください。

3. Netlet は Netlet ルールで定義されたローカルポートで待機します。
4. Netlet はリモートクライアントとホストの間で、Netlet ルールで指定されたポートを使用するチャンネルを確立します。

Netlet の操作

Netlet が異なる組織間のさまざまなユーザーの要求に合わせて機能するには、次の手順を実行する必要があります。

1. ユーザー要件に基づいて、スタティックルールとダイナミックルールのどちらを作成するかを決定します。[192 ページの「ルールのタイプ」](#)を参照してください。
2. **Access Manager** 管理コンソールの「サービス設定」タブで、Netlet テンプレートにグローバルオプションを定義します。[305 ページの第 11 章「Netlet の設定」](#)を参照してください。
3. ルールの基準を組織、ロール、ユーザーから選択し、各レベルで必要に応じて修正します。組織、ロール、ユーザーについては、『**Portal Server 管理ガイド**』を参照してください。

注 `srapNetletServlet.properties` ファイルのフレームセットパラメータの値に、英語以外の言語は使用しないでください。

リモートホストからのアプレットのダウンロード

URL から返されたページに、リモートマシンからフェッチする必要があるアプレットが埋め込まれていることがあります。ただし、Java™ のセキュリティによって、アプレットがそのアプレットのダウンロード元以外のホストと通信することは許可されていません。アプレットがローカルネットワークポートを使用してゲートウェイと通信できるようにするには、Access Manager 管理コンソールの「アプレットのダウンロード」フィールドを確認し、次の構文を指定する必要があります。

local-port:server-host:server-port

各表記の意味は次のとおりです。

local-port はローカルポートです。Netlet は、アプレットから送信されるトラフィックをここで待機します。

server-host は、アプレットのダウンロード元です。

server-port は、アプレットのダウンロードに使用されるポートです。

Netlet ルールの定義

Netlet の設定は Netlet ルールによって定義されます。このルールは、Access Manager 管理コンソールの「SRA 設定」セクションで設定されます。Netlet ルールは組織、ロール、またはユーザーのいずれかに対して設定できます。Netlet ルールをロールまたはユーザーに対して定義したときは、組織を選択してから目的のロールまたはユーザーを選択します。

警告 Netlet ルールはマルチバイトエントリをサポートしません。Netlet ルールのどの編集フィールドにもマルチバイト文字を指定しないでください。

Netlet ルールには 64000 を超えるポート番号を指定できません。

表 5-1 は、Netlet ルールのフィールドを示しています。

表 5-1 Netlet ルールのフィールド

パラメータ	説明	値
ルール名	この Netlet ルールの名前を指定します。各ルールに一意の名前を指定する必要があります。これは、特定のルールへのアクセスを定義する場合に便利です。詳細については、 314 ページの「Netlet ルールへのアクセスの定義」 を参照してください。	
暗号化方式	暗号化方式を定義するか、ユーザーが選択できる方式のリストを指定します。	<p>選択した暗号化方式は、Netlet プロバイダにリスト表示されます。ユーザーは必要な暗号化方式をリストから選択できます。</p> <p>デフォルト: Netlet 管理コンソールで指定するデフォルト VM ネイティブ暗号化方式と、デフォルト Java プラグイン暗号化方式。</p>
URL	<p>URL ユーザーが Netlet プロバイダのリンクをクリックしたときにブラウザで開かれる URL を指定します。ブラウザにはアプリケーションのウィンドウが表示され、ルールによって指定されたローカルポート番号で localhost に接続します。</p> <p>相対 URL を指定する必要があります。</p>	<p>Netlet ルールによって呼び出されるアプリケーションへの URL。</p> <p>例: telnet://localhost:30000</p> <p>アプリケーションの呼び出しにアプレットが必要な場合は、その URL を指定します。</p> <p>null: 指定した URL によってアプリケーションが起動されない、またはデスクトップで制御されない場合に設定する値。通常は Web ベース以外のアプリケーションで使用されます。</p>

表 5-1 Netlet ルールのフィールド (続き)

パラメータ	説明	値
アプレットのダウンロード	このルールでアプレットのダウンロードが必要であるかどうかを指定します。	<p>チェックマークなし: アプレットをダウンロードしない</p> <p>チェックマークあり: ループバックポートを使用してアプレットを Portal Server マシンからダウンロードする</p> <p>アプレットの詳細は、<i>local-port:server-host:server-port</i> の形式で指定します。</p> <ul style="list-style-type: none"> • <i>local-port</i> はクライアントの宛先ポートを表します。このポートは、デフォルトのループバックポートとは異なる必要があります。詳細については、第 11 章「Netlet の設定」を参照してください。各ルールに一意の <i>local port</i> を指定します。 • <i>server-host</i> はアプレットのダウンロード元のサーバー名を表します。 • <i>server-port</i> はアプレットのダウンロードに使用されるサーバー上のポートを表します。 <p>アプレットがダウンロードされる場合にサーバーが指定されていないときは、アプレットは Portal Server のホストからダウンロードされません。</p>
拡張セッション	Netlet がアクティブの場合、Portal Server セッションのアイドル時間のタイムアウトを制御します。	<p>チェックマークあり: Netlet がアクティブで、ほかのポータルアプリケーションがアイドルの場合にのみ、ポータルセッションを持続するようにする。</p> <p>チェックマークなし: Netlet アプリケーションがアクティブで、ほかのポータルアプリケーションがアイドルの場合でも、ポータルセッションのアイドル時間は、セッションに指定されたアイドル時間でタイムアウトになる。</p>

表 5-1 Netlet ルールのフィールド (続き)

パラメータ	説明	値
ローカルポート	Netlet が待機するクライアントのポート。	<p><i>local-port</i> の値は一意である必要があります。特定のポート番号を複数のルールに指定することはできません。</p> <p>複数のローカルポートを指定するのは、複数の接続に複数のホストを指定している場合です。構文については、196 ページの「複数ホスト接続のスタティックルール」を参照してください。</p> <p>FTP ルールでは、ローカルポートは 30021 である必要があります。</p>
宛先ホスト	Netlet 接続の受信者。	<p><i>host</i> : Netlet 接続を受信するホスト名。これはスタティックルールで使用されます。siroe などの簡易ホスト名、または <i>siroe.mycompany.com</i> などの完全修飾 DNS 形式のホスト名を指定します。次の場合に複数のホストを指定します。</p> <ul style="list-style-type: none"> 指定された各ホストとの接続を確立する場合。指定された各ホストに対して、対応するクライアントと宛先ポートを指定する必要があります。構文については、196 ページの「複数ホスト接続のスタティックルール」を参照してください。 指定されたホストのリストから、使用可能なホストへの接続を試みる場合。構文については、197 ページの「複数ホストを選択するスタティックルール」を参照してください。 <p>TARGET: 構文で TARGET を指定するルールはダイナミックルールです。TARGET は、デスクトップの Netlet プロバイダで、必要な宛先ホストをユーザーが 1 つ以上指定できることを示します。</p> <p>1 つのルールでスタティックホストと TARGET を組み合わせることはできません。</p>

表 5-1 Netlet ルールのフィールド (続き)

パラメータ	説明	値
宛先ポート	宛先ホスト上のポート。	<p>ホストと宛先ホストのほかに、宛先ポートを指定する必要があります。</p> <p>複数の宛先ホストがある場合は、複数の宛先ポートを指定できます。複数のポートは、port1+port2+port3-port4+port5 のように指定します。</p> <p>ポート番号間のプラス (+) 記号は、単一の宛先ホストに対する代替ポートを表します。</p> <p>異なる宛先ホストのポート番号を区切るときは、区切り文字としてポート番号間にマイナス (-) 記号を挿入します。</p> <p>この例では、Netlet は port1、port2、および port3 を順番に使用して、指定された最初の宛先ホストへの接続を試みます。これに失敗した場合、Netlet は port4 と port5 をこの順序で使用して 2 番目のホストへの接続を試みます。</p> <p>複数のポートは、スタティックルールでのみ設定できます。</p>

ゲートウェイが Portal Server からセッション通知を受け取るようにするには、次の情報を

```
com.ipplanet.am.jassproxy.trustAllServerCerts=true
```

Portal Server 上の次のプロパティファイルに追加します。

```
/etc/opt/SUNWam/config/AMConfig.instance-name.properties
```

2. Access Manager を再起動します。

ルールのタイプ

ルールで宛先ホストがどのように指定されているかにより、Netlet ルールは2つのタイプに分かれます。

スタティックルール

スタティックルールは、ルールの一部として宛先ホストを指定します。スタティックルールを作成する場合、ユーザーは必要な宛先ホストを指定することができません。次の例では、`sesta` は宛先ホストです。

ルール名	暗号化方式	URL	アプレットのダウンロード	拡張セッション	ローカルポート	宛先ホスト	宛先ポート
ftpstatic	SSL_RSA_WITH_RC4_128_MD5	null	false	true	30021	sesta	21

複数の宛先ホストおよびポートを設定できるのは、スタティックルールだけです。設定例については、[196 ページの「複数ホスト接続のスタティックルール」](#)を参照してください。

ダイナミックルール

ダイナミックルールでは、宛先ホストはルールの一部として指定されません。ユーザーは Netlet プロバイダで必要な宛先ホストを指定できます。次の例では、`TARGET` は宛先ホストの可変部分です。

ルール名	暗号化方式	URL	アプレットのダウンロード	拡張セッション	ローカルポート	宛先ホスト	宛先ポート
ftpdynamic	SSL_RSA_WITH_RC4_128_MD5	null	false	true	30021	TARGET	21

暗号化方式

暗号化方式に基づいて、Netlet ルールはさらに次のように分類されます。

- ユーザー設定可能な暗号化方式ルール**: このルールでは、ユーザーが選択できる暗号化方式のリストを指定できます。これらのオプション暗号化方式は、Netlet プロバイダにリスト表示されます。ユーザーは必要な暗号化方式をリストから選択できます。次の例では、ユーザーは複数の暗号化方式を選択できます。

ルール名	暗号化方式	URL	アプレットのダウンロード	拡張セッション	ローカルポート	宛先ホスト	宛先ポート
Telnet	SSL_RSA_WITH_RC4_128_SHA SSL_RSA_WITH_RC4_128_MD5	null	false	true	30000	TARGET	23

注 Portal Server ではさまざまな暗号化方式が有効になっている場合がありますが、ユーザーが選択できる暗号化方式は、Netlet ルールの一部として設定されている方式だけです。

Netlet でサポートされる暗号化方式のリストについては、[193 ページの「サポートされる暗号化方式」](#)を参照してください。

- 管理者設定暗号化方式ルール**: このルールでは、暗号化方式は Netlet ルールの一部として定義されます。ユーザーは必要な暗号化方式を選択できません。次の例では、暗号化方式は SSL_RSA_WITH_RC4_128_MD5 に設定されています。

ルール名	暗号化方式	URL	アプレットのダウンロード	拡張セッション	ローカルポート	宛先ホスト	宛先ポート
Telnet	SSL_RSA_WITH_RC4_128_MD5	null	false	true	30000	TARGET	23

Netlet でサポートされる暗号化方式のリストについては、[193 ページの「サポートされる暗号化方式」](#)を参照してください。

サポートされる暗号化方式

表 5-2 は、Netlet でサポートされる暗号化方式のリストを示しています。

表 5-2 サポートされる暗号化方式のリスト

暗号化方式

ネイティブ VM 暗号化方式

KSSL_SSL3_RSA_WITH_3DES_EDE_CBC_SHA

KSSL_SSL3_RSA_WITH_RC4_128_MD5

KSSL_SSL3_RSA_WITH_RC4_128_SHA

KSSL_SSL3_RSA_EXPORT_WITH_RC4_40_MD5

KSSL_SSL3_RSA_WITH_DES_CBC_SHA

Java プラグイン暗号化方式

SSL_RSA_WITH_3DES_EDE_CBC_SHA

SSL_RSA_WITH_RC4_128_MD5

SSL_RSA_WITH_RC4_128_SHA

SSL_RSA_EXPORT_WITH_RC4_40_MD5

SSL_RSA_WITH_DES_CBC_SHA

SSL_RSA_WITH_NULL_MD5

下位互換性

旧バージョンの Portal Server は、Netlet ルールの一部として暗号化方式をサポートしていません。暗号化方式を使用せずに既存のルールと下位互換を行うには、ルールでデフォルトの暗号化方式を指定します。暗号化方式を使用しない既存のルールは、次のとおりです。

ルール名	暗号化方式	URL	アプレットの ダウンロード	拡張セッ ション	ローカル ポート	宛先ホスト	宛先ポート
Telnet		telnet://localhos t:30000	false	true	30000	TARGET	23

これは次のように解釈されます。

ルール名	暗号化方式	URL	アプレットのダウンロード	拡張セッション	ローカルポート	宛先ホスト	宛先ポート
Telnet	デフォルト暗号化方式	telnet://localhost:30000	false	true	30000	TARGET	23

これは、管理者設定ルールでデフォルトとして選択した「暗号化方式」フィールドと同じです。詳細については、[309 ページ](#)の「デフォルトの暗号化方式の指定」を参照してください。

注 Netlet ルールには 64000 を超えるポート番号を指定できません。

Netlet ルールの例

ここでは、Netlet ルールの例をいくつか示し、Netlet 構文がどのように機能するかについて説明します。

- [基本的なスタティックルール](#)
- [複数ホスト接続のスタティックルール](#)
- [URL を呼び出すダイナミックルール](#)
- [アプレットをダウンロードするダイナミックルール](#)

基本的なスタティックルール

このルールは、クライアントマシンから `sesta` への Telnet 接続をサポートします。

ルール名	暗号化方式	URL	アプレットのダウンロード	拡張セッション	ローカルポート	宛先ホスト	宛先ポート
myrule	SSL_RSA_WITH_RC4_128_MD5	null	false	true	1111	sesta	23

各表記の意味は次のとおりです。

`myrule` はルール名です。

`SSL_RSA_WITH_RC4_128_MD5` は、適用される暗号化方式を示します。

`null` は、このアプリケーションが URL で呼び出されない、またはデスクトップから実行できないことを示します。

`false` は、クライアントがこのアプリケーションを実行するためにアプレットをダウンロードしないことを示します。

`true` は、Netlet 接続がアクティブになっても、Portal Server がタイムアウトにならないことを示します。

1111 は、Netlet が宛先ホストからの接続要求を待機するクライアント側のポートです。

`sesta` は Telnet 接続の受信側ホストの名前です。

23 は接続の宛先ホストのポート番号です。この例では、既知の Telnet ポートです。

デスクトップ Netlet プロバイダにはリンクが表示されませんが、Netlet は指定されたポート (1111) で自動的に起動して待機します。クライアントソフトウェア、この場合はポート 1111 で localhost に接続した Telnet セッションを開始するようにユーザーに指示してください。

たとえば、Telnet セッションを開始するには、クライアントは端末の UNIX コマンド行で次のコマンドを入力する必要があります。

```
telnet localhost 1111
```

複数ホスト接続のスタティックルール

このルールは、クライアントマシンから 2 台のマシン `sesta` および `siroe` への Telnet 接続をサポートします。

ルール名	暗号化方式	URL	アプレットのダウンロード	拡張セッション	ローカルポート	宛先ホスト	宛先ポート
myrule	SSL_RSA_WITH_RC4_128_MD5	null	false	true	1111	sesta	23
					1234	siroe	23

各表記の意味は次のとおりです。

23 は接続用の宛先ホスト上のポート番号です。Telnet の予約ポート番号です。

1111 は、Netlet が最初の宛先ホスト `sesta` からの接続要求を待機するポートです。

1234 は、Netlet が 2 番目の宛先ホスト `siroe` からの接続要求を待機するポートです。

このルールの最初の 6 つのフィールドは、195 ページの「基本的なスタティックルール」と同じです。2 番目の宛先ホストを識別するためのフィールドが 3 つ追加されている点が異なります。

ルールにターゲットを追加するときは、新しい宛先ホストごとに、ローカルポート、宛先ホスト、宛先ポートの 3 つのフィールドを追加する必要があります。

注 各宛先ホストへの接続を、3 つのフィールドのセットを使って記述することができます。2048 未満の待機ポート番号は、UNIX ベースのリモートクライアントでは使用できません。UNIX は下位数値のポートに制約され、`root` でリスナーを開始する必要があるためです。

このルールは前述のルールと同様に機能します。Netlet プロバイダはリンクを表示しませんが、Netlet は指定された 2 つのポート (1111 と 1234) で自動的に起動して待機します。ユーザーはクライアントソフトウェア、この場合は、ホストに接続するために `localhost` のポート 1111 に対して Telnet セッションを、2 番目のホストに接続するには、`localhost` のポート 1234 に対して Telnet セッションを開始する必要があります。

複数ホストを選択するスタティックルール

このルールは、複数の代替ホストを指定する場合に使用します。ルールの最初のホストへの接続に失敗した場合、Netlet は 2 番目に指定されたホストへの接続を試み、成功するまで指定の順に代替ホストへの接続を試みます。

ルール名	暗号化方式	URL	アプレットのダウンロード	拡張セッション	ローカルポート	宛先ホスト	宛先ポート
gojoe	SSL_RSA_WI TH_RC4_128 _MD5	/gojoe.ht ml	8000:gojoeser ver:8080	true	10491	siroe+sesta	35+26+491 -35+491

各表記の意味は次のとおりです。

10491 は、Netlet が宛先ホストからの接続要求を待機するクライアント側のポートです。

Netlet はポート 35、ポート 26、ポート 491 の順に使用可能なポートにアクセスし、`siroe` との接続を確立しようと試みます。

siroe との接続が確立できない場合、Netlet はポート 35、491 の順序で sesta への接続を試みます。

ホスト間のプラス (+) 記号は代替ホストを表します。

ポート番号間のプラス (+) 記号は、単一の宛先ホストに対する代替ポートを表します。

異なる宛先ホストのポート番号を区切るときは、区切り文字としてポート番号間にマイナス (-) 記号を挿入します。

URL を呼び出すダイナミックルール

このルールを使用することで、目的の宛先ホストを設定できるため、Netlet を使用してさまざまなホストへの Telnet 接続を確立できます。

ルール名	暗号化方式	URL	アプレットのダウンロード	拡張セッション	ローカルポート	宛先ホスト	宛先ポート
myrule	SSL_RSA_WITH_RC4_128_MD5	telnet://localhost:30000	false	true	30000	TARGET	23

各表記の意味は次のとおりです。

myrule はルール名です。

SSL_RSA_WITH_RC4_128_MD5 は、適用される暗号化方式を示します。

telnet://localhost:30000 はルールで呼び出される URL です。

false はアプレットがダウンロードされないことを示します。

拡張セッション (true) は、Netlet 接続がアクティブになっても、Portal Server がタイムアウトにならないことを示します。

30000 は、Netlet がこのルールの接続要求を待機するクライアント上のポートです。

TARGET は、ユーザーが Netlet プロバイダを使用して宛先ホストを設定する必要があることを示します。

23 は Netlet で開かれる宛先ホストのポートです。この例では、既知の Telnet ポートです。

▶ ルールの追加後に Netlet を実行するには

このルールが追加した後に、ユーザーは Netlet を目的どおりに稼働させるためにいくつかの手順を実行しなければなりません。ユーザーはクライアント側で次の操作を実行する必要があります。

1. 標準の Portal Server デスクトップの Netlet プロバイダセクションで、「編集」をクリックします。

新しい Netlet ルールが、「新規ターゲットの追加」セクションの「ルール名」に表示されます。

2. ルール名を選択し、宛先ホスト名を入力します。
3. 変更内容を保存します。

デスクトップに戻ります。デスクトップの Netlet プロバイダセクションに新しいリンクが表示されます。

4. 新しいリンクをクリックします。

新しいブラウザが起動し、Netlet ルールで指定した URL が表示されます。

注 同じルールに複数の宛先ホストを追加する場合は、この手順を繰り返します。選択された最後のリンクがアクティブです。

アプレットをダウンロードするダイナミックルール

このルールは、ダイナミックに割り当てられたホストとクライアント間の接続を定義します。このルールにより、アプレットのあるサーバーからクライアントに GO-Joe アプレットがダウンロードされます。

ルール名	暗号化方式	URL	アプレットのダウンロード	拡張セクション	ローカルポート	宛先ホスト	宛先ポート
gojoe	SSL_RSA_WITH_RC4_128_MD5	/gojoe.html	8000:gojoeserver:8080	true	3399	TARGET	58

各表記の意味は次のとおりです。

gojoe はルール名です。

SSL_RSA_WITH_RC4_128_MD5 は、適用される暗号化方式を示します。

/gojoe.html :たとえば、アプレットを含む HTML ページのパスや、ポータルが配備されている Web コンテナのドキュメントルートへの相対パスです。

8000:server:8080 は、クライアントでアプレットを受け取る宛先ポートがポート 8000であることを示します。gojoeserve はアプレットを送るサーバー名、8080 はアプレットのダウンロード元のサーバー上のポートです。

Extend Session(true) は、Netlet 接続がアクティブになっても、Portal Server がタイムアウトにならないことを示します。

3399 は、Netlet がこのタイプの接続要求を待機するクライアント上のポートです。

TARGET は、ユーザーが Netlet プロバイダを使用して宛先ホストを設定する必要があることを示します。

58 は Netlet で開かれる宛先サーバーのポートです。この例では、GoJoe のポートです。ポート 58 は宛先ホストが自分のトラフィックを待機するポートです。Netlet は新しいアプレットの情報をこのポートに渡します。

Netlet ルールの例

表 5-3 は、いくつかの一般的なアプリケーションの Netlet ルールの例を示しています。

この表には 7 つの列があります。それぞれ、「Netlet ルールのルール名」、「URL」、「ダウンロードアプレット」、「ローカルポート」、「宛先ホスト」、「宛先ポート」の各フィールドに対応します。最後の列は、ルールの説明を示します。

注	表 5-3 には、Netlet ルールの暗号化方式、およびセッションの延長のフィールドは示されていません。表に示される例で、それぞれが「SSL_RSA_WITH_RC4_128_MD5」および「true」に設定されていることを前提としています。
----------	----------------------------------------------------------------------------------------------------------------------------

表 5-3 Netlet ルールの例

ルール	URL	アプレットのダウンロード	ローカルポート	宛先ホスト	宛先ポート	説明
IMAP	null	false	10143	imapserver	143	クライアント側の Netlet ローカルポートはサーバー側の宛先ポートと同じである必要はありません。標準の IMAP と SMTP ポート以外を使用する場合は、標準ポートと異なるポートにクライアントが設定されていることを確認します。 Solaris クライアントユーザーは、root で実行している場合を除き、1024 未満のポート番号には接続できません。
SMTP	null	false	10025	smtpserver	25	
Lotus Web クライアント	null	false	80	lotus-server	80	このルールでは、Netlet がポート 80 でクライアントを待機し、ポート 80 でサーバー lotus-server に接続します。Lotus Web クライアント側で、待機するポートがサーバーポートと一致している必要があります。

表 5-3 Netlet ルールの例 (続き)

ルール	URL	アプレットのダウンロード	ローカルポート	宛先ホスト	宛先ポート	説明
Lotus Notes 非 Web クライアント	null	false	1352	lotus-dominio	1352	<p>このルールを使用すると、Lotus Notes クライアントは Netlet を通じて Lotus Domino サーバーに接続できます。クライアントがサーバーに接続する場合、サーバー名に localhost が指定されていないことを確認してください。これは、Lotus Domino サーバーの実際のサーバー名を指定する必要があります。サーバー名は、サーバーのシステム名と同じでなければなりません。Netlet を使用する場合、クライアントはその名前を 127.0.0.1 として解決する必要があります。その方法には次の 2 種類があります。</p> <ul style="list-style-type: none"> • クライアントホストテーブルで、127.0.0.1 をポイントするようにサーバー名を設定します。 • 127.0.0.1 をポイントするサーバー名の DNS エントリをエクスポートします。 <p>サーバー名は、設定時に Domino サーバーの設定に使用したサーバー名と同じ名前である必要があります。</p>

表 5-3 Netlet ルールの例 (続き)

ルール	URL	アプレットのダウンロード	ローカルポート	宛先ホスト	宛先ポート	説明
Microsoft Outlook および Exchange Server Microsoft Windows NT、Microsoft Windows 2000、および Microsoft Windows XP では、この設定は機能しません。Microsoft Windows NT、2000、および XP については、リライタ経由で Outlook Web Access を使用してください。	null	false	135	exchange	135	<p>このルールでは、Netlet がクライアントのポート 135 で待機し、ポート 135 のサーバー exchange に接続します。Outlook クライアントはこのポートを使用して、Exchange サーバーへの最初の接続試行を行い、失敗した場合は指定されている代替ポートを順に使用してサーバーと通信します。</p> <p>クライアントマシン上で次の操作を行います。</p> <ul style="list-style-type: none"> • ユーザーは Outlook クライアントに設定されている Exchange サーバーのホスト名を localhost に変更する必要があります。このオプションの場所は、Outlook のバージョンによって異なります。 • ユーザーはホストファイルを使用して、Exchange サーバーのホスト名 (単一の完全修飾名) を IP アドレス 127.0.0.1 にマップする必要があります。 • Microsoft Windows 95 または 98 では、このファイルは %Windows%Hosts に格納されています。 • Microsoft Windows NT 4.0 では、このファイルは %WinNT%System32%drivers%etc%Hosts に格納されています。 <p>エントリは次のようになります。</p> <pre>127.0.0.1 exchange exchange.company.com</pre> <p>Exchange サーバーは、それ自体の名前を Outlook クライアントに返します。このマッピングにより、Outlook クライアントは Netlet クライアントを使用して元のサーバーに接続できるようになります。</p>

表 5-3 Netlet ルールの例 (続き)

ルール	URL	アプレットのダウンロード	ローカルポート	宛先ホスト	宛先ポート	説明
FTP	null	false	30021	<i>your-ftp_ server.your-d omain</i>	21	<p>単一の FTP サーバーへの FTP サービスに、制御対象エンドユーザーアカウントを提供できます。これにより、エンドユーザーシステムから単一の場所への安全なリモート FTP 転送が保証されます。ユーザー名を使用しない場合、FTP の URL は匿名の FTP 接続として解釈されます。</p> <p>Netlet FTP ルールのローカルポートとして、ポート 30021 を定義する必要があります。</p> <p>Netlet 接続を使用してダイナミック FTP を使用することはできません。</p>
Netscape 4.7 Mail Client	null	false	30143, 30025	TARGET TARGET	10143 10025	<p>Netscape クライアントでは、ユーザーは次のコマンドを指定する必要があります。</p> <p>IMAP または受信メールについては localhost:30143</p> <p>SMTP または発信メールについては localhost:30025</p>
Graphon	third_ party/ xsession_ start.ht ml	true	10491	TARGET	491	<p>Netlet を通じて Graphon にアクセスするためのルール。 xsession_start.html は Graphon にバンドルされています。</p>
Citrix	third_ party/ citrix_s tart.ht ml	true	1494	TARGET	1494	<p>Netlet を通じて Citrix にアクセスするためのルール。 citrix_start.html は Citrix にバンドルされています。</p>

表 5-3 Netlet ルールの例 (続き)

ルール	URL	アプレットのダウンロード	ローカルポート	宛先ホスト	宛先ポート	説明
Remote Control	third_party/pca_start.html	true	5631 5632	TARGET TARGET	5631 5632	Netlet を通じて Remote Control にアクセスするためのルール。 pca_start.html は Remote Control にバンドルされています。

Netlet ロギングの有効化

ゲートウェイサービスで、Netlet 関連アクティビティのロギングを有効にできます。[284 ページの「Netlet ロギングの有効化」](#)を参照してください。このログファイルは、Access Manager 設定属性の「ロギング」セクションにある「ログの場所」属性で指定されたディレクトリに作成されます。

ログファイル名には、次の命名ルールがあります。

`srapNetlet_gateway-hostname_gateway-profile-name`

Netlet ログには、次の情報が記録されます。

- 開始時間
- ソースアドレス
- ソースポート
- サーバーアドレス
- サーバーポート
- 停止時間
- 状態 (起動または停止)

デバッグロギングの有効化

デバッグ情報の場所は、Portal Server ノードの `AMConfig-instance-name.properties` ファイルに設定されている `com.iplanet.services.debug.directory` 属性の値によって異なります。

たとえば、`com.iplanet.services.debug.directory` 属性に次の値が設定されているとします。

```
/var/opt/SUNWam/debug/
```

この場合、NetFile のデバッグ情報は `/var/opt/SUNWam/debug` ディレクトリの `srapNetFile` ファイルから取得できます。

詳細については、『Access Manager 管理ガイド』を参照してください。

Sun Ray 環境での Netlet の実行

Sun Ray 環境のクライアントマシンでアプレットをダウンロードする必要があるアプリケーションを実行するときは、HTML ファイルを変更する必要があります。次に、必要な変更を加えたファイルの例を示します。

新しい HTML ファイル

```
<!-- @(#)citrix_start.html 2.1      98/08/17 Copyright (c) 1998 i-Planet, Inc., All  
rights reserved.  -->
```

```
<html>
```

```
<script language="JavaScript">
```

```
var KEY_VALUES; // KEY_VALUES['key'] = 'value';
```

```
function retrieveKeyValues() {
```

```
    KEY_VALUES = new Object();
```

```
    var queryString = '' + this.location;
```

```
    queryString = unescape(queryString);
```

```
    queryString = queryString.substring((queryString.indexOf('?')) + 1);
```

```
    if (queryString.length < 1) {
```

```
        return false; } 
```

```
    var keypairs = new Object();
```

```

var numKP = 0;
while (queryString.indexOf('&') > -1) {
    keypairs[numKP] = queryString.substring(0,queryString.indexOf('&'));
    queryString = queryString.substring((queryString.indexOf('&')) + 1);
    numKP++;
}
// クエリ文字列に最後の keypairs[] データとして残されている内容を格納します。
keypairs[numKP++] = queryString;
var keyName;
var keyValue;
for (var i=0; i < numKP; ++i) {
    keyName = keypairs[i].substring(0,keypairs[i].indexOf('='));
    keyValue = keypairs[i].substring((keypairs[i].indexOf('=')) + 1);
    while (keyValue.indexOf('+') > -1) {
        keyValue = keyValue.substring(0,keyValue.indexOf('+')) + ' ' +
keyValue.substring(keyValue.indexOf('+') + 1);
    }
    keyValue = unescape(keyValue);
    // 英数字以外のエスケープを解除します。
    KEY_VALUES[keyName] = keyValue;
}
}
function getClientPort(serverPort) {
    var keyName = "clientPort['" + serverPort + "']";
    return KEY_VALUES[keyName];
}
function generateContent() {
    retrieveKeyValues();
    var newContent =
        "<html>¥n"
        + "<head></head>¥n"

```

```
+ "<body>\n"  
+ "<applet code=\"%com.citrix.JICA.class%\" archive=\"%JICAEngN.jar%\"  
width=800 height=600>\n"  
+ "<param name=\"%cabbase%\" value=\"%JICAEngM.cab%\">\n"  
+ "<param name=\"%address%\" value=\"%localhost%\">\n"  
+ "<param name=ICAPortNumber value="  
+ getClientPort('1494')  
+ ">\n"  
+ "</applet>\n"  
+ "</body>\n"  
+ "</html>\n";  
document.write(newContent);  
}  
</script>  
<body onLoad="generateContent();">  
</body>  
</html>
```

変更前の HTML ファイル

```
<html>  
<body>  
<applet code="com.citrix.JICA.class" archive="JICAEngN.jar" width=800 height=600>  
<param name="cabbase" value="JICAEngM.cab">  
<param name="address" value="localhost">  
<param name=ICAPortNumber value=1494>  
</applet>  
</body></html>
```

Netlet での PDC の使用

この章では、Netlet で PDC を使用できるように、クライアントブラウザの Java™ プラグインを設定する方法について説明します。

注 Netlet での PDC の使用は、JSSE をサポートしているクライアント仮想マシン (VM) だけでサポートされます。

PDC 用の Netlet の設定

▶ Netlet を PDC 用に設定するには

1. 次のいずれかの形式で、ブラウザからクライアント証明書をエクスポートします。
 - PKCS
 - JKS

クライアント証明書をエクスポートしたら、VM が証明書を使用するように、Java プラグインの次の JVM パラメータを設定します。

```
javax.net.ssl.keyStoreType  
javax.net.ssl.keyStorePassword  
javax.net1.ssl.keyStore
```

2. コントロールパネルから Java プラグインを起動します。
3. Java 実行時環境を設定するための「詳細」タブを選択します。
4. 「Java 実行時のパラメータ」を指定します。

例

```
Djavax.net.ssl.keyStoreType=pkcs  
Djavax.net.ssl.keyStorePassword=testing123  
Djavax.net1.ssl.keyStore="C:¥dir¥test.cert"
```

5. 「適用」をクリックします。
6. Java プラグインを閉じ、関連付けられているブラウザを再起動します。

証明書

この章では、証明書の管理、および自己署名証明書または認証局からの証明書をインストールする方法について説明します。

この章で説明する内容は次のとおりです。

- [SSL 証明書の概要](#)
- [証明書ファイル](#)
- [証明書の信頼属性](#)
- [CA の信頼属性](#)
- [certadmin スクリプト](#)
- [自己署名証明書の生成](#)
- [証明書認証局から届いた SSL 証明書のインストール](#)
- [ルート CA 証明書の追加](#)
- [証明書の信頼属性の変更](#)
- [ルート CA 証明書のリスト表示](#)
- [すべての証明書のリスト表示](#)
- [証明書の削除](#)
- [証明書の出力](#)

SSL 証明書の概要

Sun Java™ System Portal Server Secure Remote Access ソフトウェアは、証明書ベースのリモートユーザー認証を提供します。SRA では、通信の安全を確保するために SSL (Secure Sockets Layer) を使用します。SSL プロトコルを使用することで、2つのマシン間の通信がセキュリティ保護されます。

SSL 証明書は、公開鍵と秘密鍵のペアを使用した暗号化と複合化の機能を提供します。

証明書には、次の 2 種類があります。

- 自己署名証明書 (ルート CA 証明書とも呼ばれる)
- 認証局 (CA) が発行する証明書

ゲートウェイのインストール時に、デフォルトでは自己署名証明書が生成およびインストールされます。

証明書は、インストール後にいつでも生成、取得、または交換することができます。

SRA は PDC (Personal Digital Certificates) によるクライアント認証をサポートします。PDC は SSL クライアント認証を通じてユーザーを認証するメカニズムです。SSL クライアント認証を使用して、SSL ハンドシェイクがゲートウェイで終了します。ゲートウェイはユーザーの PDC を抽出し、認証されたサーバーにこれを渡します。このサーバーは、この PDC を使用してユーザーを認証します。認証連鎖における PDC の設定については、[81 ページの「認証連鎖の使用」](#)を参照してください。

SRA には、SSL 証明書を管理するための certadmin というツールが用意されています。[218 ページの「certadmin スクリプト」](#)を参照してください。

証明書ファイル

証明書関連のファイルは `/etc/opt/SUNWps/cert/default/gateway-profile-name` 内にあります。このディレクトリには、デフォルトで 5 つのファイルが格納されています。

表 7-1 は、これらのファイルの説明を示しています。

表 7-1 証明書ファイル

ファイル名	タイプ	説明
cert8.db、 key3.db、 secmod.db	バイナリ	証明書、キー、および暗号化モジュールのデータが含まれます。 certadmin スクリプトを使用して操作できます。 Sun Java System Web Server で使用されるデータベースファイルと同じ形式を持ち、 <i>portal-server-install-root/SUNWwbsvr/alias</i> に格納されます。 必要に応じて、Portal Server ホストとゲートウェイコンポーネントまたはゲートウェイの間でこれらのファイルを共有できます。
.jsspass	非表示テキストファイル	SRA 鍵データベースの暗号化されたパスワードを格納します。
.nickname	非表示テキストファイル	ゲートウェイが使用する必要のあるトークン名と証明書名を <i>token-name:certificate-name</i> の形式で格納します。 デフォルトのトークン (デフォルトの内部ソフトウェア暗号化モジュールのトークン) を使用している場合は、トークン名は省略されます。ほとんどの場合、.nickname ファイルには証明書名だけが格納されます。 管理者はこのファイルの証明書名を変更できます。ゲートウェイでは、指定した証明書が使用されます。

証明書の信頼属性

証明書の信頼属性が示す情報は、次のとおりです。

- 証明書が認証された CA から発行されているかどうか (クライアント証明書またはサーバー証明書の場合)
- 証明書をサーバーまたはクライアント証明書の発行者として信頼できるかどうか (ルート証明書の場合)

各証明書について、「SSL, 電子メール, オブジェクト署名」の順序で表される 3 つの信頼カテゴリがあります。ゲートウェイコンポーネントの場合、最初のカテゴリだけが使用されます。各カテゴリの位置に、信頼属性コードが設定されます (カテゴリにコードが設定されない場合もある)。

カテゴリの属性コードはカンマ (,) で区切られ、属性のセット全体は引用符 (") で囲まれます。たとえば、ゲートウェイのインストール時に生成、インストールされた自己署名証明書には、「u,u,u」が設定されます。これは、証明書がルート CA 証明書ではなくサーバー証明書 (ユーザー証明書) であることを示します。

表 7-2 は、属性値のリストとそれぞれの意味を示しています。

表 7-2 証明書信頼属性

属性	説明
p	有効なピア
P	認証されたピア (p のサブセット)
c	有効な CA
T	クライアント証明書の発行が認証された CA (c のサブセット)
C	サーバー証明書の発行が認証された CA (c のサブセット) (SSL のみ)
u	認証または署名に証明書を使用できる
w	警告を送信 (他の属性とともに使用され、そのコンテキストでの証明書の使用について警告を追加する)

CA の信頼属性

公開されている既知の CA のほとんどは、すでに認証データベースに含まれています。公開 CA の信頼属性の変更については、[228 ページ](#)の「[証明書](#)の信頼属性の変更」を参照してください。

表 7-3 は、代表的な認証局とその信頼属性を示しています。

表 7-3 公開されている認証局

認証局名	信頼属性
Verisign/RSA Secure Server CA	CPp,CPp,CPp
VeriSign Class 4 Primary CA	CPp,CPp,CPp
GTE CyberTrust Root CA	CPp,CPp,CPp
GTE CyberTrust Global Root	CPp,CPp,CPp
GTE CyberTrust Root 5	CPp,CPp,CPp
GTE CyberTrust Japan Root CA	CPp,CPp,CPp
GTE CyberTrust Japan Secure Server CA	CPp,CPp,CPp
Thawte Personal Basic CA	CPp,CPp,CPp
Thawte Personal Premium CA	CPp,CPp,CPp
Thawte Personal Freemail CA	CPp,CPp,CPp
Thawte Server CA	CPp,CPp,CPp
Thawte Premium Server CA	CPp,CPp,CPp
American Express CA	CPp,CPp,CPp
American Express Global CA	CPp,CPp,CPp
Equifax Premium CA	CPp,CPp,CPp
Equifax Secure CA	CPp,CPp,CPp
BelSign Object Publishing CA	CPp,CPp,CPp
BelSign Secure Server CA	CPp,CPp,CPp
TC TrustCenter, Germany, Class 0 CA	CPp,CPp,CPp
TC TrustCenter, Germany, Class 1 CA	CPp,CPp,CPp
TC TrustCenter, Germany, Class 2 CA	CPp,CPp,CPp
TC TrustCenter, Germany, Class 3 CA	CPp,CPp,CPp

表 7-3 公開されている認証局 (続き)

認証局名	信頼属性
TC TrustCenter, Germany, Class 4 CA	CPp,CPp,CPp
ABAEcom (sub., Am. Bankers Assn.) Root CA	CPp,CPp,CPp
Digital Signature Trust Co. Global CA 1	CPp,CPp,CPp
Digital Signature Trust Co. Global CA 3	CPp,CPp,CPp
Digital Signature Trust Co. Global CA 2	CPp,CPp,CPp
Digital Signature Trust Co. Global CA 4	CPp,CPp,CPp
Deutsche Telekom AG Root CA	CPp,CPp,CPp
Verisign Class 1 Public Primary Certification Authority	CPp,CPp,CPp
Verisign Class 2 Public Primary Certification Authority	CPp,CPp,CPp
Verisign Class 3 Public Primary Certification Authority	CPp,CPp,CPp
Verisign Class 1 Public Primary Certification Authority - G2	CPp,CPp,CPp
Verisign Class 2 Public Primary Certification Authority - G2	CPp,CPp,CPp
Verisign Class 3 Public Primary Certification Authority - G2	CPp,CPp,CPp
Verisign Class 4 Public Primary Certification Authority - G2	CPp,CPp,CPp
GlobalSign Root CA	CPp,CPp,CPp
GlobalSign Partners CA	CPp,CPp,CPp
GlobalSign Primary Class 1 CA	CPp,CPp,CPp
GlobalSign Primary Class 2 CA	CPp,CPp,CPp
GlobalSign Primary Class 3 CA	CPp,CPp,CPp
ValiCert Class 1 VA	CPp,CPp,CPp
ValiCert Class 2 VA	CPp,CPp,CPp
ValiCert Class 3 VA	CPp,CPp,CPp
Thawte Universal CA Root	CPp,CPp,CPp
Verisign Class 1 Public Primary Certification Authority - G3	CPp,CPp,CPp
Verisign Class 2 Public Primary Certification Authority - G3	CPp,CPp,CPp
Verisign Class 3 Public Primary Certification Authority - G3	CPp,CPp,CPp
Verisign Class 4 Public Primary Certification Authority - G3	CPp,CPp,CPp
Entrust.net Secure Server CA	CPp,CPp,CPp

表 7-3 公開されている認証局 (続き)

認証局名	信頼属性
Entrust.net Secure Personal CA	CPp,CPp,CPp
Entrust.net Premium 2048 Secure Server CA	CPp,CPp,CPp
ValiCert OCSP Responder	CPp,CPp,CPp
Baltimore CyberTrust Code Signing Root	CPp,CPp,CPp
Baltimore CyberTrust Root	CPp,CPp,CPp
Baltimore CyberTrust Mobile Commerce Root	CPp,CPp,CPp
Equifax Secure Global eBusiness CA	CPp,CPp,CPp
Equifax Secure eBusiness CA 1	CPp,CPp,CPp
Equifax Secure eBusiness CA 2	CPp,CPp,CPp
Visa International Global Root 1	CPp,CPp,CPp
Visa International Global Root 2	CPp,CPp,CPp
Visa International Global Root 3	CPp,CPp,CPp
Visa International Global Root 4	CPp,CPp,CPp
Visa International Global Root 5	CPp,CPp,CPp
beTRUSTed Root CA	CPp,CPp,CPp
Xcert Root CA	CPp,CPp,CPp
Xcert Root CA 1024	CPp,CPp,CPp
Xcert Root CA v1	CPp,CPp,CPp
Xcert Root CA v1 1024	CPp,CPp,CPp
Xcert EZ	CPp,CPp,CPp
CertEngine CA	CPp,CPp,CPp
BankEngine CA	CPp,CPp,CPp
FortEngine CA	CPp,CPp,CPp
MailEngine CA	CPp,CPp,CPp
TraderEngine CA	CPp,CPp,CPp
USPS Root	CPp,CPp,CPp
USPS Production 1	CPp,CPp,CPp
AddTrust Non-Validated Services Root	CPp,CPp,CPp

表 7-3 公開されている認証局 (続き)

認証局名	信頼属性
AddTrust External Root	CPp,CPp,CPp
AddTrust Public Services Root	CPp,CPp,CPp
AddTrust Qualified Certificates Root	CPp,CPp,CPp
Verisign Class 1 Public Primary OCSP Responder	CPp,CPp,CPp
Verisign Class 2 Public Primary OCSP Responder	CPp,CPp,CPp
Verisign Class 3 Public Primary OCSP Responder	CPp,CPp,CPp
Verisign Secure Server OCSP Responder	CPp,CPp,CPp
Verisign Time Stamping Authority CA	CPp,CPp,CPp
Thawte Time Stamping CA	CPp,CPp,CPp
E-Certify CA	CPp,CPp,CPp
E-Certify RA	CPp,CPp,CPp
Entrust.net Global Secure Server CA	CPp,CPp,CPp
Entrust.net Global Secure Personal CA	CPp,CPp,CPp

certadmin スクリプト

certadmin スクリプトを使用して、次のような証明書管理タスクを実行できます。

- [自己署名証明書の生成](#)
- [証明書署名要求 \(CSR\) の生成](#)
- [ルート CA 証明書の追加](#)
- [CA から届いた証明書のインストール](#)
- [証明書の削除](#)
- [証明書の信頼属性の変更](#)
- [ルート CA 証明書のリスト表示](#)
- [すべての証明書のリスト表示](#)
- [証明書の出力](#)

自己署名証明書の生成

各サーバーとゲートウェイの間で SSL 通信を行うには、証明書を生成する必要があります。

▶ インストール後に自己署名証明書を生成するには

1. 証明書を生成するゲートウェイマシンで、**root** として `certadmin` スクリプトを実行します。

```
portal-server-install-root/SUNWps/bin/certadmin -n gateway-profile-name
```

証明書管理メニューが表示されます。

```
1) 自己署名証明書の生成
2) 証明書署名要求 (CSR) の生成
3) ルート CA 証明書の追加
4) 証明書認証局 (CA) から証明書をインストール
5) 証明書の削除
6) 証明書の信頼属性の変更 (PDC 向けなど)
7) ルート CA 証明書のリスト
8) すべての証明書のリスト
9) 証明書の内容の出力
10) 終了
choice: [10] 1
```

2. 証明書管理メニューのオプション **1** を選択します。

既存のデータベースファイルを維持するかどうかを確認するメッセージが表示されます。

3. 組織に固有の情報、トークン名、証明書名を入力します。

注 ワイルドカード証明の場合は、ホストの完全修飾 DNS 名にアスタリスク (*) を含めます。たとえば、完全修飾ホスト名が `abc.sesta.com` の場合、`*.sesta.com` のように指定します。生成される証明書は、`sesta.com` ドメインのすべてのホストで有効になります。

このホストの完全修飾 DNS 名を指定してください [host_name.domain_name]

組織（企業など）の名前を指定してください []

組織単位（部門など）の名前を指定してください []

所在地の都市名を指定してください []

所在地の都道府県を指定してください []

2 桁の国コードを指定してください []

トークン名は、デフォルトの内部（ソフトウェア）暗号化モジュールを使用しない場合（暗号化カードを使用する場合など）にだけ必要です。トークン名は、`modutil -dbdir /etc/opt/SUNWps/cert/default -list` を実行してリスト表示できます。必要がない場合は、次でリターンキーを押します。

トークン名を入力してください []

この証明書の名前を自由に入力してください

証明書の有効期間を入力してください（月単位） [6]
自己署名証明書が生成され、プロンプトに戻ります。

トークン名（デフォルトトークンの場合は指定されない）と証明書名は、`/etc/opt/SUNWps/cert/gateway-profile-name` の `.nickname` ファイルに格納されます。

4. ゲートウェイを再起動して証明書を適用します。

```
gateway-install-root/SUNWps/bin/gateway -n new gateway-profile-name start
```

証明書署名要求 (CSR) の生成

CA に証明書を要求する前に、その CA が要求する情報を含む証明書署名要求 (CSR) を生成する必要があります。

▶ CSR を生成するには

1. root として certadmin スクリプトを実行します。

```
portal-server-install-root/SUNWps/bin/certadmin -n gateway-profile-name
```

証明書管理メニューが表示されます。

```
1) 自己署名証明書の生成
2) 証明書署名要求 (CSR) の生成
3) ルート CA 証明書の追加
4) 証明書認証局 (CA) から証明書をインストール
5) 証明書の削除
6) 証明書の信頼属性の変更 (PDC 向けなど)
7) ルート CA 証明書のリスト
8) すべての証明書のリスト
9) 証明書の内容の出力
10) 終了
choice: [10] 2
```

2. 証明書管理メニューのオプション **2** を選択します。

組織に固有の情報、トークン名、Web マスターの電子メールアドレスと電話番号を要求するプロンプトが表示されます。

ホスト名は、完全修飾 DNS 名で指定する必要があります。

このホストの完全修飾 DNS 名を指定してください [snape.sesta.com]

組織（企業など）の名前を指定してください []

組織単位（部門など）の名前を指定してください []

所在地の都市名を指定してください []

所在地の都道府県を指定してください []

2 桁の国コードを指定してください []

トークン名は、デフォルトの内部（ソフトウェア）暗号化モジュールを使用しない場合（暗号化カードを使用する場合など）にだけ必要です。トークン名は、`modutil -dbdir /etc/opt/SUNWps/cert/default -list` を実行してリスト表示できます。必要がない場合は、次でリターンキーを押します。

トークン名を入力してください []

次に、証明書の生成の対象であるコンピュータの Web マスターへの連絡先情報を入力します。

このサーバーの管理者または Web マスターの電子メールアドレスを指定してください []

このサーバーの管理者または Web マスターの電話番号を指定してください []

3. 要求されるすべての情報を入力します。

注 Web マスターの電子メールアドレスと電話番号を省略することはできません。有効な CSR を取得するには、この情報が必要です。

CSR が生成され、`portal-server-install-root/SUNWps/bin/csr.hostname.datetimestamp` ファイルに格納されます。CSR は画面にも出力されます。CA に証明書を要求するときは、CSR をコピーして直接貼り付けることができます。

ルート CA 証明書の追加

ゲートウェイの証明書データベースに登録されていない CA が署名した証明書をクライアントサイトが提示した場合、SSL ハンドシェイクは失敗します。

これを防ぐには、証明書データベースにルート CA 証明書を追加する必要があります。これにより、ゲートウェイはその CA を認識できるようになります。

ブラウザで CA の Web サイトにアクセスし、その CA のルート証明書を取得します。certadmin スクリプトを使用するときは、ルート CA 証明書のファイル名とパスを指定します。

▶ ルート CA 証明書を追加するには

1. root として certadmin スクリプトを実行します。

```
portal-server-install-root/SUNWps/bin/certadmin -n gateway-profile-name
```

証明書管理メニューが表示されます。

- 1) 自己署名証明書の生成
- 2) 証明書署名要求 (CSR) の生成
- 3) ルート CA 証明書の追加
- 4) 証明書認証局 (CA) から証明書をインストール
- 5) 証明書の削除
- 6) 証明書の信頼属性の変更 (PDC 向けなど)
- 7) ルート CA 証明書のリスト
- 8) すべての証明書のリスト
- 9) 証明書の内容の出力
- 10) 終了

```
choice: [10] 3
```

2. 証明書管理メニューのオプション **3** を選択します。
3. ルート証明書を格納したファイルの名前と証明書名を入力します。
証明書データベースにルート CA 証明書を追加します。

証明書認証局から届いた SSL 証明書のインストール

ゲートウェイのインストール時に、自己署名証明書がデフォルトで作成およびインストールされます。インストール後はいつでも、正式な認証局 (CA) が指定するベンダまたは自社の CA が署名した SSL 証明書をインストールすることができます。

この作業は、次の 3 段階で実行されます。

- [証明書署名要求 \(CSR\) の生成](#)
- [CA への証明書の要求](#)
- [CA から届いた証明書のインストール](#)

CA への証明書の要求

証明書署名要求 (CSR) を生成したら、その CSR を使用して CA に証明書を要求します。

▶ CA に証明書を要求するには

1. 認証局の Web サイトにアクセスし、証明書を要求します。
2. CA が必要とする場合は、CSR を提示します。CA によっては、その他の情報の提供も必要です。

CA から証明書が届きます。これをファイルに保存します。ファイルには、証明書の内容だけでなく、「BEGIN CERTIFICATE」および「END CERTIFICATE」という行も含めます。

次の例には、実際の証明書データは含まれていません。

```
-----BEGIN CERTIFICATE-----
```

証明書の内容

```
-----END CERTIFICATE-----
```

CA から届いた証明書のインストール

certadmin スクリプトを使用して、CA から届いた証明書を /etc/opt/SUNWps/cert/gateway-profile-name 内のローカルデータベースファイルにインストールできます。

▶ CA から届いた証明書をインストールするには

1. root として certadmin スクリプトを実行します。

```
portal-server-install-root/SUNWps/bin/certadmin -n gateway-profile-name
```

証明書管理メニューが表示されます。

- 1) 自己署名証明書の生成
- 2) 証明書署名要求 (CSR) の生成
- 3) ルート CA 証明書の追加
- 4) 証明書認証局 (CA) から証明書をインストール
- 5) 証明書の削除
- 6) 証明書の信頼属性の変更 (PDC 向けなど)
- 7) ルート CA 証明書のリスト
- 8) すべての証明書のリスト

9) 証明書の内容の出力

10) 終了

choice: [10] 4

2. 証明書管理メニューのオプション **4** を選択します。

証明書ファイル名、証明書名、トークン名の入力を求められます。

証明書が含まれているファイルの名前（パスを含む）を指定してください
この証明書に CSR を作成するとき、使用するトークン名を入力してください。 []

3. 要求されるすべての情報を入力します。

証明書が `/etc/opt/SUNWps/cert/gateway-profile-name` にインストールされ、画面はプロンプトに戻ります。

4. ゲートウェイを再起動して証明書を適用します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

証明書の削除

証明書管理スクリプトを使用して、証明書を削除することができます。

▶ 証明書を削除するには

1. `root` として `certadmin` スクリプトを実行します。

```
portal-server-install-root/SUNWps/bin/certadmin -n gateway-profile-name
```

`gateway-profile-name` は、ゲートウェイのインスタンス名です。

証明書管理メニューが表示されます。

- 1) 自己署名証明書の生成
 - 2) 証明書署名要求 (CSR) の生成
 - 3) ルート CA 証明書の追加
 - 4) 証明書認証局 (CA) から証明書をインストール
 - 5) 証明書の削除
 - 6) 証明書の信頼属性の変更 (PDC 向けなど)
 - 7) ルート CA 証明書のリスト
 - 8) すべての証明書のリスト
 - 9) 証明書の内容の出力
 - 10) 終了
- choice: [10] 5

2. 証明書管理メニューのオプション 5 を選択します。
3. 削除する証明書の名前を入力します。

証明書の信頼属性の変更

証明書の信頼属性の変更が必要となる理由の1つに、ゲートウェイでのクライアント認証の使用が挙げられます。クライアント認証には、PDC (Personal Digital Certificate) などがあります。ゲートウェイは、PDC を発行する CA を信頼する必要があり、証明書の信頼属性は、SSL 用に「T」に設定する必要があります。

ゲートウェイが HTTPS サイトとの通信を設定されている場合、ゲートウェイは、HTTPS サイトのサーバー証明書を発行する CA を信頼する必要があり、証明書の信頼属性は SSL 用に「C」に設定する必要があります。

▶ 証明書の信頼属性を変更するには

1. root として certadmin スクリプトを実行します。

```
gateway-install-root/SUNWps/bin/certadmin -n gateway-profile-name
```

gateway-profile-name は、ゲートウェイのインスタンス名です。

証明書管理メニューが表示されます。

- 1) 自己署名証明書の生成
- 2) 証明書署名要求 (CSR) の生成
- 3) ルート CA 証明書の追加
- 4) 証明書認証局 (CA) から証明書をインストール
- 5) 証明書の削除
- 6) 証明書の信頼属性の変更 (PDC 向けなど)
- 7) ルート CA 証明書のリスト
- 8) すべての証明書のリスト

9) 証明書の内容の出力

10) 終了

choice: [10] 6

2. 証明書管理メニューのオプション 6 を選択します。
3. 出力する証明書の名前を入力します。たとえば、Thawte Personal Freemail C などです。

証明書の名前を入力してください
Thawte Personal Freemail CA

4. 証明書の信頼属性を入力します。

証明書に持たせる信頼属性を入力してください [CT, CR, CT]

証明書の信頼属性が変更されます。

ルート CA 証明書のリスト表示

証明書管理スクリプトを使用して、すべての CA 証明書をリスト表示することができます。

▶ **ルート CA 証明書をリスト表示するには**

1. root として certadmin スクリプトを実行します。

```
portal-server-install-root/SUNWps/bin/certadmin -n gateway-profile-name
```

gateway-profile-name は、ゲートウェイのインスタンス名です。
証明書管理メニューが表示されます。

- 1) 自己署名証明書の生成
 - 2) 証明書署名要求 (CSR) の生成
 - 3) ルート CA 証明書の追加
 - 4) 証明書認証局 (CA) から証明書をインストール
 - 5) 証明書の削除
 - 6) 証明書の信頼属性の変更 (PDC 向けなど)
 - 7) ルート CA 証明書のリスト
 - 8) すべての証明書のリスト
 - 9) 証明書の内容の出力
 - 10) 終了
- choice: [10] 7

2. 証明書管理メニューのオプション 7 を選択します。

すべてのルート CA 証明書が表示されます。

すべての証明書のリスト表示

証明書管理スクリプトを使用して、すべての証明書とその信頼属性を表示することができます。

▶ **すべての証明書をリスト表示するには**

1. root として certadmin スクリプトを実行します。

```
portal-server-install-root/SUNWps/bin/certadmin -n gateway-profile-name
```

gateway-profile-name は、ゲートウェイのインスタンス名です。
証明書管理メニューが表示されます。

- 1) 自己署名証明書の生成
 - 2) 証明書署名要求 (CSR) の生成
 - 3) ルート CA 証明書の追加
 - 4) 証明書認証局 (CA) から証明書をインストール
 - 5) 証明書の削除
 - 6) 証明書の信頼属性の変更 (PDC 向けなど)
 - 7) ルート CA 証明書のリスト
 - 8) すべての証明書のリスト
 - 9) 証明書の内容の出力
 - 10) 終了
- choice: [10] 8

2. 証明書管理メニューのオプション **8** を選択します。
すべての CA 証明書が表示されます。

証明書の出力

証明書管理スクリプトを使用して、証明書を出力することができます。

▶ 証明書を出力するには

1. root として certadmin スクリプトを実行します。

```
portal-server-install-root/SUNWps/bin/certadmin -n gateway-profile-name
```

gateway-profile-name は、ゲートウェイのインスタンス名です。
証明書管理メニューが表示されます。

- 1) 自己署名証明書の生成
- 2) 証明書署名要求 (CSR) の生成
- 3) ルート CA 証明書の追加
- 4) 証明書認証局 (CA) から証明書をインストール
- 5) 証明書の削除
- 6) 証明書の信頼属性の変更 (PDC 向けなど)
- 7) ルート CA 証明書のリスト
- 8) すべての証明書のリスト

9) 証明書の内容の出力

10) 終了

choice: [10] 9

2. 証明書管理メニューのオプション 9 を選択します。
3. 出力する証明書の名前を入力します。

URL アクセス制御の設定

この章では、Sun Java™ System Access Manager 管理コンソールから、エンドユーザーのアクセスを許可または拒否する方法について説明します。

注 Sun Java System Portal Server Secure Remote Access (SRA) のすべての属性について簡単に調べるには、Access Manager 管理コンソールの右上に表示される「ドキュメント」をクリックします。

▶ URL アクセス制御を設定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 管理コンソールの「サービス設定」タブを選択します。
3. 「SRA 設定」の下にある「アクセスリスト」の隣の矢印をクリックします。
「アクセスリスト」ページが表示されます。

このページでは、次のタスクを実行できます。

- [拒否される URL リストの設定](#)
- [許可される URL リストの設定](#)
- [シングルサインオンの管理](#)

注 SRA のインストール後、デフォルトではすべてのユーザーがアクセスリストサービスを使用できるようなっていません。このサービスは、インストール時にデフォルトで作成された `amadmin` ユーザーだけが使用できます。その他のユーザーがゲートウェイを通じてデスクトップにアクセスするには、このサービスが必要です。`amadmin` としてログインし、このサービスをすべてのユーザーに割り当てます。

拒否される URL リストの設定

このフィールドでは、エンドユーザーがゲートウェイ経由でアクセスできないようにする URL のリストを指定できます。

ゲートウェイは、許可される URL リストをチェックする前に拒否される URL リストをチェックします。

▶ 拒否される URL リストを設定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の下にある「アクセスリスト」の隣の矢印をクリックします。
「アクセスリスト」ページが表示されます。
4. 「拒否される URL」フィールドに、ゲートウェイ経由でのアクセスを拒否する URL を指定します。入力する URL の形式は次のとおりです。
`http://abc.siroe.com`
5. 「追加」をクリックします。
「拒否される URL リスト」に URL が追加されます。
`http://*.siroe.com` のように、正規表現も使用できます。この場合、`siroe.com` ドメインのすべてのホストへのアクセスが拒否されます。
6. 「保存」をクリックし、変更内容を記録します。

許可される URL リストの設定

エンドユーザーがゲートウェイ経由でアクセスできるすべての URL を指定できます。デフォルトでは、このリストには、すべての URL へのアクセスが許可されることを意味するワイルドカード (*) が入力されています。特定の URL を除くすべての URL へのアクセスを許可する場合は、アクセスを制限する URL を「拒否される URL」リストに追加します。同様に、特定の URL に対してだけアクセスを許可する場合は、「拒否される URL」フィールドを空白にし、「許可される URL」フィールドに適切な URL を指定します。

ゲートウェイは、許可される URL をチェックする前に拒否される URL をチェックします。

▶ 許可される URL リストを設定するには

1. Access Manager 管理コンソールに管理者としてログインします。

2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の下にある「アクセスリスト」の隣の矢印をクリックします。
「アクセスリスト」ページが表示されます。
4. 「許可される URL」フィールドに、ゲートウェイ経由でのアクセスを許可する URL を指定します。入力する URL の形式は次のとおりです。
`http://abc.siroe.com`
5. 「追加」をクリックします。
「許可される URL」に URL が追加されます。

注 デフォルトでは、「許可される URL」フィールドには、すべての URL へのアクセスが許可されることを意味するワイルドカード (*) が入力されています。

6. 「保存」をクリックし、変更内容を記録します。

シングルサインオンの管理

SRA ソフトウェアのアクセスリストサービスを使用して、各種ホストのシングルサインオン (SSO) 機能を制御できます。シングルサインオン機能を有効にするには、ゲートウェイサービスで「HTTP 基本認証を有効」オプションが有効になっている必要があります。241 ページの「[HTTP 接続と HTTPS 接続の有効化](#)」を参照してください。

アクセスリストサービスを使用して、特定ホストのシングルサインオンを無効にすることができます。つまり、セッションごとにシングルサインオンを有効にしている場合を除き、HTTP 基本認証を必要とするホストに接続するエンドユーザーは、毎回、認証が必要となります。

特定ホストのシングルサインオンを無効にしている場合でも、エンドユーザーは Portal Server の単一セッション内であれば、そのホストに何度でも接続できます。たとえば、`abc.sesta.com` へのシングルサインオンを無効にすると仮定します。ユーザーがこのサイトに最初に接続するときは、認証が必要です。ユーザーが他のページを参照してからこのページに戻った場合、同じ Portal Server セッション内のページであれば、認証は必要ありません。

ユーザーは、制限付き管理コンソールでこれらの属性を設定できます。

▶ **ホストのシングルサインオンを無効にするには**

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の下にある「アクセスリスト」の隣の矢印をクリックします。
「アクセスリスト」ページが表示されます。
4. 「シングルサインオンを無効にするホスト」フィールドに、SSO を無効にするホストを指定します。
ホスト名は abc.siroe.com の形式で指定します。
5. 「追加」をクリックします。
ホスト名がリストに追加されます。
6. 「保存」をクリックし、変更内容を記録します。

▶ **セッションごとのシングルサインオンを有効にするには**

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の下にある「アクセスリスト」の隣の矢印をクリックします。
「アクセスリスト」ページが表示されます。
4. 「セッションごとのシングルサインオンを有効」チェックボックスにチェックマークを付け、シングルサインオンセッションを有効化します。
5. 「保存」をクリックし、変更内容を記録します。

▶ **認証レベルを指定するには**

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の下にある「アクセスリスト」の隣の矢印をクリックします。
「アクセスリスト」ページが表示されます。
4. 「許可される認証レベル」フィールドまでスクロールします。
5. 許可される認証を入力します。すべてのレベルを許可するときは、アスタリスク (*) を入力します。
6. 「保存」をクリックし、変更内容を記録します。

ゲートウェイの設定

この章では、Sun Java™ System Access Manager 管理コンソールからゲートウェイの属性を設定する方法について説明します。

注 Sun Java System Portal Server Secure Remote Access (SRA) のすべての属性について簡単に調べるには、Access Manager 管理コンソールの右上に表示される「ヘルプ」をクリックし、「Secure Remote Access 管理ヘルプ」をクリックします。

ゲートウェイのインスタンスの作成方法については、[51 ページ](#)の「ゲートウェイのインスタンスの作成」を参照してください。

ゲートウェイプロファイルの作成方法については、[40 ページ](#)の「ゲートウェイプロファイルの作成」を参照してください。

ゲートウェイプロファイルを作成したら、ゲートウェイの属性を設定する必要があります。

▶ **ゲートウェイの属性を設定するには**

1. Access Manager 管理コンソールに管理者としてログインします。
2. 管理コンソールの「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
このページで、適切なタブをクリックします。
 - [コアタブ](#)
 - [プロキシタブ](#)

- セキュリティタブ
- リライタタブ
- ログイングタブ

次にこれらのタブと、各タブで設定できる属性について説明します。

コアタブ

ゲートウェイサービスの「コア」タブでは、次のタスクを実行できます。

- HTTP 接続と HTTPS 接続の有効化
- リライタプロキシの有効化とリストの作成
- Netlet の有効化
- Netlet プロキシの有効化とリストの作成
- プロキシレットの有効化
- Cookie 管理の有効化
- HTTP 基本認証の有効化
- 持続 HTTP 接続の有効化
- 持続接続 1 つあたりの最大要求数の指定
- 持続ソケット接続のタイムアウトの指定
- 回復時間に必要な正常なタイムアウトの指定
- Cookie を転送する URL のリストの作成
- 最大接続キューの指定
- ゲートウェイタイムアウトの指定
- 最大スレッドプールサイズの指定
- キャッシュされたソケットのタイムアウトの指定
- Portal Server のリストの作成
- サーバーの再試行間隔の指定
- 外部サーバー Cookie の格納の有効化
- URL からのセッションを取得するには
- 安全な Cookie としてマークする

HTTP 接続と HTTPS 接続の有効化

インストール時にゲートウェイを HTTPS モードで実行するように選択している場合は、インストール後、ゲートウェイは HTTPS モードで実行されます。HTTPS モードの場合、ゲートウェイはブラウザからの SSL 接続を許可し、非 SSL 接続を拒否します。

ただし、ゲートウェイを HTTP モードで実行するように設定することもできます。HTTPS モードで発生する、SSL セッションの管理、SSL トラフィックの暗号化と復号化に伴うオーバーヘッドが取り除かれるため、ゲートウェイのパフォーマンスが向上します。

▶ HTTP モードまたは HTTPS モードで実行するようにゲートウェイを設定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 管理コンソールの「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「コア」タブで次の操作を行います。
 - 「HTTP 接続を有効」、「HTTPS 接続を有効」、または必要に応じて両方のチェックボックスにチェックマークを付けます。
 - 「HTTPS ポート」フィールドに適切な HTTPS ポートを指定します。
 - 「HTTP ポート」フィールドに適切な HTTP ポートを指定します。
6. 「保存」をクリックし、変更内容を記録します。
7. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

リライタプロキシの有効化とリストの作成

リライタプロキシを使用して、ゲートウェイとイントラネットコンピュータの間の HTTP トラフィックをセキュリティ保護することができます。リライタプロキシを指定しない場合、いずれかのイントラネットコンピュータにアクセスしようとすると、ゲートウェイコンポーネントによりイントラネットコンピュータに直接つながります。

リライタプロキシは、インストール後に自動的に起動されません。次の手順を実行して、リライタプロキシを有効にする必要があります。

▶ リライタプロキシを有効化し、リライタプロキシリストを作成するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。

注 リライタプロキシとゲートウェイが、同じゲートウェイプロファイルを使用していることを確認してください。

「ゲートウェイプロファイルを編集」ページが表示されます。

5. 「コア」タブをクリックします。
6. 「リライタプロキシを有効」チェックボックスにチェックマークを付けて、リライタプロキシを有効にします。
7. 「リライタプロキシのリスト」編集ボックスに、hostname:port という形式で適切なホスト名とポート番号を入力します。

ヒント 目的のポートが使用可能で未使用であることを確認するには、コマンド行で次のコマンドを実行します。

```
netstat -a | grep port-number | wc -l
```

port-number は、目的のポート番号です。

8. 「追加」をクリックします。
9. 「保存」をクリックし、変更内容を記録します。

10. サーバーで `portal-server-install-root/SUNWps/bin/certadmin` を実行し、リライタプロキシの証明書を作成します。

この手順が必要になるのは、リライタプロキシのインストール時に証明書の作成を選択していない場合です。

11. リライタプロキシがインストールされているマシンに `root` としてログインし、リライタプロキシを起動します。

```
rewriter-proxy-install-root/SUNWps/bin/rwproxyd -n gateway-profile-name start
```

12. ゲートウェイがインストールされているマシンに `root` としてログインし、ゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

Netlet の有効化

Netlet を使用することで、インターネットなどのセキュリティの弱いネットワークで一般的な TCP/IP サービスを安全に実行できます。TCP/IP アプリケーション (Telnet や SMTP など)、HTTP アプリケーション、同じポートを使用するすべてのアプリケーションを実行できます。

Netlet を有効にした場合は、ゲートウェイは着信トラフィックが Netlet トラフィックであるか、または Portal Server トラフィックであるかを判断する必要があります。Netlet を無効にした場合は、ゲートウェイはすべての着信トラフィックが HTTP トラフィックと HTTPS トラフィックのいずれかであると仮定するため、オーバーヘッドが低減します。Netlet は、Portal Server でアプリケーションをまったく使用しないことが確実な場合にだけ無効にしてください。

► Netlet を有効にするには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「コア」タブをクリックします。
6. 「Netlet を有効」チェックボックスにチェックマークを付けます。デフォルトでは、このチェックボックスは選択されています。チェックマークを外すと、Netlet は無効になります。

7. 「Netlet プロキシを有効」チェックボックスにチェックマークを付けて、Netlet プロキシを有効にします。
8. 「Netlet プロキシホスト」編集ボックスに、hostname:port という形式で適切なホスト名とポート番号を入力します。
9. 「保存」をクリックし、変更内容を記録します。
10. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

Netlet プロキシの有効化とリストの作成

Netlet プロキシは、ゲートウェイを経由してイントラネット内の Netlet プロキシまでクライアントからの安全なトンネルを拡張することで、ゲートウェイとイントラネットの間の Netlet トラフィックの安全性を補強します。

Netlet プロキシを有効にすると、Netlet パケットが Netlet プロキシにより解読され、送信先サーバーに送られます。これにより、ファイアウォール内で開くポート数を減らすことができます。

▶ Netlet プロキシを有効化し、Netlet プロキシリストを作成するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の右矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「Netlet プロキシを有効」チェックボックスにチェックマークを付けて、Netlet プロキシを有効にします。
6. 「Netlet プロキシホスト」フィールドに、hostname:port という形式で適切な Netlet プロキシホストの名前とポート番号を入力します。

ヒント 目的のポートが使用可能で未使用であることを確認するには、コマンド行で次のコマンドを実行します。

```
netstat -a | grep port-number | wc -l
```

port-number は、目的のポート番号です。

7. 「追加」をクリックします。
8. 「保存」をクリックし、変更内容を記録します。
9. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

プロキシレットの有効化

▶ プロキシレットを有効にするには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の右矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「コア」タブをクリックします。
6. 「プロキシレットを有効」チェックボックスにチェックマークを付けます。
7. 「プロキシ」タブをクリックします。「ドメインとサブドメインのプロキシ」フィールドまでスクロールして、ゲートウェイにリダイレクトする URL のドメインを入力します。
8. 「保存」をクリックします。

Cookie 管理の有効化

多くの Web サイトは、ユーザーセッションの追跡と管理に Cookie を使用しています。HTTP ヘッダに Cookie が設定されている Web サイトにゲートウェイが要求をルーティングする場合、ゲートウェイは次の方法でそれらの Cookie を破棄するか、またはそのまま通過させます。

- ゲートウェイサービスで「Cookie 管理を有効」属性が選択されていない場合、Cookie はリライトされません。このため、ブラウザとイントラネットホストの間で Cookie が伝達されないことがあります。
- 「Cookie 管理を有効」属性が選択されている場合、ゲートウェイは Cookie をリライトします。このリライトによって、ブラウザと目的のイントラネットホストの間で Cookie が正しく伝達されるようになります。

この設定は、Portal Server が Portal Server ユーザーセッションの追跡に使用する Cookie には適用されません。この設定は、「ユーザーセッション Cookie を転送する URL」オプションの設定によって制御されます。250 ページの「[Cookie を転送する URL のリストの作成](#)」を参照してください。

この設定は、ユーザーがアクセスを許可されたすべての Web サイトに適用されます (つまり、一部のサイトの Cookie を破棄し、別のサイトの Cookie を保持することはできない)。

注 Cookie を使用しないゲートウェイであっても、「Cookie ドメイン」リストから URL を削除しないでください。「Cookie ドメイン」リストについては、『Access Manager 管理ガイド』を参照してください。

▶ Cookie の管理を有効にするには

1. Access Manager 管理コンソールに管理者としてログインします。
1. 「サービス設定」タブを選択します。
2. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
3. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
4. 「コア」タブをクリックします。
5. 「Cookie 管理を有効」チェックボックスにチェックマークを付けて、Cookie 管理を有効化します。
6. 「保存」をクリックし、変更内容を記録します。

7. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

HTTP 基本認証の有効化

ゲートウェイサービスには HTTP 基本認証を設定できます。

Web サイトは、サイトを閲覧する前にユーザー名とパスワードの入力を要求する HTTP 基本認証で保護することができます (HTTP 応答コードは 401、WWW 認証は BASIC)。Portal Server はユーザー名とパスワードを保存するため、ユーザーは BASIC で保護された Web サイトに再びアクセスするときに証明情報を再入力する必要はありません。これらの証明情報は、ディレクトリサーバー上のユーザープロファイルに保存されます。

BASIC で保護されたサイトをユーザーが訪問できるかどうかは、この設定によって決定するわけではありませんが、ユーザーが入力する証明情報がユーザーのプロファイルに確実に保存されます。

この設定は、ユーザーがアクセスを許可されたすべての Web サイトに適用されます (つまり、一部のサイトについて HTTP 基本認証のキャッシングを有効にし、別のサイトについて無効にするということとはできない)。

注	BASIC 認証ではなく、Microsoft Windows NT challenge/response (HTTP 応答コード 401、WWW 認証は NTLM) で保護された Microsoft の IIS (Internet Information Server) が提供する URL のブラウズはサポートされません。
----------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------

また、管理コンソールのアクセスリストサービスを使用して、シングルサインオンを有効にすることができます。シングルサインオンの有効化については、[237 ページの「シングルサインオンの管理」](#)を参照してください。

▶ HTTP 基本認証を有効にするには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「コア」タブをクリックします。

6. 「HTTP 基本認証を有効」チェックボックスにチェックマークを付けて、HTTP 基本認証を有効にします。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

持続 HTTP 接続の有効化

ゲートウェイで HTTP の持続接続を有効にし、Web ページの (イメージやスタイルシートなどの) すべてのオブジェクトにソケットが開かれないように設定することができます。

▶ 持続 HTTP 接続を有効にするには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「コア」タブをクリックします。
6. 「持続 HTTP 接続を有効」チェックボックスにチェックマークを付けて、持続的な HTTP 接続を有効にします。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

持続接続 1 つあたりの最大要求数の指定

▶ 持続接続 1 つあたりの最大要求数を指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「コア」タブをクリックします。
6. 「持続接続ごとの最大要求数」フィールドまでスクロールし、適切な要求数を入力します。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

持続ソケット接続のタイムアウトの指定

▶ 持続ソケット接続のタイムアウトを指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「コア」タブをクリックします。
6. 「持続ソケット接続のタイムアウト」フィールドまでスクロールし、適切なタイムアウト時間を秒単位で指定します。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

回復時間に必要な正常なタイムアウトの指定

回復時間に必要な正常なタイムアウトとは次の合計です。

- ブラウザから送信された要求がゲートウェイに到達するまでの時間
- ゲートウェイが応答を送信してから、ブラウザが実際に受信するまでの時間

これは、ネットワークの状況やクライアントの接続スピードといった要因に影響を受けます。

▶ 回復時間に必要な正常なタイムアウトを指定するには

これはクライアント (ブラウザ) とゲートウェイの間でのネットワークトラフィックの往復時間です。

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「コア」タブをクリックします。
6. 「回復時間に必要な正常なタイムアウト」フィールドに、必要な猶予期間を秒単位で指定します。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

Cookie を転送する URL のリストの作成

Portal Server は、ユーザーセッションの追跡に Cookie を使用します。ゲートウェイがサーバーに HTTP 要求を送信すると (ユーザーのデスクトップページを生成するためにデスクトップサブレットが呼び出される場合など)、この Cookie はサーバーに転送されます。サーバー上のアプリケーションはこの Cookie を使用して、ユーザーの検証と特定を行います。

Portal Server の Cookie は、サーバー以外のマシンに送信された HTTP 要求には転送されませんが、それらのマシンの URL が「ユーザーセッション Cookie を転送する URL」リストに指定されている場合は転送されます。したがってこのリストに URL を追加すると、サーブレットと CGI が Portal Server の Cookie を受け取り、API を使用してユーザーを特定することができます。

URL は後続の暗黙的なワイルドカードを使って照合されます。たとえば、リストのデフォルトエントリを次のように指定した場合、

```
http://server:8080
```

http://server:8080 から始まるすべての URL に Cookie が転送されます。

次のように指定するとします。

```
http://newmachine.eng.siroe.com/subdir
```

この場合、この文字列から始まるすべての URL に、Cookie が転送されます。

たとえば、「http://newmachine.eng/subdir」で始まるすべての URL には Cookie は転送されません。これはこの文字列が転送リスト内の文字列と完全に一致する文字列から始まっていないためです。このようなマシン名の変形で始まる URL に Cookie を転送するには、転送リストにエントリを追加する必要があります。

同様に、リストに適切なエントリが追加されている場合を除き、「https://newmachine.eng.siroe.com/subdir」から始まる URL には Cookie は転送されません。

▶ Cookie を転送する URL を追加するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「コア」タブをクリックします。
6. 「ユーザーセッション Cookie を転送する URL」編集ボックスまでスクロールし、適切な URL を入力します。
7. 「追加」をクリックすると、「ユーザーセッション Cookie を転送する URL」リストにこのエントリが追加されます。
8. 「保存」をクリックし、変更内容を記録します。
9. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

最大接続キューの指定

ゲートウェイが受け付ける最大同時接続数を指定できます。この数を超える接続の試行は、ゲートウェイに受け付けられません。

▶ 最大接続キューを指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「コア」タブをクリックします。
6. 「最大接続キュー」フィールドまでスクロールし、適切な接続数を指定します。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

ゲートウェイタイムアウトの指定

ゲートウェイがブラウザとの接続をタイムアウトするまでの時間を、秒単位で指定できます。

▶ ゲートウェイタイムアウトを指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「コア」タブをクリックします。

6. 「ゲートウェイタイムアウト」フィールドまでスクロールし、タイムアウトまでの間隔を秒単位で指定します。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

最大スレッドプールサイズの指定

ゲートウェイスレッドプールで事前に作成できる最大スレッド数を指定できます。

▶ 最大スレッドプールサイズを指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「コア」タブをクリックします。
6. 「最大スレッドプールサイズ」フィールドまでスクロールし、適切なスレッド数を指定します。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

キャッシュされたソケットのタイムアウトの指定

ゲートウェイが Portal Server との接続をタイムアウトするまでの時間を、秒単位で指定できます。

▶ キャッシュされたソケットのタイムアウトを指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「コア」タブをクリックします。
6. 「キャッシュされたソケットのタイムアウト」フィールドまでスクロールし、タイムアウトまでの時間を秒単位で指定します。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

Portal Server のリストの作成

ゲートウェイが要求に応答するように、複数の Portal Server を設定できます。ゲートウェイのインストール時に、ゲートウェイの稼動に必要な Portal Server を指定することがあります。この Portal Server は、デフォルトでは「Portal Server」フィールドに表示されます。その他の Portal Server を `http://portal-server-name:port number` の形式でリストに追加することができます。ゲートウェイは要求を処理するために、リスト内の各 Portal Server に順次アクセスを試みます。

▶ Portal Server を指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。

「ゲートウェイプロファイルを編集」ページが表示されます。

5. 「コア」タブをクリックします。
6. 「Portal Server」フィールドまでスクロールし、Portal Server 名を入力します。
Portal Server を `http://portal-server-name:port-number` の形式で編集フィールドに指定し、「追加」をクリックします。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

サーバーの再試行間隔の指定

この属性により、Portal Server、リライタプロキシ、または Netlet プロキシがクラッシュしたり、停止したりして使用不能になった場合に、これらの再起動を要求するまでの間隔を指定します。

▶ サーバーの再試行間隔を指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「コア」タブをクリックします。
6. 「サーバーの再試行間隔」フィールドまでスクロールし、適切な秒数を指定します。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

外部サーバー Cookie の格納の有効化

「外部サーバーの Cookie を格納」オプションを有効にすると、ゲートウェイはサードパーティ製アプリケーション、またはゲートウェイ経由でアクセスするサーバーからの Cookie を格納、管理します。アプリケーションまたはサーバーが Cookie を使用しないデバイスにサービスを提供できない場合、あるいは、旧式であるため Cookie がないと状態管理ができないという場合でも、Gateway はアプリケーションまたはサーバーに認識されることなく Cookie を使用しないデバイスにサービスを提供します。Cookie を使用しないデバイスとクライアント検出については、『Access Manager Customization and API Guide』を参照してください。

▶ 外部サーバー Cookie を格納するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「コア」タブをクリックします。
6. 「外部サーバーの Cookie を格納」チェックボックスにチェックマークを付けて、外部サーバー Cookie の格納を有効にします。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

URL からのセッションを取得するには

「URL からセッションを取得」オプションを有効にすると、Cookie をサポートするかどうかに関係なく、セッション情報が URL の一部としてコード化されます。つまりゲートウェイは、クライアントのブラウザから送信されるセッション Cookie の代わりに URL に含まれるセッション情報を使用して検証を行います。

▶ URL からのセッションを取得するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。

3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「コア」タブをクリックします。
6. 「URL からセッションを取得」チェックボックスにチェックマークを付けて、URL からのセッションの取得を有効にします。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

安全な Cookie としてマークする

安全な Cookie としてマーク付けしておけば、ブラウザはセキュリティを補強した Cookie として処理します。セキュリティの実装には、ブラウザを使用します。このためには、「Cookie 管理を有効」属性を有効化しておく必要があります。

▶ 安全な Cookie としてマークするには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「コア」タブをクリックします。
6. 「安全な Cookie としてマークする」チェックボックスにチェックマークを付けて、安全な Cookie としてマークを付けます。
「Cookie 管理を有効」属性が有効になっていることを確認します。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

プロキシタブ

ゲートウェイサービスの「プロキシ」タブでは、次のタスクを実行できます。

- Web プロキシ使用の有効化
- Web プロキシを使用する URL のリストの作成
- Web プロキシを使用しない URL のリストの作成
- ドメインとサブドメインのプロキシのリストの作成
- プロキシのパスワードリストの作成
- 自動プロキシ設定サポートの有効化
- 自動プロキシ設定ファイルの場所の指定
- Web プロキシを通じた Netlet トンネリングの有効化

Web プロキシ使用の有効化

▶ Web プロキシの使用を有効にするには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「プロキシ」タブをクリックします。
6. 「プロキシを使用する」チェックボックスにチェックマークを付けて、Web プロキシの使用を有効にします。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

Web プロキシを使用する URL のリストの作成

「プロキシを使用する」オプションを無効にしている場合でも、ゲートウェイが「ドメインとサブドメインのプロキシ」リストの Web プロキシだけを使用して、特定の URL に接続するように指定できます。これらの URL は、「Web プロキシを使用する URL」フィールドに指定する必要があります。この値がプロキシの使用に与える影響についての詳細は、60 ページの「[Access Manager へアクセスするプロキシの指定](#)」を参照してください。

▶ Web プロキシを使用する URL を指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「プロキシ」タブをクリックします。
6. 「Web プロキシを使用する URL」編集ボックスに、`http://host.name.subdomain.com` という形式で適切な URL を入力します。「追加」をクリックします。
「Web プロキシを使用する URL」リストに URL が追加されます。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

Web プロキシを使用しない URL のリストの作成

「Web プロキシを使用しない URL」リストに指定されている URL に対しては、ゲートウェイは直接接続を試みます。これらの URL への接続には Web プロキシは使用されません。

▶ Web プロキシを使用しない URL を指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「プロキシ」タブをクリックします。
6. 「Web プロキシを使用しない URL」編集ボックスに適切な URL を入力し、「追加」をクリックします。
「Web プロキシを使用しない URL」リストに URL が追加されます。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。
`gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start`

ドメインとサブドメインのプロキシのリストの作成

▶ ドメインとサブドメインのプロキシを指定するには

さまざまなホストにプロキシ情報を適用する方法については、[60 ページの「Access Manager へアクセスするプロキシの指定」](#)を参照してください。

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の右矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。

「ゲートウェイプロファイルを編集」ページが表示されます。

- 「プロキシ」タブをクリックします。
- 「ドメインとサブドメインのプロキシ」編集ボックスまでスクロールし、必要な情報を入力します。「追加」をクリックします。

エントリが「ドメインとサブドメインのプロキシ」リストボックスに追加されます。

プロキシ情報は次の形式で入力します。

```
domainname proxy1:port1|subdomain1 proxy2:port2|subdomain2
proxy3:port3|* proxy4:port4
```

*は特別に指定する以外のすべてのドメインとサブドメインに対して、*の後に定義されるプロキシが適用されなければならないことを示します。

プロキシにポートを指定しない場合、デフォルトのポート 8080 が使用されます。

- 「保存」をクリックし、変更内容を記録します。
- 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

プロキシのパスワードリストの作成

プロキシサーバーが一部またはすべてのサイトへのアクセスに認証を要求する場合、指定されたプロキシサーバーでゲートウェイが認証されるために必要な、ユーザー名とパスワードを指定する必要があります。

▶ プロキシパスワードを指定するには

- Access Manager 管理コンソールに管理者としてログインします。
- 「サービス設定」タブを選択します。
- 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。

「ゲートウェイ」ページが表示されます。

- 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
- 「プロキシ」タブをクリックします。
- 「プロキシパスワードのリスト」フィールドまでスクロールし、各プロキシサーバーの情報を入力して「追加」をクリックします。

プロキシ情報は次の形式で入力します。

```
proxyserver|username|password
```

proxyserver は、「ドメインとサブドメインのプロキシ」リストに定義したプロキシサーバーです。

7. 認証を必要とするすべてのプロキシについて、手順 6 を繰り返します。
8. 「保存」をクリックし、変更内容を記録します。
9. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

自動プロキシ設定サポートの有効化

自動プロキシ設定を有効にするオプションを選択すると、「ドメインとサブドメインのプロキシ」フィールドに指定した情報が無視されます。ゲートウェイは、イントラネット設定にだけプロキシ自動設定 (PAC) ファイルを使用します。PAC ファイルについては、[67 ページの「自動プロキシ設定の使用」](#)を参照してください。

▶ 自動プロキシ設定サポートを有効にするには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「プロキシ」タブをクリックします。
6. 「自動プロキシ設定サポートを有効」チェックボックスにチェックマークを付けて、PAC サポートを有効にします。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

自動プロキシ設定ファイルの場所の指定

▶ 自動プロキシ設定ファイルの場所を指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「プロキシ」タブをクリックします。
6. 「自動プロキシ設定ファイルの位置」フィールドまでスクロールし、PAC ファイルの名前と場所を指定します。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

Web プロキシを通じた Netlet トンネリングの有効化

▶ Web プロキシを通じての Netlet トンネリングを有効にするには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「プロキシ」タブをクリックします。
6. 「Web プロキシ経由の Netlet トンネリングを有効」チェックボックスにチェックマークを付けて、トンネル化を有効にします。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

セキュリティタブ

ゲートウェイサービスの「セキュリティ」タブでは、次のタスクを実行できます。

- 非認証 URL のリストの作成
- 証明書が有効なゲートウェイホストのリストの作成
- 40 ビット暗号化接続の許可
- SSL Version 2.0 の有効化
- SSL 符号化選択の有効化
- SSL Version 3.0 の有効化
- Null 暗号化方式の有効化
- 信頼されている SSL ドメインのリストの作成
- PDC (Personal Digital Certificate) 認証の設定
- 安全な Cookie としてマークする
- HTTP 接続と HTTPS 接続の有効化

非認証 URL のリストの作成

一部の URL で認証を不要にするように指定できます。通常は、イメージを含むディレクトリおよびフォルダが該当します。

▶ 非認証 URL パスを指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「非認証 URL」フィールドまでスクロールし、folder/subfolder の形式で適切なフォルダパスを入力します。

URL が、/images など完全修飾名ではない場合、ポータル URL として処理されます。

非ポータル URL を追加するには、URL を完全修飾名にしてください。

6. 「追加」をクリックすると、「非認証 URL」リストにこのエントリが追加されます。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

証明書が有効なゲートウェイホストのリストの作成

- ▶ 証明書が有効なゲートウェイホストのリストにゲートウェイを追加するには
 1. Access Manager 管理コンソールに管理者としてログインします。
 2. 「サービス設定」タブを選択します。

左の区画にすべてのサービスが表示されます。
 3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。

右の区画に「ゲートウェイ」ページが表示されます。
 4. 証明書ベースの認証を有効にするゲートウェイプロファイルを選択します。
 5. 「セキュリティ」タブをクリックします。
 6. 「証明書が有効なゲートウェイホスト」リストにゲートウェイ名が追加されます。

host1.sesta.com の形式でゲートウェイを追加します。
 7. 「追加」をクリックします。

40 ビット暗号化接続の許可

このオプションは、40 ビットの (弱い) SSL (Secure Sockets Layer) 接続を許可する場合に選択します。このオプションを選択していない場合、128 ビット接続だけがサポートされます。

このオプションを無効にするときは、ブラウザが必要な接続タイプをサポートするように設定されていることを確認する必要があります。

-
- 注** Netscape Navigator 4.7x の場合は、次の処理が必要です。
- 「Communicator」メニューの「ツール」の「セキュリティ情報」を選択します。
 - 左の区画で「Navigator」リンクをクリックします。
 - 「詳細セキュリティ (SSL) 設定」の「SSL v2 の設定」または「SSL v3 の設定」をクリックします。
 - 適切な暗号化方式を有効にします。
-

▶ 40 ビット暗号化接続を許可するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「セキュリティ」タブをクリックします。
6. 「40 ビット暗号化を許可」チェックボックスにチェックマークを付けて、40 ビットブラウザ接続を有効にします。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

SSL Version 2.0 の有効化

SSL Version 2.0 を有効または無効にできます。SSL 2.0 を無効化すると、古い SSL 2.0 しかサポートしないブラウザは SRA に対して認証ができません。これにより、セキュリティのレベルが格段に向上します。

▶ SSL Version 2.0 を有効にするには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「セキュリティ」タブをクリックします。
6. 「SSL バージョン 2.0 を有効」チェックボックスにチェックマークを付けて、バージョン 2.0 を有効にします。
デフォルトでは、このオプションは有効に設定されています。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

SSL 符号化選択の有効化

SRA は、数多くの標準暗号化方式をサポートしています。パッケージ内のすべての暗号化方式をサポートするか、必要な暗号化方式を個別に選択するかを選択することができます。ゲートウェイインスタンスごとに、個別に SSL 暗号化方式を選択できます。選択した暗号化方式のいずれかがクライアントサイトに存在していれば、SSL ハンドシェイクは正常に行われます。

▶ 暗号化方式の個別選択を有効にするには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。

4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「SSL 暗号化方式の選択を有効」チェックボックスまでスクロールし、このオプションを選択します。
このオプションを有効にすると、SSL2、SSL3、および TLS の各暗号化方式から、適切な暗号化方式を選択できます。
6. 「保存」をクリックし、変更内容を記録します。
7. 端末ウィンドウからゲートウェイを再起動します。
`gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start`

SSL Version 3.0 の有効化

SSL Version 3.0 を有効または無効にできます。SSL 3.0 を無効化すると、SSL 3.0 しかサポートしないブラウザは SRA ソフトウェアに対して認証ができません。これにより、セキュリティのレベルが格段に向上します。

▶ SSL Version 3.0 を有効にするには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「SSL バージョン 3.0 を有効」チェックボックスにチェックマークを付けて、バージョン 3.0 を有効にします。
6. 「保存」をクリックし、変更内容を記録します。
7. 端末ウィンドウからゲートウェイを再起動します。
`gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start`

Null 暗号化方式の有効化

▶ Null 暗号化方式を有効にするには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「Null 暗号化方式を有効」チェックボックスにチェックマークを付けて、Null 暗号化方式を有効にします。
6. 「保存」をクリックし、変更内容を記録します。
7. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

信頼されている SSL ドメインのリストの作成

▶ 信頼されている SSL ドメインのリストを作成するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「セキュリティ」タブをクリックします。
6. 「信頼できる SSL ドメイン」フィールドまでスクロールし、ドメイン名を入力して「追加」をクリックします。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

PDC (Personal Digital Certificate) 認証の設定

PDC は認証局 (CA) が発行し、CA の非公開鍵で署名されます。CA は証明書を発行する前に要求本文の ID を検証します。この場合 PDC が存在すると、強力な認証メカニズムとして機能します。

PDC には所有者の公開鍵、所有者名、有効期限、デジタル証明書を発行した認証局の名前、シリアル番号、その他の情報が収められています。

Portal Server での認証には、PDC とスマートカードや Java カードなどのコード化されたデバイスを使用できます。コード化されたデバイスは、カードに保存された PDC と電子的に同等のものを搬送します。ユーザーがこれらのメカニズムのいずれかを使用してログインすると、ログイン画面も認証画面も表示されません。

PDC 認証プロセスには、いくつかの手順が伴います。

1. ブラウザから、<https://my.sesta.com> のような接続要求を入力します。

この要求への応答は、my.sesta.com までのゲートウェイが証明書を受け付けるように設定されているかどうかによって異なります。

注 ゲートウェイが証明書を受け付けるように設定されている場合、ゲートウェイは証明書付きのログインだけを受け付け、その他のログインを拒否します。

ゲートウェイは、証明書が既知の認証局から発行されたものであるか、有効期限内であるか、変更されていないかどうかをチェックします。証明書が有効であれば、ユーザーが認証プロセスの次の手順に進むことを許可します。

2. ゲートウェイはサーバー内の PDC 認証モジュールに証明書を渡します。

▶ PDC とコード化されたデバイスを設定するには

PDC とコード化されたデバイスを設定するには、次の手順を実行します。

1. Portal Server マシンで、
`portal-server-install-root/SUNWam/config/AMConfig-instance-name.properties` ファイルに次の行を追加します。

`com.ipplanet.authentication.modules.cert.gwAuthEnable=yes`

(ファイルのどの場所に追加してもかまわない)
2. PDC を有効にするゲートウェイの認証データベースに、適切な証明書をインポートします。

詳細については、[第7章「証明書」](#)を参照してください。
3. 証明書を登録します。

- a. Access Manager 管理コンソールに管理者としてログインします。
 - b. 「アイデンティティ管理」タブを選択します。
 - c. 「組織」を選択します。
 - d. 「表示」ドロップダウンメニューから「サービス」を選択します。
 - e. 「コア」の隣の矢印をクリックします。
 - f. 「組織認証モジュール」リストボックスの「証明書」と「LDAP」を選択します。
 - g. 「ユーザープロファイル」ドロップダウンメニューから「ダイナミック」を選択します。
 - h. 「保存」をクリックします。
4. 「信頼できるリモートホスト」リストを作成します。
 - a. 「サービス設定」タブをクリックします。
 - b. 「認証設定」の隣の矢印をクリックします。
 - c. 「信頼できるリモートホスト」リストボックスまでスクロールします。
 - d. 何も強調表示せずに「削除」をクリックします。
 - e. テキストボックスに何らかの文字列を入力します。
 - f. 「追加」をクリックします。
 - g. 「保存」をクリックします。
 5. 新しいインスタンスを作成します。
 - a. 「アイデンティティ管理」タブをクリックします。
 - b. 「表示」ドロップダウンメニューから「サービス」を選択します。
 - c. 「認証設定」の隣の矢印をクリックします。
「サービスインスタンスリスト」が表示されます。
 - d. 「新規」をクリックします。
「新規サービスインスタンス」ページが表示されます。
 - e. サービスインスタンス名に「gatewaypcd」と入力します。
注：この名前を使用する必要があります。
 - f. 「送信」をクリックします。
「gatewaypcd」がサービスインスタンスに表示されます。

- g. 「gatewaypdc」をクリックし、サービスを編集します。
「gatewaypdc プロパティを表示」ページが表示されます。
 - h. 右の区画の「認証設定」の隣の「編集」リンクをクリックします。
ポップアップウィンドウが表示されます。
 - i. 「追加」をクリックします。
「<組織名> の認証設定」ページが表示されます。
 - j. 「追加」をクリックします。
「認証モジュールを追加」ページが表示されます。
 - k. 「モジュール名」の「証明書」と、「適用基準」の「REQUIRED」を選択します。
 - l. 「了解」をクリックします。
 - m. 「了解」をもう一度クリックし、ポップアップウィンドウを閉じます。
6. 証明書をゲートウェイホストに関連付けます。
- a. 「サービス設定」タブを選択します。
 - b. 「ゲートウェイ」の隣の矢印をクリックします。
ゲートウェイプロファイルが右の区画に表示されます。
 - c. ゲートウェイプロファイルを選択します。
 - d. 「セキュリティ」タブをクリックします。
 - e. 「証明書が有効なゲートウェイホスト」リストボックスにゲートウェイ名が追加されます。
 - f. 「保存」をクリックします。
 - g. サーバーを再起動します。
 - h. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```
7. PDC を有効にしたゲートウェイへのアクセス権を必要とするブラウザに対して、CA から発行されたクライアント証明書をインストールします。
8. 次のゲートウェイプロファイルと組織にアクセスします。

```
https://gateway:instance-port/YourOrganization
```


ユーザー名とパスワードを要求するプロンプトが表示されずに、証明書の名前を使用してログインできます。

リライタタブ

ゲートウェイサービスの「リライタ」タブでは、次のタスクを実行できます。

- すべての URL のリライトの有効化
- URI とルールセットのマッピングリストの作成
- パースする MMI タイプリストの作成
- デフォルトドメインの指定
- リライトしない URI のリストの作成
- MIME 推測の有効化
- パースする URI マッピングリストの作成
- マスキングの有効化
- マスキングのためのシード文字列の指定
- マスクしない URI のリストの作成
- ゲートウェイプロトコルと元の URI プロトコルの同一化

すべての URL のリライトの有効化

ゲートウェイサービスで「すべての URL のリライトを有効」オプションを有効にすると、「ドメインとサブドメインのプロキシ」リストのエントリをチェックせずに、リライタはすべての URL をリライトします。「ドメインとサブドメインのプロキシ」リストのエントリは無視されます。

- ▶ ゲートウェイによるすべての URL のリライトを有効にするには
1. Access Manager 管理コンソールに管理者としてログインします。
 2. 「サービス設定」タブを選択します。
 3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
 4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
 5. 「リライタ」タブをクリックし、「基本」サブセクションを表示します。
 6. 「すべての URL のリライトを有効」チェックボックスにチェックマークを付け、ゲートウェイによるすべての URL のリライトを有効にします。
 7. 「保存」をクリックし、変更内容を記録します。

8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

URI とルールセットのマッピングリストの作成

ルールセットは、Access Manager 管理コンソールの「Portal Server 設定」の下のリライタサービスに作成されます。詳細については、『Portal Server 管理ガイド』を参照してください。

ルールセットを作成したら、「URI をルールセットにマップ」フィールドを使用してドメインとルールセットを関連付けます。デフォルトでは、「URI をルールセットにマップ」リストに次の 2 つのエントリが追加されます。

- `*:/*Sun.COM/portal/*|default_gateway_ruleset`

この `sun.com` はポータルインストールドメインで、`/portal` はポータルのインストールコンテキストです。

- `*|generic_ruleset`

デフォルトドメインのすべてのページに対して、デフォルトゲートウェイのルールセットが適用されます。他のすべてのページには、汎用ルールセットが適用されます。デフォルトのゲートウェイルールセットと汎用ルールセットはパッケージ内のルールセットです。

注 デスクトップに表示されるすべてのコンテンツについて、コンテンツがフェッチされる場所にかかわらず、デフォルトドメインのルールセットが使用されます。

たとえば、URL `yahoo.com` のコンテンツを集めるようにデスクトップを設定すると仮定します。Portal Server は `sesta.com` 内にあります。フェッチされたコンテンツに `sesta.com` のルールセットが適用されます。

注 ルールセットを指定するドメインは、「ドメインとサブドメインのプロキシ」リストに含まれている必要があります。

► URI をルールセットにマッピングするには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。

「ゲートウェイ」ページが表示されます。

4. 編集する属性が含まれるゲートウェイプロファイルをクリックします。
「ゲートウェイ-(ゲートウェイプロファイル名)」ページが表示されます。
5. 「リライタ」タブをクリックし、「基本」サブセクションを表示します。
6. 「URI をルールセットにマップ」フィールドまでスクロールします。
7. 「URI をルールセットにマップ」フィールドに適切なドメイン名またはホスト名とルールセットを入力し、「追加」をクリックします。

「URI をルールセットにマップ」リストにエントリが追加されます。

ドメインまたはホスト名とルールセットは次の形式で指定します。

ドメイン名 | ルールセット名

例

eng.sesta.com|default

注 ルールセットを適用する順序は、ホスト名 - サブドメイン - ドメインの順です。

たとえば、「ドメインベースのルールセット」リストに次のエントリを指定していると仮定します。

sesta.com|ruleset1

eng.sesta.com|ruleset2

host1.eng.sesta.com|ruleset3

ruleset3 は host1 のすべてのページに適用されます。

ruleset2 は、host1 から取得されたページを除く eng のすべてのページに適用されます。

ruleset1 は、eng サブドメインおよび host1 から取得されたページを除く、sesta.com ドメインのすべてのページに適用されます。

8. 「保存」をクリックし、変更内容を記録します。
9. 端末ウィンドウからゲートウェイを再起動します。

`gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start`

Outlook Web Access 用のルールセット

SRA ソフトウェアでは、OWA (Outlook Web Access) から MS Exchange 2000 SP3 インストールおよび MS Exchange 2003 にアクセスする機能がサポートされます。

▶ OWA のルールセットを設定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 編集する属性が含まれるゲートウェイプロファイルをクリックします。
「ゲートウェイ-(ゲートウェイプロファイル名)」ページが表示されます。
5. 「URI をルールセットにマップ」フィールドで、Exchange 2000 がインストールされているサーバー名を入力し、それに続けて Exchange 2000 Service Pack 4 OWA ルールセットを入力します。

例

```
exchange.domain.com|exchange_2000sp3_owa_ruleset
```

パースする MIME タイプリストの作成

リライタでは、コンテンツタイプ、つまり HTML、JavaScript、CSS、XML に基づいて Web ページをパースするために、4 つのパーサーが使用されます。デフォルトでは、これらのパーサーには一般的な MIME タイプが関連付けられています。新しい MIME タイプとこれらのパーサーの関連付けは、ゲートウェイサービスの「パーサーを MIME タイプにマップ」フィールドで行います。これにより、リライタ機能を他の MIME タイプに拡張できます。

複数のエントリーは、セミコロン (;) またはカンマ (,) で区切ります。

例

```
HTML=text/html;text/htm;text/x-component;text/wml;text/vnl/wap.wml
```

これは、これらの MIME が HTML リライタに送られ、URL のリライトに HTML ルールを適用することを指定しています。

ヒント

MIME マッピングリストから不要なパーサーを削除すると、処理速度が向上します。たとえば、特定のイントラネットのコンテンツに JavaScript が含まれないことが確実な場合は、MIME マッピングリストから JavaScript エントリーを削除できます。

▶ MIME のマッピングを指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 編集する属性が含まれるゲートウェイプロファイルをクリックします。
「ゲートウェイ-(ゲートウェイプロファイル名)」ページが表示されます。
5. 「リライタ」タブをクリックし、「基本」サブセクションを表示します。
6. 「パーサーを MIME タイプにマップ」フィールドまでスクロールし、編集ボックスに必要な MIME タイプを追加します。複数のエントリを区切るときは、セミコロンまたはカンマを使用します。

エントリは `HTML=text/html;text/htm` の形式で指定します。

7. 「追加」をクリックし、必要なエントリをリストに追加します。
8. 「保存」をクリックし、変更内容を記録します。
9. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

リライトしない URI のリストの作成

▶ リライトしない URI を指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 編集する属性が含まれるゲートウェイプロファイルをクリックします。
「ゲートウェイ-(ゲートウェイプロファイル名)」ページが表示されます。
5. 「リライタ」タブをクリックし、「基本」サブセクションを表示します。
6. 「リライトしない URI」フィールドまでスクロールし、編集ボックスに URI を追加します。

注：このリストに #* を追加することで、href ルールがルールセットの一部である場合でも URI をリライトできます。

7. 「保存」をクリックし、変更内容を記録します。

8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

デフォルトドメインの指定

デフォルトのドメインは、URL にホスト名だけが含まれ、ドメインとサブドメインが指定されていない場合に便利です。この場合、ゲートウェイはホスト名がデフォルトのドメインリストにあるものと仮定し、そのように処理を進めます。

たとえば、URL のホスト名が `host1`、デフォルトのドメインとサブドメインが `red.sesta.com` のように指定されている場合、ホスト名は `host1.red.sesta.com` として解決されます。

▶ デフォルトドメインを指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブをクリックします。
3. 「SRA 設定」の「ゲートウェイ」の隣の右矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「リライタ」タブをクリックし、「基本」サブセクションを表示します。
6. 「デフォルトのドメイン」フィールドまでスクロールし、必要なデフォルト値を `subdomain.domain` の形式で入力します。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

MIME 推測の有効化

リライタは、パーサーの選択にページの MIME タイプを使用します。WebLogic や Oracle などの一部の Web サーバーは MIME タイプを送信しません。これに対応するには、「パーサーと URI のマッピング」リストボックスにデータを追加して、MIME 推測機能を有効にします。

▶ MIME 推測を有効にするには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 編集する属性が含まれるゲートウェイプロファイルをクリックします。
「ゲートウェイ-(ゲートウェイプロファイル名)」ページが表示されます。
5. 「リライタ」タブをクリックし、「基本」サブセクションを表示します。
6. 「MIME 推測を有効」チェックボックスにチェックマークを付け、MIME 推測を有効にします。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

パースする URI マッピングリストの作成

MIME 推測機能が有効で、サーバーが MIME タイプを送信しない場合は、このリストを使用してパーサーと URI がマッピングされます。

複数の URI はセミコロンで区切られます。

たとえば、HTML=*.html;*.htm;*Servlet のように指定します。

この例の設定では、HTML リライタは拡張子が html、htm、Servlet のすべてのページのコンテンツをリライトします。

▶ パーサーを URI にマッピングするには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 編集する属性が含まれるゲートウェイプロファイルをクリックします。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「リライタ」タブをクリックし、「基本」サブセクションを表示します。

6. 「パーサーと URI のマッピング」フィールドまでスクロールし、編集ボックスにデータを追加します。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

マスキングの有効化

マスキングを有効にすることで、リライタはページのイントラネット URL が判読されないように URI をリライトします。

▶ マスキングを有効にするには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 編集する属性が含まれるゲートウェイプロファイルをクリックします。
「ゲートウェイ-(ゲートウェイプロファイル名)」ページが表示されます。
5. 「リライタ」タブをクリックし、「基本」サブセクションを表示します。
6. 「マスキングを有効」チェックボックスにチェックマークを付け、マスキングを有効にします。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

マスキングのためのシード文字列の指定

URI のマスキングには、シード文字列が使用されます。マスキングアルゴリズムにより文字列が生成されます。

注 マスクされた URI をブックマークしても、このシード文字列が変更されたり、ゲートウェイが再起動された場合は機能しなくなります。

▶ マスキングのためのシード文字列を指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 編集する属性が含まれるゲートウェイプロファイルをクリックします。
「ゲートウェイ-(ゲートウェイプロファイル名)」ページが表示されます。
5. 「リライタ」タブをクリックし、「基本」サブセクションを表示します。
6. 「マスキングのシード文字列」フィールドまでスクロールし、編集ボックスに文字列を追加します。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

マスクしない URI のリストの作成

アプレットなどの一部のアプリケーションはインターネット URI を必要とし、マスクすることができません。これらのアプリケーションを指定するには、リストボックスに URI を追加します。

たとえば、次のように追加します。

```
*/Applet/Param*
```

リストボックスに追加した URL は、コンテンツの URI

`http://abc.com/Applet/Param1.html` がルールセット内のルールと一致する場合にマスクされません。

▶ マスクしない URI を作成するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 編集する属性が含まれるゲートウェイプロファイルをクリックします。
「ゲートウェイ-(ゲートウェイプロファイル名)」ページが表示されます。
5. 「リライタ」タブをクリックし、「基本」サブセクションを表示します。

6. 「マスクしない URI」のリストフィールドまでスクロールし、編集ボックスに URI を追加します。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

ゲートウェイプロトコルと元の URI プロトコルの同一化

ゲートウェイが HTTP と HTTPS の両方のモードで稼動する場合、HTML コンテンツ内で参照されるリソースへのアクセスに同じプロトコルを使用するようにリライタを設定できます。

たとえば、元の URL が `http://intranet.com/Public.html` であれば、HTTP ゲートウェイが追加されます。元の URL が `https://intranet.com/Public.html` であれば、HTTPS ゲートウェイが追加されます。

注 これは、スタティックな URI だけに適用され、JavaScript によって生成されるダイナミック URI には適用されません。

▶ ゲートウェイプロトコルと元の URI プロトコルを同一化するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 編集する属性が含まれるゲートウェイプロファイルをクリックします。
「ゲートウェイ -(ゲートウェイプロファイル名)」ページが表示されます。
5. 「リライタ」タブをクリックし、「基本」サブセクションを表示します。
6. 「ゲートウェイプロトコルを元の URI プロトコルと同じにする」チェックボックスにチェックマークを付けます。
7. 「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

ロギングタブ

ゲートウェイサービスの「ロギング」タブでは、次のタスクを実行できます。

- [ロギングの有効化](#)
- [Netlet ロギングの有効化](#)

ロギングの有効化

ゲートウェイログファイルが、各セッションの最少情報または詳細情報のどちらを取り込むか指定できます。このログ情報は、Access Manager 設定属性の「ロギング」セクションに含まれる「ログの場所」属性で指定されたディレクトリに保存されます。このログは、Portal Server マシン上に置かれます。

ログ名には次の命名ルールがあります。

`srapGateway_gatewayhostname_gateway-profile-name`

ログ情報は Access Manager の設定に基づいて、ファイルまたはデータベースとして保存されます。ログのフィールドはカンマ区切りの ASCII 値で、他のデータ分析ツールにエクスポートできます。

▶ ゲートウェイのロギングを有効にするには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「リライタ」タブをクリックします。
6. 「ロギングを有効」チェックボックスにチェックマークを付けて、ゲートウェイのロギングを有効にします。

注 ログ情報が取り込まれるのは、「ロギングを有効」フィールドがすでに有効になっている場合だけです。

7. クライアントアドレス、要求タイプ、宛先ホストなどの最低限のログ情報を取り込むには、「セッション単位のロギングを有効」チェックボックスにチェックマークを付けます。

8. ゲートウェイがクライアント、要求型、宛先ホスト、要求のタイプ、クライアント要求 URL、クライアントポート、データサイズ、セッション ID、応答結果コード、完全応答サイズなどの詳細情報を取り込むようにするには、「セッション単位の詳細なロギングを有効」を選択します。

注 詳細なログ情報が取り込まれるのは、「セッション単位のロギングを有効」フィールドがすでに有効になっている場合だけです。

9. 「保存」をクリックし、変更内容を記録します。
10. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

Netlet ロギングの有効化

このオプションを選択すると、Netlet に関連するアクティビティのロギングを有効にできます。Netlet ログには、Netlet セッションに関する次の詳細情報が記録されます。

- 開始時間
- ソースアドレス
- ソースポート
- サーバーアドレス
- サーバーポート
- 停止時間
- 状態 (起動または停止)

► Netlet ロギングを有効にするには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「サービス設定」タブを選択します。
3. 「SRA 設定」の「ゲートウェイ」の隣の矢印をクリックします。
「ゲートウェイ」ページが表示されます。
4. 属性を設定するゲートウェイプロファイルを選択します。
「ゲートウェイプロファイルを編集」ページが表示されます。
5. 「ロギング」タブをクリックします。

6. 「Netlet ロギングを有効」チェックボックスにチェックマークを付けて、Netlet ロギングを有効にします。
7. ページの下部にある「保存」をクリックし、変更内容を記録します。
8. 端末ウィンドウからゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

ロギングタブ

NetFile の設定

この章では、Sun Java™ System Access Manager 管理コンソールから NetFile を設定する方法について説明します。

注 SRA のすべての属性について簡単に調べるには、Access Manager 管理コンソールの右上に表示される「ヘルプ」をクリックし、「Secure Remote Access 管理ヘルプ」をクリックします。

► **NetFile の属性を設定するには**

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。
5. 「表示」リストボックスから「サービス」を選択します。
6. 「SRA 設定」の下の「NetFile」の隣にある矢印をクリックします。

このページで、適切なタブをクリックします。

- [ホストタブ](#)
- [権限タブ](#)
- [表示タブ](#)
- [操作タブ](#)
- [一般タブ](#)

次に、これらのタブと、各タブで設定できる属性について説明します。

ホストタブ

NetFile サービスの「ホスト」タブでは、次のタスクを実行できます。

- OS の文字セットの指定
- ホスト検出順序の指定
- 共通ホストのリストの設定
- デフォルトドメインの指定
- Microsoft Windows のドメイン / ワークグループの指定
- デフォルトの WINS/DNS サーバーの指定
- 異なるタイプのホストへのアクセスの指定
- 「許可されるホスト」リストの設定
- 「拒否されるホスト」リストの設定

OS の文字セットの指定

ホストとの対話にデフォルトエンコーディングとして使用する文字セットを指定できます。デフォルト値は UTF-8 です。

警告	文字セットが正しく指定されていないと、エラーメッセージが正しく表示されず、マシンの動作も予測できません。
-----------	------------------------------------------------------

▶ OS の文字セットを指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。
5. 「表示」リストボックスから「サービス」を選択します。
6. 「SRA 設定」の下の「NetFile」の隣にある矢印をクリックします。
「NetFile」ページが表示されます。
7. 「ホスト」タブをクリックし、「設定」サブセクションを表示します。
8. 「OS 文字セット」フィールドまでスクロールし、文字セットコードを選択します。

9. 「保存」をクリックし、変更内容を記録します。

ホスト検出順序の指定

▶ ホスト検出順序を指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。
5. 「表示」リストボックスから「サービス」を選択します。
6. 「SRA 設定」の下の「NetFile」の隣にある矢印をクリックします。
「NetFile」ページが表示されます。
7. 「ホスト」タブをクリックし、「設定」サブセクションを表示します。
8. 「ホスト検出順序」フィールドまでスクロールし、ホストのタイプを選択します。
9. 「上に移動」ボタンと「下に移動」ボタンを使用して、ホストの検出順序を変更します。
10. 「保存」をクリックし、変更内容を記録します。

共通ホストのリストの設定

すべてのリモート NetFile ユーザーが NetFile を通じて使用できるホストのリストを設定できます。追加する各ホストについて、次の情報を指定します。

ホスト名 : ホスト名または完全修飾名を入力します。指定したホスト名がユーザーが設定したホスト名と一致する場合、両方の情報が統合され、指定した値がユーザーが指定した値に上書きされます。

たとえば、共通の 4 つのホスト `sesta`、`siroe`、`florizon`、および `abc` を設定しているとします。ユーザーはそのうち 2 つのホスト、`sesta` と `siroe` を設定します。この場合、ユーザーが指定した値は管理者が指定した値よりも優先されます。ユーザーの NetFile には、`florizon` と `abc` もリストされ、ユーザーは 2 つのホストでさまざまな処理を実行できます。「拒否されたホスト」リストに `florizon` を指定している場合、ユーザーの NetFile に `florizon` がリストされますが、`florizon` については処理が実行できません。

ホストのタイプ: ユーザーが「共通ホスト」リスト内にあるマシンをすでに追加している場合、ユーザーの設定が優先されます。タイプが競合する場合、管理者が追加した共有はそのユーザーには追加されません。ユーザーと管理者が同じ共通を追加した場合、その共有は追加されますがユーザーが設定したパスワードが優先されます。

エンコーディング: ここで指定した値とユーザーの設定が競合する場合、ユーザーの設定が優先されます。設定を空白にしている、または無効な値を指定している場合、クライアントの OS (ユーザーのマシン) の文字セットが使用されます。

注 ユーザーは NetFile クライアントアプリケーションでこれらの値を編集できます。ただし、編集した値が有効なのは、現在のセッションだけです。ユーザーがログアウトし再びログインすると、編集された値は保持されません。

▶ **共通ホストのリストを設定するには**

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。
5. 「表示」リストボックスから「サービス」を選択します。
6. 「SRA 設定」の下の「NetFile」の隣にある矢印をクリックします。
「NetFile」ページが表示されます。
7. 「ホスト」タブをクリックし、「設定」サブセクションを表示します。
「NetFile」ページが表示されます。
8. 共通ホストを追加するには
 - a. 「共通ホスト」フィールドの「新規」をクリックします。
 - b. 次のフィールドに適切な情報を入力します。
 - ホスト名
 - ホストタイプ
 - 暗号化
 - Microsoft Windows ドメイン / ワークグループ
 - ユーザー名
 - パスワード

- c. 追加する共有ごとに次のフィールドに適切な情報を入力し、「リストに追加」をクリックします。
 - o 共有リスト
 - o 共有名
 - o 共有パスワード
 - d. 「了解」をクリックします。
 - e. 追加または削除する共通ホストごとに、この一連の情報を繰り返します。
9. 「共通ホスト」リストから共通ホストを削除するには
- a. 「削除」をクリックし、「共有リスト」のホスト名を選択します。次に、「削除」をクリックします。
 - b. 「了解」をクリックし、変更内容を記録します。

デフォルトドメインの指定

NetFile が許可されたホストへのアクセスに使用するデフォルトドメインを指定できません。

このデフォルト値が適用されるのは、ユーザーが NetFile を使用してホストを追加するときに、完全修飾ホスト名を指定していない場合です。

警告 「デフォルトドメイン」フィールドが空ではなく、有効なドメイン名が指定されていることを確認してください。

▶ デフォルトドメインを指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。
5. 「表示」リストボックスから「サービス」を選択します。
6. 「SRA 設定」の下の「NetFile」の隣にある矢印をクリックします。
「NetFile」ページが表示されます。
7. 「ホスト」タブをクリックし、「設定」サブセクションを表示します。

8. 「デフォルトドメイン」フィールドまでスクロールし、デフォルトのドメイン名を入力します。
9. 「保存」をクリックし、変更内容を記録します。

Microsoft Windows のドメイン / ワークグループの指定

これは、ユーザーが Microsoft Windows ホストにアクセスするときに使用する、デフォルトの Microsoft Windows ドメインまたはワークグループです。

ユーザーはマシンを追加するときに別の値を指定し、この値を上書きできます。

▶ デフォルトの Microsoft Windows ドメインまたはワークグループを指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。
5. 「表示」リストボックスから「サービス」を選択します。
6. 「SRA 設定」の下の「NetFile」の隣にある矢印をクリックします。
「NetFile」ページが表示されます。
7. 「ホスト」タブをクリックし、「設定」サブセクションを表示します。
8. 「デフォルトの Windows ドメイン / ワークグループ」フィールドまでスクロールし、デフォルトのドメイン名またはワークグループ名を入力します。
9. 「保存」をクリックし、変更内容を記録します。

デフォルトの WINS/DNS サーバーの指定

これは、Microsoft Windows ホストへのアクセスで NetFile が使用する WINS/DNS サーバーです。

- ▶ デフォルトの WINS/DNS サーバーを指定するには
 1. Access Manager 管理コンソールに管理者としてログインします。
 2. 「アイデンティティ管理」タブを選択します。
 3. 「表示」 ドロップダウンリストから「組織」を選択します。
 4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。
 5. 「表示」 リストボックスから「サービス」を選択します。
 6. 「SRA 設定」の下の「NetFile」の隣にある矢印をクリックします。
「NetFile」ページが表示されます。
 7. 「ホスト」タブをクリックし、「設定」サブセクションを表示します。
 8. 「デフォルトの WINS/DNS サーバー」フィールドまでスクロールし、デフォルトの Microsoft Windows または DNS サーバー名を入力します。
 9. 「保存」をクリックし、変更内容を記録します。

異なるタイプのホストへのアクセスの指定

Microsoft Windows、FTP、NFS、または Netware ホストなどの特定のホストへのユーザーのアクセスを指定できます。各タイプのホストへのアクセスを許可または拒否するオプションを設定できます。デフォルトでは、これらのオプションはすべて有効になっています。

- ▶ 異なるタイプのホストへのアクセスを指定するには
 1. Access Manager 管理コンソールに管理者としてログインします。
 2. 「アイデンティティ管理」タブを選択します。
 3. 「表示」 ドロップダウンリストから「組織」を選択します。
 4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。
 5. 「表示」 リストボックスから「サービス」を選択します。
 6. 「SRA 設定」の下の「NetFile」の隣にある矢印をクリックします。
「NetFile」ページが表示されます。

7. 「ホスト」タブをクリックし、「アクセス」サブセクションを表示します。
8. アクセスを可能にするホストタイプをクリックします。次のオプションから選択できます。
 - Windows ホストへのアクセスを許可
 - FTP ホストへのアクセスを許可
 - NFS ホストへのアクセスを許可
 - Netware ホストへのアクセスを許可オプションを選択すると、そのタイプのホストへのアクセスが可能になります。チェックボックスのチェックマークを外すと、そのタイプのホストにアクセスできなくなります。
9. 「保存」をクリックし、変更内容を記録します。

「許可されるホスト」リストの設定

デフォルトでは、このリストに * が指定されているため、ユーザーは NetFile を通じてすべてのホストにアクセスできます。この設定を変更する場合、* を削除し、ユーザーが NetFile を通じてアクセスする必要のあるホストだけをこのリストに指定します。または、この * エントリを残し、「拒否されたホスト」リストでアクセスを拒否するホストを指定します。その場合、「拒否されたホスト」リストで指定したホストを除きすべてのホストへのアクセスが許可されます。

詳細については、[295 ページの「拒否されるホスト」リストの設定](#)を参照してください。

注 「許可されたホスト」と「拒否されたホスト」リストがいずれも空白の場合、どのホストにもアクセスできません。

▶ 許可されたホストリストを作成するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。
5. 「表示」リストボックスから「サービス」を選択します。
6. 「SRA 設定」の下の「NetFile」の隣にある矢印をクリックします。

「NetFile」ページが表示されます。

7. 「ホスト」タブをクリックし、「アクセス」サブセクションを表示します。
8. 「許可されたホスト」フィールドまでスクロールします。編集フィールドに、アクセスを許可するホストの名前を入力し、「追加」をクリックします。
「許可されたホスト」リストボックスにホスト名が追加されます。
9. 「保存」をクリックし、変更内容を記録します。

「拒否されるホスト」リストの設定

289 ページの「共通ホストのリストの設定」で共通に使用できるホストのリストを指定した後に、NetFile を通じたユーザーのアクセスを拒否するホストのリストも指定できます。

注	<p>ホストへのアクセスを拒否し、ユーザーがすでに NetFile ウィンドウでこのホストを追加している場合、ユーザーの NetFile ウィンドウには、その後も拒否されたホストが表示されます。ただし、ユーザーはこのホストでは操作を行えません。</p> <p>NetFile Java2 では、アプリケーションに拒否されたホストが表示されるときに、そのホストに赤の十字がマークされ、アクセスできないことを示します。</p>
----------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

注	<p>「許可されたホスト」と「拒否されたホスト」リストがいずれも空白の場合、どのホストにもアクセスできません。</p>
----------	-------------------------------------------------------------

▶ 拒否されたホストリストを作成するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。
5. 「表示」リストボックスから「サービス」を選択します。
6. 「SRA 設定」の下の「NetFile」の隣にある矢印をクリックします。
「NetFile」ページが表示されます。

7. 「ホスト」タブをクリックし、「アクセス」サブセクションを表示します。
8. 「拒否されたホスト」フィールドまでスクロールします。編集フィールドに、アクセスを拒否するホストの名前を入力します。
9. 「追加」をクリックします。
「拒否されたホスト」リストボックスにホスト名が追加されます。
10. 「保存」をクリックし、変更内容を記録します。

権限タブ

NetFile サービスの「権限」タブでは、次のタスクを実行するためのユーザーのアクセス権をリモートホストから許可または拒否できます。

- ファイル名の変更
- ファイルとフォルダの削除
- ファイルのアップロード
- ファイルとフォルダのダウンロード
- ファイルの検索
- ファイルのメール送信
- ファイルの圧縮
- ユーザー ID の変更

このオプションを使用することで、ユーザーが NetFile を使用してホストに接続する場合に、異なる ID を使用できるかどうかを指定できます。大規模な組織では、ユーザーは複数のユーザー ID を持つことができます。ユーザーが単一のユーザー ID を使用するように制限する場合は、「ユーザー ID の変更を許可」オプションを無効にします。これにより、特定の組織のすべてのユーザーがユーザー ID を変更できなくなり、NetFile を使用してホストに接続するときに使用する ID が単一の ID (デスクトップログイン ID) に制限されます。また、ユーザーがマシンごとに異なるログイン ID を持つことがあります。この場合、必要に応じてユーザーによる ID の変更を許可することができます。

- Microsoft Windows ドメインの変更

このオプションは、NT ドメインだけに適用されます。

ユーザーがシステムを追加するときに「ユーザーの NT ドメイン名」フィールドに無効なドメイン名を指定すると、エラーメッセージが表示されます。ユーザーが後でホスト情報を編集し、無効なドメイン名を指定しても、エラーメッセージは表示されません。

ユーザーがドメイン名を指定するときは、そのドメインのユーザー名とパスワードも指定する必要があります。ホストのユーザー名とパスワードを使用する必要がある場合、ユーザーは「ユーザーの NT ドメイン名」フィールドからドメインを削除しなければなりません。

デフォルトでは、アクセス権のオプションはすべて有効になっています。

▶ **アクセス権を有効化または無効化するには**

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。
5. 「表示」リストボックスから「サービス」を選択します。
6. 「SRA 設定」の下の「NetFile」の隣にある矢印をクリックします。
「NetFile」ページが表示されます。
7. 「権限」タブをクリックします。
8. 適切な「許可」フィールドまでスクロールし、チェックボックスにチェックマークを付けてアクセス権を有効にします。
9. 「保存」をクリックし、変更内容を記録します。

注 ユーザーが NetFile の使用を開始した後にこのオプションを無効にすると、ユーザーがログアウトし再びログインした後に変更内容が有効になります。

表示タブ

NetFile サービスの「表示」タブでは、次のタスクを実行できます。

- [NetFile のウィンドウサイズの指定](#)
- [NetFile ウィンドウの位置の指定](#)

NetFile のウィンドウサイズの指定

ユーザーのデスクトップの NetFile ウィンドウのサイズを、ピクセル単位で指定できます。デフォルト値は 700|400 ピクセルです。無効な値を入力した場合、NetFile はデフォルトの値を使用します。

注 ユーザーが使用できる制限付きの管理コンソールでも、この値を編集することができます。指定した値は、ユーザーがデスクトップで NetFile ウィンドウのサイズを変更したときに新しい値に置き換わります。

▶ NetFile ウィンドウのサイズを指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。
5. 「表示」リストボックスから「サービス」を選択します。
6. 「SRA 設定」の下の「NetFile」の隣にある矢印をクリックします。
「NetFile」ページが表示されます。
7. 「表示」タブをクリックします。
8. 「ウィンドウのサイズ」フィールドまでスクロールし、適切なウィンドウサイズをピクセル単位で入力します。
値は 700|400 の形式で入力し、空白文字を挿入しません。座標軸は x|y です。他の文字を区切り文字として使用することはできません。
9. 「保存」をクリックし、変更内容を記録します。

NetFile ウィンドウの位置の指定

NetFile ウィンドウがユーザーのデスクトップに表示される位置を指定できます。デフォルト値は 100|50 ピクセルです。無効な値を入力した場合、NetFile はデフォルトの値を使用します。

注 ユーザーが使用できる制限付きの管理コンソールでも、この値を編集することができます。指定した値は、ユーザーがデスクトップで NetFile ウィンドウの位置を変更したときに新しい値に置き換わります。

► NetFile ウィンドウの位置を指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。
5. 「表示」リストボックスから「サービス」を選択します。
6. 「SRA 設定」の下の「NetFile」の隣にある矢印をクリックします。
「NetFile」ページが表示されます。
7. 「表示」タブをクリックします。
8. 「ウィンドウの位置」フィールドまでスクロールし、適切なウィンドウ位置の座標を入力します。
値は 100|50 の形式で入力し、空白文字を挿入しません。座標軸は x|y です。他の文字を区切り文字として使用することはできません。
9. 「保存」をクリックし、変更内容を記録します。

操作タブ

NetFile サービスの「操作」タブでは、次のタスクを実行できます。

- 一時ファイルディレクトリの指定
- ファイルアップロードサイズの制限の設定
- 検索ディレクトリ制限の指定
- 圧縮の指定

一時ファイルディレクトリの指定

NetFile は、ファイルのメール送信など、さまざまなファイル操作に一時ディレクトリを必要とします。デフォルトの一時ディレクトリは /tmp です。一時ファイルは、必要な操作が実行された後に削除されます。

指定された一時ディレクトリがサーバー上に存在しない場合は作成されます。

Web サーバーが実行時に使用する ID (nobody または noaccess) に、指定されたディレクトリに対するアクセス権 `rwX` が割り当てられていることを確認します。また、要求される一時ディレクトリへの完全パスに対するアクセス権 `rx` が ID に割り当てられていることを確認します。

ヒント NetFile の一時ディレクトリを個別に作成する場合があります。Portal Server のすべてのモジュールに共通な一時ディレクトリを指定すると、ディスクの容量がすぐに足りなくなります。ファイルのメール送信など、NetFile の一部の操作は、一時ディレクトリの容量がなくなると機能しません。

▶ 一時ディレクトリを指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」ドロップダウンリストから「組織」を選択します。
4. 「表示」リストボックスから「サービス」を選択します。
5. 「SRA 設定」の下の「NetFile」の隣にある矢印をクリックします。
「NetFile」ページが表示されます。
6. 「操作」タブをクリックし、「トラフィック」サブセクションを表示します。

7. 「一時ディレクトリの場所」フィールドまでスクロールし、適切な一時ディレクトリの場所を入力します。
8. 「保存」をクリックし、変更内容を記録します。

ファイルアップロードサイズの制限の設定

このフィールドに、アップロードできる最大ファイルサイズを指定できます。アップロードするファイルのサイズがここで指定した制限を超えると、エラーメッセージが表示され、ファイルはアップロードされません。デフォルト値は 5M バイトです。無効な値を入力すると、NetFile は値をデフォルト値にリセットします。

ユーザーごとに異なるファイルアップロードサイズ制限を指定できます。

注 アップロードの最大ファイルサイズは、M バイト単位で指定します。整数値で指定する必要があります。

▶ ファイルアップロードサイズの制限を設定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。
5. 「表示」リストボックスから「サービス」を選択します。
6. 「SRA 設定」の下の「NetFile」の隣にある矢印をクリックします。
「NetFile」ページが表示されます。
7. 「操作」タブをクリックし、「トラフィック」サブセクションを表示します。
8. 「ファイルのアップロード制限」フィールドまでスクロールします。適切なサイズ制限を M バイト単位で入力します。
9. 「保存」をクリックし、変更内容を記録します。

検索ディレクトリ制限の指定

1 回の検索操作で検索できるディレクトリの最大数を設定できます。この制限により、ネットワークの停滞が軽減され、複数のユーザーが同時にログインした場合のアクセス速度も速くなります。デフォルト値は 100 です。無効な値を入力すると、NetFile は値をデフォルト値にリセットします。

ユーザーが A というディレクトリを使用しているとします。A には 100 のサブディレクトリがあります。検索するディレクトリの最大数を 100 に指定した場合、ディレクトリ A 全体の検索が行われ処理が停止します。ディレクトリ A で検索の制限数 100 に達したため、他のディレクトリの検索は行われません。検索の制限数を超えるまでに累積された検索結果と、検索の制限数を超えたことを示すエラーメッセージが表示されます。検索を続けるためには、ユーザーは次のディレクトリで手動で検索を再開する必要があります。

検索操作は、深度優先で行われます。つまり、検索の処理はユーザーが選択したディレクトリのすべてのサブディレクトリを実行し、その後次に次のディレクトリに移動します。

▶ ディレクトリ検索の制限を指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。
5. 「表示」リストボックスから「サービス」を選択します。
6. 「SRA 設定」の下の「NetFile」の隣にある矢印をクリックします。
「NetFile」ページが表示されます。
7. 「操作」タブをクリックし、「検索」サブセクションを表示します。
8. 「検索ディレクトリ制限」フィールドまでスクロールし、適切な数値を入力します。

注 このフィールドには、整数値を入力してください。

9. 「保存」をクリックし、変更内容を記録します。

圧縮の指定

圧縮の属性は、NetFile Java2 だけに適用されます。

▶ デフォルトの圧縮タイプを指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。
5. 「表示」リストボックスから「サービス」を選択します。
6. 「SRA 設定」の下の「NetFile」の隣にある矢印をクリックします。
「NetFile」ページが表示されます。
7. 「操作」タブをクリックし、「圧縮」サブセクションを表示します。
8. 「デフォルトの圧縮タイプ」フィールドまでスクロールします。
「Zip」または「GZip」を選択します。
9. 「保存」をクリックし、変更内容を記録します。

一般タブ

NetFile サービスの「一般」タブでは、MIME タイプ設定ファイルの場所を指定できません。

MIME タイプ設定ファイルの場所の指定

この情報はクライアントブラウザに送信する応答コンテンツのタイプを判断する場合に必要となります。ブラウザは NetFile を開くとき、またはダウンロード操作を行うときに、ファイルの関連付けが必要なアプリケーションを決定するために、この情報を必要とします。これは、インストール時に設定されます。

Portal Server の Web サーバーの MIME タイプファイルを使用する必要があるときは、ファイルの場所を指定します。

`portal-server-install-root/SUNWam/servers/instance-name-of-web-server-machine/config`

注 MIME タイプ設定ファイルの場所は、組織レベルだけで設定可能です。

▶ MIME タイプ設定ファイルの場所を指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。
5. 「表示」リストボックスから「サービス」を選択します。
6. 「SRA 設定」の下の「NetFile」の隣にある矢印をクリックします。
「NetFile」ページが表示されます。
7. 「一般」タブをクリックします。
8. 「MIME タイプ設定ファイルの場所」フィールドまでスクロールし、MIME タイプ設定ファイルが格納されている場所の完全パスを入力します。
9. 「保存」をクリックし、変更内容を記録します。

Netlet の設定

この章では、Sun Java™ System Access Manager 管理コンソールから Netlet の属性を設定する方法について説明します。

注 SRA のすべての属性について簡単に調べるには、Access Manager 管理コンソールの右上に表示される「ヘルプ」をクリックし、「Secure Remote Access 管理ヘルプ」をクリックします。

組織レベルで設定できるすべての属性は、ユーザーレベルでも設定できます。組織、ロール、ユーザーの各レベルの属性については、『Access Manager 管理ガイド』を参照してください。

Netlet 属性を設定するには、次の手順を実行し、組織レベルの属性を設定します。

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。
5. 「表示」ドロップダウンリストから「サービス」を選択します。
6. 「SRA 設定」の下に「Netlet」の隣にある矢印をクリックします。

このページでは、次のタスクを実行できます。

- [Netlet ルールの追加](#)
- [ユーザーへの Netlet サービスの割り当て](#)
- [Netlet ルールの追加](#)
- [既存の Netlet ルールの変更](#)
- [Netlet ルールの削除](#)

ユーザープロファイルの設定と Netlet ルールの作成以外の操作では、サイトの要件に基づいて次の属性を設定する必要があります。これらの属性は組織レベルまたはユーザーレベルで設定できます。

- デフォルトの暗号化方式の指定
- デフォルトループバックポートの割り当て
- 接続の再認証の有効化
- 接続の警告ポップアップを表示
- 「ポート警告ダイアログにチェックボックスを表示」の有効化
- キープアライブ間隔の設定
- 「Portal のログアウト時に Netlet を終了」オプションの設定
- Netlet ルールへのアクセスの定義
- Netlet ルールへのアクセスの拒否
- ホストへのアクセスの許可ホストへのアクセスの拒否

ユーザーへの Netlet サービスの割り当て

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。
選択した組織名が、管理コンソールの左上に場所として表示されます。
5. 選択した組織の「表示」ドロップダウンリストから「ユーザー」を選択します。
6. 左の区画で目的のユーザーの隣にある矢印をクリックします。
7. このユーザーが使用できる Netlet サービスが割り当てられていない場合は、このユーザーの「表示」ドロップダウンリストから「サービス」を選択します。
8. 「追加」をクリックします。
9. 「サービス」のリストから「Netlet」を選択します。
10. 「了解」をクリックします。
11. このユーザーの「表示」ドロップダウンリストから「Netlet」サービスを選択することで、Netlet 属性を変更できます。

Netlet ルールの追加

Access Manager 管理コンソールの「アイデンティティ管理」タブでは、Netlet ルールをグローバルレベルで追加または作成できます。これらのルールは、新しい組織を作成すると、その組織に継承されます。

新しいルールの作成または既存のルールの修正は、組織、ロール、ユーザーレベルで行えます。

▶ Netlet ルールを追加するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. ルールを作成する組織を選択します。
4. 「表示」ドロップダウンリストから「サービス」を選択します。
5. 「SRA 設定」の下の「Netlet」の隣にある矢印をクリックします。
右の区画に「Netlet」ページが表示されます。
6. 「Netlet ルール」フィールドの「追加」をクリックします。
Netlet ルールの追加ページが表示されます。ルールすべてのフィールドに同じ値が入力されていますが、必要に応じて変更できます。
7. 「ルール名」フィールドに一意の名前を入力します。
8. 適切な暗号化方式を指定します。デフォルトの暗号化方式を使用する場合は、「デフォルト」を選択します。使用できる暗号化方式のリストから選択するときは、「その他」を選択します。
デフォルトの暗号化方式については、[309 ページの「デフォルトの暗号化方式を指定するには」](#)を参照してください。
9. 呼び出すアプリケーションの URL を「URL」フィールドに入力します。
10. アプレットをダウンロードする必要がある場合は、「アプレットのダウンロード」チェックボックスにチェックマークを付けます。対応する編集ボックスに、`local-port:server-host:server-port` の形式でアプレットの詳細を入力します。

注 各ルールに一意のローカルポートを指定します。

アプレットの詳細を指定する必要があるのは、アプレットを Portal Server ホスト以外のホストからダウンロードする必要がある場合だけです。チェックボックスにチェックマークを付けていない場合、編集ボックスは無効になっています。詳細については、[187 ページの「リモートホストからのアプレットのダウンロード」](#)を参照してください。

11. このルールに対応する Netlet セッションの実行中は Portal Server セッション時間が延長されるようにするときは、「拡張セッション」チェックボックスにチェックマークを付けます。
12. Netlet が待機するローカルポートを「ローカルポート」フィールドに入力します。
FTP ルールでは、ローカルポートは 30021 である必要があります。
13. 「宛先ホスト」フィールドにエントリを入力します。
スタティックルールでは、Netlet 接続のターゲットマシンのホスト名を入力します。
ダイナミックルールでは、「TARGET」と入力します。
14. 宛先ホストのポートを宛先ポートフィールドに入力します。
15. 「リストに追加」をクリックして、「ローカルポートと宛先サーバーポートのマップ」フィールドに最後の 3 つのエントリを反映させます。
16. 「了解」をクリックします。
ルールが保存され、「Netlet」ページに戻ります。「Netlet ルール」リストに新しいルールが表示されます。

既存の Netlet ルールの変更

管理コンソールの「アイデンティティ管理」タブでは、既存のルールを組織、ロール、ユーザーレベルで変更できます。これらのルールは、新しい組織を作成すると、その組織に継承されます。

► Netlet ルールを変更するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. ルールを修正する組織を選択します。
4. 「表示」ドロップダウンリストから「サービス」を選択します。
5. 「SRA 設定」の下の「Netlet」の隣にある矢印をクリックします。
右の区画に「Netlet」ページが表示されます。
6. 変更するルールの隣のチェックボックスにチェックマークを付けます。
「Netlet ルールの編集」ページが表示されます。
7. 必要な変更を行い、「了解」をクリックします。
修正されたルールが保存され、「Netlet」ページに戻ります。

Netlet ルールの削除

管理コンソールの「アイデンティティ管理」タブで、Netlet ルールをグローバルレベルで削除できます。

▶ Netlet ルールを削除するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. ルールを削除する組織を選択します。
4. 「SRA 設定」の下の「Netlet」の隣にある矢印をクリックします。
右の区画に「Netlet」ページが表示されます。
5. 「Netlet ルール」リストから削除するルールの横のチェックボックスを選択します。
6. 「削除」をクリックします。
選択したルールが「Netlet ルール」リストから削除されます。

注 ここでは、すべての属性の組織レベルでの設定について説明します。

デフォルトの暗号化方式の指定

Netlet ルールにはデフォルトの暗号化方式を指定する必要があります。これはルールの一部として暗号化方式が指定されていない既存のルールを使用する場合に便利です。このフィールドの設定は必須です。[194 ページの「下位互換性」](#)を参照してください。

▶ デフォルトの暗号化方式を指定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。
5. 「表示」ドロップダウンリストから「サービス」を選択します。
6. 「SRA 設定」の下の「Netlet」の隣にある矢印をクリックします。
右の区画に「Netlet」ページが表示されます。

7. 「デフォルトのネイティブ VM 暗号化方式」フィールド、または「デフォルトの Java プラグイン暗号化方式」フィールドまでスクロールし、ドロップダウンリストから適切な暗号化方式を選択します。サポートされる暗号化方式のリストについては、[193 ページの「サポートされる暗号化方式」](#)を参照してください。
8. 「保存」をクリックし、変更内容を記録します。

デフォルトループバックポートの割り当て

この属性は、Netlet を通じてアプレットがダウンロードされるときにローカルマシンで使用されるポートを指定します。Netlet ルールの設定値が優先される場合を除き、デフォルトの 58000 が使用されます。

▶ デフォルトループバックポートを割り当てるには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。
5. 「表示」ドロップダウンリストから「サービス」を選択します。
6. 「SRA 設定」の下の「Netlet」の隣にある矢印をクリックします。
右の区画に「Netlet」ページが表示されます。
7. 「デフォルトのループバックポート」フィールドまでスクロールし、適切なポート番号を入力します。
8. 「保存」をクリックし、変更内容を記録します。

接続の再認証の有効化

Netlet 接続を確立しようとするユーザーに、その都度 Netlet パスワードの入力を要求する場合は、このオプションを有効にします。このオプションを有効にすると、ユーザーのデスクトップに接続の警告ポップアップが表示されなくなります。詳細については、[311 ページの「接続の警告ポップアップを表示」](#)を参照してください。

このオプションを有効にすると、ユーザーは Netlet チャネルの編集オプションを使用して再認証パスワードを変更できるようになります。デフォルトでは、最初のパスワードは srap-Netlet です。

▶ 接続の再認証を有効にするには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。
5. 「表示」ドロップダウンリストから「サービス」を選択します。
6. 「SRA 設定」の下の「Netlet」の隣にある矢印をクリックします。
右の区画に「Netlet」ページが表示されます。
7. 「接続の再認証」フィールドまでスクロールし、オプションを選択します。
8. 「保存」をクリックし、変更内容を記録します。

接続の警告ポップアップを表示

この属性が有効になっている場合は、Netlet を使用してアプリケーションを実行しているときに、他のユーザーが待機ポートを通じて Netlet に接続しようとする、デスクトップに警告ポップアップダイアログボックスが表示されます。ユーザーのデスクトップにポップアップを表示しないようにするときは、この属性の選択を解除します。

▶ 接続の警告ポップアップを有効にするには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。

5. 「表示」ドロップダウンリストから「サービス」を選択します。
6. 「SRA 設定」の下の「Netlet」の隣にある矢印をクリックします。
右の区画に「Netlet」ページが表示されます。
7. 「接続の警告ポップアップを表示」チェックボックスにチェックマークを付けて、警告ポップアップを有効にします。
8. 「保存」をクリックし、変更内容を記録します。

「ポート警告ダイアログにチェックボックスを表示」の有効化

この属性が管理コンソールで有効になっている場合は、Netlet がローカルマシン上の自由に使用できるポートを通じて宛先ホストに接続しようとしたときに、ユーザーのデスクトップの警告ポップアップにチェックボックスが表示されます。このチェックボックスをデスクトップ上で切り替えることによって、ユーザーは警告ポップアップを有効または無効にすることができます。

管理コンソールの「ポート警告ダイアログにチェックボックスを表示」オプションを無効にすると、ユーザーはこの警告ポップアップを非表示にできるようになります。

▶ ユーザーによるポート警告ダイアログの非表示を許可するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。
5. 「表示」ドロップダウンリストから「サービス」を選択します。
6. 「SRA 設定」の下の「Netlet」の隣にある矢印をクリックします。
右の区画に「Netlet」ページが表示されます。
7. 「ポート警告ダイアログにチェックボックスを表示」フィールドまでスクロールし、チェックボックスのチェックマークを外します。
8. 「保存」をクリックし、変更内容を記録します。

キープアライブ間隔の設定

クライアントが Web プロキシを通じてゲートウェイに接続している場合は、アイドル状態の Netlet 接続はプロキシタイムアウトによって切断されます。切断されないようにするには、このパラメータにプロキシタイムアウトより小さい値を指定してください。

▶ キープアライブ間隔を設定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。
5. 「表示」ドロップダウンリストから「サービス」を選択します。
6. 「SRA 設定」の下の「Netlet」の隣にある矢印をクリックします。
右の区画に「Netlet」ページが表示されます。
7. 「キープアライブ間隔 (分)」フィールドまでスクロールし、適切な時間を入力します。
8. 「保存」をクリックし、変更内容を記録します。

「Portal のログアウト時に Netlet を終了」オプションの設定

ユーザーが Portal Server をログアウトしたときにすべての接続を終了させるときは、このオプションを有効にします。これにより、セキュリティが向上します。デフォルトでは、このオプションは有効に設定されています。

このオプションを無効にすると、ユーザーが Portal Server デスクトップからログアウトした後も、有効な Netlet 接続が持続します。

注 このオプションを無効にしても、Portal Server からログアウトしたユーザーは Netlet 接続を新たに確立できません。既存の接続が持続するだけです。

▶ 「ポータルログアウト時に Netlet を終了」オプションを設定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。
5. 「表示」ドロップダウンリストから「サービス」を選択します。
6. 「SRA 設定」の下の「Netlet」の隣にある矢印をクリックします。
右の区画に「Netlet」ページが表示されます。
7. 「ポータルログアウト時に Netlet を終了」フィールドまでスクロールし、必要に応じてチェックボックスを選択または選択解除します。
8. 「保存」をクリックし、変更内容を記録します。

206 ページの「Sun Ray 環境での Netlet の実行」も参照してください。

Netlet ルールへのアクセスの定義

特定の組織、ロール、ユーザーに対して特定の Netlet ルールへのアクセスを定義できます。

▶ Netlet ルールへのアクセスを定義するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。
5. 「表示」ドロップダウンリストから「サービス」を選択します。
6. 「SRA 設定」の下の「Netlet」の隣にある矢印をクリックします。
右の区画に「Netlet」ページが表示されます。
7. 「Netlet ルールにアクセス」フィールドまでスクロールします。
8. 「Netlet ルールにアクセス」フィールドで、選択している組織が使用できるようにするルールの名前を入力します。

このフィールドにアスタリスク (*) を指定すると、選択している組織は、定義されているすべての Netlet ルールを使用できるようになります。

9. 「追加」をクリックします。
指定したルールが「Netlet ルールにアクセス」リストに追加されます。
10. 使用可能にする各 Netlet ルールについて、手順 7、8、9 を繰り返します。
11. 「保存」をクリックし、変更内容を記録します。

Netlet ルールへのアクセスの拒否

特定の組織、ロール、ユーザーに対して特定の Netlet ルールへのアクセスを拒否できます。

▶ Netlet ルールへのアクセスを拒否するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。
5. 「表示」ドロップダウンリストから「サービス」を選択します。
6. 「SRA 設定」の下の「Netlet」の隣にある矢印をクリックします。
右の区画に「Netlet」ページが表示されます。
7. 「Netlet ルールの拒否」フィールドまでスクロールします。
8. 「Netlet ルールの拒否」フィールドで、選択している組織がアクセスを拒否されるルールの名前を入力します。

このフィールドにアスタリスク (*) を指定すると、選択している組織は、定義されているすべての Netlet ルールへのアクセスが拒否されるようになります。
9. 「追加」をクリックします。
指定したルールが「Netlet ルールの拒否」リストに追加されます。
10. アクセスを拒否する各 Netlet ルールについて、手順 7、8、9 を繰り返します。
11. 「保存」をクリックし、変更内容を記録します。

ホストへのアクセスの許可

特定の組織、ロール、ユーザーに対して特定のホストへのアクセスを定義できます。この定義により、特定のホストへのアクセスを許可できます。たとえば、ユーザーが telnet 接続する 5 つのホストを「許可」リストに設定できます。

▶ ホストへのアクセスを許可するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。
5. 「表示」ドロップダウンリストから「サービス」を選択します。
6. 「SRA 設定」の下の「Netlet」の隣にある矢印をクリックします。
右の区画に「Netlet」ページが表示されます。
7. 「許可されたホスト」フィールドまでスクロールします。
8. 「許可されたホスト」フィールドに、アクセスを許可するホストの名前を入力します。

このフィールドにアスタリスク (*) を指定すると、指定されたドメインのすべてのホストへのアクセスが可能になります。たとえば、*.sesta.com と指定した場合、ユーザーは sesta.com ドメイン内のすべての Netlet ターゲットを実行できます。また、xxx.xxx.xxx.* のように、ワイルドカードを含む IP アドレスも指定できます。
9. 「追加」をクリックします。
指定したホストが「許可されたホスト」リストに追加されます。
10. アクセス可能にする各ホストについて、手順 7 と 8 を繰り返します。
11. 「保存」をクリックし、変更内容を記録します。

ホストへのアクセスの拒否

組織内の特定のホストへのアクセスを拒否することができます。アクセスを拒否するホストを「拒否されたホスト」リストに指定します。

▶ ホストへのアクセスを拒否するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」 ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。
5. 「表示」 ドロップダウンリストから「サービス」を選択します。
6. 「SRA 設定」の下の「Netlet」の隣にある矢印をクリックします。
右の区画に「Netlet」ページが表示されます。
7. 「拒否されたホスト」フィールドまでスクロールします。
8. アクセスを拒否するホストの名前を「拒否されたホスト」フィールドに入力します。

このフィールドにアスタリスク (*) を指定すると、ユーザーは選択している組織内のすべてのホストにアクセスできなくなります。たとえば、組織 *sesta* のすべてのホストへのアクセスを拒否するには、「拒否されたホスト」フィールドに `*.sesta.com` と入力します。

特定のホストへのアクセスを拒否するには、完全修飾名を指定します。たとえば、ホスト *abc* へのアクセスを拒否する場合は、`abc.sesta.com` と入力します。

9. 「追加」をクリックします。
指定したドメインが「拒否されたホスト」リストに追加されます。
10. アクセス可能にする各ドメインについて、手順 7 と 8 を繰り返します。
11. 「保存」をクリックし、変更内容を記録します。

プロキシの設定

次の属性は、ユーザーレベルで設定できます。

- ブラウザのプロキシタイプ
- ブラウザのプロキシホスト
- ブラウザのプロキシポート
- ブラウザのプロキシ無効化リスト

管理コンソールでこれらの値を指定していないため、Netlet がブラウザのプロキシ設定を判断できない場合は、最初に Netlet を通じて接続が確立されるときに、この情報の入力を要求するプロンプトが表示されます。入力した情報は格納され、そのユーザーが次回以降に接続するときに使用されます。

次の場合には、Netlet はブラウザのプロキシ設定を判断できません。

- ユーザーが Java プラグイン (1.4.0 より前のバージョン) を使用する Internet Explorer 4.x、5.x、または 6.x を使用し、Java プラグインコントロールパネルの「プロキシ」タブで「ブラウザ設定を使用」オプションを有効にし、Internet Explorer の「ローカルエリアネットワーク (LAN) の設定」ダイアログの「自動構成スクリプトを使用する」フィールドで追加製品または INS ファイルを指定している場合。
- ユーザーが Java プラグイン (Version 1.3.1_01 以降) を使用する Netscape 6.2 を使用し、Java プラグインコントロールパネルの「プロキシ」タブで「ブラウザ設定を使用」オプションを有効にしている場合。

いずれの場合も、Netlet はブラウザ設定を特定できない場合があります、次の情報の指定がユーザーに求められます。

- ブラウザのプロキシタイプ
この属性は値 DIRECT または MANUAL です。ドロップダウンリストから DIRECT を選択すると、Netlet はゲートウェイホストに直接接続します。
- ブラウザのプロキシホスト
Netlet の接続で経由する必要があるプロキシホストを指定します。
- ブラウザのプロキシポート
Netlet の接続で経由する必要があるプロキシホストのポートを指定します。
- ブラウザのプロキシ無効化リスト (カンマ区切り)
プロキシを通じた Netlet 接続を必要としないホストを指定します。このリストには、複数のホスト名をカンマ区切りで指定できます。

プロキシレットの設定

この章では、Sun Java™ System Access Manager 管理コンソールからプロキシレットを設定する方法について説明します。

注 SRA のすべての属性について簡単に調べるには、Access Manager 管理コンソールの右上に表示される「ヘルプ」をクリックし、「Secure Remote Access 管理ヘルプ」をクリックします。

プロキシレットの設定

ユーザーがログインしたときにプロキシレットが自動的に起動するようにプロキシレットを設定するには、「プロキシレットのチャンネルを編集する」ページの「プロキシレットアプレットを自動的にダウンロード」チェックボックスにチェックマークを付けます。「プロキシレットアプレットを自動的にダウンロード」チェックボックスにチェックマークが付いていない場合には、標準のポータルデスクトップのプロキシレットチャンネルの「プロキシレットの起動」をクリックすれば、必要に応じてプロキシレットを取得できます。

▶ プロキシレットの属性を設定するには

1. Access Manager 管理コンソールに管理者としてログインします。
2. 「アイデンティティ管理」タブを選択します。
3. 「表示」ドロップダウンリストから「組織」を選択します。
4. 目的の組織名をクリックします。選択した組織名が、管理コンソールの左上に場所として表示されます。
5. 「表示」ドロップダウンリストから「サービス」を選択します。
6. 「SRA 設定」の「プロキシレット」の隣の矢印をクリックします。
7. 選択した組織のユーザーに適した競合の解決レベルを選択します。

8. 必要に応じて、「プロキシレットアプレットを自動的にダウンロード」チェックボックスをクリックします。
9. 「デフォルトのプロキシレットアプレットバインド IP」アドレス (プロキシレットが動作する場所) を入力します。
10. プロキシレットが待機するポート番号 (必須) を「プロキシレットアプレットのデフォルトのポート」フィールドに入力します。この設定はクライアントレベルの編集ウィンドウでも行うことができます。
11. 「保存」をクリックします。

注 Portal Server にログインしてプロキシレットを起動してから、Java プラグインをインストールした場合は、Netscape ブラウザを再起動する必要があります。

注 プロキシレットが有効になっている場合は、gateway.httpsurl エントリではなく、platform.conf ファイルの gateway.httpurl を使用します。

SSL アクセラレータの設定

この章では、Sun Java™ System Portal Server Secure Remote Access の各種アクセラレータを設定する方法について説明します。

この章で説明する内容は次のとおりです。

- [Sun Crypto Accelerator 1000](#)
- [Sun Crypto Accelerator 4000](#)
- [外部 SSL デバイスとプロキシアクセラレータ](#)

概要

外部アクセラレータは、SSL 機能をサーバーの CPU からオフロードする専用のハードウェアプロセッサです。これを使用することで、CPU は別のタスクを実行できるようになるので、SSL トランザクションの処理速度が向上します。

Sun Crypto Accelerator 1000

Sun™ Crypto Accelerator 1000 (Sun CA1000) ボードは、公開鍵と対称暗号化を実行する暗号化コプロセッサとして機能するショート PCI ボードです。この製品には外部インタフェースがありません。ボードは内部 PCI バスインタフェースを通じてホストと対話します。このボードの目的は、電子商取引アプリケーションのセキュリティプロトコルのために、計算を中心とするさまざまな暗号化アルゴリズムを高速化することです。

RSA [7] や Triple-DES (3DES) [8] など、多くの重要暗号化機能がアプリケーションから Sun CA1000 にオフロードされ、並行処理されます。これにより、CPU を他のタスクに振り分けられるようになり、SSL トランザクションの処理速度が向上します。

Sun Crypto Accelerator 1000 の有効化

Portal Server Secure Remote Access がインストールされていること、およびゲートウェイサーバー証明書 (自己署名した、または任意の CA が発行した証明書) がインストールされていることを確認します。詳細については、[211 ページの第 7 章「証明書」](#)を参照してください。

表 13-1 は、SSL アクセラレータをインストールする前に、必要な情報を確認するためのチェックリストです。このリストには、Crypto Accelerator 1000 のパラメータと値が示されています。

表 13-1 Crypto Accelerator 1000 のインストールチェックリスト

パラメータ	値
SRA インストールのベースディレクトリ	/opt
SRA の証明書データベースへのパス	/etc/opt/SUNWps/cert/default
SRA サーバー証明書のニックネーム	server-cert
レルム	sra-keystore
レルムユーザー	crypta

Sun Crypto Accelerator 1000 の設定

► Sun Crypto Accelerator 1000 を設定するには

1. ユーザーガイドの指示に従って、ハードウェアをインストールします。次の情報を参照してください。
<http://www.sun.com/products-n-solutions/hardware/docs/pdf/816-2450-11.pdf>
2. CD から次のパッケージをインストールします。
SUNWcryptm、SUNWcrypu、SUNWcrysu、SUNWdcar、SUNWcrypr、SUNWcrysl、SUNWdcamn、SUNWdcav
3. <http://sunsolve.sun.com> から入手できる次のパッチをインストールします。
110383-01, 108528-05, 112438-01
4. pk12util および modutil というツールがインストールされていることを確認します。

これらのツールは /usr/sfw/bin の下にインストールされます。ツールが /usr/sfw/bin ディレクトリにない場合は、Sun Java System の配布メディアから SUNWtisu パッケージを手動で追加する必要があります。

```
Solaris_[sparc/x86]/Product/shared_components/
```

5. スロットファイルを作成します。

```
vi /etc/opt/SUNWconn/crypto/slots
```

このファイルの唯一の行として、「crypta@sra」を入力します。

6. レalmを作成し、設定します。

- a. root としてログインします。

- b. 次のコマンドを入力します。

```
cd /opt/SUNWconn/bin/secadm
```

```
secadm> create realm=sra
```

```
Realm sra created successfully.
```

7. ユーザーを作成します。

- a. 次のコマンドを入力し、問い合わせに回答します。

```
secadm> set realm=sra
```

```
secadm{srap}> su
```

```
secadm{root@sra}>create user=crypta
```

```
Initial password:
```

```
Confirm password:
```

```
User crypta created successfully.
```

8. 作成したユーザーとしてログインします。

```
secadm{root@sra}> login user=crypta
```

```
Password:
```

```
secadm{crypta@sra}> show key
```

```
No keys exist for this user.
```

9. Sun Crypto モジュールをロードします。

環境変数 LD_LIBRARY_PATH が /usr/lib/mps/secv1/ をポイントする必要があります。

次のように入力します。

```
modutil -dbdir /etc/opt/SUNWps/cert/default -add "Sun Crypto  
Module" -libfile /opt/SUNWconn/crypto/lib/libpkcs11.so
```

次のコマンドを実行して、このモジュールがロードされたことを確認します。

```
modutil -list -dbdir /etc/opt/SUNWps/cert/default
```

10. 次のコマンドを実行し、ゲートウェイ証明書と鍵を「Sun Crypto モジュール」にエクスポートします。

環境変数 LD_LIBRARY_PATH が /usr/lib/mps/secv1/ をポイントする必要があります。

次のように入力します。

```
pk12util -o servercert.pl2 -d /etc/opt/SUNWps/cert/default -n server-cert
```

```
pk12util -i servercert.pl2 -d /etc/opt/SUNWps/cert/default -h "crypta@sra"
```

次に、show key コマンドを実行します。

```
secadm{crypta@sra}> show key
```

このユーザーの 2 つの鍵が表示されます。

11. /etc/opt/SUNWps/cert/default/.nickname ファイルでニックネームを変更します。

```
vi /etc/opt/SUNWps/cert/default/.nickname
```

server-cert を crypta@sra:server-cert に置き換えます。

12. 高速化する暗号化方式を有効化します。

Sun CA1000 は RSA 機能をアクセラレートしますが、アクセラレーションがサポートされる暗号化方式は DES と 3DES だけです。

13. /etc/opt/SUNWps/platform.conf.gateway-profile-name を変更してアクセラレータを有効化します。

```
gateway.enable.accelerator=true
```

14. 端末ウィンドウから、次のコマンドを指定してゲートウェイを再起動します。

```
portal-server-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

注 ゲートウェイは、ゲートウェイプロファイルの HTTPS ポートとして指定されているポートで、プレーンサーバーソケット (非 SSL) にバインドします。

着信するクライアントトラフィックに対して、非 SSL 暗号化または復号化が行われます。この処理は、アクセラレータ側で行われます。

このモードでは、PDC は機能しません。

Sun Crypto Accelerator 4000

Sun™ Crypto Accelerator 4000 ボードは、ギガビット Ethernet ベースのネットワーク インタフェースカードで、Sun サーバーでの IPsec および SSL (どちらも対称および非対称) の暗号化ハードウェアアクセラレーションをサポートします。

暗号化されていないネットワークトラフィックの標準ギガビットイーサネットネットワーク インタフェースカードとして機能するほかに、このボードには、暗号化された IPsec トラフィックのスループット向上をサポートする暗号化ハードウェアも含まれます。

Crypto Accelerator 4000 ボードは、ハードウェアとソフトウェアの両方の暗号化アルゴリズムをアクセラレートします。また、DES および 3DES 暗号化方式の一括暗号化もサポートします。

Sun Crypto Accelerator 4000 の有効化

SRA がインストールされていること、およびゲートウェイサーバー証明書 (自己署名した、または任意の CA が発行した証明書) がインストールされていることを確認します。SSL アクセラレータをインストールする前に、次のチェックリストに基づいて必要な情報を入手してください。

表 13-2 は、Crypto Accelerator 4000 のパラメータと値を示しています。

表 13-2 Crypto Accelerator 4000 のインストールチェックリスト

パラメータ	値
Portal Server Secure Remote Access インストールのベースディレクトリ	/opt
SRA インスタンス	デフォルト
SRA の証明書データベースへのパス	/etc/opt/SUNWps/cert/default
SRA サーバー証明書のニックネーム	server-cert
CA4000 キーストア	srap
CA4000 キーストアユーザー	crypta

Sun Crypto Accelerator 4000 の設定

▶ Sun Crypto Accelerator 4000 を設定するには

1. ユーザーガイドの指示に従って、ハードウェアとソフトウェアパッケージをインストールします。次の情報を参照してください。

<http://www.sun.com/products-n-solutions/hardware/docs/pdf/816-2450-11.pdf>

2. <http://sunsolve.sun.com> から入手できる次のパッチをインストールします:
114795
3. certutil, pk12util および modutil というツールがインストールされていることを確認します。

これらのツールは /usr/sfw/bin の下にインストールされます。

ツールが /usr/sfw/bin ディレクトリにない場合は、

Sun Java System の配布メディアから SUNWtisu パッケージを手動で追加する必要があります。

Solaris_[sparc/x86]/Product/shared_components/

4. ボードを初期化します。

/opt/SUNWconn/bin/vcadm ツールを実行して Crypto ボードを初期化し、次の値を設定します。

Initial Security Officer Name: sec_officer

Keystore name: sra-keystore

Run in FIPS 140-2 Mode: No

5. ユーザーを作成します。
vcaadm{vca0@localhost, sec_officer}> create user

New user name: crypta

Enter new user password:

Confirm password:

User crypta created successfully.

6. キーストアにトークンをマッピングします。
vi /opt/SUNWconn/cryptov2/tokens
次に、このファイルに sra-keystore を追加します。
7. 一括暗号化を有効にします。

```
touch /opt/SUNWconn/cryptov2/sslreg
```

8. Sun Crypto モジュールをロードします。

環境変数 `LD_LIBRARY_PATH` が `/usr/lib/mps/secv1/` をポイントする必要があります。

次のように入力します。

```
modutil -dbdir /etc/opt/SUNWps/cert/default -add "Sun Crypto Module" -libfile /opt/SUNWconn/cryptov2/lib/libvpkcs11.so
```

次のコマンドを実行することで、このモジュールがロードされたことを確認できます。

```
modutil -list -dbdir /etc/opt/SUNWps/cert/default
```

9. 次のコマンドを実行し、ゲートウェイ証明書と鍵を「Sun Crypto モジュール」にエクスポートします。

環境変数 `LD_LIBRARY_PATH` が `/usr/lib/mps/secv1/` をポイントする必要があります。

```
pk12util -o servercert.pl2 -d /etc/opt/SUNWps/cert/default -n server-cert
```

```
pk12util -i servercert.pl2 -d /etc/opt/SUNWps/cert/default -h "sra-keystore"
```

次のコマンドを実行することで、鍵がエクスポートされたことを確認できます。

```
certutil -K -h "sra-keystore" -d /etc/opt/SUNWps/cert/default
```

10. `/etc/opt/SUNWps/cert/default/.nickname` ファイルでニックネームを変更します。

```
vi /etc/opt/SUNWps/cert/default/.nickname
```

`server-cert` を `sra-keystore:server-cert` に置き換えます。

11. 高速化する暗号化方式を有効化します。

[267 ページの「SSL 符号化選択の有効化」](#) を参照してください。

12. 端末ウィンドウから、次のコマンドを指定してゲートウェイを再起動します。

```
portal-server-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

ゲートウェイは、キーストアのパスワードを要求します。

"sra-keystore":crypta:crytpa-password のパスワードまたは Pin を入力します。

- 注** ゲートウェイは、ゲートウェイプロファイルの HTTPS ポートとして指定されているポートで、プレーンサーバーソケット (非 SSL) にバインドしません。
- 着信するクライアントトラフィックに対して、非 SSL 暗号化または復号化が行われます。この処理は、アクセラレータ側で行われます。
- このモードでは、PDC は機能しません。

外部 SSL デバイスとプロキシアクセラレータ

オープンモードの Secure Remote Access (SRA) の前段で外部 SSL デバイスを実行できます。これは、クライアントと SRA の間に SSL リンクを提供します。

外部 SSL デバイスアクセラレータの有効化

▶ 外部 SSL デバイスアクセラレータの有効化

1. SRA がインストールされ、ゲートウェイがオープンモード (HTTP モード) で稼働していることを確認します。
2. HTTP 接続を有効にします。247 ページの「[HTTP 基本認証の有効化](#)」を参照してください。

表 13-3 は、外部 SSL デバイスとプロキシアクセラレータのパラメータと値を示しています。

表 13-3 外部 SSL デバイスとプロキシアクセラレータのチェックリスト

パラメータ	値
SRA インスタンス	デフォルト
ゲートウェイのモード	http
ゲートウェイのポート	880
外部デバイス / プロキシのポート	443

外部 SSL デバイスアクセラレータの設定

▶ 外部 SSL デバイスアクセラレータを設定するには

1. ユーザーガイドの指示に従って、ハードウェアとソフトウェアパッケージをインストールします。
2. 必須のパッチがあれば、それをインストールします。
3. HTTP を使用するために、ゲートウェイインスタンスを設定します。
4. `platform.conf` ファイルに次の値を入力します。

```
gateway.enable.customurl=true
gateway.enable.accelerator=true
gateway.httpurl=https://external-device-URL:port-number
```

5. ゲートウェイ通知は、次の 2 つの方法で設定できます。
 - Access Manager がポート 880 でゲートウェイマシンにアクセスできる場合 (セッション通知の形式は HTTP) は、`platform.conf` ファイルに次の値を入力します。

```
vi /etc/opt/SUNWps/platform.conf.default
gateway.protocol=http
gateway.port=880
```

- Access Manager がポート 443 で外部デバイス / プロキシにアクセスできる場合 (セッション通知の形式は HTTPS) は、`platform.conf` ファイルに次の値を入力します。

```
vi /etc/opt/SUNWps/platform.conf.default
gateway.host=External Device/Proxy Host Name
gateway.protocol=https
gateway.port=443
```

6. SSL デバイス、プロキシが稼動し、トラフィックがゲートウェイポートにトンネルされるように設定されたことを確認します。
7. 端末ウィンドウから、次のコマンドを指定してゲートウェイを再起動します。

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```


ログファイル

次のファイルはデフォルトの `/var/opt/SUNWps/debug` ディレクトリに格納されるログファイルで、デバッグ情報などの情報が記録されます。

表 A-1 情報ファイルとデバッグファイル

ファイル名	内容
次のログファイルは、デフォルトディレクトリ <code>/etc/opt/SUNWam/debug/</code> の <code>AMConfig-instance-name.properties</code> ファイルのデバッグパラメータによって制御されます。Linux のパス名については、24 ページの「 Solaris と Linux のパス名の比較 」を参照してください。	
<code>amconsole</code>	Netfile、Netlet、および Gateway Admin ファイル
<code>srapNetFile</code>	NetFile 情報ファイル
<code>srapNetlet</code>	Netlet 情報ファイル
<code>srapProxylet</code>	プロキシレットの情報ファイル
次のログファイルは、デフォルトディレクトリ <code>/etc/opt/SUNWps/</code> の <code>platform.conf.gateway-profile-name</code> ファイルのデバッグパラメータ <code>gateway.debug</code> によって制御されます。Linux のパス名については、24 ページの「 Solaris と Linux のパス名の比較 」を参照してください。	
<code>srapGateway.gateway-profile-name</code>	ゲートウェイ情報
<code>Gateway_to_from_server.gateway-profile-name</code>	
<code>Gateway_to_from_browser.gateway-profile-name</code>	
<code>srapNetletProxy.gateway-profile-name</code>	
<code>srapRewriterProxy.gateway-profile-name</code>	
<code>rwproxy.log.rewriter-proxy-instance-name</code>	リライタープロキシの開始時刻と停止時刻
<code>nlproxy.log.netlet-proxy-instance-name</code>	Netlet プロキシの開始時刻と停止時刻

表 A-1 情報ファイルとデバッグファイル (続き)

ファイル名	内容
gateway.log.gateway.instance.name	ゲートウェイの開始時刻と停止時刻
	次のリライターファイルは、デフォルトディレクトリ /var/opt/SUNWam/config/ の AMConfig-instance-name.properties ファイルのデバッグパラメータによって制御されます。詳細については、137 ページの「デバッグログを使用した障害追跡」を参照してください。
RuleSetInfo	リライトに使用されたすべてのルールは、このファイルに記録されます。
Original Pages	<p>ページの URI、解決された URI (解決された URI がページ URI と異なる場合)、コンテンツの MIME、ページに適用されたルールセット、パーサー MIME、元のコンテンツが記録されます。</p> <p>このファイルには、パースに関連する具体的な error/warning/message も記録されません。</p> <p>message モードではすべての内容が記録され、warning モードと error モードではリライト時に発生した例外だけが記録されます。</p>
Rewritten Pages	<p>ページの URI、解決された URI (解決された URI がページ URI と異なる場合)、コンテンツの MIME、ページに適用されたルールセット、パーサー MIME、リライトされたコンテンツが記録されます。</p> <p>この情報は、デバッグモードを message に設定した場合にだけ記録されます。</p>
Unaffected Pages	このファイルには、変更されなかったページのリストが含まれます。
URIInfo Pages	<p>このファイルには、検出され、変換された URL が含まれます。コンテンツが元のデータと同じ状態で残されたすべてのページの詳細が記録されます。</p> <p>記録される詳細情報は ページの URI、MIME、符号化データ、リライト時に適用されたルールセットの ID、およびパーサー MIME です。</p>

設定属性

この付録では、Access Manager 管理コンソールを使用して各 Portal Server Secure Remote Access コンポーネントの「サービス設定」タブから設定できる、Sun Java™ System Portal Server Secure Remote Access の属性を説明します。

- [アクセスリストサービス](#)
- [ゲートウェイサービス](#)
- [NetFile サービス](#)
- [Netlet サービス](#)
- [プロキシレットサービス](#)

アクセスリストサービス

表 B-1 は、アクセスリストサービスの属性を示しています。

表 B-1 アクセスリストサービスの属性

属性	デフォルト値	説明
拒否される URL		エンドユーザーがゲートウェイを通じてアクセスできない URL のリスト。
許可される URL	*	エンドユーザーがゲートウェイを通じてアクセスできる URL のリスト。
シングルサインオンを無効にするホスト		リスト内のホストに対して、シングルサインオンを無効にします。
セッションごとのシングルサインオンを有効		セッションでのシングルサインオンを有効にします。

表 B-1 アクセスリストサービスの属性 (続き)

属性	デフォルト値	説明
許可される認証レベル	*	認証を信頼する程度を指定します。すべての認証レベルを許可するときは、アスタリスク(*)を入力します。認証レベルについては、『Access Manager 管理ガイド』を参照してください。

ゲートウェイサービス

「ゲートウェイサービス」をクリックすると、右の区画に新規プロファイルを作成するためのボタンと、すでに作成されているゲートウェイプロファイルのリストが表示されます。

「新規」をクリックすると、隣の区画に、新規ゲートウェイプロファイルの名前を入力するように表示されます。デフォルトテンプレートを使用するか、以前作成したゲートウェイプロファイルをテンプレートとして使用するかを選択するオプションがあります。

表示されているゲートウェイプロファイル名をクリックすると、タブのリストが表示されます。次のタブが表示されます。

- [コア](#)
- [プロキシ](#)
- [セキュリティ](#)
- [リライタ](#)
- [ロギング](#)

コア

表 B-2 は、ゲートウェイサービスのコア属性を示しています。

表 B-2 ゲートウェイサービスのコア属性

属性	デフォルト値	説明
HTTPS 接続を有効		HTTPS 接続を有効にします。
HTTPS ポート	443	HTTPS ポートを指定します。
HTTP 接続を有効	*	HTTP 接続を有効にします。
HTTP ポート	80	HTTP ポートを指定します。

表 B-2 ゲートウェイサービスのコア属性 (続き)

属性	デフォルト値	説明
リライタプロキシを有効	*	ゲートウェイとイントラネットの間の HTTP トラフィックをセキュリティ保護できます。このリライタプロキシとゲートウェイでは、同じゲートウェイプロファイルが使用されます。
リライタプロキシのリスト		リライタプロキシをリストします。リライタプロキシのインスタンスが複数存在する場合には、 <i>host-name:port</i> の形式で個別に詳細を入力します。
Netlet を有効	選択	TCP/IP アプリケーション (Telnet や SMTP など)、HTTP アプリケーション、同じポートを使用するすべてのアプリケーションをセキュリティ保護できます。
プロキシレットを有効	選択	クライアントマシン上でプロキシレットのダウンロードを有効にします。
Netlet プロキシを有効		クライアントからの安全なトンネルを、ゲートウェイを経由してイントラネット内の Netlet プロキシまで拡張することで、ゲートウェイとイントラネット間の Netlet トラフィックのセキュリティを補強します。Portal Server でアプリケーションを使用しない場合は、無効にします。
Netlet プロキシホスト		Netlet プロキシホストを <i>hostname:port</i> の形式でリストします。
Cookie 管理を有効		ユーザーがアクセスを許可されたすべての Web サイトに対して、ユーザーセッションを追跡および管理します。Portal Server ユーザーセッションを追跡するために Portal Server で使用される Cookie には、この設定は適用されません。
持続 HTTP 接続を有効	選択	ゲートウェイで HTTP の持続的接続を有効にし、Web ページのイメージやスタイルシートなどのすべてのオブジェクトにソケットが開かれないように設定することができます。
持続接続ごとの最大要求数	10	持続的接続 1 つあたりの要求数を指定します。

表 B-2 ゲートウェイサービスのコア属性 (続き)

属性	デフォルト値	説明
持続ソケット接続のタイムアウト	50	ソケットを閉じるまでに必要な時間を指定します。
回復時間に必要な正常なタイムアウト	20	ブラウザが要求を送信してからゲートウェイに到達するまでの猶予時間と、ゲートウェイが応答を送信してからブラウザが実際に受信するまでの時間を指定します。
ユーザーセッション Cookie を転送する URL		サーバーおよび CGI で、Portal Server の Cookie を受信し、API を使用してユーザーを特定することができます。
最大接続キュー	50	ゲートウェイが受け付ける最大同時接続数を指定します。
ゲートウェイタイムアウト (秒)	120	ゲートウェイがブラウザとの接続をタイムアウトするまでの時間を、秒単位で指定します。
最大スレッドプールサイズ	200	ゲートウェイスレッドプールで事前に作成できる最大スレッド数を指定します。
キャッシュされたソケットのタイムアウト	200	ゲートウェイが Portal Server との接続をタイムアウトするまでの時間を、秒単位で指定します。
Portal Server		<code>http://portal server name:port -number</code> の形式で Portal Server を指定します。ゲートウェイは要求を処理するために、リスト内の各 Portal Server にラウンドロビン式にアクセスを試みます。
サーバーの再試行間隔 (秒)	120	Portal Server、リライタプロキシ、Netlet プロキシがクラッシュやパフォーマンス低下で利用できなくなったために、起動しようとする要求を行う間隔を指定します。
外部サーバーの Cookie を格納		ゲートウェイで、サードパーティ製アプリケーション、またはゲートウェイ経由でアクセスするサーバーからの Cookie を格納、管理できます。

表 B-2 ゲートウェイサービスのコア属性 (続き)

属性	デフォルト値	説明
URL からセッションを取得		Cookie をサポートするかどうかに関係なく、セッション情報を URL の一部としてコード化します。ゲートウェイでは、クライアントのブラウザから送信されるセッション Cookie の代わりに、URL に含まれるこのセッション情報を使用して検証を行います。

プロキシ

表 B-3 は、ゲートウェイサービスのプロキシ属性を示しています。

表 B-3 ゲートウェイサービスのプロキシ属性

属性	デフォルト値	説明
プロキシを使用する		Web プロキシの使用を有効にします。
Web プロキシを使用する URL		「プロキシを使用する」オプションを無効にしている場合でも、ゲートウェイが「ドメインとサブドメインのプロキシ」リストの Web プロキシだけを使用して接続するのに必要な URL をリストします。
Web プロキシを使用しない URL		ゲートウェイが直接接続できる URL をリストします。
ドメインとサブドメインのプロキシ	iportal.com sun.com	特定のドメインの特定のサブドメインへのアクセスに使用するプロキシを指定します。
プロキシパスワードのリスト		プロキシサーバーが一部またはすべてのサイトへのアクセスに認証を要求する場合、指定されたプロキシサーバーでゲートウェイが認証されるために必要なサーバー名、ユーザー名、およびパスワードを指定します。
自動プロキシ設定サポートを有効		「ドメインとサブドメインのプロキシ」フィールドで渡された情報を無視するよう指定します。

表 B-3 ゲートウェイサービスのプロキシ属性 (続き)

属性	デフォルト値	説明
自動プロキシ設定ファイルの位置		PAC サポートで使用されるファイルの場所を指定します。
Web プロキシ経由の Netlet トンネリングを有効		クライアントから、ゲートウェイを通してイントラネット内の Web プロキシまでの、安全なトンネルを拡張します。

セキュリティ

表 13-4 は、ゲートウェイサービスのセキュリティ属性を示しています。

表 13-4 ゲートウェイサービスのセキュリティ属性

属性	デフォルト値	説明
HTTP 基本認証を有効	選択	ユーザー名とパスワードを保存します。ユーザーは BASIC で保護された Web サイトに再びアクセスするときに証明情報を再入力する必要はありません。
非認証 URL	/portal/desktop/images /amserver/login_images /portal/desktop/css /amserver/jss /amconsole/console/css /portal/searchadmin/console/js /amconsole/console/js /amserver/css	画像を格納したディレクトリのように、認証を必要としない URL を指定します。
証明書が有効なゲートウェイホスト		証明書が有効なゲートウェイホストをリストします。

表 13-4 ゲートウェイサービスのセキュリティ属性 (続き)

属性	デフォルト値	説明
40 ビット暗号化を許可		40 ビットの (弱い) SSL (Secure Sockets Layer) 接続を許可します。このオプションを選択していない場合、128 ビット接続だけがサポートされます。
SSL バージョン 2.0 を有効	選択	SSL バージョン 2.0 を有効にします。 SSL 2.0 を無効化すると、古い SSL 2.0 しかサポートしないブラウザは SRA に対して認証ができません。これにより、セキュリティのレベルが格段に向上します。
SSL 符号化選択を有効		SSL の暗号化方式を選択できるようにします。パッケージ内のすべての暗号化方式をサポートするか、必要な暗号化方式を個別に選択するかを選択することができます。ゲートウェイインスタンスごとに、個別に SSL 暗号化方式を選択できます。
SSL2 符号化		選択した SSL バージョン 2 の暗号化方式をリストします。
SSL3 符号化		選択した SSL バージョン 3 の暗号化方式をリストします。
TLS 符号化		TLS 暗号化方式をリストします。
SSL バージョン 3.0 を有効	選択	SSL バージョン 3.0 を有効にします。 SSL 3.0 を無効化すると、SSL 3.0 しかサポートしないブラウザは SRA に対して認証ができません。これにより、セキュリティのレベルが格段に向上します。
Null 暗号化方式を有効		Null 暗号化を有効にします。
信頼できる SSL ドメイン		信頼されている SSL ドメインをリストします。
安全な Cookie としてマークする		安全な Cookie としてマークします。「Cookie 管理を有効」オプションが有効である必要があります。

リライト

「リライト」タブは、さらに2つに分かれています。

- [基本](#)
- [詳細](#)

基本

表 B-4 は、ゲートウェイサービスのリライト基本属性を示しています。

表 B-4 ゲートウェイサービスのリライト属性 - 基本

属性	デフォルト値	説明
すべての URI のリライトを有効		「ドメインとサブドメインのプロキシ」リストのエントリをチェックせずに、すべての URL がリライトされます。
URI をルールセットにマップ	<pre>*//*.iportal.com*/portal/* default_gateway_ruleset */portal/NetFileOpenFileServlet* null_ruleset * generic_ruleset REPLACE_WITH_IPLANET_MAIL_SERVER_NAME iportal_net_mail_ruleset REPLACE_WITH_EXCHANGE_SERVER_NAME exchange_2000sp3_owa_ruleset *//*.iportal.com*/amconsole/* default_gateway_ruleset REPLACE_WITH_NOTES_SERVER_NAME inotes_ruleset http://*/portal/NetFileController* null_ruleset</pre>	「URI をルールセットにマップ」リストを使用して、ドメインとルールセットを関連付けます。ルールセットは、Access Manager 管理コンソールの「Portal Server 設定」で作成されます。

表 B-4 ゲートウェイサービスのリライト属性 - 基本 (続き)

属性	デフォルト値	説明
パーサーを MIME タイプにマップ	JAVASCRIPT=application/x-java XML=text/xml HTML=text/html;text/html;text/x-component;text/wml;text/vnd.wap.wml CSS=text/css	新規 MIME タイプを HTML、JAVASCRIPT、CSS、または XML に関連付けます。複数のエントリは、セミコロンまたはカンマで区切ります。
リライトしない URI		リライトしない URI をリストします。 注：このリストに #* を追加することで、href ルールがルールセットの一部である場合でも URI をリライトできます。
デフォルトのドメイン		ホスト名をデフォルトのドメインおよびサブドメインに解決します。これは、インストール時に指定されます。

詳細

表 B-5 は、ゲートウェイサービスのリライト詳細属性を示しています。

表 B-5 ゲートウェイサービスのリライト属性 - 詳細

属性	デフォルト値	説明
MIME 推測を有効		MIME が送信されないときの MIME 推測機能を有効にします。「パーサーと URI のマッピング」リストボックスにデータを追加する必要があります。
パーサーと URI のマッピング		パーサーと URI をマッピングします。複数の URI はセミコロンで区切られます。 たとえば、HTML=*.html;*.htm;*Servlet のように指定します。 この例の設定では、リライトは拡張子が html、htm、Servlet のすべてのページのコンテンツをリライトします。
マスキングを有効		リライトはページのイントラネット URL が判読されないように URI をリライトします。

表 B-5 ゲートウェイサービスのリライト属性 - 詳細 (続き)

属性	デフォルト値	説明
マスキングのシード文字列		URI のマスキングに使用するシード文字列を指定します。マスキングアルゴリズムにより、このランダム文字列が生成されます。
マスクしない URI		<p>マスクしないインターネット URI を指定します。アプリケーション (アプレットなど) がインターネット URI を要求するときに使用します。</p> <p>たとえば、次のように追加します。</p> <p>*/Applet/Param*</p> <p>リストボックスに追加した URL は、コンテンツの URI http://abc.com/Applet/Param1.html がルールセット内のルールと一致する場合にマスクされません。</p>
ゲートウェイプロトコルを元の URI プロトコルと同じにする		<p>HTML コンテンツ内で参照されるリソースへのアクセスに、リライトは同じプロトコルを使用できます。</p> <p>これは、スタティックな URI だけに適用され、JavaScript によって生成されるダイナミック URI には適用されません。</p>

ロギング

表 B-6 は、ゲートウェイサービスのロギング属性を示しています。

表 B-6 ゲートウェイサービスロギングの属性

属性	デフォルト値	説明
ロギングを有効		ロギングを有効化します。
セッション単位のロギングを有効		クライアントアドレス、要求タイプ、宛先ホストなどの最低限のログ情報を取り込めるようにします。

表 B-6 ゲートウェイサービスログギングの属性 (続き)

属性	デフォルト値	説明
セッション単位の詳細なログギングを有効		クライアント、要求タイプ、宛先ホスト、要求のタイプ、クライアント要求 URL、クライアントポストデータサイズ、セッション ID、応答結果コード、完全応答サイズなどの詳細情報を取り込めるようにします。 注：「セッション単位のログギングを有効」を有効にする必要があります。
Netlet ログギングを有効		ログギングを有効にする場合に指定します。その場合、開始時刻、ソース、アドレス、ソースポート、サーバーアドレス、サーバーポート、停止時刻、状態 (起動または停止) の各情報が取り込まれます。

NetFile サービス

「NetFile サービス」をクリックすると、右の区画にタブが表示されます。次のタブが表示されます。

- [ホスト](#)
- [権限](#)
- [表示](#)
- [操作](#)
- [一般](#)

ホスト

「ホスト」タブは、さらに2つに分かれています。

- [設定](#)
- [アクセス](#)

設定

表 B-7 は、NetFile サービスのホスト設定属性を示しています。

表 B-7 NetFile サービスのホスト設定属性

属性	デフォルト値	説明
OS 文字セット	Unicode (UTF-8)	ホストとの対話にデフォルトエンコーディングとして使用する文字セットを指定します。
ホスト検出順序	WIN、 NETWARE、 FTP、NFS	ホストの検出順序を指定します。
共通ホスト		すべてのリモート NetFile ユーザーが NetFile を通じて使用できるホストを指定します。
デフォルトドメイン		NetFile が許可されたホストへのアクセスに使用するデフォルトドメインを指定します。
デフォルトの Microsoft Windows ドメイン / ワークグループ		ユーザーが Microsoft Windows ホストにアクセスするときに使用する、デフォルトの Microsoft Windows ドメインまたはワークグループを指定します。
デフォルトの WINS/DNS サーバー		Microsoft Windows ホストへのアクセスで NetFile が使用する WINS/DNS サーバーを指定します。

アクセス

表 B-8 は、NetFile サービスのホストアクセス属性を示しています。

表 B-8 NetFile サービスのホストアクセス属性

属性	デフォルト値	説明
Microsoft Windows ホストへのアクセスを許可	選択	Microsoft Windows ホストにアクセスできるようにします。
FTP ホストへのアクセスを許可	選択	FTP ホストにアクセスできるようにします。
NFS ホストへのアクセスを許可	選択	NFS ホストにアクセスできるようにします。
Netware ホストへのアクセスを許可	選択	Netware ホストにアクセスできるようにします。
許可されるホスト	*	NetFile を通じてユーザーがアクセスできるホストを指定します。
拒否されるホスト		NetFile を通じてユーザーがアクセスできないホストを指定します。

権限

ユーザーが NetFile の使用を開始した後にこのオプションを無効にすると、ユーザーがログアウトし再びログインした後に変更内容が有効になります。

表 B-9 は、NetFile サービスの権限属性を示しています。

表 B-9 NetFile サービスの権限属性

属性	デフォルト値	説明
ファイル名の変更を許可	選択	ユーザーがファイル名を変更できるようにします。
ファイル / フォルダの削除を許可	選択	ユーザーがファイルおよびフォルダを削除できるようにします。
ファイルアップロードを許可	選択	ユーザーがファイルをアップロードできるようにします。
ファイル / フォルダのダウンロードを許可	選択	ユーザーがファイルおよびフォルダをダウンロードできるようにします。
ファイル検索を許可	選択	ユーザーが検索できるようにします。
ファイルのメール送信を許可	選択	ファイルをメール送信できるようにします。

表 B-9 NetFile サービスの権限属性 (続き)

属性	デフォルト値	説明
ファイルの圧縮を許可	選択	ファイルを圧縮できるようにします。
ユーザー ID の変更を許可	選択	ユーザーが別の ID を使用できるようにします。
Microsoft Windows ドメインの変更を許可	選択	ユーザーが Microsoft Windows ドメインを変更できるようにします。

表示

表 B-10 は、NetFile サービスの表示属性を示しています。

表 B-10 NetFile サービスの表示属性

属性	デフォルト値	説明
ウィンドウサイズ	700 400	ユーザーのデスクトップの NetFile ウィンドウのサイズを、ピクセル単位で指定します。無効な値を入力した場合、NetFile はデフォルトの値を使用します。
ウィンドウの位置	100 50	NetFile ウィンドウがユーザーのデスクトップに表示される位置を指定します。無効な値を入力した場合、NetFile はデフォルトの値を使用します。

操作

「操作」タブは、さらに次のように分かれています。

- [トラフィック](#)
- [検索](#)
- [圧縮](#)

トラフィック

表 B-11 は、NetFile サービスの操作トラフィック属性を示しています。

表 B-11 NetFile サービスの操作トラフィック属性

属性	デフォルト値	説明
一時ディレクトリの場所	/tmp	<p>NetFile のファイル操作で使用する一時ディレクトリを指定します。</p> <p>Web サーバーが実行時に使用する ID (nobody または noaccess) に、指定されたディレクトリに対するアクセス権 rwX が割り当てられていることを確認してください。また、要求される一時ディレクトリへの完全パスに対するアクセス権 rx が ID に割り当てられていることを確認してください。</p> <p>NetFile の一時ディレクトリを個別に作成する場合があります。Portal Server のすべてのモジュールに共通な一時ディレクトリを指定すると、ディスクの容量がすぐに足りなくなります。NetFile は一時ディレクトリの容量がなくなると機能しません。</p>
ファイルのアップロード制限 (M バイト)	5	<p>アップロードできるファイルの最大サイズを指定します。無効な値を入力すると、NetFile は値をデフォルト値にリセットします。整数値で指定する必要があります。</p> <p>ユーザーごとに異なるファイルアップロードサイズ制限を指定できます。</p>

検索

表 B-12 は、NetFile サービスの操作検索属性を示しています。

表 B-12 NetFile サービスの操作検索属性

属性	デフォルト値	説明
検索ディレクトリ制限	100	1 回の検索操作で検索できるディレクトリの最大数を指定します。

圧縮

表 B-13 は、NetFile サービスの操作圧縮属性を示しています。

表 B-13 NetFile サービスの操作圧縮属性

属性	デフォルト値	説明
デフォルトの圧縮タイプ	Zip	圧縮のタイプとして Zip または Gzip を指定します。
デフォルトの圧縮レベル	6	圧縮のレベルを 1～9 の番号で指定します。

一般

表 B-14 は、NetFile サービスの一般属性を示しています。

表 B-14 NetFile サービスの一般属性

属性	デフォルト値	説明
MIME タイプ設定ファイルの場所	/opt/S1PS62/SUNWps/samples/config/netfile	クライアントブラウザに送信する応答コンテンツのタイプを指定します。

Netlet サービス

表 B-15 は、Netlet サービスの属性を示しています。

表 B-15 Netlet サービスの属性

属性	デフォルト値	説明
Netlet ルール		ルールを追加するか削除するかを選択します。
ルールを追加する場合は、次の 9 個の属性が必要です。		
-- ルール名		一意のルール名を指定します。
-- 暗号化方式		適切な暗号化方式を指定します。
-- URL		呼び出すアプリケーションの URL を指定します。

表 B-15 Netlet サービスの属性 (続き)

属性	デフォルト値	説明
-- アプレットのダウンロード		アプレットをダウンロードする必要があるかどうかを指定します。アプレットを使用する場合、関連する編集ボックスには次の構文で入力します。 <code>local-port:server-host:server-port</code>
-- 拡張セッション		このルールに対応する Netlet セッションの実行中は Portal Server セッション時間が延長されるようにします。
-- ローカルポートと宛先サーバーポートのマッピング		ローカルポート、ターゲットホスト、およびターゲットポートを指定します。これらの値 (この表の次の 3 項目) の入力後、「追加」をクリックすると、入力した値がリストに表示されます。
-- ローカルポート		Netlet が待機するローカルポートを指定します。FTP ルールでは、ローカルポートは 30021 である必要があります。
-- 宛先ホスト		スタティックルールの場合は、Netlet 接続での宛先マシンのホスト名。 ダイナミックルールの場合は、「TARGET」。
-- 宛先ポート		宛先ホスト上のポートを指定します。
デフォルトのネイティブ VM 暗号化方式		Netlet ルールのデフォルトの暗号化方式を指定します。これはルールの一部として暗号化方式が指定されていない既存のルールを使用する場合に便利です。
デフォルトの Java™ プラグイン暗号化方式		Netlet ルールのデフォルトの暗号化方式を指定します。これはルールの一部として暗号化方式が指定されていない既存のルールを使用する場合に便利です。
デフォルトのループバックポート	58000	Netlet を通じてアプレットがダウンロードされるときにクライアントで使用されるポートを指定します。デフォルト値は、Netlet ルール内で上書きできます。
接続の再認証		Netlet 接続を確立しようとするユーザーに、その都度 Netlet パスワードの入力を要求します。

表 B-15 Netlet サービスの属性 (続き)

属性	デフォルト値	説明
接続の警告ポップアップを表示	選択	ユーザーが Netlet でアプリケーションを実行する場合、または侵入者が待機ポートを通じてデスクトップにアクセスしようとしている場合に、メッセージを表示します。
ポート警告ダイアログにチェックボックスを表示	選択	Netlet がユーザーの標準ポータルデスクトップ上の宛先ホストに接続しようとしたときに、警告ダイアログポップアップの表示を抑制することができます。
キープアライブ間隔 (分)	0	クライアントが Web プロキシを通じてゲートウェイに接続している場合は、アイドル状態の Netlet 接続はプロキシタイムアウトによって切断されます。切断されないようにするには、このパラメータにプロキシタイムアウトより小さい値を指定してください。
ポータルのログアウト時に Netlet を終了	選択	ユーザーが Portal Server をログアウトしたときにすべての接続を終了するようにします。
Netlet ルールにアクセス	*	特定の組織、ロール、ユーザーに対して特定の Netlet ルールへのアクセスを定義します。
Netlet ルールの拒否		特定の組織、ロール、ユーザーに対して特定の Netlet ルールへのアクセスを拒否します。
許可されるホスト	*	特定の組織、ロール、ユーザーに対して特定のホストへのアクセスを定義します。
拒否されるホスト		組織内の特定のホストへのアクセスを拒否します。

プロキシレットサービス

表 B-16 は、プロキシレットサービスの属性を示しています。

表 B-16 プロキシレットサービスの属性

属性	デフォルト値	説明
プロキシレットアプレットを自動的にダウンロード		このチェックボックスにチェックマークが付いている場合には、ユーザーがログオンしたときに、クライアントマシンにプロキシレットがダウンロードされます。
プロキシレットアプレットのデフォルトのバインド IP	127.0.0.1	プロキシレットアプレットが存在する IP アドレス。
プロキシレットアプレットのデフォルトのポート	58080	プロキシレットが待機するポート。

国コード

次の表は、認証管理時に指定する2文字の国コードを示しています。

表 C-1 2文字の国コード(1 / 9)

ad	アンドラ公国
ae	アラブ首長国連邦
af	アフガニスタン
ag	アンティグアおよびバーブーダ
ai	アンギラ
al	アルバニア
am	アルメニア
an	オランダ領アンティル
ao	アンゴラ
aq	南極大陸
ar	アルゼンチン
arpa	旧 Arpanet
as	アメリカ領サモア
at	オーストリア
au	オーストラリア
aw	アルバ
az	アゼルバイジャン
ba	ボスニアヘルツェゴビナ
bb	バルバドス
bd	バングラデシュ

表 C-1 2文字の国コード(2/9)

be	ベルギー
bf	ブルキナファソ
bg	ブルガリア
bh	バーレーン
bi	ブルンジ
bj	ベニン
bm	バーミューダ
bn	ブルネイ
bo	ボリビア
br	ブラジル
bs	バハマ
bt	ブータン
bv	ブーベ島
bw	ボツワナ
by	ベラルーシ
bz	ベリーズ
ca	カナダ
cc	ココス諸島
cf	中央アフリカ共和国
cd	コンゴ民主共和国
cg	コンゴ
ch	スイス
ci	コートジボアール
ck	クック諸島
cl	チリ
cm	カメルーン
cn	中国
co	コロンビア
com	商用
cr	コスタリカ

表 C-1 2文字の国コード(3/9)

cs	旧チェコスロバキア
cu	キューバ
cv	カーボヴェルデ
cx	クリスマス諸島
cy	キプロス
cz	チェコ共和国
de	ドイツ
dj	ジブチ
dk	デンマーク
dm	ドミニカ
do	ドミニカ共和国
dz	アルジェリア
ec	エクアドル
edu	北米4年制大学
ee	エストニア
eg	エジプト
eh	西サハラ
er	エリトリア
es	スペイン
et	エチオピア
fi	フィンランド
fj	フィジー
fk	フォークランド諸島
fm	ミクロネシア
fo	フェロー諸島
fr	フランス
fx	フランス(欧州領域)
ga	ガボン
gb	イギリス
gd	グレナダ

表 C-1 2文字の国コード(4 / 9)

ge	グルジア
gf	仏領ギアナ
gh	ガーナ
gi	ジブラルタル
gl	グリーンランド
gm	ガンビア
gn	ギニア
gov	米国政府
gp	グアドループ(仏領)
gq	赤道ギニア
gr	ギリシャ
gs	サウスジョージア島、およびサウスサンドウィッチ島
gt	グアテマラ
gu	グアム(米国)
gw	ギニアビサオ
gy	ガイアナ
hk	香港
hm	ハードおよびマクドナルド諸島
hn	ホンジュラス
hr	クロアチア
ht	ハイチ
hu	ハンガリー
id	インドネシア
ie	アイルランド
il	イスラエル
in	インド
int	国際機関
io	英インド洋領
iq	イラク
ir	イラン

表 C-1 2文字の国コード(5/9)

is	アイスランド
it	イタリア
jm	ジャマイカ
jo	ヨルダン
jp	日本
ke	ケニア
kg	キルギス共和国(キルギスタン)
kh	カンボジア王国
ki	キリバス
km	コモロス
kn	セントクリストファーおよびネイビス
kp	北朝鮮
kr	韓国
kw	クウェート
ky	ケイマン諸島
kz	カザフスタン
la	ラオス
lb	レバノン
lc	セントルシア
li	リヒテンシュタイン
lk	スリランカ
lr	リベリア
ls	レソト
lt	リトアニア
lu	ルクセンブルク
lv	ラトビア
ly	リビア
ma	モロッコ
mc	モナコ
md	モルダビア

表 C-1 2文字の国コード(6 / 9)

mg	マダガスカル
mh	マーシャル諸島
mil	米軍
mk	マケドニア
ml	マリ
mm	ミャンマー
mn	モンゴル
mo	マカオ
mp	北マリアナ諸島
mq	マルチニーク (仏領)
mr	モーリタニア
ms	モントセラト
mt	マルタ
mu	モーリシャス
mv	モルジブ
mw	マラウイ
mx	メキシコ
my	マレーシア
mz	モザンビーク
na	ナミビア
nato	NATO (1996年に廃止、 hq.nato.int を参照)
nc	ニューカレドニア (仏領)
ne	ニジェール
net	ネットワーク
nf	ノーフォーク諸島
ng	ナイジェリア
ni	ニカラグア
nl	オランダ
no	ノルウェー
np	ネパール

表 C-1 2文字の国コード(7/9)

nr	ナウル
nt	中立地帯
nu	ニウエ
nz	ニュージーランド
om	オマーン
org	非営利組織 (sic)
pa	パナマ
pe	ペルー
pf	ポリネシア (仏領)
pg	パプアニューギニア
ph	フィリピン
pk	パキスタン
pl	ポーランド
pm	サンピエールおよびミクロン諸島
pn	ピトケルン諸島
pr	プエルトリコ
pt	ポルトガル
pw	パラウ
py	パラグアイ
qa	カタール
re	レユニオン (仏領)
ro	ルーマニア
ru	ロシア連邦
rw	ルワンダ
sa	サウジアラビア
sb	ソロモン諸島
sc	セーシェル
sd	スーダン
se	スウェーデン
sg	シンガポール

表 C-1 2文字の国コード(8/9)

sh	セントヘレナ島
si	スロベニア
sj	スヴァールバルおよびヤンマイエン諸島
sk	スロバキア共和国
sl	シエラレオネ
sm	サンマリノ
sn	セネガル
so	ソマリア
sr	スリナム
st	サントメおよびプリンシペ
su	旧ソビエト連邦
sv	エルサルバドル
sy	シリア
sz	スワジランド
tc	タークス諸島およびカイコ諸島
td	チャド
tf	フランス南方領
tg	トーゴ
th	タイ
tj	タジキスタン
tk	トケラウ
tm	トルクメニスタン
tn	チュニジア
to	トンガ
tp	東ティモール
tr	トルコ
tt	トリニダードおよびトバゴ
tv	ツバル
tw	台湾
tz	タンザニア

表 C-1 2文字の国コード(9/9)

ua	ウクライナ
ug	ウガンダ
uk	英国
um	米島嶼部(ミッドウェー、ジョンストン、ウェーク諸島)
us	米国
uy	ウルグアイ
uz	ウズベキスタン
va	教皇庁(バチカン市国)
vc	セントヴィンセント、およびグレナディン諸島
ve	ベネズエラ
vg	バージン諸島(英領)
vi	バージン諸島(米領)
vn	ベトナム
vu	バヌアツ
wf	ワリスフツナ諸島
ws	サモア
ye	イエメン
yt	マヨット
yu	ユーゴスラビア
za	南アフリカ
zm	ザンビア
zr	ザイール
zw	ジンバブエ

用語集

このドキュメンテーションセットで使用されているすべての用語の一覧については、『Java™ Enterprise System 用語集』(<http://docs.sun.com/doc/819-1933?l=ja>) を参照してください。

索引

A

AMConfig プロパティファイル, 52
デフォルト, 50

C

Calendar, 37
certadmin スクリプト, 218
chroot, 54
Citrix
html ファイル, 206
Communication Express, 37
Cookie
安全としてマーク, 257
外部の格納, 256
有効化, 246

D

DMZ, 30
DNS, 202

F

FTP
NetFile でのサポート, 180

H

hostproxy
作成, 60
HTML
リライタのルール, 103
HTTP
基本認証, 247
ヘッダー, 79
リソース、Web プロキシの使用, 61
リソースへのアクセス, 61
HTTP メタタグ, 94

I

iNotes, 37

J

Java Enterprise System アクセサリ CD
jchdt パッケージ, 94
SUNWrhino パッケージ, 67
JavaScript
リライタのルール, 110
Jcharset
PAC ファイルの使用, 67
jCIFS
Microsoft Windows アクセス, 181

NetFile でのサポート, 180

M

Messenger Express, 37

Microsoft Exchange Server, 203

Microsoft Windows

 jCIFS が必要, 181

 ドメイン, 292, 293

Microsoft Windows ドメイン

 指定, 292

MIME

 推測, 133, 278

 パースするタイプ, 276

MIME タイプ

 指定, 304

 リストの作成, 276

N

NetFile, 179

 Novell Netware の使用, 181

 ProFTPD サーバーの使用, 181

 アクセスの有効化, 181

 圧縮, 303

 アップロードサイズの制限, 301

 一時ファイルディレクトリ, 300

 ウィンドウサイズ, 298

 ウィンドウの位置, 299

 概要, 179

 共通ホストのリスト, 289

 サポートされるプロトコル, 180

 設定, 287

 デバッグ, 182

 ホスト検出順序, 180

 ホストへのアクセス, 293

 ホストへのアクセスの許可, 294

 ホストへのアクセスの拒否, 295

 ログイン, 182

NetFile サービス, 34

Netlet, 184

 PAC ファイルの使用, 67

 PDC 用の設定, 209

 Sun Ray 環境, 206

 Web プロキシ経由のトンネル化, 263

 アプレット, 184

 アプレットのダウンロードチェックボックス,
 307

 概要, 183

 キープアライブ間隔, 313

 コンポーネント, 184

 使用例, 186

 接続の再認証の有効化, 311

 設定, 305

 待機ポート, 184

 プロバイダ, 185

 ポート番号, 195

 ホストへのアクセス, 316

 ホストへのアクセスの拒否, 317

 有効化, 243

 リモートホストからのアプレットのダウンロード,
 187

 ルール, 185, 187

 ルールの追加, 307

 ログイン, 205, 284

 ログアウト時の終了, 313

Netlet サービス, 34

 ユーザーへの割り当て, 306

Netlet プロキシ, 185

 再起動, 74

 作成, 73

 使用, 70

 有効化, 74

 利点, 70

Netlet ルール, 308

 アクセスの拒否, 315

 アクセスの指定, 314

 削除, 309

 スタティックルール, 192

 ダイナミック, 192

 変更, 308

Netlet ルールの例

FTP, 204

IMAP, 201

Lotus Notes 非 Web クライアント, 202

Lotus Web クライアント, 201

Microsoft Outlook および Exchange Server, 203

Netscape 4.7 Mail Client, 204

SMTP, 201

Net ルール

サンプル, 200

NFS

NetFile でのサポート, 180

nlpmultiinstance スクリプト, 73

Novell Netware

NetFile のプロトコル, 181

O

OS 文字セット, 288

Outlook Web Access, 203

設定, 176

ルールセット, 176, 275

P

PAC

設定, 67

PAC ファイル

Rhino ソフトウェアの使用, 67

場所, 263

PDC

設定, 209, 270

認証, 212

認証の連鎖, 81

platform.conf, 42

プロパティ, 44

Portal Server

リストの作成, 254

ProFTPD

NetFile の使用, 181

R

Rhino ソフトウェア

PAC ファイルのパス, 67

rwpmultiinstance, 76

S

Secure Sockets Layer, 32

SMB

Microsoft Windows アクセス, 181

SMTP, 243

Solaris

サポート, 27

パッチ, 27

SRA

SRA コアへのアクセス, 60

サービス, 33

すべての属性のリスト, 333

ソフトウェア, 29

SSL, 212

SSL Version 3.0, 268

SSO, 237

SUNWjchdt パッケージ, 94

T

TCP/IP, 184, 243

Telnet, 243

U

UNIX

コマンド行, 34

URL

- Web プロキシのリストの作成, 259
- 許可される, 236
- 拒否される, 236
- セッションの取得, 256
- 動的 Netlet ルールによる呼出し, 198
- 非認証, 264
- リライトしない, 277
- リライトの有効化, 273

URL スクレイパー, 94

W

watchdog

- Netlet プロキシ, 74
- リライタプロキシ, 77

Web プロキシ, 61

- Netlet トンネリング, 263
- 有効化, 258

WINS/DNS サーバー

- 指定, 293

WML

- リライタのルール, 127

X

XML ルール

- リライタ, 124

あ

アクセス

- ホストへのアクセスの許可, 316
- ホストへのアクセスの拒否, 317

アクセスリスト

- 許可される URL, 236
- 拒否される URL, 236
- シングルサインオン, 237

アクセラレータ

Sun Crypto 1000, 321

Sun Crypto 4000, 325

外部 SSL デバイス, 328

プロキシ, 328

圧縮

- NetFile の, 303

アップロードサイズの制限

- NetFile の, 301

アプリケーション

- サポート, 37
- 実行, 183

アプレット, 184

- ダウンロード, 199

暗号化

- null, 269
- 管理者設定, 193
- サポート, 193
- 選択, 267
- デフォルト暗号化方式, 309
- ユーザー設定可能, 193

い

一時ファイルディレクトリ, 300

う

ウィンドウサイズ

- NetFile の, 298

ウィンドウの位置

- NetFile の, 299

お

オープンモード, 30

か

カスケードスタイルシート

リライタ, 127

カスタマイズ

ゲートウェイのユーザーインターフェース, 83

管理コンソール, 34

管理者設定暗号化方式, 193

き

キープアライブ間隔

設定, 313

起動

ゲートウェイ, 57

逆プロキシ, 78

有効化, 78

キャッシュされたソケットのタイムアウト, 254

競合の解決, 36

共通ホストのリスト

設定, 289

許可

40 ビットブラウザ接続, 266

許可される URL, 236

拒否

URL, 236

く

国コード

2 文字の値, 353

け

警告ポップアップダイアログボックス, 311

ケーススタディ

リライタ, 172

ゲートウェイ

chroot モード, 54

HTTPS モード, 241

HTTP モード, 241

PAC ファイルの使用, 67

URL からのセッションの取得, 256

概要, 39

起動, 57

ゲートウェイプロファイル, 40

再起動, 59

証明書が有効, 265

スレッドプールの指定, 253

接続の有効化, 241

設定, 239

タイムアウト, 252

停止, 58

マルチホーム, 52

ロギング, 283

ゲートウェイサービス, 33

検索

ディレクトリの制限, 302

こ

コンポーネント

Netlet, 184

さ

サーバーの再試行間隔, 255

サービス

SRA, 33

再起動

Netlet プロキシ, 74

ゲートウェイ, 59

リライタプロキシ, 77

最大要求数, 249

再認証

接続, 311

削除

- Netlet ルール , 309
- 作成
 - hostproxy, 60
 - Portal Server のリスト , 254
 - URI とルールセットのマッピングリスト , 274
 - ゲートウェイプロファイル , 40
 - 証明書が有効なゲートウェイホストのリスト , 265
 - 信頼されている SSL ドメインのリスト , 269
 - パーサーと URI のマッピングリスト , 133
 - パースするマッピングのリスト , 279
 - 非認証 URL のリスト , 264
 - マスクしない URI のリスト , 281
 - リライトプロキシ , 76
 - リライトしない URI のリスト , 129,277
- サポート
 - Solaris, 27
- サポートされる暗号化方式 , 193
- サンプル
 - リライト , 140

し

- 自己署名証明書 , 219
- 実行
 - HTTPS モード , 241
 - HTTP モード , 241
 - アプリケーション , 183
- 指定 , 238
 - Microsoft Windows ドメイン , 292
 - MIME タイプファイル , 304
 - NetFile ウィンドウの位置 , 299
 - NetFile のウィンドウサイズ , 298
 - OS 文字セット , 288
 - 一時ディレクトリ , 300
 - キープアライブ間隔 , 313
 - キャッシュされたソケットのタイムアウト , 254
 - 競合の解決 , 36
 - ゲートウェイスレッドプールサイズ , 253
 - ゲートウェイタイムアウト , 252
 - 検索ディレクトリの制限 , 302

- 最大接続キュー , 252
- 直接接続 , 260
- デフォルトの WINS/DNS , 293
- デフォルトのドメイン , 278,291
- 認証レベル , 238
- プロキシ , 259
- プロキシ認証 , 261
- ホストへのアクセス , 293
- ループバックポート , 310
- 自動検出
 - Netfile 内 , 180
- 自動プロキシ設定
 - ファイルの場所 , 263
- 終了
 - Netlet, 313
- 障害追跡 , 137
- 証明書
 - CA から届いた証明書のインストール , 224
 - certadmin スクリプト , 218
 - SSL, 212
 - 公開証明書 , 215
 - 削除 , 226
 - 自己署名 , 219
 - 出力 , 232
 - 証明書署名要求 , 221
 - 信頼属性 , 214
 - 信頼属性の変更 , 228
 - すべてをリスト表示 , 231
 - ファイル , 212
 - 要求 , 224
 - ルート CA 証明書 , 223
 - ルート CA 証明書のリスト表示 , 230
 - ワイルドカード , 82
- 処理順序
 - プロキシ , 63
- シングルサインオン , 237
- 信頼されている SSL ドメイン , 269
- 信頼属性 , 214

す

スタティックルール, 192

せ

正常なタイムアウト, 250

生成

自己署名証明書, 219

セキュアモード, 31

接続

警告ポップアップダイアログボックスの有効化,
311

持続, 248

設定

Netlet, 305

Outlook Web Access, 176

Personal Digital Certificates, 270

Secure Remote Access, 35

共通ホストのリスト, 289

許可されたホストのリスト, 294

拒否されたホストのリスト, 295

拒否される URL, 236

ゲートウェイ, 239

持続 HTTP 接続, 248

プロキシレット, 319

リライト, 128

選択

暗号化, 267

そ

属性

SRA, 333

設定, 35

ソケット

接続のタイムアウト, 249

た

ダイナミックルール

Netlet, 192

アプレットのダウンロード, 199

呼出し, 198

タイムアウト

キャッシュされたソケット, 254

ゲートウェイ, 252

正常な, 250

ソケット接続, 249

ち

チェックボックス

ポート警告ダイアログで有効化, 312

つ

通知, 36

て

停止

ゲートウェイ, 58

デバッグ

NetFile, 182

情報の場所, 206

デバッグログ

リライト, 137

デフォルト

Microsoft Windows ドメイン, 292, 293

Microsoft Windows ワークグループ, 292

ゲートウェイプロファイル, 40

ドメイン, 66

デフォルトの暗号化方式, 309

デフォルトのドメイン

指定, 278, 291

リライト, 66

と

ドメインとサブドメインのプロキシ, 63

に

認証

PDC, 81, 212

連鎖, 81

認証レベル, 238

は

パーサーと URI のマッピング, 133

パスワード

プロキシ, 261

ひ

非表示

ポート警告, 312

非武装ゾーン, 30

ふ

ファイルアップロードの制限, 301

複数インスタンス

ゲートウェイ, 52

リライタプロキシ, 75

ブラウザキャッシング

無効化, 82

プロキシ

hostproxy の指定, 60

Netlet, 185, 244

Web, 61

アクセラレータ, 328

逆, 78

指定, 259

認証, 261

パスワード, 261

リライタ, 242

プロキシ自動設定, 67

プロキシレット

PAC ファイルの使用, 67

設定, 319

有効化, 245

利点, 93

プロキシレットサービス, 34

プロトコル

NetFile, 180

NetFile でのサポート, 180

プロパティ

platform.conf, 44

へ

ヘッダー

HTTP, 79

変更

Netlet ルール, 308

ほ

ポート

loopback, 310

Netlet, 184

ポート番号

Netlet, 195

ホスト

アクセスの許可, 316

アクセスの拒否, 317

アクセスの指定, 293

許可されたホストのリストの設定, 294, 295

ホスト検出順序

NetFile で使用, 180

設定, 289

ま

マスキング

シード文字列, 280

有効化, 134, 280

マルチホームゲートウェイ, 52

む

無効化

Netlet プロキシ, 244

SSL Version 2.0, 267

シングルサインオン, 238

ブラウザキャッシング, 82

も

モード

HTTP, 241

HTTPS, 241

オープン, 30

セキュア, 31

文字セットのエンコーディング, 94

ゆ

有効化

40 ビットブラウザ接続, 266

Cookie, 246

HTTP 基本認証, 247

HTTP 接続, 248

MIME 推測, 133, 278

NetFile アクセス, 181

Netlet プロキシ, 74, 244

Netlet ログイン, 205, 284

Null 暗号化方式, 269

PDC 認証, 270

SSL Version 2.0, 267

SSL Version 3.0, 268

Web プロキシ, 258

Web プロキシの使用, 258

安全な Cookie としてマーク, 257

外部サーバー Cookie の格納, 256

逆プロキシ, 78

警告ポップアップダイアログボックス, 311

シングルサインオン, 238

すべての URL のリライト, 128

接続, 241

接続の再認証, 311

デバッグ, 182

認証の連鎖, 81

符号化選択, 267

プロキシレット, 245

ポート警告ダイアログにチェックボックスを表示, 312

マスキング, 134, 280

リライトプロキシ, 77, 242

ログイン, 283

ユーザー設定可能な暗号化方式, 193

り

リライト

6.x と 3.0 のルールセットのマッピング, 177

HTML ルール, 103

JavaScript ルール, 110

URI とルールセットのマッピングリストの作成, 274

URL スクレイパー, 94

XML ルール, 124

同じプロトコル, 282

ケーススタディ, 172

サンプル, 140

サンプルの操作, 140

すべての URL のリライトの有効化, 273

すべての URL をリライト, 128

設定, 128

デバッグログの使用, 137

ドメインとサブドメインのプロキシリスト, 66

パーサーと URI のマッピングリストの作成, 133

- マスクングの有効化, 134
- ライトしない URI のリストの作成, 129
- ルールセット DTD, 96
- ルールでのパターンマッチング, 108
- ルールの記述, 100
- ワイルドカードの使用, 130

リライターサービス, 33

リライタープロキシ

- 再起動, 77
- 作成, 76
- 有効化, 77
- 利点, 75

る

ループバックポート

- 割り当て, 310

ルール

- Netlet, 187
- WML, 127
- カスケードスタイルシート, 127
- 追加, 307
- リライター, 100
- リライターでの HTML, 103
- リライターでの JavaScript, 110

ルールセット

- generic, 129
- OWA, 275

ルールセットのマッピング

- URI のリストの作成, 274

れ

連携管理, 84

ろ

ログイン

- NetFile, 182

Netlet, 205

- ゲートウェイ, 283

- デバッグの有効化, 206

- リライター, 137

ログファイル

- ファイル名, 331

わ

ワイルドカード

- Web プロキシ, 63

- リライター, 130

ワイルドカードの証明書, 82

割り当て

- Netlet サービスをユーザーに, 306

- デフォルトのループバックポート, 310