Sun Java System Portal Server Secure Remote Access 7.2 管理ガイド

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

Part No: 820–4821
2008 年 5 月

080627@20490

目次

図目次

表目次

例目次

はじめに

このガイドでは、Sun Java™ System Portal Server Secure Remote Access 7.2 サーバーの管理方法について説明します。

Sun Java System Portal Server Secure Remote Access (SRA) サーバーは、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、SRA は、組織にセキュリティー保護された内部ポータルを提供し、従業員、ビジネスパートナー、一般の人々など、あらゆるターゲットユーザー向けのコンテンツ、アプリケーション、データへのアクセスを提供します。

この章で説明する項目は次のとおりです。

対象読者

『Sun Java System Portal Server Secure Remote Access 7.2 管理ガイド』は、Secure Remote Access サーバーを設定し、管理するユーザーを対象としています。

『Sun Java System Portal Server Secure Remote Access 7.2 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。Secure Remote Access サーバー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。

お読みになる前に

Portal Secure Remote Access サーバーの管理者は、次のテクノロジを理解している必要があります。

- Sun Java System Portal Server
- Sun Java System Directory Server
- Sun Java System Access Manager
- 使用している次のような Web コンテナ
  - Sun Java System Application Server 8.2
  - Sun Java System Web Server 7.0
- 使用しているオペレーティングシステム
- 基本的な UNIX® の管理手順
- LDAP (Lightweight Directory Access Protocol)
- Web Services for Remote Portlets (WSRP)

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

内容の紹介

本書は次の章で構成されています。

- パート I「Secure Remote Access サーバーコンポーネント」
  - 第 1 章Portal Server Secure Remote Access サーバーの概要では、Sun Java System Portal Server と Portal Server Secure Remote Access の関係について説明します。
  - 第 2 章ゲートウェイの操作では、ゲートウェイに関連する概念と、ゲートウェイの管理タスクについて説明します。
  - 第 3 章プロキシレットの操作では、Web ページを解析せずにゲートウェイ経由でイントラネット Web ページにアクセスできるようにするプロキシレットについて説明します。
  - 第 4 章リライタの操作では、プロキシレットおよびリライタを使用して、ゲートウェイ経由でイントラネット Web ページにアクセスする方法について説明します。
  - 第 5 章ネットファイルの操作では、ネットファイルを使用してリモートファイルシステムおよびリモートディレクトリへのアクセスと操作を行う方法について説明します。

- 第 6 章ネットレットの操作では、ネットレットを使用して、インターネットなどのセキュリティー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する方法について説明します。

- パート II「Secure Remote Access サーバーの設定」
  - 第 7 章Secure Remote Access サーバーのアクセス制御の設定では、Portal Server 管理コンソールへのアクセスを管理する方法について説明します。
  - 第 8 章Secure Remote Access ゲートウェイの設定では、Portal Server 管理コンソールからゲートウェイ属性を設定する方法について説明します。
  - 第 9 章ゲートウェイサービスのリライタの設定では、「リライタ」タブのゲートウェイサービスを使用してさまざまなタスクを実行する方法について説明します。
  - 第 10 章証明書の操作では、証明書の管理、および認証局からの自己署名証明書のインストールについて説明します。
  - 第 11 章ネットレットの設定では、Portal Server 管理コンソールからネットレット属性を設定する方法ついて説明します。
  - 第 12 章PDC (Private Domain Certificates) を使用する場合のネットレットの設定では、ネットレットで 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。
  - 第 13 章プロキシレットの設定では、Portal Server 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。
  - 第 14 章ネットファイルの設定では、Portal Server 管理コンソールを使用してネットファイルのオプションや権限などを設定する方法について説明します。
  - 第 15 章Secure Socket Layer アクセラレータの設定では、Portal Server Secure Remote Access サーバーの各種アクセラレータの設定について説明します。
- パート III「Secure Remote Access サーバーの管理」
  - 第 16 章ゲートウェイの管理では、ゲートウェイプロファイルとゲートウェイインスタンスの作成方法について説明します。
  - 第 17 章連携管理の例では、ネットワークアイデンティティー管理のさまざまなシナリオについて説明します。
- 付録 A 設定属性では、Portal Server 管理コンソールから各 Portal Server Secure Remote Access コンポーネントに対して設定できる、Sun Java System Portal Server Secure Remote Access の属性について説明します。
- 付録 B ログファイルでは、デバッグ情報などの情報について説明します。
- 付録 C 国コードでは、認証管理の際に指定する必要がある 2 文字の国コードの一覧を示します。

関連マニュアル

- 『Sun Java System Portal Server 7.2 Deployment Planning Guide 』
- 『Sun Java System Portal Server 7.2 Technical Overview 』
- 『Sun Java System Portal Server 7.2 管理ガイド』
- 『Sun Java System Portal Server 7.2 Command-Line Reference 』
- 『Sun Java System Portal Server 7.2 リリースノート』
- 『Sun Java System Portal Server 7.1 Community Sample Guide 』
- 『Sun Java System Portal Server 7.2 Technical Reference 』
- 『Sun Java System Portal Server 7.2 Developer's Guide 』

Portal Server の概念とコンポーネントの概要については、『Sun Java System Portal Server 7.2 Technical Overview 』を参照してください。

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/1224.1`
- Access Manager ドキュメント: `http://docs.sun.com/coll/1292.2`
- Web Server ドキュメント: `http://docs.sun.com/coll/1308.3`
- Application Server ドキュメント: `http://docs.sun.com/coll/1310.3`
- Web Proxy Server ドキュメント: `http://docs.sun.com/coll/1311.4`

関連する **Sun** 以外の **Web** サイト情報

このマニュアルでは、第三者が提供している URL で関連する追加情報を参照します。

注 – このマニュアルで紹介する Sun 以外の Web サイトが使用可能かどうかについては、Sun は責任を負いません。このようなサイトやリソース上、またはこれらを経由して利用できるコンテンツ、広告、製品、またはその他の資料についても、Sun は保証しておらず、法的責任を負いません。こうしたサイトやリソース上で、またはこれらを経由して利用できるコンテンツ、製品、サービスを利用または信頼したことに伴って発生した (あるいは発生したと主張される) 実際の (あるいは主張される) 損害や損失についても、Sun は一切の責任を負いません。

マニュアル、サポート、およびトレーニング

Sun のサービス	URL	内容
マニュアル	http://jp.sun.com/documentation/	PDF 文書および HTML 文書をダウンロードできます。
サポートおよびトレーニング	http://jp.sun.com/supportraining/	技術サポート、パッチのダウンロード、および Sun のトレーニングコース情報を提供します。

表記上の規則

このマニュアルでは、次のような字体や記号を特別な意味を持つものとして使用します。

表 P–1　表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 machine_name% you have mail.
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	machine_name% **su** Password:
aabbcc123	変数を示します。実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、rm *filename* と入力します。
『』	参照する書名を示します。	『コードマネージャー・ユーザーズガイド』を参照してください。
「」	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第 5 章「衝突の回避」を参照してください。 この操作ができるのは、「スーパーユーザー」だけです。

表 P–1 表記上の規則 (続き)

字体または記号	意味	例
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	`sun% grep '^#define \` `XV_VERSION_STRING'`

コマンド例におけるシェルプロンプト

コード例は次のように表示されます。

- C シェル

  ```
  machine_name% command y|n [filename]
  ```

- C シェルのスーパーユーザー

  ```
  machine_name# command y|n [filename]
  ```

- Bourne シェルおよび Korn シェル

  ```
  $ command y|n [filename]
  ```

- Bourne シェルおよび Korn シェルのスーパーユーザー

  ```
  # command y|n [filename]
  ```

[ ] は省略可能な項目を示します。上記の例は、*filename* は省略してもよいことを示しています。

| は区切り文字 (セパレータ) です。この文字で分割されている引数のうち 1 つだけを指定します。

キーボードのキー名は英文で、頭文字を大文字で示します (例: Shift キーを押します)。ただし、キーボードによっては Enter キーが Return キーの動作をします。

ダッシュ (-) は 2 つのキーを同時に押すことを示します。たとえば、Ctrl-D は Control キーを押したまま D キーを押すことを意味します。

パート I

Secure Remote Access サーバーコンポーネント

◆ ◆ ◆ 第 1 章

Portal Server Secure Remote Access サーバーの概要

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 と Sun Java System Portal Server Secure Remote Access コンポーネントの関係について説明します。

この章の内容は次のとおりです。

Secure Remote Access の概要

Secure Remote Access を使えば、リモートエンドユーザーは所属する組織のネットワークとサービスに、インターネット経由で安全にアクセスできます。また、セキュリティー保護されたインターネットポータルを組織に提供し、従業員、ビジネスパートナー、一般ユーザーなど、ターゲットとするユーザーがコンテンツ、アプリケーション、およびデータを利用できるようにします。

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、Secure Remote Access ではブラウザによる、セキュリティー保護されたリモートアクセスが提供されます。Secure Remote Access は、Java™ テクノロジに対応したブラウザを使用するすべてのデバイスからユーザーへのアクセスが可能な、セキュリティー保護された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する必要はありません。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への暗号化されたセキュリティー保護されたアクセスが、ユーザーに対して保証されます。

Secure Remote Access 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提供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ー、保護、およびプライバシに重点が置かれています。Secure Remote Access のアーキテクチャーは、これらのタイプのポータルに適しています。

ユーザーは Secure Remote Access ソフトウェアを利用することで、イントラネットリソースをインターネットに公開しなくても、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、次の節で説明するオープンモードとセキュリティー保護されたモードの 2 つのモードで動作します。

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 Secure Remote Access 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、セキュリティー保護されたリモートアクセスは使用できません。したがって、ユーザーはセキュリティー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リティー保護された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。

公開情報の配布、無償アプリケーションへのアクセス許可について、ポータルに機密情報が含まれていない場合、大量のアクセス要求への応答は、セキュリティー保護されたモードに比べて速くなります。

オープンモードでは、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。

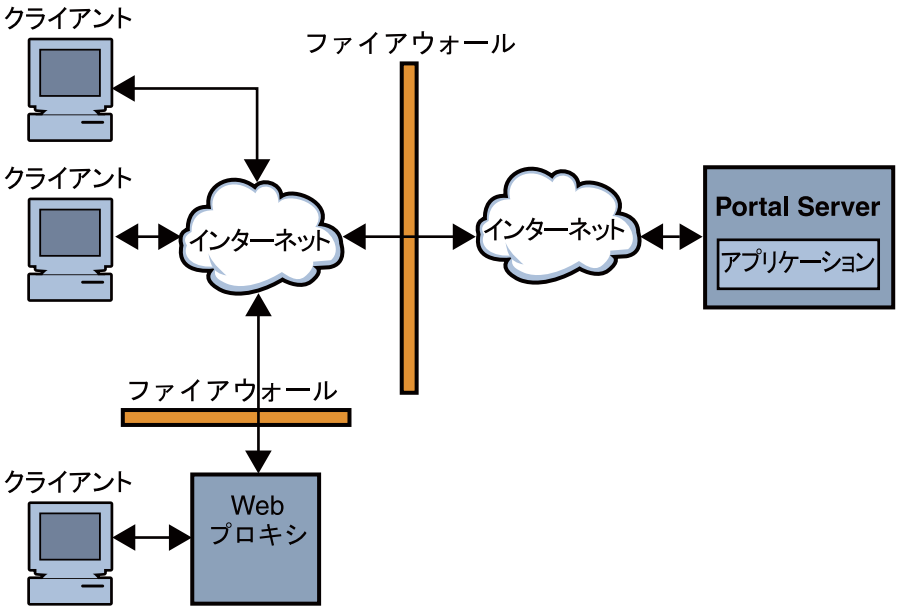


図 1–1　Secure Remote Access を使用するオープンモードの Portal Server

セキュリティー保護されたモード

セキュリティー保護されたモードは、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ー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リティー保護された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 または HTTPS が使用されます。

セキュリティー保護されたモードでは、SSL を使用してクライアントとゲートウェイ間のインターネット上の接続を暗号化しています。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。

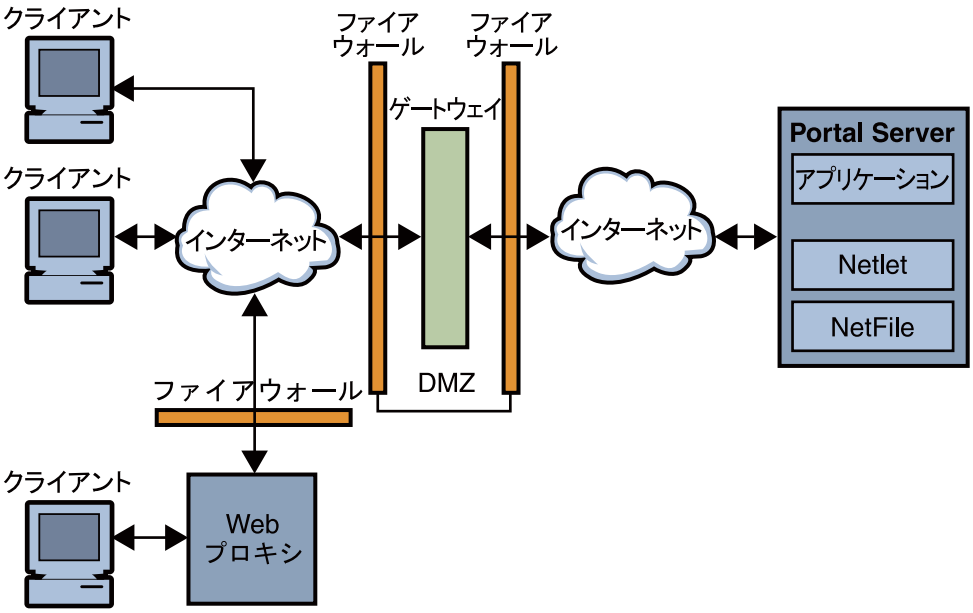


図 1–2　Secure Remote Access を使用するセキュリティー保護されたモードの Portal Server

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。Secure Remote Access 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。ビジネス要件への対応方法の詳細については、『Sun Java System Portal Server 7.2 Deployment Planning Guide 』を参照してください。

Secure Remote Access サービス

Secure Remote Access 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

  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ー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 サーバーでは、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 ベースのリソースが使用されます。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ネットレットは、この目的で使用されます。詳細については、第 2 章ゲートウェイの操作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 4 章リライタの操作を参照してください。

- ネットファイル

  ネットファイルは、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ーアプリケーションです。ネットファイルには Java ベースのユーザーインタフェースが含まれます。詳細については、第 5 章ネットファイルの操作を参照してください。

- ネットレット

  ネットレットは、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かつ効率的に実行できるようにします。サイトにネットレットを実装すると、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 6 章ネットレットの操作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するために、クライアントマシン上のブラウザのプロキシ設定を読み込み、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。

Secure Remote Access 属性の設定

Secure Remote Access の属性は、Portal Server 管理コンソールで次のサービスを使用して設定します。

- アクセス制御

  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 7 章Secure Remote Access サーバーのアクセス制御の設定を参照してください。

- ゲートウェイ

コンポーネントの有効化、Cookie 管理、プロキシ管理、セキュリティー設定、パフォーマンスチューニング、リライタのマッピング管理など、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 8 章Secure Remote Access ゲートウェイの設定を参照してください。

- ネットファイル

  共通ホスト、MIME タイプ、および異なる種類のホストへのアクセスなど、ネットファイル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 14 章ネットファイルの設定を参照してください。

- ネットレット

  ネットレットルール、必須ルールへのアクセス、組織とホスト、およびデフォルトアルゴリズムなど、ネットレット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 章ネットレットの設定を参照してください。

- リライタ

  すべてのリライタルールセットのダウンロード、アップロード、および削除を行う場合に使用します。

- 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 13 章プロキシレットの設定を参照してください。

注意 – 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有効にするには、ゲートウェイを再起動します。詳細については、191 ページの「コマンド行オプションによるゲートウェイ属性の設定」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

1. 『**Sun Java System Portal Server 7.2** 管理ガイド』の「管理コンソールにログインする」
2. 「**Secure Remote Access**」タブを選択し、適切なサービスタブ (「ネットレット」、「ネットファイル」、または「プロキシレット」) をクリックします。
3. 「**DN** を選択」ドロップダウンメニューから「組織」または「ロール」を選択します。
4. 「**COS** 優先順位」ドロップダウンボックスで適切な競合解決レベルを選択します。

5 「保存」をクリックして終了します。

サポートされるアプリケーション

SRA は、次のアプリケーションをサポートします。

- 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

準備

▼ ポータル用の **SRA** を有効にする

1 `PortalServer_base/bin/psadmin switch-sra-status -u amadmin -f <passwordfile> on` コマンドを使用して、**SRA** の状態を切り替えます。

2 `PortalServer_base/bin/psadmin provision-sra -u amadmin -f <passwordfile> -p <portal-id> --gateway-profile <profile-name> --enable` コマンドを使用して、**SRA** の状態をプロビジョニングします。

◆ ◆ ◆ 第 2 章

2 ゲートウェイの操作

この章では、ゲートウェイに関連する概念について説明します。ゲートウェイの管理については、第 16 章ゲートウェイの管理を参照してください。ゲートウェイの設定については、第 8 章Secure Remote Access ゲートウェイの設定を参照してください。

この章の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ー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。

ゲートウェイインスタンスごとに、次のタスクを完了してください。

- 34 ページの「ゲートウェイプロファイルの作成」
- 35 ページの「複数のゲートウェイインスタンスの作成」
- 第 8 章Secure Remote Access ゲートウェイの設定

そのほかに、次のゲートウェイ関連トピックがあります。

- 35 ページの「ゲートウェイの再起動」
- 35 ページの「ゲートウェイウォッチドッグの設定」
- 36 ページの「仮想ホストの指定」
- 36 ページの「Access Manager へアクセスするプロキシの設定」

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、および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」という名前の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/SUNWportal/platform.conf.default`

`/etc/opt/SUNWportal` 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` ファイルの詳細については、36 ページの「platform.conf ファイルの概要」を参照してください。

プロファイルを操作するときは、次の作業を実行できます。

- 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。

注意 – 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は、同じプロファイルを割り当てないでください。このように設定すると、ポート番号が同じになることにより競合が発生します。

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、競合が発生します。

複数のゲートウェイインスタンスの作成

複数のゲートウェイインスタンスの作成については、『Sun Java System Portal Server 7.2 Installation and Configuration Guide』の第 4 章「Installing and Configuring a Gateway With Portal Server」を参照してください。

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とは、1 つの Portal Server 上に作成された複数のゲートウェイです。これらのインスタンスを作成するには、`platform.conf` ファイルを次のように変更します。

```
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。

次の部分を参照しながら、`/etc/opt/SUNWam/config/` の `AMConfig-`*`instance-name`*`.properties` 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。

261 ページの「同じ LDAP を使用するゲートウェイインスタンスを作成する」を参照してください。

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場合
- 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場合
- OutOfMemory エラーなどによりゲートウェイがクラッシュする場合
- ゲートウェイが応答を停止し、要求を一切処理しない場合

ゲートウェイウォッチドッグの設定

ウォッチドッグ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。ウォッチドッグを開始または停止するには、次のコマンドを実行します。`./psadmin sra-watchdog -u amadmin -f <password-file> -t <type> on|off`この間隔は、デフォルトでは 60 秒に設定されています。この値を変更する場合は、crontab ユーティリティーで次の行を編集します。

```
0-59 * * * * gateway-install-root/SUNWportal/bin/
/var/opt/SUNWportal/.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す。たとえば、ホスト名 abc 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 cde を追加できます。

Access Manager へアクセスするプロキシの設定

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。264 ページの「プロキシを指定する」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、デフォルトでは次の場所にあります。
/etc/opt/SUNWportal

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 platform.conf ファイルのサンプルを示します。

```
Tue May 30 11:51:23 IST 2006
debug.com.sun.portal.rewriter.original.level=INFO
gateway.favicon=
gateway.bindipaddress=10.12.154.236
debug.com.sun.portal.sra.rproxy.toFromServer.handler.java.util.logging.FileHandler.pattern=
/var/opt/SUNWportal/logs/sra/default/Gateway.toFromServer.%u.%g.log
gateway.port=443
rewriterproxy.jvm.flags=-ms64m -mx128m
portal.server.instance=default
debug.com.sun.portal.handler.java.util.logging.FileHandler.filter=
```

```
gateway.jdk.dir=/usr/jdk/entsys-j2se
gateway.ignoreURIList=/MSOffice/cltreq.asp,/_vti_bin/owssvr.dll
debug.com.sun.portal.rewriter.rest.level=INFO
gateway.trust_all_server_certs=true
debug.com.sun.portal.handler.java.util.logging.FileHandler.append=true
gateway.cdm.cacheCleanupTime=300000
gateway.httpurl=
debug.com.sun.portal.handler.java.util.logging.FileHandler.count=1
gateway.jvm.classpath=
debug.com.sun.portal.setserverlogs=false
gateway.protocol=https
debug.com.sun.portal.sra.rproxy.toFromServer=java.util.logging.FileHandler
rewriterproxy.jvm.classpath=
gateway.enable.customurl=false
debug.com.sun.portal.sra.rproxy.toFromBrowser=java.util.logging.FileHandler
debug.com.sun.portal.handler.java.util.logging.FileHandler.formatter=com.sun.portal.
log.common.PortalLogFormatter
debug.com.sun.portal.sra.rproxy.toFromBrowser.handler.java.util.logging.FileHandler.pattern=
/var/opt/SUNWportal/logs/sra/default/Gateway.toFromBrowser.%u.%g.log
debug.com.sun.portal.level=INFO
debug.com.sun.portal.rewriter.unaffected.separatefile=true
gateway.enable.accelerator=false
debug.com.sun.portal.rewriter.original.separatefile=true
gateway.virtualhost=nicp236.india.sun.com 10.12.154.236
debug.com.sun.portal.stacktrace=true
gateway.host=nicp236.india.sun.com
debug.com.sun.portal.handler.java.util.logging.FileHandler.pattern=
/var/opt/SUNWportal/logs/sra/default/%logger.%sraComponentType.%u.%g.log
gateway.certdir=/etc/opt/SUNWportal/cert/default
gateway.sockretries=3
gateway.allow.client.caching=true
debug.com.sun.portal.rewriter.unaffected.level=INFO
debug.com.sun.portal.rewriter.uriinfo.separatefile=true
log.config.check.period=2000
debug.com.sun.portal.rewriter.rewritten.level=INFO
gateway.userProfile.cacheSize=1024
debug.com.sun.portal.rewriter.rulesetinfo.level=INFO
netletproxy.jvm.classpath=
gateway.userProfile.cacheSleepTime=60000
debug.com.sun.portal.rewriter.uriinfo.level=INFO
debug.com.sun.portal.rewriter.rest.separatefile=true
gateway.notification.url=notification
debug.com.sun.portal.rewriter.rulesetinfo.separatefile=true
gateway.logdelimiter=&&
gateway.ignoreServerList=false
gateway.jvm.flags=-ms64m -mx128m
debug.com.sun.portal.handler.java.util.logging.FileHandler.limit=5000000
gateway.dsame.agent=http\://sunone216.india.sun.com\:8080/portal/RemoteConfigServlet
```

```
gateway.httpsurl=
gateway.retries=6
gateway.userProfile.cacheCleanupTime=300000
gateway.logging.password=X03MO1qnZdYdgyfeuILPmQ\=\= UX9x0jIua3hx1YOVRG/TLg\=\=
netletproxy.jvm.flags=-ms64m -mx128m
debug.com.sun.portal.rewriter.rewritten.separatefile=true
gateway.user=noaccess
gateway.external.ip=10.12.154.236
debug.com.sun.portal.handler=java.util.logging.FileHandler
gateway.cdm.cacheSleepTime=60000
rewriterproxy.accept.from.gateways=
rewriterproxy.checkacl=false
```

次の表は、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 ファイルのプロパティー

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、初期化のあと、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、およびポート。
gateway.protocolgateway.hostgateway.port		ゲートウェイのプロトコル、ホスト、およびポート。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、または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。

表 2–1 ファイルのプロパティー (続き)

エントリ	デフォルト値	説明
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検査します。 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。
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ープロバイダアドレスを指定できます。

gateway.virtualhost.defaultOrg=org		ユーザーがログインするデフォルトの org を指定します。
		たとえば、仮想ホストフィールドのエントリが次のようになっていると仮定します。
		`gateway.virtualhost=test.com employee.test.com`
		`Managers.test.com`
		この場合、デフォルトの org エントリは次のようになります。
		`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`
		ユーザーは `https://manager.test.com` を使用して、`https://test.com/o=Manager,dc=test,dc=com` ではなくマネージャーの org にログインできます。
		注 – virtualhost と defaultOrg は `platform.conf file` 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。

gateway.notification.url		ゲートウェイのホスト、プロトコル、および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ときに使用されます。 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 ファイルのプロパティー (続き)

エントリ	デフォルト値	説明
gateway.debug	error	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、*debug-directory*/files です。デバッグファイルの場所は、gateway.debug.dir エントリで指定されます。 次のデバッグレベルがあります。 ■ 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。 ■ warning: 警告メッセージがログとして記録される。 ■ 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。 ■ 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。 次のデバッグファイルがあります。 srapGateway.*gateway-profile-name*: ゲートウェイデバッグメッセージを格納する。 Gateway_to_from_server.*gateway-profile-name*: 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。 このファイルを生成するには、/var/opt/SUNWportal/debug ディレクトリの書き込み権を変更します。 Gateway_to_from_browser.*gateway-profile-name*: 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。 このファイルを生成するには、/var/opt/SUNWportal/debug ディレクトリの書き込み権を変更します。
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 アドレスをここで指定する必要があります。この IP は、ネットレットが 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。

表 2–1 ファイルのプロパティー (続き)

エントリ	デフォルト値	説明
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では、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ーは高まりますが、ネットワークの負荷が増え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、キャッシュのクリーンアップが頻繁に再試行されます。
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)。
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書き換えるためのカスタム URL を指定できます。
gateway.httpurl		ゲートウェイがページを書き換えるためのカスタム 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書き換えるためのカスタム 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。

表 2–1 ファイルのプロパティー (続き)

エントリ	デフォルト値	説明
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。
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、このプロパティーを 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*		このプロパティーの値には、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で、Access Manager からゲートウェイに送信される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で、Access Manager からゲートウェイに送信されるキャッシュのタイムアウト値。
netletproxy.port	10555	ネットレット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、このプロパティーを true に設定します。
rewriterproxy.accept.from.gateways		これは、リライタプロキシでの要求の受け入れを可能にする IP アドレスのリストです。HTTP モードと HTTPS モードの両方で機能します。これはセキュリティーを高める目的で使用されます。このリストのアドレスからの要求のみが受け入れられ、その他の要求は一切処理されません。各 IP アドレスをコンマで区切って指定できます。デフォルト値は空で、その場合は旧バージョンモードとして扱われます。つまり、リライタプロキシへのすべての要求が受け入れられます。

表 2–1 ファイルのプロパティー (続き)

エントリ	デフォルト値	説明
rewriterproxy.checkacl=	false	このプロパティーが有効になっている場合、リライタプロキシは、ゲートウェイと同じように ACL の値をチェックします。旧バージョンモードの値は false です。true に設定すると、リライタプロキシは指定された DN で、URL をゲートウェイアクセスサービスに指定された値と照合し、リストの設定に従って要求を許可または拒否します。この値は、HTTP モードと HTTPS モードの両方で機能します。

Web プロキシの使用

SUN 以外の 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。

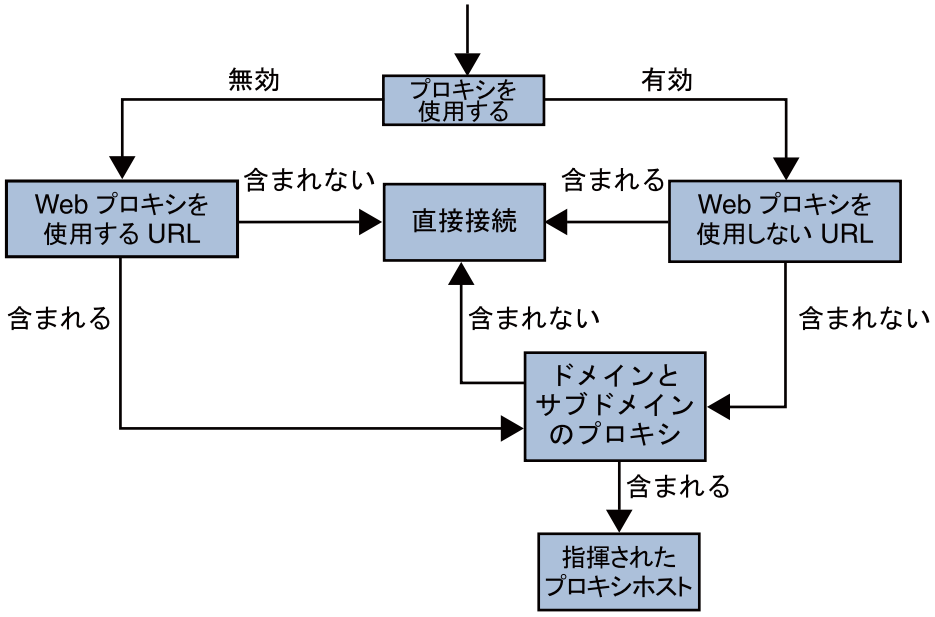
- ゲートウェイサービスの「ドメインとサブドメインのプロキシ」フィールドで、必要なプロキシとドメインおよびサブドメインのリストを作成します。
- 「プロキシを使用する」オプションを選択すると、次のような設定になります。
  - 指定されたホストに、「ドメインとサブドメインのプロキシ」フィールドで指定したプロキシが使用されます。
  - 「ドメインとサブドメインのプロキシ」リストで指定したドメインとサブドメイン内の、特定の URL に直接接続できるようにするには、「Web プロキシを使用しない URL」フィールドにその URL を指定します。

  「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。

    「プロキシを使用する」オプションが無効になっていても、「Web プロキシを使用する URL」リスト内の URL への接続には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。

次の図は、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。



43 ページの「Web プロキシの設定」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。

前述した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、前述した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、ほか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もっと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します。一致するエントリが見つかるまで、次の検索が順に行われます。

- 「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ると、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。エントリが見つかると、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。エントリが見つかると、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。エントリが見つかると、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。エントリが見つかると、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。エントリが見つかると、対応するプロキシが使用されます。
- 一致するエントリが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、次の表に示すように、これらのエントリを内部的に 1 つのテーブル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す。
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

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
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。
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す。
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。

ヒント– 「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、個々のエントリをリスト内の各行に配置できます。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
```

この情報は次のように指定できます。

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

このようなリスト形式にすると、反復されたエントリやその他のあいまいなエントリを追跡しやすくなります。

ドメインとサブドメインのプロキシリストに基づく書き換え

リライタも、「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。リライタは、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 を書き換えます。

注意– 「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、書き換えの対象と見なされません。たとえば、エントリ 24 は書き換えの対象になりません。

リライタについては、第 4 章リライタの操作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。

前述した例では、`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。

要求された URL が host1 の場合、このエントリ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」リストに host1.red.sesta.com があるかどうかが確認されます。

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。

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、ネットレット、およびプロキシレットは、*Rhino* ソフトウェアを使用して PAC ファイルを解析します。SUNWrhino パッケージは、Java 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、ネットレット、およびプロキシレットは PAC ファイルを解析できません。
- 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 ファイルを解析できません。
- 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 ファイルを取得します。
- 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。

  1. コマンド行で、次のファイルを編集します。

     /etc/opt/SUNWportal/platform.conf.*gateway-profile-name*
  2. 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
  3. 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。

     ./psadmin start-sra-instance –u amadmin – f <*password file*> –N <*profile name*>–t <*gateway*>
- 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。50 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負荷分散は行われません。
- 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。
- イントラ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。このプロキシタイプは、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。51 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

次のプロキシをドメインとサブドメインに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

次のプロキシをドメインとサブドメインに使用する場合を考えます。

```
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 ドメイン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 にアクセスします。

PAC ファイルの場所の指定

PAC ファイルの場所を指定するための形式は、その場所により次のようになります。

- PAC ファイルが Web サーバーに常駐している場合、PAC URL は次のようになります。

  http://hostname/*pacfile_name* .pac

- PAC ファイルがローカルファイルの場合 (たとえば、c:\pacfile\sample.pac)、Java 1.4.1_x では PAC URL を次のように入力します。

  file://c:/pacfile/sample.pac

- PAC ファイルがローカルファイルの場合 (たとえば、c:\pacfile\sample.pac)、Java 1.4.2_x では PAC URL を次のように入力します。

  file:///c:/pacfile/sample.pac

個別のセッションにおけるサービスの追加

個別のセッションで Portal Server サービスを追加する場合は、次の操作を行います。

- 管理コンソールの「ゲートウェイ」>「コア」の下に Portal Server を一覧表示する。
- 「ゲートウェイ」>「セキュリティー」の下にある「非認証 URL」に Portal Server の URL を一覧表示する。

ネットレットプロキシの使用

ネットレット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ネットレット接続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。このように設定するには、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。ネットレット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。

ネットレットプロキシは、ゲートウェイを経由してクライアントからのセキュリティー保護されたトンネルをイントラネット内のネットレット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ーを補強します。プロキシを使用すると、ネットレットパケットがネットレットプロキシにより解読され、送信先に送られます。

ネットレットプロキシを使用する利点を次に示します。

- セキュリティー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。
- 52 ページの「ネットレットプロキシの使用」の「ネットレット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セキュリティー保護されたチャネルを Portal Server まで延長します。ネットレットプロキシはデータの暗号化によってセキュリティーを改善しますが、システムリソースの使用を増やす場合があります。ネットレットプロキシのインストールについては、『Sun Java System インストールガイド』を参照してください。

次の作業を実行できます。

- Portal Server ノードまたは別のノードでネットレットプロキシをインストールします。
- 複数のネットレット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。これは負荷分散に役立ちます。

- 単一のマシンでネットレット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、ネットレットプロキシの単一のインストールを設定します。
- Web プロキシを通じたネットレットトンネリング

ネットレット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、およびネットレット宛先サーバーから構成されます。

最初の例では、ネットレット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。データの暗号化はクライアントとゲートウェイの間だけで行われます。ネットレット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、ネットレット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。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。すべてのネットレットがネットレットプロキシを通じてルーティングされているため、ネットレット要求に対して 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、ネットレット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。別のノードにネットレット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。この場合も、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 つのポートはネットレットの要求をネットレットプロキシサーバーにルーティングします。

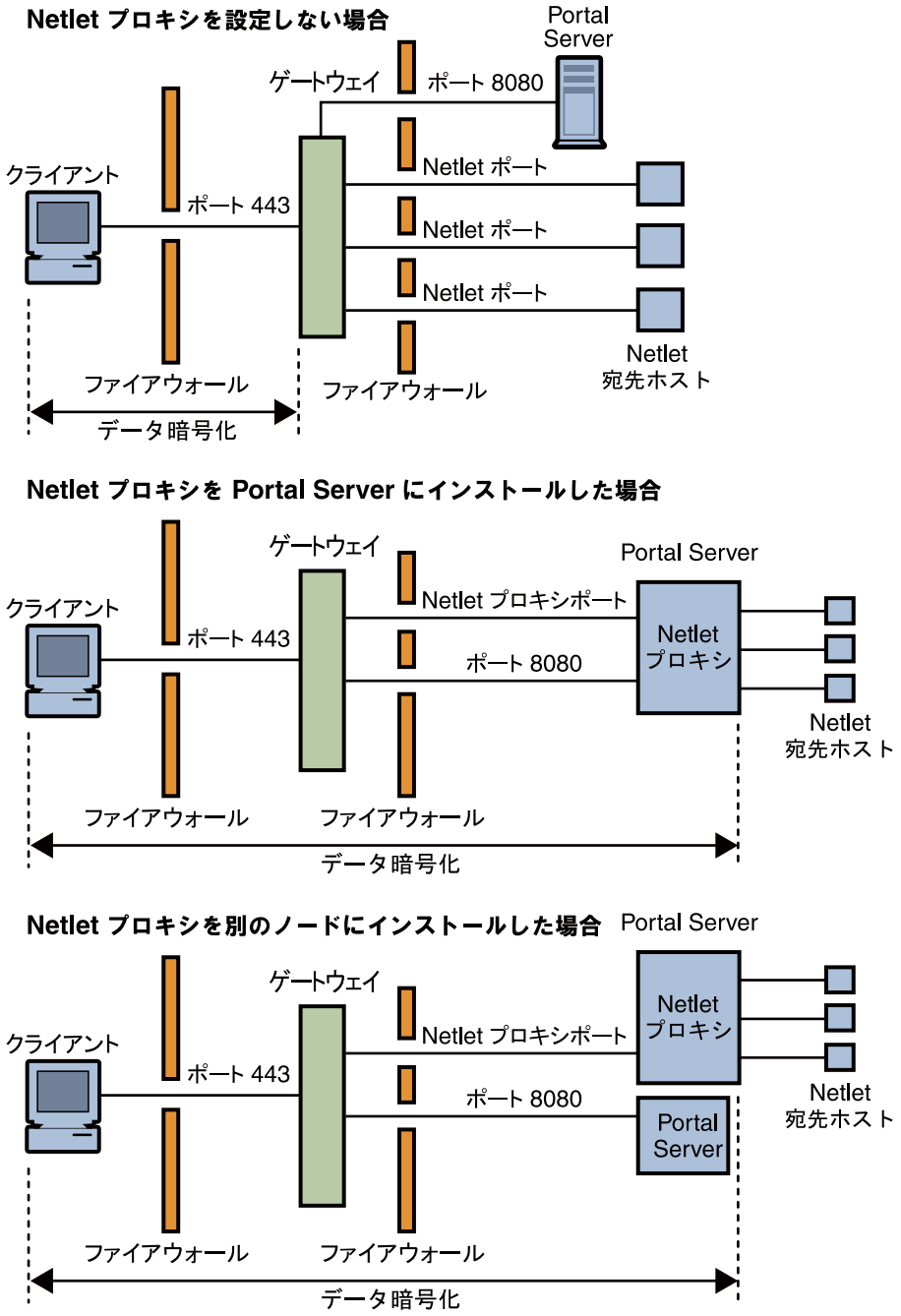


図 2–1　ネットレットプロキシの実装

ネットレットプロキシの有効化

ネットレットプロキシは、Portal Server 管理コンソールを使用して、ゲートウェイサービスから有効にします。

ネットレットプロキシの再起動

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、ネットレットプロキシを設定することができます。ネットレットプロキシを監視し、ネットレットプロキシが停止したときに再起動するように ウォッチドッグプロセスをスケジューリングできます。

ネットレットプロキシは手動で再起動することもできます。手順については、265 ページの「ネットレットプロキシを再起動する」を参照してください。

ネットレットプロキシのウォッチドッグを設定する

ウォッチドッグがネットレット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。この間隔を変更するには、`crontab` ファイルに次の行を追加します。

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

注– ウオッチドッグを開始または停止するには、次のコマンドを実行します。

```
./psadmin sra-watchdog -u amadmin -f <password-file> -t <type> on|off
```

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取得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。
- 宛先サーバーが HTTPS をサポートせず、HTTP しかサポートしていなくても、ゲートウェイとインターネットの間の HTTP トラフィックはセキュリティー保護されます。

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を Portal Servers リストに指定してください。

ゲートウェイインスタンスごとに (Portal Server ノード上でなくてもかまわない) リライタプロキシの複数インスタンスがある場合は、`platform.conf` ファイルに、リライタプロキシに対して 1 つのポートエントリを指定するのではなく、*host-name:port* の形式でリライタプロキシごとに詳細を指定します。

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あとで実行します。

265 ページの「リライタプロキシインスタンスを作成する」を参照してください。

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。

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 ウォッチドッグ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。

266 ページの「リライタプロキシを再起動する」を参照してください。

リライタプロキシのウォッチドッグの設定

ウォッチドッグ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。この間隔を変更するには、`crontab` ファイルに次の行を追加します。

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

注 – ウオッチドッグを開始または停止するには、次のコマンドを実行します。
`./psadmin sra-watchdog -u amadmin -f <password-file> -t <type> on|off`

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、負荷分散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 で書き換えられる必要があります。このためには、逆プロキシを有効化する必要があります。

266 ページの「逆プロキシを有効化する」を参照してください。

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。それら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` に設定します。次に、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。

表 2–3 HTTP ヘッダーの情報 (続き)

ヘッダー	構文	説明
PS-Netlet	`X-PS-Netlet:enabled=true/false`	ゲートウェイでネットレット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。 ネットレット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。 次に例を示します。 ■ `PS-Netlet: enabled=false` ネットレットは無効化されています。 ■ `PS-Netlet: enabled=true; encryption=ssl` ネットレットは有効で、ゲートウェイは SSL モードで稼動しています。 ネットレットが有効でない場合は、`encryption=ssl` または `encryption=plain` は生成されません。
PS-GW-URL	`X-PS-GW-URL: http(s)://gatewayURL(:port)`	クライアントが接続している URL を示します。 ポートが標準ポートでない場合 (ゲートウェイが HTTP/HTTPS モードで稼動し、ポートが 80/443 でない場合など) は、`:port` が追加されます。

表 2–3 HTTP ヘッダーの情報 *(続き)*

ヘッダー	構文	説明
PS-GW-Rewriting-URL	`X-PS-GW-URL: http(s)://gatewayURL(:port)/[SessionInfo]`	ゲートウェイがすべてのページを書き換える URL を示します。 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。 ■ 接続先ホストが「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書き換える実際の URL (符号化されたセッション ID 情報が含まれる) ■ 接続先ホストが「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 `$SessionID` となる 注 – 認証ページからの応答のように、応答の過程でユーザーの Access Manager のセッション ID が変更された場合、ページは、それまでヘッダーに指定されていた値ではなくその値で書き換えられます。次に例を示します。 ■ ブラウザが Cookie をサポートする場合は、次のようになります。 PS-GW-Rewriting-URL: https://siroe.india.sun.com:10443/ ■ ブラウザが Cookie をサポートせず、エンドサーバーが「ユーザーセッション Cookie を転送する URL」フィールドに含まれる場合は、次のようになります。 PS-GW-Rewriting-URL: https://siroe.india.sun.com:10443/SessIDValCustomEncodedValue/ ■ ブラウザが Cookie をサポートしないが、エンドサーバーが「ユーザーセッション Cookie を転送する URL」フィールドに含まれない場合は、次のようになります。 PS-GW-Rewriting-URL: https://siroe.india.sun.com:10443/$SessionID
PS-GW-CLientIP	`X-PS-GW-CLientIP:` *IP*	ゲートウェイが `recievedSocket.getInetAddress().getHostAddress()` から取得した IP を示します。 クライアントがゲートウェイに直接接続する場合、この値によって 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より高いレベルのセキュリティーがもたらされます。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。

手順については、267 ページの「既存の PDC インスタンスに認証モジュールを追加する」を参照してください。

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。

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。

この証明書によって、同じドメイン内の複数のホストがセキュリティーで保護されます。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。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セキュリティー保護されたアクセスを提供するため、クライアントが情報をローカルにキャッシュする必要はありません。

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セキュリティー保護された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。この要因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 の管理下に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。

268 ページの「ブラウザキャッシングを無効にする」を参照してください。

ゲートウェイサービスのユーザーインタフェースのカスタマイズ

ここでは、編集可能な各種のゲートウェイプロパティーファイルについて説明します。

srapGateway.properties ファイルの編集

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*`/SUNWportal/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は、メッセージの言語にかかわりなく、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には、このファイルを *portal-server-install-root*`/SUNWportal/srapGateway_<locale>.properties` などの各ロケール用ファイルを変更します。

srapgwadminmsg.properties ファイルの編集

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。

LDAP ディレクトリの共有

Portal Server と Access Manager サーバーの 2 つのインスタンスが同じ LDAP ディレクトリを共有している場合、それ以後のすべての Portal Server インスタンス、Access Manager インスタンス、およびゲートウェイインスタンスで LDAP ディレクトリが共有されます。268 ページの「LDAP ディレクトリを共有する」を参照してください。

◆ ◆ ◆ 第 3 章

プロキシレットの操作

この章では、Web ページを解析せずにゲートウェイ経由でイントラネット Web ページにアクセスできるようにするプロキシレットについて説明します。

プロキシレットの操作

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 (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または宛先サーバーの間でチャネルのセキュリティーが確保されます。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。復号化は、クライアントマシン上で行われます。

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、プロキシレットはクライアントのプロキシ設定を、そのゲートウェイをポイントするように再設定します。

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 が有効化どうかを確認する方法については、57 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 アドレスまたはプロキシのホスト名とポートが指定されている Portal Server 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が確認されます。

- ブラウザのアクセス権が適切かどうか
- ブラウザが IE 6.0 SP2、IE 7、Firefox 2.0 であるかどうか
- 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

これらの要件をすべて満たしている場合は、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。クライアントに JRE 1.4.2 以降がインストールされていない場合、ユーザーがインターネット接続と管理者権限の両方を持っていれば、プロキシレットによって 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、PAC (Proxy Auto Configuration) ファイルまたはプロキシ設定リストからプロキシ設定が取得されます。

注 – プロキシレットアプレットを使用する場合は、ブラウザのポップアップブロッカを無効にするようにユーザーに通知してください。

HTTPS のサポート

プロキシレットによる HTTPS のサポートは、次のようになっています。

- 復号化は、クライアントマシンで行われます。
- SSL モードで稼働している場合、接続先サーバーにアクセスできます。
- クライアント証明書は、接続先サーバーに直接提示されます。
- ゲートウェイでは、基本認証のシングルサインオン (SSO) はサポートされていません (ゲートウェイは SSO 情報を HTTP ヘッダーに挿入できない)。
- URL ベースのアクセス制御はサポートされていないため、ホストベースのアクセス制御のみが可能です。
- ゲートウェイの前にある外部アクセラレータと外部逆プロキシは、現時点ではサポートされていません。

注 – このサポートは、Portal Server が HTTPS を使用する場合のプロキシレットに対するものではありません。

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ー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

プロキシレットの有効化と設定については、第 13 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 の Web 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。

◆ ◆ ◆ 第 4 章

4 リライタの操作

Secure Remote Access のリライタコンポーネントは、Web ページを解析して、ユーザーがゲートウェイ経由でイントラネット Web ページにアクセスできるようにします。

この章の内容は次のとおりです。

リライタの概要

Secure Remote Access 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) 参照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、登録されている名前空間にネームをカプセル化し、それに名前空間のラベルを付ける方法を定義します。もっとも一般的な URI は URL (Uniform Resource Locator) です。リライタは HTTP または HTTPS だけをサポートします。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。

例 4–1 URL の書き換え

`http://abc.sesta.com\\index.html` は書き換えられます。

書き換えられない URL は `http:\\\\abc.sesta.com`、*http:/abc.com* などです。

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、パフォーマンスに影響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、URL スクレーパーとゲートウェイによって使用されます。

URL スクレーパー

URL スクレーパープロバイダは、設定されている URI からコンテンツを取得します。それらの URI 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、ユーザーが次のサイトにアクセスするとします。

```
<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">
```

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

URL スクレーパープロバイダの詳細については、『Sun Java System 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ます。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次の HTML ページ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` のコンテンツを取得します。

ゲートウェイはいくつかのルールを使用して、取得された Web ページの書き換える要素を判断します。

ルールセットの記述

ルールセットの定義の詳細については、『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。

この節の内容は、次のとおりです。

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 of programmatically
    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 'URL'>
<!ENTITY % eEXPRESSION 'EXPRESSION'>
<!ENTITY % eDHTML 'DHTML'>
<!ENTITY % eDJS 'DJS'>
<!ENTITY % eSYSTEM 'SYSTEM'>

<!ENTITY % ruleSetElements '(HTMLRules | JSRules | XMLRules)?'>
<!ENTITY % htmlElements '(Form | Applet | Attribute)*'>
<!ENTITY % jsElements '(Variable | Function)*'>
<!ENTITY % xmlElements '(Attribute | TagText)*'>

<!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、必須属性の値を * だけに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、99 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、135 ページの「ケーススタディー」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。

- コンテンツの書き換えが必要な 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、書き換えが必要なページを特定します。

- 各ページで書き換えが必要な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の書き換え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

ルールセットを作成する場合は、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コンテンツの書き換え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コンテンツの書き換え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コンテンツの書き換えに必要なすべてのルールを指定します。
- イントラネットページで、書き換えの必要のある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。
- ゲートウェイを再起動して変更を適用します。

  gateway-install-root`/SUNWportal/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 などがあります。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解析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書き換えます。

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

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、および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。

- 75 ページの「HTML コンテンツの属性ルール」
- 77 ページの「HTML コンテンツのフォームルール」
- 78 ページの「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

このルールは値を書き換え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。次に例を示します。

- 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)

この節では、次の項目について説明します。

- 75 ページの「属性ルールの構文」
- 76 ページの「属性ルールの例」
- 76 ページの「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、80 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)

`DHMTL` : 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

書き換えられる 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(\q/focus.html\q,\qfocus\q);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(\q
gateway-URL
/http://abc.sesta.com/focus.html\q,\qfocus\q);return;">

</Form>
```

説明

指定されたページコンテンツを書き換えるには、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、書き換えが必要な 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書き換えられます。

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。

- 77 ページの「フォームルールの構文」
- 77 ページの「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 は値を書き換え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、80 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 /text.html を書き換えるとします。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書き換えられます。

入力タグ属性値の value は、出力に示されるように書き換えられます。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 を追加する方法で書き換えられます。80 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。リライタは、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、それを書き換えます。

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

```
<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 は値を書き換える必要のあるパラメータの名前です (必須)。

valuePatterns については、80 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 は書き換えられます。

名前が Test で始まるすべてのパラメータが書き換えられます。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、書き換え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書き換えられます。

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 は値を書き換える必要があるフォームのフィールドです。

valuePatterns は、書き換えが必要な部分文字列を示します。valuePatterns のあとに表示されるすべてのコンテンツは書き換えられます (省略可能、デフォルト "" は値全体の書き換え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。次に例を示します。

```
<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

ワイルドカードのアスタリスク (*) を使用して、書き換え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。* はすべてのテキストとの一致を示すため、valuePattern に続くテキストがなくなります。したがって、リライタが書き換えるテキストもなくなります。* は *abc のように、ほかの文字列と組み合わせて使用する必要があります。この場合、*abc に続くすべてのコンテンツが書き換えられます。

注 – アスタリスク (*) はルールのどのフィールドでも、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、セミコロンやコ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 個のアスタリスク (*) は書き換えられないフィールドと一致し、2 個のアスタリスク (**) は書き換えが必要なフィールドと一致します。

80 ページの「valuePatterns でのワイルドカードの使用」は、* ワイルドカードの使用例を示しています。

表 4–1 * ワイルドカードの使用例

URL	valuePatterns	説明
url1, url2, url3, url4	valuePatterns = "**, *, **, *"	** が書き換えられる部分を表すため、url1 と url3 が書き換えられます。
XYZABChttp://host1.sesta.com/dir1.html	valuePatterns = "*ABC"	http://host1.sesta.com/dir1.html の部分だけが書き換えられます。*ABC のあとのすべてを書き換える必要があります。
"0|dir1|dir2|dir3|dir4|test|url1	valuePatterns = "*|*|**|*|**|*|"	dir2、dir4、および url1 が書き換えられます。書き換え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 を直接解析できないため、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。

- 82 ページの「変数」
- 88 ページの「関数の引数」

変数

変数の汎用構文は次のとおりです。

<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。

- 82 ページの「URL 変数」
- 83 ページの「EXPRESSION 変数」
- 85 ページの「DHTML (ダイナミック HTML) 変数」
- 86 ページの「DJS (ダイナミック JavaScript) 変数」
- 87 ページの「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。

- 82 ページの「URL 変数の構文」
- 82 ページの「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` の値が書き換えられます (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 imgsrc3=imgsrc2;
//-->
</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書き換えられます。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 のみがプレフィックスとして追加されます。3 番目の変数 `imagsrc2` は、値が文字列ではなく別の JavaScript 値であるため書き換えられません。

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。

- 84 ページの「EXPRESSION 変数の構文」
- 84 ページの「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 variables
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、コンテンツを書き換えます。3 行目では、値が簡易 URL に書き換えられます。

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

```
<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/test.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 が書き換えられます。

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。

- 86 ページの「DJS 変数の構文」
- 86 ページの「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=\q/tmp/tmp.jpg\q;"
var dJSVar="var dJSimgsrc=\q../tmp/tmp.jpg\q;"
var dJSVar="var dJSimgsrc=
\qhttp://abc.sesta.com/tmp/tmp.jpg\q;"
```

ルール

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

出力

```
//DJS Var - need 2 rules
var dJSVar="var dJSimgsrc=\q gateway-URL
/http://abc.sesta.com/tmp/tmp.jpg\q;"var dJSVar="var dJSimgsrc=\q
gateway-URL/http
://abc.sesta.com/dir1/dir2/dir3/jscript/tmp/tmp.jpg\q;"
var dJSVar="var dJSimgsrc=\q gateway-URL/
http://abc.sesta.com/tmp/tmp.jpg\q;"
```

説明

ここでは、2 つのルールが必要です。最初のルールはダイナミック 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 変数の値が書き換えられます。

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。

- 87 ページの「SYSTEM 変数の構文」
- 88 ページの「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、doument.location、location.pathname、location.href、location.protocol、location.hostname、location.host、および 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 はそのページの URL です (省略可能、デフォルト * は任意のページを意味する)。

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 Var
alert(window.location.pathname);
//-->
</SCRIPT>
```

ルール

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

出力

```
</SCRIPT>
<SCRIPT LANGUAGE="Javascript">
<!--
//SYSTEM Var
alert(psSRAPRewriter_convert_pathname(window.location.pathname));
//-->
</SCRIPT>
```

説明

リライタは、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 を書き換えます。

関数の引数

値の書き換え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。

- 89 ページの「URL パラメータ」
- 90 ページの「EXPRESSION パラメータ」
- 92 ページの「DHTML パラメータ」
- 94 ページの「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、書き換え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、書き換え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書き換えるが、2 番目のパラメータは書き換えない、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。

- 89 ページの「URL パラメータの構文」
- 89 ページの「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、書き換え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、書き換え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書き換えるが、2 番目のパラメータは書き換えない、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 つのパラメータを書き換え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 つのパラメータが書き換えられます。2 番目のルールは、window.open 関数の最初のパラメータを書き換え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。

- 91 ページの「EXPRESSION パラメータの構文」
- 91 ページの「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、書き換え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、書き換え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書き換えられます。

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書き換え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書き換えます。

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。書き換えは、`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 ページをダイナミックに生成する `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。

- 92 ページの「DHTML パラメータの構文」
- 93 ページの「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、書き換え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、書き換え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書き換えられます。

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(\q<a href="/index.html">write</a><BR>\q)
document.writeln(\q<a href="index.html">writeln</a><BR>\q)
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(\q<a href="gateway-URL/
http://xyz.siroe.com/index.html">write</a><BR>\q)
document.writeln(\q<a href="gateway-URL/
http://xyz.siroe.com/test/rewriter/test1/
jscript/JSFUNC/index.html">writeln</a><BR>\q)
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書き換え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書き換え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書き換え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書き換え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書き換えられます。

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、次の項目から構成されています。

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、書き換え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書き換えられます。

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=\qhttp://abc.sesta.co
m\q"));
</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=\qgateway-URL/
http://abc.sesta.com\q"));
</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書き換える必要があることを指定します。JavaScript 内で、変数 top.location も書き換える必要があります。この変数は 2 番目のルールを使用して書き換えられます。

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。書き換えが必要な XML コンテンツは、2 つのカテゴリに分類されます。

- 95 ページの「タグテキスト」(タグの PCDATA または CDATA と同様)
- 96 ページの「属性」

タグテキスト

このルールは、タグ要素の PCDATA または CDATA を書き換えるためのものです。

この節は、次の項目から構成されています。

- 95 ページの「タグテキストの構文」
- 95 ページの「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/test.html</attribute><attribute>abc.html</attribute>
</xml>
```

説明

ページコンテンツの最初の行には97 ページの「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、書き換えは行われません。これを書き換えるには、`<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。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

`<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、80 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/string.html"/></xml>
```

説明

前述した例では、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、書き換えられます。80 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。75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解析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書き換えます。

```
<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>
```

デバッグログを使用したトラブルシューティング

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。リライタは画像を書き換え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

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.iplanet.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ーに、デバッグファイルのディレクトリを指定します。

   com.iplanet.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。

./psadmin start-sra-instance –u amadmin – f <*password file*> –N <*profile name*>– t <*gateway*>

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。99 ページの「デバッグファイル名」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 4–2 リライタのデバッグファイル

ファイル名	説明
RuleSetInfo	書き換え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。

表 4–2 リライタのデバッグファイル (続き)

ファイル名	説明
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、解析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、すべてのコンテンツが記録されます。warning モードと error モードでは、書き換え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、および書き換えら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、書き換え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*_OrginalPages.*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 スクレー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 スクレー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タ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

この節では、次の点を説明します。

- 書き換えが必要なコンテンツを含む簡単な HTML ページ
- コンテンツの書き換えに必要なルール
- 書き換えられた 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、書き換えら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、該当箇所で説明します。

注 – 太字で表示されている文は、書き換えら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

- 103 ページの「HTML 属性のサンプル」
- 107 ページの「HTML フォームのサンプル」
- 109 ページの「HTML アプレットのサンプル」

JavaScript

- 変数
  - 111 ページの「JavaScript URL 変数のサンプル」
  - 111 ページの「JavaScript コンテンツのサンプル」
  - 115 ページの「JavaScript DHTML 変数のサンプル」
  - 118 ページの「JavaScript DJS 変数のサンプル」
  - 120 ページの「JavaScript SYSTEM 変数のサンプル」

関数

- 121 ページの「JavaScript URL 関数のサンプル」
- 123 ページの「JavaScript EXPRESSION 関数のサンプル」
- 124 ページの「JavaScript DHTML 関数のサンプル」
- 126 ページの「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

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。

書き換え前の 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://..</a>
<br><br>
2. href <a href="https://host1.siroe.com">https://..</a>
<br><br>
3. href <a href="../images/logo.gif">../images/</a>
<br><br>
4. href <a href="images/logo.gif">images/..</a> <br><br>
5. href <a href="../../images/logo.gif">../../images/</a> <br><br>
Rewriting ends
</html>
```

ルール

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

書き換え後の HTML

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

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 は書き換えられます。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 <a href="gateway-URL/https://host1.siroe.com">https://..</a>
```

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

// 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とともにゲートウェイ 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 ディレクトリの下の 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とともにゲートウェイ 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とともにゲートウェイ 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 ディレクトリの下の 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

ここでは、HTML JavaScript トークンのサンプルの使用について説明します。

▼ 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 ソースを書き換えるためのルール (Rules for Rewriting JavaScript Source)**」セクションに追加します。
3. **Portal Server** 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. 端末ウィンドウからゲートウェイを再起動します。

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

書き換え前の HTML

```
<html>
<head>
Rewriting starts
<script language="javascript">
function Check(test,ind){
if (ind == \qblur\q)
{alert("testing onBlur")}
if (ind == \qfocus\q)
{alert("testing onFocus")}
}
</SCRIPT>
</head>
<body>
<form>
<input TYPE=TEXT SIZE=20 value=blur onAbort="Check
(\q/indexblur.html\q,\qblur\q);return;">
```

```
<input TYPE=TEXT SIZE=20 value=blur onBlur="Check
(\q/indexblur.html\q,\qblur\q);return;">
<input TYPE=TEXT SIZE=20 value=focus onFocus="Check
(\q/focus.html\q,\qfocus\q);return;">
<input TYPE=TEXT SIZE=20 value=focus onChange="Check
(\q/focus.html\q,\qfocus\q);return;">
<input TYPE=TEXT SIZE=20 value=focus onClick="Check
(\q/focus.html\q,\qblur\q);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。

書き換え後の HTML

```
<html>
<head>
Rewriting starts
<script language="javascript">
function Check(test,ind){
if (ind == \qblur\q)
{alert("testing onBlur")}
if (ind == \qfocus\q)
{alert("testing onFocus")}
}
</SCRIPT>
</head>
<body>
<form>
<input TYPE=TEXT SIZE=20 value=blur onAbort="Check
(\qgateway URL/portal-server-URL/indexblur.html\q,\qblur\q);return;">
<input TYPE=TEXT SIZE=20 value=blur onBlur="Check
(\qgateway URL/portal-server-URL/indexblur.html\q,\qblur\q);return;">
<input TYPE=TEXT SIZE=20 value=focus onFocus="Check
(\qgateway URL/portal-server-URL/focus.html\q,\qfocus\q);return;">
<input TYPE=TEXT SIZE=20 value=focus onChange="Check
(\qgateway URL/portal-server-URL/focus.html\q,\qfocus\q);return;">
<input TYPE=TEXT SIZE=20 value=focus onClick="Check
```

```
(\qgateway URL/portal-server-URL/focus.html\q,\qblur\q);return;">
```

// このサンプルではすべての文が書き換えられます。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。リライタは JavaScript トークンを検出し、あとの処理のために JavaScript 関数ルールに渡します。サンプルの 2 番目のルールは、書き換えるパラメータをリライタに伝えます。

```
</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

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** 属性を書き換えるためのルール **(Rules for Rewriting HTML Attributes)**」セクションに追加します。
4. **Portal Server** 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. 端末ウィンドウからゲートウェイを再起動します。

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

書き換え前の 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|"/>
```

書き換え後の 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 は書き換えられます。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 のあとの URL が書き換えられます。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、80 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 は書き換えられません。

```
</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 は書き換えられます。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 は書き換えられません。

```
</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

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** 属性を書き換えるためのルール **(Rules for Rewriting HTML Attributes)**」セクションに追加します。

3. **Portal Server** 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。

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

書き換え前の 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*" />
```

書き換え後の HTML

```
<HTML>
Rewriting starts
<BR>
<APPLET codebase=gateway-URL/portal-server-URL
/rewriter/HTML/applet/appletcode=RewriteURLinApplet.class archive=/test>
```

<Attribute 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 は書き換えられます。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 は書き換えられます。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 は書き換えられます。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 は書き換えられます。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

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** ソースを書き換えるためのルール **(Rules for Rewriting JavaScript Source)**」セクションに追加します。
4. **Portal Server** 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。

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

書き換え前の 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>
<img src="images/logo.gif">
<br>
Image
</body>
<br>
Rewriting ends
</html>
```

ルール

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

書き換え後の 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.jpg";
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 のあとにパスが追加されます。

```
//-->
</SCRIPT>
<br>
Testing JavaScript variables!
<br>
<img src="gateway URL/portal-server-URL/rewriter
/JavaScript/variables/url/images/logo.gif">
```

default_gateway_ruleset に <Attribute name="src"/> ルールが定義されているので、この行は書き換えられます。

```
<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

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** ソースを書き換えるためのルール **(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. **Portal Server** 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。

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

書き換え前の 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"/>
```

書き換え後の 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、必要に応じて書き換えられます。

```
document.write("<A HREF="+expvar+">EXPRESSION</A><P>")
```

// 前の文の書き換え後の値 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、直接書き換えられます。

```
document.write("<A HREF="+expvar+">EXPRESSION</A><P>")
```

// 前の文の書き換え後の値 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

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** ソースを書き換えるためのルール **(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)。**Portal Server** 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。

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

書き換え前の 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>
<img src="images/logo.gif">IMAGE
</body>
</html>
```

ルール

```
<Variable name="DHTML">dhtmlVar</Variable>
```

書き換え後の 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 の右側をダイナミック 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。ダイナミック 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 属性の値が書き換えられます。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、最終的な書き換え出力となります。

```
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 ルールは右側をダイナミック 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、文は次のように書き換えられます。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 ルールは右側のダイナミック 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 が書き換えられることを確認してください。

```
//-->
</SCRIPT>
<br><br>
Testing DHTML Variables
<br><br>
<img src="gateway-URL/portal-server-URL/
rewriter/JavaScript/variables/dhtml/images/logo.gif">
```

`default_gateway_ruleset` に `<Attribute name="src"/>` ルールが定義されているので、この行は書き換えられます。

```
<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

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 ソースを書き換えるためのルール (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)。Portal Server 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。

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

書き換え前の HTML ページ

```
<html>
<head>
<title>Dynamic JavaScript Variable Test Page</title>
</head>
<body>
<script LANGUAGE="Javascript">
<!--
var dJSVar="var dJSimgsrc=\q/tmp/tmp/jpg\q;"
var dJSVar="var dJSimgsrc=\q../../../tmp/tmp/jpg\q;"
var dJSVar="var dJSimgsrc=\qhttp://abc.sesta.com/tmp/tmp/jpg\q;"
//-->
</SCRIPT>
<br>
Testing Dynamic JavaScript Variables
<br>
<img src="images/logo.gif">
<br>
Image
</body>
</html>
```

ルール

```
<Variable name="dJSVar" type="DJS"/>
<Variable name="dJSimgsrc “ type=URL"/>
```

書き換え後の HTML ページ

```
<html>
<head>
<title>Dynamic JavaScript Variable Test Page</title>
</head>
<body>
<script LANGUAGE="Javascript">
<!--
var dJSVar="var dJSimgsrc=\qgateway-URL
/portal-server-URL/tmp/tmp/jpg\q;"
var dJSVar="var dJSimgsrc=\qgateway-URL
/portal-server-URL/rewriter/tmp/tmp/jpg\q;"
var dJSVar="var dJSimgsrc=\qgateway-URL
/http://abc.sesta.com/tmp/tmp/jpg\q;"
```

// 上のすべての文は、ゲートウェイ URL と Portal Server URL で書き換えられます。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 の右側をダイナミック 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、文は次のように書き換えられます。

```
//-->
</SCRIPT>
<br>
Testing Dynamic JavaScript Variables
<br>
<img src="gateway-URL/portal-server-URL
/rewriter/JavaScript/variables/djs/images/logo.gif">
```

default_gateway_ruleset に <Attribute name="src"/> ルールが定義されているので、この行は書き換えられます。

```
<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

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/system/system.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** ソースを書き換えるためのルール **(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. **Portal Server** 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。

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

書き換え前の 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"/>
```

書き換え後の HTML

```
<html>
<head>
<title>JavaScript SYSTEM Variables Test Page</title>
</head>
<body>
<SCRIPT>
convertsystem 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、必要に応じて書き換えます。

```
//-->
</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

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 ソースを書き換えるためのルール (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)。**Portal Server** 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。

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

書き換え前の 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"/>
```

書き換え後の 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

1 このサンプルには次の場所からアクセスできます。

```
<portal-install-location>/SUNWportal/samples/rewriter
```

2 このサンプルで指定されているルールを、default_gateway_ruleset の「**JavaScript** ソースを書き換えるためのルール **(**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 **Portal Server** 管理コンソールを使用して、リライタサービスの default_gateway_ruleset を編集します。

4 ゲートウェイを再起動します。

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

書き換え前の 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"/>
```

書き換え後の 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書き換え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

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 ソースを書き換えるためのルール (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. **Portal Server** 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。

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

書き換え前の HTML ページ

```
<html>
<head>
Testing JavaScript DHTML Functions
<br>
<br>
<script>
<!--
document.write(\q<a href="/index.html">write</a><BR>\q)
document.writeln(\q<a href="index.html">writeln</a><BR>\q)
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"/>
```

書き換え後の HTML ページ

```
<html>
<head>
Testing JavaScript DHTML Functions
<br>
<br>
<script>
<!--
document.write(\q<a href="gateway-URL
/portal-server-URL/index.html">write</a><BR>\q)
```

// 最初のルールは、DHTML JavaScript 関数 `document.write` の最初のパラメータを書き換え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、書き換えが必要な文はこのルールによって決定されます。

```
document.writeln(\q<a href="gateway-URL
/portal-server-URL/rewriter/JavaScript/functions/dhtml/index.html">writeln</a><BR>\q)
```

// 2 番目のルールは、DHTML JavaScript 関数 document.writeln の最初のパラメータを書き換え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、書き換え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、上の文は書き換えられません。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書き換え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

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 ソースを書き換えるためのルール (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)。**Portal Server** 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。

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

書き換え前の HTML ページ

```
<html>
Test for JavaScript DJS Functions
<br>
<script>
menu.addItem(new NavBarMenuItem("All Available
Information","JavaScript:top.location=\qhttp://abc.sesta.com\q"));
//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"/>
```

書き換え後の HTML ページ

```
<html>
Testing JavaScript DJS Functions
<br>
<script>
menu.addItem(new NavBarMenuItem
("All Available Information","javaScript:top.location=
\qgateway-URL/http://abc.sesta.com\q"));
```

`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 を書き換える必要があります。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書き換え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、この変数の値を書き換える場合に必要となります。2 番目のルールは、JavaScript 変数 `top.location` の値を書き換え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 が書き換えられます。

```
//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書き換える必要があることを指定しますが、この文は書き換えられません。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

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** ソースを書き換えるためのルール **(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. **Portal Server** 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。

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

書き換え前の 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|"/>
```

書き換え後の 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、書き換えられます。Attribute name は href、tag は check、valuePatterns は 1234 です。valuePatterns よりもあとの文字列は書き換えられます。valuePatterns の詳細については、80 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>
```

ケーススタディー

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ーでは、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ーは、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&amp;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

131 ページの「説明」は、サンプルルールセットとケーススタディーの間のマッピングを示しています。

表 4–3 サンプルルールセットとケーススタディーのマッピング

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
`var g_szVirtualRoot= "http://abc.siroe.com/mailweb";`	<Variable name="URL"> g_szVirtualRoot </Variable>	`var g_szVirtualRoot= "http://gateway.sesta.com /http://abc.siroe.com/mailweb";`	`g_szVirtualRoot` は単一の URL を値に持つ変数です。 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。
`src="/destin_files/logo-ie5.gif"`	<Attribute name="src" />	`src="http://gateway.sesta.com/ http://abc.siroe.com/ destin_files/logo-ie5.gif`	src 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、`src` という名前の属性をすべて検索し、その属性の値を書き換えるようにリライタに指示します。

表 4–3 サンプルルールセットとケーススタディーのマッピング (続き)

ページコンテンツ	適用されるルール	リライタの出力	説明
href="http://abc.siroe.com /mailclient/destin/Inbox/ ?Cmd=contents&Page=1"	<Attribute name="href"/>	href="http://gateway.sesta.com/ http://abc.siroe.com /mailclient/destin/ 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、その属性の値を書き換えるようにリライタに指示します。

注 – ルールセットを適用する優先順位は、ホスト名 - サブドメイン - ドメインです。

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 サブドメイン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。

1. 「保存」をクリックして終了します。
2. 端末ウィンドウからゲートウェイを再起動します。

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

Outlook Web Access 用のルールセット

Secure Remote Access サーバーでは、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

1 **Portal Server** 管理コンソールに管理者としてログインします。

2 「**Secure Remote Access**」タブを選択し、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。

3 「**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。

次に例を示します。

```
exchange.domain.com|exchange_2000sp3_owa_ruleset.
```

パブリックフォルダの使用

Exchange 側では、パブリックフォルダは NTLM 認証を使用するように設定されています。これは、HTTP 基本認証を使用するように変更する必要があります。

この変更を行うには、Exchange サーバーで「コントロールパネル」 >「管理ツール」の順に選択し、「インターネットインフォメーションサービス」を開きます。

「既定の Web サイト」の下に、パブリックフォルダに関する「パブリック」というタブがあります。タブを右クリックし、プロパティーを選択します。「ディレクトリセキュリティー」タブをクリックします。「匿名アクセスと認証」コントロールパネルで「編集」を選択します。「基本認証」チェックボックスのみを選択し、ほかのすべてのチェックボックスの選択を解除します。

6.x と 3.0 のルールセットのマッピング

次の表は、Secure Remote Access サーバーのリライタ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 4–4 SP3 のルールのマッピング

リライタ 6.0 の DTD 要素	リライタ 3.0 リストボックス名
HTML コンテンツのルール	
Attribute: URL	HTML 属性の書き換え
Attribute: DJS	JavaScript を含む HTML 属性の書き換え
Form	フォーム入力タグリストの書き換え
アプレット	アプレット/オブジェクトパラメータ値リストの書き換え
JavaScript コンテンツのルール	
Variable: URL	URL タイプの JavaScript 変数の書き換え
Variable: EXPRESSION	JavaScript 変数関数の書き換え
Variable: DHTML	HTML タイプの JavaScript 変数の書き換え

表 4–4　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
Variable: DJS	JavaScript タイプの JavaScript 変数の書き換え
Variable: SYSTEM	JavaScript システム変数の書き換え
Function: URL	JavaScript 関数パラメータの書き換え
Function: EXPRESSION	JavaScript 関数パラメータ関数の書き換え
Function: DHTML	HTML タイプの JavaScript 関数パラメータの書き換え
Function: DJS	JavaScript タイプの JavaScript 関数パラメータの書き換え
XML コンテンツのルール	
Attribute: URL	XML ドキュメントの属性値の書き換え
TagText	XM1 ドキュメントのテキストデータの書き換え
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。	

◆ ◆ ◆ 第 5 章

ネットファイルの操作

この章では、ネットファイルおよびその操作について説明します。ネットファイルの設定については、第 14 章ネットファイルの設定を参照してください。

ネットファイルの概要

ネットファイルは、ユーザーがリモートのファイルシステムとディレクトリにアクセスして操作できるようにするファイルマネージャーアプリケーションです。

Secure Remote Access のネットファイルコンポーネントは、Java2 アプレットとして使用できます。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。

ネットファイル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
- ネットファイル環境内でのメール機能
- 現在のネットファイルセッション情報の保存
- ファイルのドラッグ＆ドロップ

サポートされるファイルアクセスプロトコル

ネットファイルでは 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。ネットファイルには、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、ネットファイル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、Portal Server 管理コンソール (PSConsole) 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。

- ネットファイルでは、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 5–1 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 Windows NT 4.0 の MS FTP サーバー 4.0 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 vsFTPd 1.2.0
NFS	Solaris 2.6 以降
jCIFS	Windows 95/98/NT/2000/ME/XP

注 – ネットファイルを使用して 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。

Novell 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。

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。

▼ ネットファイルポリシーを作成する

1. **Portal Server** 管理コンソールに管理者としてログインします。
2. 「**Secure Remote Access**」タブを選択し、「ネットファイル」タブを選択します。
3. 「**DN** を選択」ドロップダウンボックスで、「組織」、「ロール」、または「ユーザー」を選択します。
4. ホストおよびサービスへのアクセスや拒否に関する権限を設定します。
5. 「保存」をクリックします。
6. ゲートウェイを再起動します。

◆ ◆ ◆ 第 6 章

ネットレットの操作

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、ネットレット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。ネットレットの設定については、第 11 章ネットレット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。

ネットレットの概要

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ネットレットを設定すると、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。

ネットレットを使用することで、インターネットなどのセキュリティー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、ネットレット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。

ネットレットアプレットを使用する場合は、ブラウザのポップアップブロッカを無効にするようにユーザーに通知してください。

ネットレットのコンポーネント

140 ページの「ネットレットのコンポーネント」は、ネットレットで使用される各種コンポーネントを示しています。

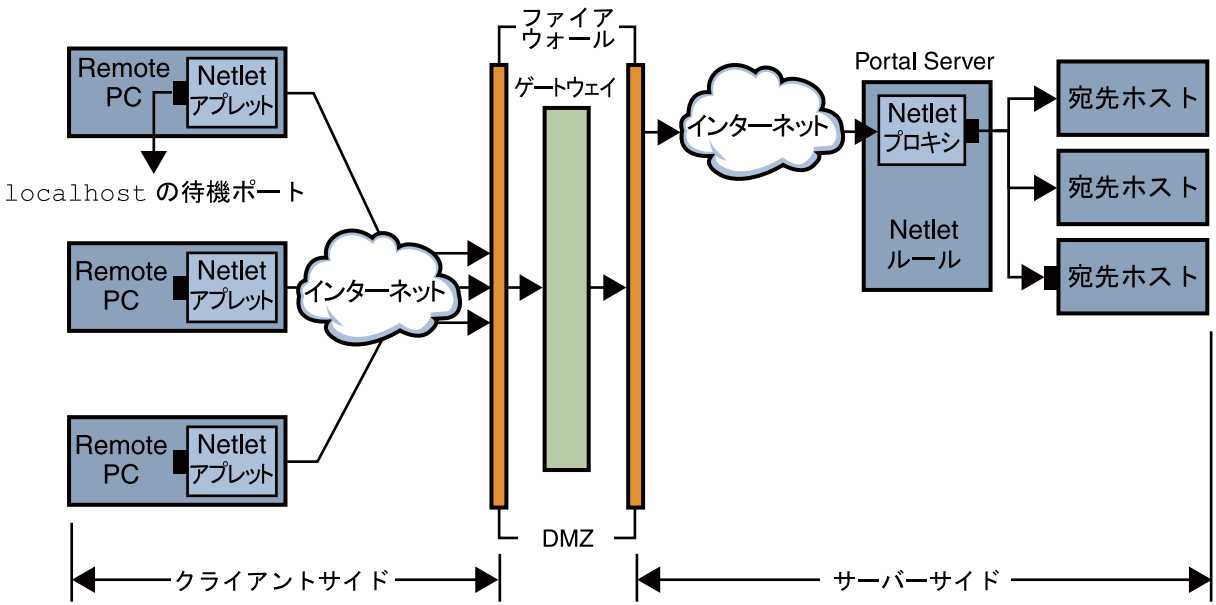


図 6–1 ネットレットのコンポーネント

localhost の待機ポート

これはネットレット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。

ネットレットアプレット

ネットレットアプレットは、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、ネットレットアプレットは、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の詳細については、146 ページの「ルールのタイプ」を参照してください。

Sun Ray 環境でのネットレットの実行については、160 ページの「Sun Ray 環境でのネットレットの実行」を参照してください。

ネットレットルール

ネットレットルールでは、クライアントマシンで実行する必要のあるアプリケーションが、対応する接続先ホストにマッピングされます。つまりネットレットは、ネットレット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、セキュリティーが向上します。

管理者はネットレット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、接続先ポート、接続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ネットレット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すみやかに決定されます。詳細については、143 ページの「ネットレットルールの定義」を参照してください。

ネットレットプロバイダ

これはネットレットの 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、デスクトップネットレットプロバイダで、ダイナミックルールの接続先ホストを指定できます。143 ページの「ネットレットルールの定義」を参照してください。

ネットレットプロキシ(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セキュリティー保護されたトンネルを保証します。ネットレット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。ネットレットプロキシについては、52 ページの「ネットレットプロキシの使用」を参照してください。

ネットレットの使用例

ネットレット使用時には、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ネットレット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ネットレット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、ネットレット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。ネットレットアプレットは、ユーザーがネットレット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の詳細については、143 ページの「ネットレットルールの定義」を参照してください。
3. ネットレットはネットレットルールで定義されたローカルポートで待機します。
4. ネットレットはリモートクライアントとホストの間で、ネットレットルールで指定されたポートを使用するチャネルを確立します。

ネットレットの操作

ネットレット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。146 ページの「ルールのタイプ」を参照してください。
2. Portal Server 管理コンソールから、ネットレットサービスのオプションを設定します。ネットレットの設定については、第 11 章ネットレット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、およびユーザーの詳細については、『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取得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 のセキュリティーによって、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。ネットレットは、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。

ネットレットルールの定義

ネットレットの設定はネットレットルールによって定義されます。このルールは、Portal Server 管理コンソールの「Secure Remote Access」設定タブを使用して設定されます。ネットレットルールは組織、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。ネットレット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。

注意 – ネットレットルールはマルチバイトエントリをサポートしません。ネットレットルールのどの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。

ネットレットルールには 64000 を超えるポート番号を指定できません。

143 ページの「ネットレットルールの定義」は、ネットレットルールのフィールドを示しています。

表6–1 ネットレットルールのフィールド

パラメータ	説明	値
ルール名	このネットレット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。	
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。	選択した暗号化方式は、ネットレット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。 デフォルト: ネットレット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。
リモートアプリケーションの URL	URL ユーザーがネットレット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。 相対 URL を指定する必要があります。	ネットレットルールによって呼び出されるアプリケーションへの URL。たとえば、`telnet://localhost:30000` などです。 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。 `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。
ダウンロードアプレットの有効化	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。	■ *Client Port* はクライアントの接続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。各ルールに一意の `local port` を指定します。 ■ *Server Host* 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。 ■ *Server Port* 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。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す。
拡張セッションを有効	ネットレット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。	ネットレットだけがアクティブで、ほかのポータルアプリケーションがアイドル状態の場合にポータルセッションを持続する場合は、この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、このオプションは選択されていません。

表 6–1 ネットレットルールのフィールド (続き)

パラメータ	説明	値
ローカルポートと宛先サーバーポートのマップ	ローカルポート	ネットレットが待機するクライアントのポート。 *local-port* 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。 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、151 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。 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。
	接続先ホスト	ネットレットが待機するクライアントのポート。 ネットレット接続の受信者。 *host* : ネットレット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 `siroe.mycompany.com` 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。 *local-port* 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。 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、151 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。 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。 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接続先ポートを指定する必要があります。構文については、151 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、152 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。 `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、デスクトップのネットレットプロバイダで、必要な接続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。 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。

表 6–1 ネットレットルールのフィールド (続き)

パラメータ	説明	値
接続先ポート	接続先ホスト上のポート。 ホストと接続先ホストのほかに、接続先ポートを指定する必要があります。 複数の接続先ホストがある場合は、複数の接続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。 ポート番号間のプラス (+) 記号は、単一の接続先ホストに対する代替ポートを表します。 異なる接続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。 この例では、ネットレットは `port1`、`port2`、および `port3` を順番に使用して、指定された最初の接続先ホストへの接続を試みます。これに失敗した場合、ネットレットは `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。 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。	

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.iplanet.am.jassproxy.trustAllServerCerts=true
```

Portal Server 上の次のプロパティーファイルに追加します。

```
/etc/opt/SUNWam/config/AMConfig.instance-name .properties
```

ルールのタイプ

ルールで接続先ホストがどのように指定されているかにより、ネットレットルールは 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接続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、ユーザーは必要な接続先ホストを指定することができません。次の例では、sesta は接続先ホストです。

ルール名	暗号化方式	URL	ダウンロードアプレットの有効化	拡張セッションを有効	ローカルポートと宛先サーバーポートのマップ
ftpstatic	SSL_RSA_WITH_RC 4_128_MD5	null	false	true	■ ローカルポート 30021 ■ 接続先ホスト: sesta ■ 接続先ポート: 21

複数の接続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、151 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、接続先ホストはルールの一部として指定されません。ユーザーはネットレットプロバイダで必要な接続先ホストを指定できます。次の例では、TARGET は接続先ホストの可変部分です。

ルール名	暗号化方式	リモートアプリケーションの URL	ダウンロードアプレットの有効化	拡張セッションを有効	ローカルポートと宛先サーバーポートのマップ
ftpdynamic	SSL_RSA_WIT H_RC4_128_MD5	null	チェックボックスにチェックマークを付ける	チェックボックスにチェックマークを付ける	■ ローカルポート 30021 ■ 接続先ホスト: TARGET ■ 接続先ポート: 21

暗号化方式

暗号化方式に基づいて、ネットレット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、ネットレット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	リモートアプリケーションの URL	ダウンロードアプレットの有効化	拡張セッションを有効	ローカルポートと宛先サーバーポートのマップ
Telnet	SSL_RSA_WITH_RC4 _128_SHA	null	チェックボックスにチェックマークを付ける	チェックボックスにチェックマークを付ける	■ ローカルポート 30000 ■ 接続先ホスト: TARGET ■ 接続先ポート: 23
	SSL_RSA_WITH_RC4 _128_MD5				

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、ネットレットルールの一部として設定されている方式だけです。

ネットレットでサポートされる暗号化方式のリストについては、148 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 ネットレット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	リモートアプリケーションの URL	ダウンロードアプレットの有効化	拡張セッションを有効	ローカルポートと宛先サーバーポートのマップ
Telnet	SSL_RSA_WITH_RC4_128_MD5	null	チェックボックスにチェックマークを付ける	チェックボックスにチェックマークを付ける	■ ローカルポート 30000 ■ 接続先ホスト: TARGET ■ 接続先ポート: 23

ネットレットでサポートされる暗号化方式のリストについては、148 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

148 ページの「サポートされる暗号化方式」は、ネットレット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 6–2 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
TLS_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_256_CBC_SHA

下位互換性

旧バージョンの Portal Server は、ネットレット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	リモートアプリケーションの **URL**	ダウンロードアプレットの有効化	拡張セッションを有効	ローカルポートと宛先サーバーポートのマップ
Telnet		`telnet://localhost:30000`	チェックボックスにチェックマークを付けない	チェックボックスにチェックマークを付ける	■ ローカルポート 30000 ■ 接続先ホスト: TARGET ■ 接続先ポート: 23

これは次のように解釈されます。

ルール名	暗号化方式	リモートアプリケーションの URL	ダウンロードアプレットの有効化	拡張セッションを有効	ローカルポートと宛先サーバーポートのマップ
Telnet	デフォルト暗号化方式	telnet://localhost:30000	チェックボックスにチェックマークを付けない	チェックボックスにチェックマークを付ける	■ ローカルポート 30000 ■ 接続先ホスト: TARGET ■ 接続先ポート: 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。

注 – ネットレットルールには 64000 を超えるポート番号を指定できません。

ネットレットルールの例

ここでは、ネットレットルールの例をいくつか示し、ネットレット構文がどのように機能するかについて説明します。

- 150 ページの「基本的なスタティックルール」
- 151 ページの「複数ホスト接続のスタティックルール」
- 153 ページの「URL を呼び出すダイナミックルール」
- 154 ページの「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	リモートアプリケーションの URL	アプレットのダウンロード	拡張セッション	ローカルポートと宛先サーバーポートのマップ
myrule	SSL_RSA_WITH_RC4_128_MD5	null	チェックボックスにチェックマークを付けない	true	■ ローカルポート 1111 ■ 接続先ホスト: sesta ■ 接続先ポート: 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、ネットレット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、ネットレットが接続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 は接続先ホストの接続用ポート番号です。この例では、既知の Telnet ポートです。

デスクトップネットレットプロバイダにはリンクが表示されませんが、ネットレット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	リモートアプリケーションの URL	ダウンロードアプレットの有効化	拡張セッションを有効	ローカルポートと宛先サーバーポートのマップ
myrule	SSL_RSA_WITH_RC4_128_MD5	null	チェックボックスにチェックマークを付けない	チェックボックスにチェックマークを付ける	■ ローカルポート: 1111 ～ 1234 ■ 接続先ホスト: sesta-siroe ■ 接続先ポート: 23

各表記の意味は次のとおりです。

23 は接続先ホストの接続用ポート番号、つまり、Telnet の予約ポートです。

1111 は、ネットレットが最初の接続先ホスト siroe 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。

1234 は、ネットレットが 2 番目の接続先ホスト siroe 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。

このルールの最初の 6 つのフィールドは、150 ページの「基本的なスタティックルール」と同じです。2 番目の接続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、新しい接続先ホストごとに、`local port`、`destination host`、`destination port` の 3 つのフィールドを追加する必要があります。

注 – 各接続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。ネットレットプロバイダはリンクを表示しませんが、ネットレットは指定された 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、ネットレットは 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	リモートアプリケーションの **URL**	ダウンロードアプレットの有効化	拡張セッションを有効	ローカルポートと宛先サーバーポートのマップ
gojoe	`SSL_RSA_WITH_RC4_128_MD5`	/gojoe.html	■ クライアントポート: 8000 ■ サーバーホスト: gojoeserver ■ サーバーポート: 8080	チェックボックスにチェックマークを付ける	■ ローカルポート 10491 ■ 接続先ホスト: siroe+sesta ■ 接続先ポート: 35+26+491-35+491

各表記の意味は次のとおりです。

`10491` は、ネットレットが接続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。

ネットレットはポート 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、ネットレットはポート 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、単一の接続先ホストに対する代替ポートを表します。

異なる接続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。

注 – 指定された一連のホストへの接続が順次に試行されます。たとえば、siroe+sesta が指定されたルールの場合、最初に siroe への接続が試行されます。この接続に失敗すると、次に sesta への接続が試行されます。ルール内で最初にリストされたホストがアクティブネットワーク内で物理的に使用できない場合、ルール内の使用不能ホストの数が多いほど、次の使用可能ホストへの接続にかかる時間が長くなります。

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、目的の接続先ホストを設定できるため、ネットレット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	リモートアプリケーションの URL	ダウンロードアプレットの有効化	拡張セッションを有効	ローカルポートと宛先サーバーポートのマップ
myrule	SSL_RSA_WITH_RC4_128_MD5	telnet://localhost:30000	チェックボックスにチェックマークを付けない	チェックボックスにチェックマークを付ける	■ ローカルポート 30000 ■ 接続先ホスト: TARGET ■ 接続先ポート: 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。

true は、ネットレット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、ネットレット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、ユーザーがネットレットプロバイダを使用して接続先ホストを設定する必要があることを示します。

23 はネットレットで開かれる接続先ホストのポートです。この例では、既知の Telnet ポートです。

▼ ルールの追加後にネットレットを実行する

このルールを追加したあとに、ユーザーはネットレットを目的どおりに稼動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** デスクトップのネットレットプロバイダセクションで、「編集」をクリックします。
   新しいネットレットルールが、「新しいターゲットの追加」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、接続先ホスト名を入力します。
3. 変更を保存します。
   デスクトップに戻ります。デスクトップのネットレット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、ネットレットルールで指定した URL が表示されます。

注 – 同じルールに複数の接続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のみがアクティブです。

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	リモートアプリケーションの URL	ダウンロードアプレットの有効化	拡張セッション	ローカルポートと宛先サーバーポートのマップ
gojoe	SSL_RSA_WITH_RC4_128_MD5	/gojoe.html	■ クライアントポート: 8000 ■ サーバーホスト: gojoeserver ■ サーバーポート: 8080	チェックボックスにチェックマークを付ける	■ ローカルポート 3399 ■ 接続先ホスト: TARGET ■ 接続先ポート: 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 は、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:gojoeserver:8080 は、クライアントでアプレットを受け取る接続先ポートがポート 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。

Extended Session (true) は、ネットレット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、ネットレット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、ユーザーがネットレットプロバイダを使用して接続先ホストを設定する必要があることを示します。

58 はネットレットで開かれる接続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 は接続先ホストが自分のトラフィックを待機するポートです。ネットレットは新しいアプレットの情報をこのポートに渡します。

ネットレットルールの例

156 ページの「ネットレットルールの例」は、いくつかの一般的なアプリケーションのネットレットルールの例を示しています。

この表には、ネットレットルールごとに 7 項目あり、それぞれ、「ルール名」、「リモートアプリケーションの URL」、「ダウンロードアプレットの有効化」、「ローカルポート」、「接続先ホスト」、「接続先ポート」の各フィールドに対応しています。最後の列は、ルールの説明を示します。

注 – 156 ページの「ネットレットルールの例」には、ネットレットルールの暗号化方式および拡張セッションのフィールドは示されていません。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 6–3 ネットレットルールの例

ルール名	リモートアプリケーションの URL	ダウンロードアプレットの有効化	ローカルポートと宛先サーバーポートのマップ	説明
IMAP	null	チェックボックスにチェックマークを付けない	■ ローカルポート 10143 ■ 接続先ホスト: imapserver ■ 接続先ポート: 143	クライアント側のネットレット `local port` はサーバー側の `destination port` 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	チェックボックスにチェックマークを付けない	■ ローカルポート 10025 ■ 接続先ホスト: smtpserver ■ 接続先ポート: 25	
Lotus Web クライアント	null	チェックボックスにチェックマークを付けない	■ ローカルポート 80 ■ 接続先ホスト: lotus-server ■ 接続先ポート: 80	このルールでは、ネットレットがポート 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 6–3 ネットレットルールの例 *(続き)*

ルール名	リモートアプリケーションの URL	ダウンロードアプレットの有効化	ローカルポートと宛先サーバーポートのマップ	説明
Lotus Notes 非 Web クライアント	null	チェックボックスにチェックマークを付けない	■ ローカルポート 1352 ■ 接続先ホスト: lotus-domino ■ 接続先ポート: 1352	このルールを使用すると、Lotus Notes クライアントはネットレットを通じて 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。ネットレットを使用する場合、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。 ■ 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。 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。

表 6–3 ネットレットルールの例 *(続き)*

ルール名	リモートアプリケーションの URL	ダウンロードアプレットの有効化	ローカルポートと宛先サーバーポートのマップ	説明
Microsoft Outlook および Exchange Server Windows NT、Windows 2000、および Windows XP では、この設定は機能しません。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	チェックボックスにチェックマークを付けない	■ ローカルポート 135 ■ 接続先ホスト: exchange ■ 接続先ポート: 135	このルールでは、ネットレット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。 クライアントマシン上で次の操作を行います。 ■ 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。 ■ 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。 ■ Windows NT4 では、このファイルは `\\WinNT\\System32\\drivers\\etc\\Hosts` に格納されています。 エントリは次のようになります。 `127.0.0.1 exchange exchange.company.com` 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 クライアントはネットレット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。

表 6–3 ネットレットルールの例 (続き)

ルール名	リモートアプリケーションの URL	ダウンロードアプレットの有効化	ローカルポートと宛先サーバーポートのマップ	説明
FTP	null	チェックボックスにチェックマークを付けない	■ ローカルポート 30021 ■ 接続先ホスト: *your-ftp_server.your-domain* ■ 接続先ポート: 21	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セキュリティー保護されたリモート 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。 ネットレット 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。 ネットレット接続でダイナミック FTP を使用できます。
Netscape 4.7 Mail Client	null	チェックボックスにチェックマークを付けない	■ ローカルポート 30143、30025 ■ 接続先ホスト: TARGET ■ 接続先ポート: 10143	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。 IMAP または受信メールについては `localhost:30143` SMTP または発信メールについては `localhost:30025`
Graphon	third_party/xsession_start.html	チェックボックスにチェックマークを付ける	■ ローカルポート 10491 ■ 接続先ホスト: TARGET ■ 接続先ポート: 491	ネットレットを通じて 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。
Citrix	third_party/citrix_start.html	チェックボックスにチェックマークを付ける	■ ローカルポート 1494 ■ 接続先ホスト: TARGET ■ 接続先ポート: 1494	ネットレットを通じて 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。

表 6–3 ネットレットルールの例 (続き)

ルール名	リモートアプリケーションの URL	ダウンロードアプレットの有効化	ローカルポートと宛先サーバーポートのマップ	説明
Remote Control	third_party/pca_start.html	チェックボックスにチェックマークを付ける	■ ローカルポート 5631 5632 ■ 接続先ホスト: TARGET TARGET ■ 接続先ポート: 5631 5632	ネットレットを通じて 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。

ネットレットのログ情報

ネットレットアプレットまたは jws のクライアント側ログは、クライアントの Java コンソールに表示されます。

ネットレットのサーバー側ログは、
/var/opt/SUNWportal/portals/<portal_ID>/logs/<INSTANCE_ID> ディレクトリの下にある portal.0.0.log ファイルに表示されます。

Sun Ray 環境でのネットレットの実行

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[\qkey\q] = \qvalue\q;
function retrieveKeyValues() {
      KEY_VALUES = new Object();
      var queryString  = \q\q + this.location;
      queryString = unescape(queryString);
      queryString = queryString.substring((queryString.indexOf(\q?\q)) + 1);
      if (queryString.length < 1) {
          return false; }
```

```
      var keypairs = new Object();
      var numKP = 0;
      while (queryString.indexOf(\q&\q) > -1) {
        keypairs[numKP] = queryString.substring(0,queryString.indexOf(\q&\q));
        queryString = queryString.substring((queryString.indexOf(\q&\q)) + 1);
        numKP++;
      }
      // クエリー文字列に最後の 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(\q=\q));
        keyValue = keypairs[i].substring((keypairs[i].indexOf(\q=\q)) + 1);
        while (keyValue.indexOf(\q+\q) > -1) {
          keyValue = keyValue.substring(0,keyValue.indexOf(\q+\q)) + \q \q
           + keyValue.substring(keyValue.indexOf(\q+\q) + 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[\q" + serverPort +"\q]";
    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(\q1494\q)
          + ">\\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>
```

パート II

Secure Remote Access サーバーの設定

ほとんどの属性は、Portal Server 管理コンソールの「Secure Remote Access」タブに表示されるオプションを使って設定できます。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。

Secure Remote Access に関連する属性は、組織、ロール、およびユーザーのレベルで設定できます。ただし、次のような例外があります。

- 競合の解決レベルは、ユーザーレベルでは設定できません。30 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。

組織の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、組織を新規で追加した場合にだけ適用されます。

このパートは、次の章から構成されています。

◆ ◆ ◆ 第 7 章

7 Secure Remote Access サーバーのアクセス制御の設定

この章では、Sun Java System Portal Server 管理コンソールから、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。

アクセス制御の設定

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。

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。

SRA ソフトウェアのアクセス制御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。

アクセス制御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。ユーザーがほかの

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。

▼ アクセス制御を設定する

1. **Portal Server** 管理コンソールに管理者としてログオンします。
2. 「**Secure Remote Access**」タブを選択します。
3. 「アクセス制御」タブを選択します。
4. 次の属性を変更します。

属性名	説明
COS 優先順位	属性値を継承するかどうかの決定に使用される値を指定します。この属性の詳細については、『Sun Java System Directory Server 管理ガイド』を参照してください。
セッションごとのシングルサインオン	セッションでのシングルサインオンを有効にする場合は、「有効」チェックボックスにチェックマークを付けます。
シングルサインオンを無効にするホスト	ホスト名を `abc.siroe.com` の形式で指定します。
許可される認証レベル	許可される認証レベルを入力します。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。デフォルト値は * です。
アクセスを許可/拒否する URL	URL フィールドに、ゲートウェイ経由でのアクセスを許可または拒否する URL を入力します。URL の入力形式は `http://abc.siroe.com` です。「アクション」ドロップダウンリストで、「許可」または「拒否」オプションをクリックします。 `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。 ゲートウェイはアクセスが拒否された URL を最初にチェックしてから、許可されている URL リストをチェックします。 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。

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制御サービスを使用できるようにはなっていません。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。

5 「保存」をクリックして終了します。

◆ ◆ ◆ 第 8 章

Secure Remote Access ゲートウェイの設定

この章では、Sun Java System Portal Server 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。

この章で説明する内容は次のとおりです。

始める前に

- ゲートウェイプロファイルの作成については、34 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。

プロファイルのコアオプションの設定

この節では、次の操作について説明します。

起動モードの設定

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。

▼ 起動モードを設定する

1. **Portal Server** 管理コンソールに管理者としてログオンします。
2. 「**Secure Remote Access**」タブを選択し、属性を変更するプロファイル名をクリックします。
3. 「コア」タブを選択します。
4. 次の属性を変更します。

HTTP 接続	ゲートウェイでの非 SSL 接続の受け付けを許可する場合は「HTTP 接続」チェックボックスにチェックマークを付けます。
HTTP ポート	HTTP ポート番号を入力します。デフォルト値は 80 です。
HTTPS 接続	ゲートウェイでの SSL 接続の受け付けを許可する場合は「HTTPS 接続」チェックボックスにチェックマークを付けます。このオプションは、デフォルトで選択されています。
HTTPS ポート	HTTPS ポート番号を入力します。デフォルト値は 443 です。

注 – 次の属性は、『Sun Java System Portal Server 7.2 Command-Line Reference』の「psadmin set-attribute」を使用して変更できます。

/space/PS/portal/bin/`psadmin set-attribute -u amadmin -f` */space/PS/portal/bin/ps_password -p portal1* `-m gateway --gateway-profile` *profileID* `-a` *sunPortalGatewayDomainsAndRulesets -A $entry*

- sunPortalGatewayDefaultDomainAndSubdomains=Default Domains
- sunPortalGatewayLoggingEnabled=Enable Logging
- sunPortalGatewayEProxyPerSessionLogging=Enable per Session Logging
- sunPortalGatewayEProxyDetailedPerSessionLogging=Enable Detailed per Session Logging
- sunPortalGatewayNetletLoggingEnabled=Enable Netlet Logging
- sunPortalGatewayEnableMIMEGuessing=Enable MIME Guessing
- sunPortalGatewayParserToURIMap=Parser to URI Mappings
- sunPortalGatewayEnableObfuscation=Enable Masking
- sunPortalGatewayObfuscationSecretKey=Seed String for Masking
- sunPortalGatewayNotToObscureURIList=URIs not to Mask
- sunPortalGatewayUseConsistentProtocolForGateway=Make
- Gateway protocol Same as \n Original URI Protocol sunPortalGatewayEnableCookieManager=Store External Server Cookies
- sunPortalGatewayMarkCookiesSecure=Mark Cookies as secure

5 端末ウィンドウからゲートウェイを再起動します。

`./psadmin start-sra-instance -u amadmin -f` *passwordfile* `-N` *profilename* `-t gateway`

コアコンポーネントの設定

ネットレットを使用することで、インターネットなどのセキュリティー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。ネットレット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ネットレット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。ネットレット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。ネットレットは、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。

▼ コンポーネントを設定する

1. **Portal Server** 管理コンソールに管理者としてログオンします。
2. 「**Secure Remote Access**」タブを選択し、属性を変更するプロファイル名をクリックします。
3. 「コア」タブを選択します。
4. 次の属性を変更します。

属性名	説明
ネットレット	ネットレットサービスを開始する場合は「有効」チェックボックスにチェックマークを付けます。このオプションは、デフォルトで選択されています。
プロキシレット	プロキシレットサービスを開始する場合は「有効」チェックボックスにチェックマークを付けます。このオプションは、デフォルトで選択されています。

5. 端末ウィンドウから、次のコマンドオプションを使用してゲートウェイを再起動します。

   `./psadmin start-sra-instance -u amadmin -f` *passwordfile* `-N` *profilename* `-t gateway`

基本オプションの設定

「Cookie 管理」属性について

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 ヘッダーに 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 は書き換えられません。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 を書き換えます。この書き換えによって、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。

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。

「HTTP 基本認証」属性について

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 サイトにふたた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はできない)。

注 – 基本認証ではなく、Windows NT challenge/response (HTTP 応答コードは 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制御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。

「Portal Server」属性について

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。

「ユーザーセッション Cookie を転送する URL」属性について

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ー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するとします。

```
http://server:8080
```

この場合、`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。

「URL からセッションを取得」属性について

「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 の一部として符号化されます。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。

▼ 基本オプションを設定する

1 **Portal Server** 管理コンソールに管理者としてログオンします。

2 「**Secure Remote Access**」タブを選択し、属性を変更するプロファイル名をクリックします。

3 「コア」タブを選択します。

4 次の属性を変更します。

属性名	説明
Cookie 管理	Cookie 管理を有効にする場合は「有効」チェックボックスにチェックマークを付けます。 このオプションは、デフォルトで選択されています。
HTTP 基本認証	「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。
Portal Server	Portal Server を `http://portal-server-name:port-number` の形式でフィールドに指定し、「追加」をクリックします。 「Portal Server」リストにさらに Portal Server を追加する場合は、この手順を繰り返します。
ユーザーセッション Cookie を転送する URL	ユーザーセッション Cookie を転送する URL を入力し、「追加」をクリックします。 「ユーザーセッション Cookie を転送する URL」リストにさらに URL を追加する場合は、この手順を繰り返します。
ゲートウェイ最小認証レベル	認証レベルを入力します。 デフォルトでは、すべてのレベルの認証を許可するようにアスタリスク (*) が追加されます。
URL からセッションを取得	URL からセッションの情報を取得する場合は「はい」を選択します。 デフォルトでは、「いいえ」オプションが選択されています。

配備オプションの設定

プロキシの設定

▼ プロキシの設定を行うには

1 **Portal Server** 管理コンソールに管理者としてログオンします。

2 「**Secure Remote Access**」タブを選択し、属性を変更するプロファイル名をクリックします。

3 「配備」タブを選択します。

4 次の属性を変更します。

属性名	説明	
プロキシを使用する	「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。	
Web プロキシ URL	「Web プロキシを使用する URL」編集ボックスに、適切な URL を `http://hostname.subdomain.com` の形式で入力し、「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。	「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、36 ページの「Access Manager へアクセスするプロキシの設定」を参照してください。
ドメインとサブドメインのプロキシ	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、* のあと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。	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、36 ページの「Access Manager へアクセスするプロキシの設定」を参照してください。
プロキシパスワードのリスト	「プロキシパスワードのリスト」に各プロキシサーバーの情報を入力し、「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。	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。

属性名	説明	
自動プロキシ設定サポート	「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。	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、49 ページの「自動プロキシ設定の使用」を参照してください。
自動プロキシ設定ファイルの位置	「場所」フィールドに、PAC ファイルの名前と場所を入力します。	

リライタプロキシおよびネットレットプロキシの設定

ネットレットプロキシについて

ネットレットプロキシは、ゲートウェイを経由してイントラネット内のネットレットプロキシまでクライアントからのセキュリティー保護された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ネットレットトラフィックの安全性を補強します。ネットレットプロキシを有効にすると、ネットレットパケットがネットレット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。

リライタプロキシについて

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の間の HTTP トラフィックをセキュリティー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、イントラネット上のマシン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。

▼ リライタプロキシとネットレットプロキシを設定する

1 **Portal Server** 管理コンソールに管理者としてログオンします。

2 「**Secure Remote Access**」タブを選択し、属性を変更するプロファイル名をクリックします。

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。

3 「配備」タブを選択します。

4 次の属性を変更します。

属性名	説明
リライタプロキシ	リライタプロキシサービスを有効にする場合は、「リライタプロキシ」チェックボックスにチェックマークを付けます。
リライタプロキシのリスト	a. 「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。 b. 「追加」をクリックします。
ネットレットプロキシ	「ネットレットプロキシを有効」チェックボックスにチェックマークを付けて、ネットレットプロキシサービスを有効にします。
ネットレットプロキシホスト	a. 「ネットレットプロキシホスト」フィールドに、`hostname:port` という形式でネットレット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。 b. 「追加」をクリックします。
Web プロキシ経由のネットレットトンネリング	「Web プロキシ経由のネットレット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。

5 サーバーで *portal-server-install-root*/SUNWportal/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。

6 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/SUNWportal/bin/rwproxyd -n *gateway-profile-name* start

7 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。

./psadmin start-sra-instance -u amadmin -f *passwordfile* -N *profilename* -t gateway

セキュリティーオプションの設定

PDC および非認証 URL の設定

▼ PDC および非認証 URL を設定する

1 **Portal Server** 管理コンソールに管理者としてログオンします。

2 「**Secure Remote Access**」タブを選択し、属性を変更するプロファイル名をクリックします。

3 「セキュリティー」タブを選択します。

4 次の属性を変更します。

属性名	説明
証明書が有効なゲートウェイホスト	a. 「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。 b. 「追加」をクリックします。

属性名	説明
非認証 URL	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が該当します。 「非認証 URL」フィールドに、`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、「追加」をクリックして、このエントリを「非認証 URL」リストに追加します。
信頼できる SSL ドメイン	「信頼できる SSL ドメイン」フィールドに、ドメイン名を入力して「追加」をクリックします。

TLS および SSL オプションの設定

▼ TLS および SSL オプションを設定する

1. **Portal Server** 管理コンソールに管理者としてログオンします。
2. 「**Secure Remote Access**」タブを選択し、属性を変更するプロファイル名をクリックします。
3. 「セキュリティー」タブを選択します。
4. 次の属性を変更します。

属性名	説明
40 ビット暗号化	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。 a. 「Communicator」メニューの「ツール」の「セキュリティー情報」を選択します。 b. 左の区画で「Navigator」リンクをクリックします。 c. 「詳細セキュリティー (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。 d. 適切な暗号化方式を有効にします。
Null 暗号化方式	「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。
SSL 暗号化方式の選択	Secure Remote Access 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 ハンドシェークは正常に行われます。
SSL Version 2.0	「SSL Version 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、Version 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。 SSL version 2.0 を有効または無効にできます。SSL 2.0 を無効にすると、古い SSL 2.0 しかサポートしないブラウザは Secure Remote Access に対して認証ができません。これにより、セキュリティーのレベルが格段に向上します。
SSL2 暗号化方式	「SSL 暗号化方式の選択の有効化」チェックボックスにチェックマークを付けます。 SSL 暗号化方式のリストから、必要な暗号化方式を選択できます。
SSL Version 3.0	SSL version 3.0 を有効または無効にできます。SSL 3.0 を無効にすると、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、セキュリティーのレベルが格段に向上します。 「SSL Version 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、Version 3.0 を有効にします。
SSL3 暗号化方式	「SSL 暗号化方式の選択の有効化」チェックボックスにチェックマークを付けます。 SSL3 暗号化方式のリストから、必要な暗号化方式を選択できます。

属性名	説明
TLS 暗号化方式	「SSL 暗号化方式の選択の有効化」チェックボックスにチェックマークを付けます。 TLS 暗号化方式のリストから、必要な暗号化方式を選択できます。

パフォーマンスオプションの設定

タイムアウトおよび再試行の設定

▼ タイムアウトおよび再試行を設定する

1. **Portal Server** 管理コンソールに管理者としてログオンします。
2. 「**Secure Remote Access**」タブを選択し、属性を変更するプロファイル名をクリックします。
3. 「パフォーマンス」タブを選択します。
4. 次の属性を変更します。

属性名	説明
サーバーの再試行間隔 (秒)	Portal Server、リライタプロキシ、またはネットレットプロキシがクラッシュや停止などで使用できなくなった場合に開始の試行を要求する時間間隔を秒単位で指定します。
ゲートウェイタイムアウト (秒)	ゲートウェイとブラウザの接続がタイムアウトになるまでの時間間隔を秒単位で指定します。 「ゲートウェイタイムアウト」フィールドに、タイムアウトまでの間隔を秒単位で指定します。
キャッシュされたソケットのタイムアウト (秒)	ゲートウェイと Portal Server の接続がタイムアウトになるまでの時間間隔を秒単位で指定します。

HTTP オプションの設定

▼ HTTP オプションを設定する

1. **Portal Server** 管理コンソールに管理者としてログオンします。

2 「**Secure Remote Access**」タブを選択し、属性を変更するプロファイル名をクリックします。

3 「パフォーマンス」タブを選択します。

4 次の属性を変更します。

属性名	説明
最大スレッドプールサイズ	適切なスレッド数を指定します。 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。
持続 HTTP 接続	「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。 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。
持続接続ごとの最大要求数	最大要求数を入力します。
持続ソケット接続のタイムアウト (秒)	適切なタイムアウトを秒単位で入力します。
回復時間に必要な正常なタイムアウト (秒)	必要な猶予時間を秒単位で入力します。 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。 ■ ブラウザが要求を送信してから要求がゲートウェイに到達するまでにかかる時間 ■ ゲートウェイが応答を送信してからブラウザがその応答を実際に受信するまでの時間 これは、ネットワーク状態やクライアントの接続速度などの要因によって決まります。
最大接続キュー	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。 適切な接続数を指定します。

Secure Remote Access のパフォーマンスの監視

管理者は Secure Remote Access の各種コンポーネントにアクセスしてパフォーマンスを監視できます。

▼ Secure Remote Access のパフォーマンスを監視する

1 Portal Server 管理コンソールにログオンします。

2 「Secure Remote Access」タブを選択し、サブメニューの「監視」をクリックします。

3 「監視」ページで、ドロップダウンメニューからプロキシインスタンスを選択します。

4 「MBeans」テーブル内で、パフォーマンス値を表示する属性を選択します。

リライタオプションの設定

基本オプションの設定

▼ 基本オプションを設定する

1 Portal Server 管理コンソールに管理者としてログオンします。

2 「Secure Remote Access」タブを選択し、属性を変更するプロファイル名をクリックします。

3 「リライタ」タブを選択します。

4 次の属性を変更します。

属性名	説明
すべての URI のリライト	「すべての URI の書き換えを有効」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 の書き換えを有効にします。 ゲートウェイサービスで「すべての URI の書き換えを有効」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 を書き換えます。「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。
書き換えない URI	編集ボックスで URI を追加します。 注 – このリストに #* を追加すると、href ルールがルールセットの一部であっても、URI を書き換えできるようにできます。

「URI をルールセットにマップ」の設定

ルールセットは、Portal Server 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 はポータルの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。ほか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、コンテンツが取得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。取得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プ」を設定する

1. **Portal Server** 管理コンソールに管理者としてログオンします。
2. 「**Secure Remote Access**」タブを選択し、属性を変更するプロファイル名をクリックします。
3. 「リライタ」タブを選択します。
4. 次の属性を変更します。

属性名	説明

URI	「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
	次に例を示します。
	`eng.sesta.com|default`
	注 – ルールセットを適用する優先順位は、ホスト名 - サブドメイン - ドメインです。
	ドメインベースのルールセットリストのエントリ例を次に示します。
	`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` サブドメイン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。

「パーサーを MIME タイプにマップ」の設定

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 ページを解析するために、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、リライタ機能をほかの MIME タイプに拡張できます。

複数のエントリは、セミコロン (;) またはコンマ (,) で区切ります。

次に例を示します。

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 の書き換えに 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。

▼ 「パーサーを **MIME** タイプにマップ」を設定する

1. **Portal Server** 管理コンソールに管理者としてログオンします。
2. 「**Secure Remote Access**」タブを選択し、属性を変更するプロファイル名をクリックします。
3. 「リライタ」タブを選択します。
4. 次の属性を変更します。

属性名	説明
パーサー	a. 「パーサーを MIME タイプにマップ」フィールドで、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、セミコロンまたはコンマを使用します。 エントリは `HTML=text/html;text/htm` の形式で指定します。 b. 「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。

PDC (Personal Digital Certificate) 認証の設定

PDC は認証局 (CA) が発行し、CA の非公開錠で署名されます。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 カードなどの符号化されたデバイスを使用できます。符号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 と符号化されたデバイスを設定する

1 **Portal Server** マシンで、`/etc/opt/SUNWam/config/AMConfig.properties` ファイルに次の行を追加します。`com.iplanet.authentication.modules.cert.gwAuthEnable=yes`。

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。証明書の設定については、**190** ページの「ゲートウェイマシンでルート **CA** 証明書をインポートする」を参照してください。

3 **Access Manager** 管理コンソールに管理者としてログインし、次の操作を行います。

   a. 「アイデンティティー管理」タブを選択し、「組織」を選択します。

   b. 「表示」ドロップダウンメニューから該当の組織に使用するサービスを選択します。

   c. 「追加」をクリックして証明書を登録します。

4 **Access Manager** 管理コンソールで、次の操作を行います。

   a. 適切な組織を選択し、「証明書」の隣の矢印をクリックします。

   b. 「信頼できるリモートホスト」リストボックスで、何も強調表示せずに「削除」をクリックします。

   c. テキストボックスに何らかの文字列を入力し、「追加」をクリックします。

d. 「保存」をクリックします。

5 **Access Manager** 管理コンソールで、次の操作を行います。

a. 適切な組織を選択し、「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。
サービスのリストが表示されます。

b. 「認証設定」コアサービスの隣の矢印をクリックし、「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。

c. サービスインスタンス名に「`gatewaypdc`」と入力します。

d. 「送信」をクリックします。
「gatewaypdc」がサービスインスタンスに表示されます。

e. 「**gatewaypdc**」をクリックし、サービスを編集します。
「gatewaypdc プロパティーを表示」ページが表示されます。

f. 「認証設定」の隣の「編集」リンクをクリックし、「追加」をクリックします。
「モジュールの追加」ページが表示されます。

g. 「モジュール名」フィールドの「証明書」と、「適用基準」の「必須」を選択し、「了解」をクリックします。

h. 「了解」をクリックして終了します。

6 **Access Manager** 管理コンソールで、次の操作を行います。

a. 「コア」の隣の矢印をクリックします。

b. 「組織認証モジュール」リストボックスで、「**gatewaypdc**」を選択します。

c. 「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。

d. 「保存」をクリックして終了します。

7 **Portal Server** 管理コンソールに管理者としてログインし、次の操作を行います。

a. 「**Secure Remote Access**」タブを選択し、適切なゲートウェイプロファイルを選択します。

b. 「セキュリティー」タブを選択します。

c. 「証明書が有効なゲートウェイホスト」リストボックスで、ゲートウェイ名を追加します。

d. 「保存」をクリックします。

8 端末ウィンドウからゲートウェイプロファイルを再起動します。

```
./psadmin start-sra-instance -u amadmin -f passwordfile -N profilename -t
gateway
```

9 **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。

10 **JVM** キーストアに、クライアント証明書をインストールします。**Windows** マシンから「スタート」>「設定」>「コントロールパネル」>「**Java**」の順にクリックして、**JVM** コントロールパネルにアクセスできます。

アプレットランタイムパラメータに、次のパラメータを追加します。

- `Djavax.net.ssl.keyStore=Path to Keystore`
- `Djavax.net.ssl.keyStorePassword=password`
- `Djavax.net.ssl.keyStoreType=type`

11 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。

▼ ゲートウェイマシンでルート **CA** 証明書をインポートする

1 ゲートウェイマシンでルート **CA** 証明書をインポートします。

a. `<Gateway-Install-Dir>/SUNWportal/bin/certadmin -n <gw-profile-name>`

Certadmin メニューが表示されます。

b. オプション **3** を選択し、証明書のパスを入力します。

詳細については、第 10 章証明書の操作を参照してください。

2 **CA** に送信する証明書署名要求を生成します。

   a. `<Gateway-Install-Dir>/SUNWportal/bin/certadmin -n <gw-profile-name>`
   Certadmin メニューが表示されます。

   b. オプション **2** を選択し、適切な情報を入力します。

   c. ファイルを保存します。

3 証明書署名要求を **CA** に送信し、承認を受けます。**CA** の署名後に証明書応答を保存します。

4 **CA** による承認が得られたら、サーバー証明書をインポートします。

   a. `<Gateway-Install-Dir>/SUNWportal/bin/certadmin -n <gw-profile-name>`
   Certadmin メニューが表示されます。

   b. オプション **4** を選択します。

   c. サーバー証明書が格納されているファイルの場所を指定します。

5 **Portal Server** マシンでルート **CA** 証明書をインポートします。

コマンド行オプションによるゲートウェイ属性の設定

ここでは、端末ウィンドウからゲートウェイ属性を設定するためのコマンド行オプションについて説明します。次の操作について説明します。

▼ 外部サーバー Cookie の格納を管理する

「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ー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、Cookie がないと状態管理ができないという場合でも、ゲートウェイ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** の格納を管理するには、次のコマンドを入力して **Enter** キーを押します。

  - 有効にするには、次のコマンドを使用します。

    PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a CookieManagement true`

  - 無効にするには、次のコマンドを使用します。

    PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a CookieManagement false`

  - 属性値を取得するには、次のコマンドを使用します。

    PS_INSTALL_DIR `/bin/psadmin get-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a CookieManagement`

参考 関連項目

『Sun Java System Portal Server 7.2 Command-Line Reference』の「psadmin set-attribute」および『Sun Java System Portal Server 7.2 Command-Line Reference』の「psadmin get-attribute」

▼ セキュリティー保護された Cookie としてのマーク付けを有効にする

セキュリティー保護された 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ーを補強した Cookie として処理します。セキュリティーの実装には、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。

- セキュリティー保護された **Cookie** としてマーク付けをするには、次のコマンドを入力して **Enter** キーを押します。

  - 有効にするには、次のコマンドを使用します。

    PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a MarkCookiesSecure true

  - 無効にするには、次のコマンドを使用します。

    PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a MarkCookiesSecure false

  - 属性値を取得するには、次のコマンドを使用します。

    PS_INSTALL_DIR /bin/psadmin get-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a MarkCookiesSecure

参考 関連項目

『Sun Java System Portal Server 7.2 Command-Line Reference』の「psadmin set-attribute」および『Sun Java System Portal Server 7.2 Command-Line Reference』の「psadmin get-attribute」

▼ プロキシを使用しない **URL** のリストを作成する

「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。

- プロキシを使用しない **URL** を管理するには、次のコマンドを入力して **Enter** キーを押します。

注 – 複数の URL がある場合は、各 URL をスペースで区切ります。

  - 使用しない **URL** を指定するには、次のコマンドを使用します。

    PS_INSTALL_DIR /bin/psadmin set-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a DontUseWebProxyURL -A *"LIST_OF_URLS"*

  - 既存のリストに **URL** を追加するには、次のコマンドを使用します。

    PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a DontUseWebProxyURL -A *"LIST_OF_URLS"*

- 既存のリストから **URL** を削除するには、次のコマンドを使用します。

  PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a DontUseWebProxyURL -E *"LIST_OF_URLS"*

- 既存の **URL** リストを取得するには、次のコマンドを使用します。

  PS_INSTALL_DIR /bin/psadmin get-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a DontUseWebProxyURL

参考 関連項目

『Sun Java System Portal Server 7.2 Command-Line Reference』の「psadmin set-attribute」および『Sun Java System Portal Server 7.2 Command-Line Reference』の「psadmin get-attribute」

▼ ルールセットと **URI** のマッピングを管理する

Secure Remote Access では、OWA (Outlook Web Access) から Microsoft 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。

1. 既存のリストに **URI** を追加するには、次のコマンドを使用します。

   PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile default -a DomainsAndRulesets -A "*URI*|*RULE_SET_NAME URI*|*RULE_SET_NAME*"

2. 既存のリストから **URI** を削除するには、次のコマンドを使用します。

   PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile default -a DomainsAndRulesets -E "*URI*|*RULE_SET_NAME URI*|*RULE_SET_NAME*"

3. 既存のリストを取得するには、次のコマンドを使用します。

   PS_INSTALL_DIR /bin/psadmin get-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a DomainsAndRulesets

4. **Outlook Web Access** のルールセットを管理するには、次のコマンドを入力して **Enter** キーを押します。

   - ルールセットを追加するには、次のコマンドを使用します。

     PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile default -a DomainsAndRulesets -A "*EXCHANGE2000_SERVER_NAME* exchange_2000sp3_owa_ruleset"

- ルールセットを削除するには、次のコマンドを使用します。

  PS_INSTALL_DIR `/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile default -a DomainsAndRulesets -E "`*EXCHANGE2000_SERVER_NAME* `exchange_2000sp3_owa_ruleset"`

- **URI** とルールセットのマッピングリストを設定するには、次のコマンドを使用します。

  PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a DomainsAndRulesets "`*URI*`|`*RULE_SET_NAME URI*`|`*RULE_SET_NAME* `"`

参考 関連項目

『Sun Java System Portal Server 7.2 Command-Line Reference』の「psadmin set-attribute」および『Sun Java System Portal Server 7.2 Command-Line Reference』の「psadmin get-attribute」

▼ デフォルトドメインを指定する

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、次のコマンドを入力して **Enter** キーを押します。

  - デフォルトドメインを設定するには、次のコマンドを使用します。

    PS_INSTALL_DIR `/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a DefaultDomainsAndSubdomains "`*DOMAIN_NAME*`"`

  - デフォルトドメインを取得するには、次のコマンドを使用します。

    PS_INSTALL_DIR `/bin/psadmin get-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a DefaultDomainsAndSubdomains`

参考 関連項目

『Sun Java System Portal Server 7.2 Command-Line Reference』の「psadmin set-attribute」および『Sun Java System Portal Server 7.2 Command-Line Reference』の「psadmin get-attribute」

▼ MIME 推測を管理する

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** 推測を管理するには、次のコマンドを入力して **Enter** キーを押します。
  - **MIME** 推測を有効にするには、次のコマンドを使用します。

    PS_INSTALL_DIR `/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a EnableMIMEGuessing true`
  - **MIME** 推測を無効にするには、次のコマンドを使用します。

    PS_INSTALL_DIR `/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a EnableMIMEGuessing false`
  - 値を取得するには、次のコマンドを使用します。

    PS_INSTALL_DIR`/bin/psadmin get-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a EnableMIMEGuessing`

参考 関連項目

『Sun Java System Portal Server 7.2 Command-Line Reference』の「psadmin set-attribute」および『Sun Java System Portal Server 7.2 Command-Line Reference』の「psadmin get-attribute」

▼ 解析する URI マッピングのリストを作成する

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書き換えます。

- 解析する **URI** マッピングのリストを作成するには、次のコマンドを入力して **Enter** キーを押します。

  - 解析する **URI** マッピングのリストを設定するには、次のコマンドを使用します。

    PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a MIMEMap

  - 既存のリストに追加するには、次のコマンドを使用します。

    PS_INSTALL_DIR /bin/psadmin set-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a MIMEMap -A *LIST*

  - 既存のリストから削除するには、次のコマンドを使用します。

    PS_INSTALL_DIR /bin/psadmin set-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a MIMEMap -E *LIST*

  - 既存のリストを取得するには、次のコマンドを使用します。

    PS_INSTALL_DIR /bin/psadmin get-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME*-a MIMEMap

参考 関連項目

『Sun Java System Portal Server 7.2 Command-Line Reference』の「psadmin set-attribute」

▼ マスキングを管理する

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 を書き換えます。

- マスキングを管理するには、次のコマンドを入力して **Enter** キーを押します。

  - マスキングを有効にするには、次のコマンドを使用します。

    PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a EnableObfuscation true

  - マスキングを無効にするには、次のコマンドを使用します。

    PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a EnableObfuscation false

  - 値を取得するには、次のコマンドを使用します。

    PS_INSTALL_DIR/bin/psadmin get-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a EnableObfuscation

参考 関連項目

『Sun Java System Portal Server 7.2 Command-Line Reference』の「psadmin set-attribute」および『Sun Java System Portal Server 7.2 Command-Line Reference』の「psadmin get-attribute」

▼ マスキングのためのシード文字列を指定する

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。

- マスキングのシード文字列を指定するには、次のコマンドを入力して **Enter** キーを押します。
  - マスキングのシード文字列を設定するには、次のコマンドを使用します。

    PS_INSTALL_DIR `/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a ObfuscationSecretKey` *SECRET_KEY*
  - 値を取得するには、次のコマンドを使用します。

    PS_INSTALL_DIR`/bin/psadmin get-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a ObfuscationSecretKey`

参考 関連項目

『Sun Java System Portal Server 7.2 Command-Line Reference』の「psadmin set-attribute」および『Sun Java System Portal Server 7.2 Command-Line Reference』の「psadmin get-attribute」

▼ マスクしない **URI** のリストを作成する

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、リストボックスに `*/Applet/Param*` を追加すると、この 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はマスクされません。

注 – 複数の URI がある場合は、各 URI をスペースで区切ります。

- マスクしない **URI** のリストを作成するには、次のコマンドを入力して **Enter** キーを押します。

  - マスクしない **URI** のリストを作成するには、次のコマンドを使用します。

    PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a NotToObscureURIList` *LIST_OF_URI*

  - 既存のリストに追加するには、次のコマンドを使用します。

    PS_INSTALL_DIR `/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a NotToObscureURIList -A` *LIST_OF_URI*

  - 既存のリストから削除するには、次のコマンドを使用します。

    PS_INSTALL_DIR `/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a NotToObscureURIList -E` *LIST_OF_URI*

  - 既存の値を取得するには、次のコマンドを使用します。

    PS_INSTALL_DIR `/bin/psadmin get-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a NotToObscureURIList`

参考 関連項目

『Sun Java System Portal Server 7.2 Command-Line Reference』の「psadmin set-attribute」および『Sun Java System Portal Server 7.2 Command-Line Reference』の「psadmin get-attribute」

▼ 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

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、次のコマンドを入力して **Enter** キーを押します。

  - 有効にするには、次のコマンドを使用します。

    PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a UseConsistentProtocolForGateway true`

  - 無効にするには、次のコマンドを使用します。

    PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a UseConsistentProtocolForGateway false`

  - 値を取得するには、次のコマンドを使用します。

    PS_INSTALL_DIR`/bin/psadmin get-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a UseConsistentProtocolForGateway`

参考 関連項目

『Sun Java System Portal Server 7.2 Command-Line Reference』の「psadmin set-attribute」および『Sun Java System Portal Server 7.2 Command-Line Reference』の「psadmin get-attribute」

◆ ◆ ◆ 第 9 章

9 ゲートウェイサービスのリライタの設定

この章で説明する内容は、次のとおりです。

- 201 ページの「URI とルールセットのマッピングリストの作成」
- 202 ページの「ゲートウェイサービスのリライタの設定」

リライタルールの詳細については、74 ページの「言語ベースのルールの定義」を参照してください。

リライタに関する問題の詳細については、98 ページの「デバッグログを使用したトラブルシューティング」を参照してください。

リライタの例については、101 ページの「サンプルの操作」を参照してください。

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` はポータルの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。ほか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` はパッケージ内のルールセットです。

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、それがどこから取得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 が 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。取得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
```

ゲートウェイサービスのリライタの設定

「リライタ」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

- 203 ページの「ゲートウェイによるすべての URL の書き換えを有効にする」
- 203 ページの「書き換えない URI を指定する」
- 204 ページの「URI をルールセットにマッピングする」
- 204 ページの「MIME のマッピングを指定する」
- 205 ページの「デフォルトドメインを指定する」

▼ ゲートウェイによるすべての URL の書き換えを有効にする

ゲートウェイサービスで「すべての URI の書き換えを有効」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 を書き換えます。「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。

1. **Portal Server** 管理コンソールに管理者としてログインします。
2. 「**Secure Remote Access**」タブを選択し、属性を変更するゲートウェイプロファイルを選択します。
3. 「リライタ」タブを選択します。
4. 「基本オプション」の「すべての **URI** の書き換えを有効」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** の書き換えを有効にします。
5. 「保存」をクリックして終了します。
6. 端末ウィンドウからゲートウェイを再起動します。

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

▼ 書き換えない URI を指定する

1. **Portal Server** 管理コンソールに管理者としてログインします。
2. 「**Secure Remote Access**」タブを選択し、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。
3. 「リライタ」タブを選択します。
4. 「基本オプション」の「追加」テキストフィールドに **URI** を入力し、「追加」をクリックします。
   「リライトしない URI」ボックスに URI 値が表示されます。

   注 – このリストに #* を追加すると、href ルールがルールセットの一部であっても、URI を書き換えできるようにできます。

   ---

5. 「保存」をクリックして終了します。

6 端末ウィンドウからゲートウェイを再起動します。

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

▼ URI をルールセットにマッピングする

1 Portal Server 管理コンソールに管理者としてログインします。

2 「Secure Remote Access」タブを選択し、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。

3 「リライタ」タブを選択します。

4 「リライタオプション」の「URI をルールセットにマップ」をクリックし、「行を追加」をクリックします。

5 「URI」フィールドに、適切なドメインまたはホスト名を入力し、「ルールセット」フィールドに、そのドメインに適したルールセットを入力します。

「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

次に例を示します。

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

6 「保存」をクリックして終了します。

7 端末ウィンドウからゲートウェイを再起動します。

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

▼ MIME のマッピングを指定する

リライタには、コンテンツタイプ (HTML、JAVASCRIPT、CSS、および XML) に基づいて Web ページを解析するための 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、リライタ機能をほかの MIME タイプに拡張できます。

複数のエントリは、セミコロン (;) またはコンマ (,) で区切ります。次に例を示します。

```
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の書き換えに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。

---

1 **Portal Server**管理コンソールに管理者としてログインします。

2 「**Secure Remote Access**」タブを選択し、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。

3 「リライタ」タブを選択します。

4 「リライタオプション」の「パーサーを**MIME**タイプにマップ」をクリックします。
エントリは`HTML=text/html;text/htm`の形式で指定します。

5 「行を追加」をクリックして、リストにエントリを追加します。「**MIME**タイプ」フィールドに、パーサー値およびマップ先の対応する**MIME**値を入力します。

6 「保存」をクリックして終了します。

7 端末ウィンドウからゲートウェイを再起動します。

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

▼ デフォルトドメインを指定する

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。

1 **Portal Server**管理コンソールに管理者としてログインします。

2 「**Secure Remote Access**」タブを選択し、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。

3 「配備」タブを選択します。

4. 「ドメインとサブドメインのプロキシ」フィールドに、必須ドメイン名をプロキシなしで入力します。

5. 「保存」をクリックして終了します。

6. 端末ウィンドウからゲートウェイを再起動します。

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

◆ ◆ ◆ 第 10 章

10 証明書の操作

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。

この章の内容は、次のとおりです。

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ー保護されます。

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 ハンドシェー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、60 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。213 ページの「certadmin スクリプト」を参照してください。

注 – 「証明書」ポップアップウィンドウは、SSL アプリケーションに共通のものです。警告を受け入れて処理を進めるように、ユーザーに通知してください。

証明書ファイル

証明書関連のファイルは `/etc/opt/SUNWportal/cert/`*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。

208 ページの「証明書ファイル」は、これらのファイルの説明を示しています。

表 10–1 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。 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。

表 10–1 証明書ファイル (続き)

ファイル名	タイプ	説明
.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 *token-name:certificate-name* 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)。

各証明書には 3 つの利用可能な信頼カテゴリがあり、SSL、電子メール、オブジェクト署名の順に表されています。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コンマ (,) 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。

209 ページの「証明書の信頼属性」は、属性値のリストとそれぞれの意味を示しています。

表 10–2 証明書の信頼属性

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)

表 10–2 証明書の信頼属性 (続き)

属性	説明
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ます。
w	警告を送信 (ほか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、219 ページの「証明書の信頼属性の変更」を参照してください。

210 ページの「CA の信頼属性」は、代表的な認証局とその信頼属性を示しています。

表 10–3 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

表 10–3 公開されている認証局 (続き)

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
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

表 10–3 公開されている認証局 (続き)

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
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

表 10–3 公開されている認証局 (続き)

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
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。

- 214 ページの「自己署名証明書の生成」
- 215 ページの「証明書署名要求 (CSR) の生成」
- 216 ページの「ルート CA 証明書の追加」
- 218 ページの「CA から届いた証明書のインストール」
- 219 ページの「証明書の削除」
- 219 ページの「証明書の信頼属性の変更」
- 220 ページの「ルート CA 証明書のリスト表示」
- 221 ページの「すべての証明書のリスト表示」
- 222 ページの「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

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/SUNWportal/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
選択: [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
```

使用する場合 (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/SUNWportal/cert/gateway-profile-name -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/SUNWportal/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。

```
./psadmin start-sra-instance -u amadmin -f passwordfile -N profilename -t gateway
```

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

1 **root** として certadmin スクリプトを実行します。

portal-server-install-root/SUNWportal/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
選択: [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する場合
(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/SUNWportal/cert -list
を実行してリストを表示できます。
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/SUNWportal/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 ハンドシェークは失敗します。

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

1 **root** として certadmin スクリプトを実行します。

portal-server-install-root/SUNWportal/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
選択: [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) サービスを提供するベンダーまたは自社の 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。

- 215 ページの「証明書署名要求 (CSR) の生成」
- 217 ページの「CA への証明書の要求」
- 218 ページの「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

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/SUNWportal/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

1 root として certadmin スクリプトを実行します。

portal-server-install-root/SUNWportal/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
選択: [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
この証明書の証明書署名要求 (CSR) の作成時に使用した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/SUNWportal/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。

```
./psadmin start-sra-instance -u amadmin -f passwordfile -N profilename -t gateway
```

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

1 **root** として certadmin スクリプトを実行します。

portal-server-install-root/SUNWportal/bin/certadmin -n

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
選択: [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

1 **root** として certadmin スクリプトを実行します。

gateway-install-root/SUNWportal/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
選択: [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 CA** 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

1 **root** として certadmin スクリプトを実行します。

portal-server-install-root/SUNWportal/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
選択: [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

1 **root** として certadmin スクリプトを実行します。

portal-server-install-root
/SUNWportal/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
選択: [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

1 **root** として certadmin スクリプトを実行します。

portal-server-install-root/SUNWportal/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
選択: [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。

◆ ◆ ◆ 第 11 章

11 ネットレットの設定

この章では、Sun Java System Portal Server 管理コンソールでのネットレット属性の設定について説明します。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、および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。

この章で説明する内容は、次のとおりです。

ネットレット属性の設定

次のタスクを実行することで、ネットレットを設定できます。

▼ 基本属性を設定する

1. **Portal Server** 管理コンソールに管理者としてログオンします。
2. 「**Secure Remote Access**」タブを選択し、「ネットレット」タブを選択します。
3. 「**DN** を選択」リストからユーザーまたは組織の **DN** を選択するか、または **DN** を追加します。
4. 次の属性を変更します。

属性名	説明
COS 優先順位	属性値を継承するかどうかの決定に使用される値を指定します。この属性の詳細については、『Sun Java System Directory Server 管理ガイド』を参照してください。
以下を使用してネットレットを起動します。	ネットレットサービスの起動モードとして「Java Web Start」または「アプレット」のいずれかのオプションを選択します。
デフォルトのループバックポート	ネットレット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。ネットレットルールの設定値が優先される場合を除き、デフォルト値の 58000 が使用されます。 適切なポート番号を入力します。
キープアライブ間隔 (秒)	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、アイドル状態のネットレット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、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。

5 「保存」をクリックして終了します。

▼ 詳細属性の設定

1 **Portal Server** 管理コンソールに管理者としてログオンします。

2 「**Secure Remote Access**」タブを選択し、「ネットレット」タブを選択します。

3 「**DN** を選択」リストからユーザーまたは組織の **DN** を選択するか、または **DN** を追加します。

4 次の属性を変更します。

属性名	説明
ポータルのログアウト時にネットレットを終了	ユーザーが Portal Server からログアウトしたときにすべての接続を終了する場合は、「はい」を選択します。これにより、セキュリティーが向上します。このオプションは、デフォルトで選択されています。 ユーザーが Portal Server デスクトップからログアウトしたあともネットレット接続を維持する場合は、「いいえ」を選択します。 注 – 「いいえ」を選択した場合でも、Portal Server からログアウトしたユーザーはネットレット接続を新たに確立できません。既存の接続だけが保持されます。

属性名	説明
接続の再認証	ネットレット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する場合は、「はい」を選択します。ネットレットルールの設定値が優先される場合を除き、デフォルト値の 58000 が使用されます。デフォルトでは、「いいえ」オプションが選択されています。
接続の警告ポップアップを表示	ネットレットを使用したアプリケーションの実行中、ほかのユーザーが待機ポートを通じてネットレットに接続しようとしたときに、デスクトップに警告ポップアップダイアログボックスを表示する場合は、「はい」を選択します。デフォルトでは、「はい」オプションが選択されています。
ポート警告ダイアログにチェックボックスを表示	この属性が管理コンソールで有効になっている場合にネットレットがローカルマシン上の使用可能なポートを通じて接続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を表示する場合は、「はい」を選択します。デフォルトでは、「はい」オプションが選択されています。
ネットレットルール	ネットレットルールをグローバルレベルで作成します。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。ネットレットルールの作成、変更、および削除の詳細については、226 ページの「ネットレットルールを作成、変更、削除する」を参照してください。
デフォルトのネイティブ VM 暗号化方式	ドロップダウンボックスから、ネットレットルールのデフォルトの暗号化方式を選択します。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。詳細については、149 ページの「下位互換性」を参照してください。
デフォルトの Java プラグイン暗号化方式	ドロップダウンボックスから、デフォルトの Java プラグイン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、148 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。

属性名	説明
許可/拒否されたホスト	ホストアドレスのチェックボックスにチェックマークを付けて、ユーザーまたは組織のタイプに基づいてアクセスを許可または拒否するホストを選択し、ドロップダウンボックスから「許可」または「拒否」を選択します。 新しいホストを追加するには、次の手順に従います。 a. 「行を追加」をクリックします。 b. 完全指定のホストアドレスを入力します。たとえば、abc の場合は abc.sesta.com のように入力します。 注 – 既存のホストを削除するには、「ホスト」リストからホストを選択し、「削除」をクリックします。 特定の組織、ロール、またはユーザーに対して特定のホストへのアクセスの許可または拒否を定義できます。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。組織内の特定のホストへのアクセスを拒否できます。各ルールに一意の local port を指定します。 注 – 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ネットレット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。
ネットレットルールのアクセス/拒否	Nelet ルールを選択し、ドロップダウンリストから「許可」または「拒否」オプションを選択します。 特定の組織、ロール、ユーザーに対して特定のネットレットルールへのアクセスを定義できます。 特定の組織、ロール、ユーザーに対して特定のネットレットルールへのアクセスを拒否できます。 注 – 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ネットレットルールを使用できるようになります。

5 「保存」をクリックして終了します。

▼ ネットレットルールを作成、変更、削除する

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。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。

1 **Portal Server** 管理コンソールに管理者としてログオンします。

2 「**Secure Remote Access**」タブを選択し、「ネットレット」タブを選択します。

3 「DN を選択」リストからユーザーまたは組織の DN を選択するか、または DN を追加します。

4 「拡張」>「ネットレットルール」の「新規ルール」をクリックします。

- ルールを削除する場合は、ルールを選択し、「削除」をクリックします。
- ルールを変更する場合は、ルール名をクリックします。
  「ネットレット」ページで、次の手順で説明されているようにパラメータを変更します。

5 「ルール名」フィールドにルール名を入力します。

6 使用できる暗号化方式のリストから「その他」を選択し、「暗号化方式」リストで 1 つ以上の暗号化方式を選択します。デフォルトの暗号化方式のままにする場合は「デフォルト」を選択します。

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。詳細は「下位互換性」の節を参照してください。暗号化方式の詳細については、「デフォルトの暗号化方式の指定」を参照してください。

7 「リモートアプリケーション URL」フィールドに、呼び出すアプリケーションの URL を入力します。

8 アプレットをダウンロードする必要がある場合は、「クライアントポート」チェックボックスにチェックマークを付けます。「クライアントポート」、「サーバーホスト」、および「サーバーポート」の各フィールドに、クライアントポート番号、サーバーホストアドレス、およびサーバーポート番号を入力します。各ルールに一意の `local port` を指定します。

デフォルトでは、「アプレットのダウンロードを有効」は無効になっています。アプレットの詳細は、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にのみ指定してください。詳細については、143 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。

9 このルールに対応するネットレットセッションの実行中に Portal Server セッション時間が延長されるようにするときは、「拡張セッションを有効」チェックボックスにチェックマークを付けます。

10 「ローカルポートと宛先サーバーポートのマップ」で、次の操作を行います。

a. ネットレット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。

b. 「接続先ホスト」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、ネットレット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。

c. 「接続先ポート」フィールドに、ターゲットホストのポートを入力します。

11 「保存」をクリックして終了します。
ルール名がネットレットのホームページに表示されます。

ネットレットのプロキシ設定

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、ネットレットがブラウザのプロキシ設定を判断できない場合は、最初にネットレット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、ネットレット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、ネットレットはブラウザ設定を特定できない場合があり、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、ネットレット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
  ネットレットの接続で経由する必要の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

  ネットレットの接続で経由する必要の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 (コンマ区切り)

  プロキシを通じたネットレット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。

◆ ◆ ◆ 第 12 章

PDC (Private Domain Certificates) を使用する場合のネットレットの設定

この章では、ネットレットで PDC を使用できるようにするための、クライアントブラウザの Java プラグインの設定について説明します。

注 – ネットレットでの 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 用のネットレットの設定

▼ ネットレットを PDC 用に設定する

1 **Portal Server** マシン上の `/ect/opt/SUNWam/config/AMConfig.properties` ファイル内の任意の場所に、**`com.iplanet.authentication.modules.cert.gwAuthEnable=yes`** を追加します。

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。

3 ゲートウェイマシンでルート **CA** 証明書をインポートします。

4 ゲートウェイプロファイルに **CA** 証明書を追加します。

ヒント – PDC をテストするには、独自のゲートウェイプロファイルを作成してください。

ゲートウェイプロファイルに証明書を追加するには、次の手順を実行します。

a. *Gateway Install Directory*/**SUNWportal/bin/certadmin -n gateway profile name** を実行します。
   Certadmin メニューが表示されます。

b. オプション **3** を選択します。

c. 証明書のパスを入力します。
   証明書が追加されたことを示すメッセージが表示されます。

5 **CA** に送信する証明書署名要求を生成します。
証明書署名要求を生成するには、次の手順を実行します。

a. *Gateway Install Directory*/**SUNWportal/bin/certadmin -n gateway profile name** を実行します。
   Certadmin メニューが表示されます。

b. オプション **2** を選択します。

c. 各質問に対して、適切な答えを入力します。

d. 要求をファイルに保存します。

6 証明書署名要求を **CA** に送信し、承認を受けます。

ヒント – CA の署名後に証明書応答を保存します。

---

7 **CA** で承認されたサーバー証明書をインポートします。
サーバー証明書をインポートするには、次の手順を実行します。

a. *Gateway Install Directory*/**SUNWportal/bin/certadmin -n gateway profile name** を実行します。
   Certadmin メニューが表示されます。

b. オプション **4** を選択します。

c. サーバー証明書が格納されているファイルの場所を入力します。

8 **Portal Server** マシンにルート **CA** 証明書をインポートします。

- **Application Server** では、次のコマンドを使用して `root-ca` を追加します。

  ```
  ./certutil -A -n rootca -t "TCu,TCu,TCuw" -d
  /var/opt/SUNWappserver/domains/domain1/config -a -i path to root-ca
  ```

◆ ◆ ◆ 第 13 章

13 プロキシレットの設定

この章では、Sun Java System Portal Server 管理コンソールでのプロキシレットの設定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。

プロキシレット属性の設定

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、「配備」オプションの「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

1. **Portal Server** 管理コンソールに管理者としてログインします。
2. 「**Secure Remote Access**」タブを選択し、「プロキシレット」タブを選択します。
3. 「**DN** を選択」リストボックスから適切な **DN** を選択するか、または特定のユーザーまたは組織の既存の **DN** を追加します。

4 「プロキシレット」ページで、次の属性を設定します。

属性名	説明
COS 優先順位	オプションのリストからプロキシレットトラフィックのサービスクラスを選択します。
プロキシレットアプレットを自動的にダウンロード	プロキシアプレットをクライアントマシンに自動的にダウンロードする場合は、「はい」をクリックします。プロキシレットアプレットをダウンロードするための基本的な要件は次のとおりです。 クライアントマシンでサーバーアプリケーションを実行できる クライアントマシンの Java バージョンが 1.4 以降である ブラウザが IE 6.0 SP2 または Firefox 2.0 である 正しいブラウザ権限
プロキシレット経由でポータルを更新	プロキシレットの起動後にポータルデスクトップを更新し、トラフィックがプロキシレットを経由するようにする場合は、「はい」をクリックします。「プロキシレットの起動後にポータルを更新」と「プロキシレットアプレットを自動的にダウンロード」の両方が有効になっている場合、「App Urls」は機能しません。
起動モード	「Java Web Start」または「アプレット」を選択します。
プロキシレットアプレットのデフォルトのバインド IP	プロキシレットがブラウザからの要求をバインドおよび待機する IP アドレスを入力します。
プロキシレットアプレットのデフォルトのポート	プロキシレットがブラウザからの要求を待機するポート番号を入力します。
自動プロキシ設定ファイルの位置	プロキシ自動設定 (PAC) ファイルから、またはプロキシ設定リストからのプロキシ設定が格納された設定ファイルの場所を入力します。

5 「プロキシレットルール」オプションで、次の操作を行います。

a. プロキシレットサービスから起動するアプリケーションのルールを指定します。

b. 「追加」をクリックします。

c. 「ドメイン」フィールドに、**`www.google.com`** のようなドメイン名を入力します。

d. プロキシレットで処理するドメインのホストおよび対応するポート番号を入力します。これにより、プロキシレットで **HTTP** 要求を解決し、要求がゲートウェイを経由しないようにします。

6 「保存」をクリックして終了します。

ポータルデスクトップのアプリケーションの設定

HTTP、FTP などの要求はプロキシレットサービスを経由します。管理者はプロキシレットルールにより、プロトコル、ホスト、またはドメインへのポートに基づいてマッピングを指定できます。プロキシレットルールにより、プロキシ自動設定 (PAC) ファイルのドメインとプロキシの設定値を指定できます。たとえば、すべての FTP トラフィックがネットレットを経由したり、すべての HTTP トラフィックがプロキシレットを経由するように経路ルールを作成することが可能です。プロキシレットサービスでレンダリングする必要がある定義済みアプリケーションを設定できます。これは、ユーザーまたは組織の設定に基づいて行うことができます。プロキシレットで処理するようにアプリケーションを追加すると、ユーザーデスクトップが管理しやすくなり、パフォーマンスも向上します。

▼ ポータルデスクトップのアプリケーションを設定する

始める前に
- 「プロキシレット」オプションが有効になっていることを確認します。プロキシレットの有効化の詳細については、「ゲートウェイプロファイル」の章を参照してください。

1 **Portal Server** 管理コンソールに管理者としてログインします。

2 「ポータル」タブを選択し、変更するポータルインスタンスを選択します。
「デスクトップ」ページが表示されます。

3 「**DN** を選択」リストボックスから適切な **DN** を選択するか、または特定のユーザーまたは組織の既存の **DN** を追加します。

4 「コンテナとチャネルを管理」リンクをクリックします。
「コンテナとチャネルを管理」ページが表示されます。

5 左の区画で、「プロキシレット」を選択します。

6 右の区画で、`appurl` リンクを選択します。

7 プロパティーウィザードで、アプリケーション名と値を入力します。必要に応じてアプリケーションのプロパティーを変更します。たとえば、アプリケーションの名前と、「**http://www.example.com**」を入力します。

8 「閉じる」をクリックして完了します。
これで、ユーザーまたは組織のレベルでポータルデスクトップにアプリケーションリンクを表示できるようになりました。

Java Web Start モードまたはアプレットモードでのプロキシレットの起動

ポータルデスクトップから、Java Web Start モードかアプレットモードのどちらかでプロキシレットを起動できます。

▼ Java Web Start モードまたはアプレットモードでプロキシレットを起動する

1 ポータルデスクトップにプロキシレットユーザーとしてログオンします。

2 フロントページで、プロキシレットチャネルに移動し、「編集」アイコンをクリックします。

3 「起動モード」リストボックスで、「**Java Web Start**」または「アプレット」オプションを選択します。

4 「完了」をクリックします。
プロキシレットを起動するには、「プロキシレットチャネル」からアプリケーションを選択します。これで、そのアプリケーションが Java Web Start モードまたはアプレットモードで起動します。

- 「自動ダウンロード」が選択されている場合は、プロキシレットチャネルの下のアプリケーションをクリックします。
- ユーザー設定に基づいて、プロキシレットコンソールが Java Web Start モードまたはアプレットモードの選択に応じて表示されます。すべての証明書を受け入れて、アプリケーションでの作業を続けます。

◆ ◆ ◆ 第 14 章

ネットファイルの設定

この章では、Sun Java System Portal Server 管理コンソールでのネットファイルの設定について説明します。

この章は、次の節で構成されています。

ネットファイルのタスクの設定

この節では、次のタスクについて説明します。

▼ 基本オプションを設定する

1. **Portal Server** 管理コンソールに管理者としてログオンします。
2. 「**Secure Remote Access**」タブを選択し、「ネットファイル」タブを選択します。
3. 「**DN** を選択」リストからユーザーまたは組織の **DN** を選択するか、または **DN** を追加します。
4. 次の属性を変更します。

属性名	説明
COS 優先順位	属性値を継承するかどうかの決定に使用される値を指定します。この属性の詳細については、『Sun Java System Directory Server 管理ガイド』を参照してください。
ドメイン/ホストの設定	ネットファイルが許可されたホストにアクセスするために必要なデフォルトドメインを入力します。 このデフォルト値が適用されるのは、ユーザーがネットファイル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。 注– 「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。
デフォルトの WINS/DNS サーバー	Microsoft Windows ホストへのアクセスでネットファイルが使用する WINS/DNS サーバーホストアドレスを入力します。 注– 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。
ホスト検出順序	「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指定します。
共通ホスト	ホスト名または完全修飾名を入力して、「追加」をクリックします。 指定したホスト名がユーザーの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の指定した値に上書きされます。 ネットファイルを使用してすべてのリモートネットファイルユーザーが利用できるホストのリストを設定します。

注 – 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。ユーザーのネットファイルには、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、ユーザーのネットファイルに 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。

5 「保存」をクリックして終了します。

▼ アクセス権限を設定する

1 **Portal Server** 管理コンソールに管理者としてログオンします。

2 「**Secure Remote Access**」タブを選択し、「ネットファイル」タブを選択します。

3 「**DN** を選択」リストからユーザーまたは組織の **DN** を選択するか、または **DN** を追加します。

4 「アクセス権限」をクリックし、次の属性を変更します。

属性名	説明
Windows ホストへのアクセス	ユーザーが Windows ホストにアクセスできるようにする場合は、「許可」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、「許可」チェックボックスにチェックマークが付いています。
FTP ホストへのアクセス	ユーザーが FTP ホストにアクセスできるようにする場合は、「許可」チェックボックスにチェックマークを付けます。
NFS ホストへのアクセス	ユーザーが NFS ホストにアクセスできるようにする場合は、「許可」チェックボックスにチェックマークを付けます。

属性名	説明
Netware ホストへのアクセス	ユーザーが Netware ホストにアクセスできるようにする場合は、「許可」チェックボックスにチェックマークを付けます。

5 「保存」をクリックして終了します。

▼ 「ホストの設定」を設定する

1 **Portal Server** 管理コンソールに管理者としてログオンします。

2 「**Secure Remote Access**」タブを選択し、「ネットファイル」タブを選択します。

3 「**DN** を選択」リストからユーザーまたは組織の **DN** を選択するか、または **DN** を追加します。

4 デフォルトでは、「ホストの許可/拒否」リストに * が指定されているため、ユーザーはネットファイル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、ユーザーがネットファイル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。

注 – ホストへのアクセスを拒否し、ユーザーがすでにネットファイルウィンドウでこのホストを追加している場合、ユーザーのネットファイルウィンドウには、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。ネットファイル 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。「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。

---

5 「保存」をクリックして終了します。

▼ 「処理の設定」を設定する

1 **Portal Server** 管理コンソールに管理者としてログオンします。

2 「**Secure Remote Access**」タブを選択し、「ネットファイル」タブを選択します。

3 「DN を選択」リストからユーザーまたは組織の DN を選択するか、または DN を追加します。

4 次の属性を変更します。

属性名	説明
デフォルトの圧縮タイプ	デフォルトのファイル圧縮形式として、ドロップダウンボックスから ZIP または GZ を選択します。
デフォルトの圧縮レベル	ドロップダウンボックスからデフォルトの圧縮レベルを選択します。デフォルトは 6 です。
一時ディレクトリの場所	一時ファイル用のディレクトリの場所を入力します。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。 一時ディレクトリは、ファイルのメール送信など、いくつかのファイル操作で必要とされます。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、必要な操作の完了後に削除されます。 注 – 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 – ネットファイル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、ネットファイルの一部の操作は、一時ディレクトリの容量がなくなると機能しません。
ファイルのアップロード制限 (M バイト)	このフィールドでアップロードできるファイルの最大サイズを入力します。デフォルト値は 5M バイトです。 アップロードされるファイルのサイズがここで指定した制限を超える場合は、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。無効な値を入力すると、ネットファイル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。

属性名	説明
検索ディレクトリ制限	1 回の検索操作で検索できるディレクトリの最大数を入力します。この制限は、ネットワークの渋滞を緩和するのに役立ち、多数のユーザーが同時にログインする場合のアクセス速度が向上します。デフォルト値は 100 です。 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、ほか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、その後に次のディレクトリに移動します。

5 「保存」をクリックして終了します。

▼ 処理権限を設定する

ユーザーがリモートホストから次のタスクを実行する権限を許可または拒否できます。

1 **Portal Server** 管理コンソールに管理者としてログオンします。

2 「**Secure Remote Access**」タブを選択し、「ネットファイル」タブを選択します。

3 「**DN** を選択」リストからユーザーまたは組織の **DN** を選択するか、または **DN** を追加します。

4 次の属性を変更します。

属性名	説明
ファイル名の変更	ユーザーがファイル名を変更できるようにする場合は、「許可」チェックボックスにチェックマークを付けます。このオプションはデフォルトで選択されています。

属性名	説明
ファイル/フォルダの削除	ユーザーがファイルおよびディレクトリを削除できるようにする場合は、「許可」チェックボックスにチェックマークを付けます。このオプションはデフォルトで選択されています。
ファイルのアップロード	ユーザーがファイルをアップロードできるようにするには、「許可」チェックボックスにチェックマークを付けます。このオプションはデフォルトで選択されています。
ファイル/フォルダのダウンロード	ユーザーがファイルまたはディレクトリをダウンロードできるようにする場合は、「許可」チェックボックスにチェックマークを付けます。このオプションはデフォルトで選択されています。
ファイル検索	ユーサーがファイル検索操作を実行できるようにする場合は、「許可」チェックボックスにチェックマークを付けます。このオプションはデフォルトで選択されています。
ファイルのメール送信	ユーザーがメールにアクセスできるようにする場合は、「許可」チェックボックスにチェックマークを付けます。このオプションはデフォルトで選択されています。
ファイルの圧縮	ユーザーが圧縮タイプを選択できるようにする場合は、「許可」チェックボックスにチェックマークを付けます。このオプションはデフォルトで選択されています。
ユーザー ID の変更	ユーザーが自分のユーザー ID を変更できるようにする場合は、「許可」チェックボックスにチェックマークを付けます。ユーザーは、ネットファイル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 を持つ場合があります。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、ネットファイル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が、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。

属性名	説明
Microsoft Windows ドメインの変更	ユーザーがデフォルトの Microsoft Windows ドメインホストを変更できるようにする場合は、「許可」チェックボックスにチェックマークを付けます。このオプションはデフォルトで選択されています。 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。

注 – これらのオプションのいずれかの選択を解除した場合、ユーザーが Portal Server デスクトップに再度ログオンするまで変更は有効になりません。

5 「保存」をクリックして終了します。

◆ ◆ ◆ 第 15 章

Secure Socket Layer 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の設定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。

アクセラレータの概要

外部アクセラレータは、Secure Socket Layer (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コプロセッサです。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加速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ー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 をほか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。

手順については、248 ページの「Crypto Accelerator 1000 を設定する」を参照してください。

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、第 10 章証明書の操作を参照してください。

248 ページの「Crypto Accelerator 1000 の有効化」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 15–1 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/SUNWportal/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

▼ Crypto Accelerator 1000 を設定する

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。

`SUNWcrypm`、`SUNWcrypu`、`SUNWcrysu`、`SUNWdcar`、`SUNWcrypr`、`SUNWcrysl`、`SUNWdcamn`、`SUNWdcav`

3 次のパッチをインストールします。これらのパッチは `http://sunsolve.sun.com` から入手できます。

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 `/usf/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 SUNWtlsu 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 レルムを作成し、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、問い合わせに応答します。

```
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/SUNWportal/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/SUNWportal/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.p12 -d /etc/opt/SUNWportal/cert/default -n server-cert
```

```
pk12util -i servercert.p12 -d /etc/opt/SUNWportal/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/SUNWportal/cert/default/.nickname** ファイルでニックネームを変更します。

```
vi /etc/opt/SUNWportal/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/SUNWportal/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。

```
./psadmin start-sra-instance -u amadmin -f passwordfile -N profilename -t gateway
```

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 (どちらも対象および非対称) 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。

手順については、252 ページの「Sun Crypto Accelerator 4000 を設定する」を参照してください。

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。

248 ページの「Crypto Accelerator 1000 の有効化」は、Crypto Accelerator 4000 のパラメータと値を示しています。

表 15–2 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

表 15–2 Crypto Accelerator 4000 のインストールチェックリスト (続き)

パラメータ	値
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/SUNWportal/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 を設定する

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 次のパッチをインストールします。このパッチは **http://sunsolve.sun.com** から入手できます。`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 `/usf/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 SUNWtlsu 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/SUNWportal/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/SUNWportal/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.p12 -d /etc/opt/SUNWportal/cert/default -n server-cert
```

```
pk12util -i servercert.p12 -d /etc/opt/SUNWportal/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/SUNWportal/cert/default
```

10 /etc/opt/SUNWportal/cert/default/.nickname ファイルでニックネームを変更します。

```
vi /etc/opt/SUNWportal/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。

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。

```
./psadmin start-sra-instance -u amadmin -f passwordfile -N profilename -t gateway
```

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。

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 Portal Server 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。

次のタスクを実行できます。

- 254 ページの「外部 SSL デバイスアクセラレータを有効化する」
- 255 ページの「外部 SSL デバイスアクセラレータを設定する」

▼ 外部 SSL デバイスアクセラレータを有効化する

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) で稼動していることを確認します。**

2 **HTTP 接続を有効にします。**

次の表は、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。

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を設定する

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/SUNWportal/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/SUNWportal/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。

   ```
   ./psadmin start-sra-instance -u amadmin -f passwordfile -N profilename -t gateway
   ```

パート III

Secure Remote Access サーバーの管理

Secure Remote Access サーバーには、管理に使用する次の 2 つのインタフェースがあります。

- Portal Server 管理コンソール
- psadmin コマンド行ユーティリティー (『Sun Java System Portal Server 7.2 Command-Line Reference』の第 1 章「psadmin Utility」)

管理作業の大半は、Web ベースの Portal Server 管理コンソールを通じて行います。この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。詳細については、『Sun Java System Portal Server 7.2 管理ガイド』の「Portal Server 管理コンソールの使用」を参照してください。

ただし、ファイルの修正などの管理作業には UNIX コマンド行インタフェースを使用します。

◆ ◆ ◆ 第 16 章

ゲートウェイの管理

ゲートウェイの管理タスク

ここでは、Portal Server ゲートウェイを管理するための次のタスクについて説明します。

▼ ゲートウェイプロファイルを作成する

1. **Portal Server** 管理コンソールに管理者としてログインします。
2. 「**Secure Remote Access**」タブをクリックして、「新規プロファイル」をクリックします。
   「新規プロファイル」ページが表示されます。
3. 新規ゲートウェイプロファイルの名前を入力します。
4. 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。

5 「了解」をクリックします。
新しいプロファイルが作成され、「プロファイル」ページに表示されます。

注意 – インスタンスのポートを変更して、使用中の既存ポートと競合しないようにしてください。

6 インスタンスを作成する必要があるマシンに **telnet** で接続します。デフォルトのゲートウェイインスタンスはこのマシンで稼働しています。

7 「今すぐ設定」モードで **AM-SDK** をインストールします。

8 **UI** インストーラを使用して、「今すぐ設定」モードまたは「あとで設定を選択」モードでゲートウェイをインストールします。

9 `/opt/SUNWportal/template/sra/GWConfig.properties.template` ファイルを一時的な場所にコピーします。たとえば、`/tmp` などにコピーします。

10 必要に応じて値を修正します。

注 – 各値が新しいプロファイルのゲートウェイインスタンスのポート番号と一致するようにしてください。

11 完了したら、次のコマンドを実行します。

```
./psadmin create-sra-instance -u amadmin -f <passwordfile> -S <template file location>.template -t gateway
```

12 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。

`./psadmin start-sra-instance –u amadmin – f` *<password file>* `–N` *<profile name>*`– t` *<gateway>*

ゲートウェイの起動および停止の詳細については、261 ページの「ゲートウェイインスタンスを起動する」を参照してください。ゲートウェイの設定については、第 8 章Secure Remote Access ゲートウェイの設定を参照してください。

▼ 同じ LDAP を使用するゲートウェイインスタンスを作成する

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.iplanet.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 **Access Manager** サービスを再起動します。

▼ ゲートウェイインスタンスを起動する

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。

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あと、次のコマンドを実行してゲートウェイを起動します。

```
./psadmin start-sra-instance –u amadmin – f <password file> –N <profile name> – t <gateway>
```

`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。あとで独自に新しいプロファイルを作成し、そのプロファイルを指定してゲートウェイを再起動することもできます。34 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。

注 – 別のゲートウェイインスタンスを起動する場合は、<*profile name*> を該当するプロファイル名に置き換えてください。

サーバー (ゲートウェイインスタンスが設定されているマシン) を再起動すると、ゲートウェイのすべてのインスタンスが再起動されます。

バックアップしたプロファイルが `/etc/opt/SUNWportal` ディレクトリに存在し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動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 -an | 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

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。

`./psadmin stop-sra-instance –u amadmin – f` <*password file*> `–N` <*profile name*> `– t` <*gateway*>

注 – 別のゲートウェイインスタンスを起動する場合は、<*profile name*> を該当するプロファイル名に置き換えてください。

2 まだ稼働しているゲートウェイプロセスがあ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。

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

▼ 管理コンソールを使用してゲートウェイを起動および停止する

1 『**Sun Java System Portal Server 7.2** 管理ガイド』の「管理コンソールにログインする」

2 「**Secure Remote Access**」タブを選択します。

3 「インスタンスを管理」サブメニューをクリックします。

4 「**SRA** プロキシインスタンス」で、インスタンスを選択します。

- インスタンスを起動する場合は「開始」をクリックします。

- インスタンスを停止する場合は「停止」をクリックします。

▼ 別のプロファイルでゲートウェイを再起動する

- ゲートウェイを再起動します。

  `./psadmin start-sra-instance –u amadmin – f <password file> –N <profile name> – t <gateway>`

▼ ゲートウェイを再起動する

- 端末ウィンドウで **root** として接続し、次の操作を行います。
  - 次の方法でウォッチドッグプロセスを開始します。

```
./psadmin sra-watchdog -u uid -f password-filename -t instance-type on
```

`[--adminuser | -u] uid`	管理者の識別名 (DN) またはユーザー ID を指定します。
`[-passwordfile | -f] password-filename`	パスワードファイル内の管理者のパスワードを指定します。
`[--type | -t] instance-type`	Secure Remote Access インスタンスのタイプを指定します。指定する値は、gateway、nlproxy、または rwproxy です。

ウォッチドッグコマンドについては、『Sun Java System Portal Server Command Line Reference Guide』を参照してください。

crontab ユーティリティーでエントリが作成され、ウォッチドッグプロセスが有効になります。ウォッチドッグは、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。

▼ 仮想ホストを指定する

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/SUNWportal/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。

`./psadmin start-sra-instance –u amadmin – f <password file> –N <profile name>– t <gateway>`

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。

▼ プロキシを指定する

1 コマンド行で、次のファイルを編集します。

`/etc/opt/SUNWportal/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。

`./psadmin start-sra-instance –u amadmin – f <password file> –N <profile name>– t <gateway>`

▼ ネットレットプロキシインスタンスを作成する

1 インスタンスを作成する必要があるマシンに **telnet** で接続します。デフォルトのゲートウェイインスタンスはこのマシンで稼働しています。

2 `/opt/SUNWportal/template/sra/NLPConfig.properties.template` ファイルを一時的な場所にコピーします。たとえば、`/tmp` などにコピーします。

3 新しいプロファイル用のファイルで必要に応じて値を変更します。

4 完了したら、次のコマンドを実行します。

```
./psadmin create-sra-instance -u amadmin -f <passwordfile> -S <template file location>.template -t nlproxy
```

5 変更を有効にするには、目的のゲートウェイプロファイル名でネットレットプロキシの新しいインスタンスを起動します。

`./psadmin start-sra-instance –u amadmin – f <password file> –N <profile name> – t nlproxy`

▼ ネットレットプロキシを再起動する

- 端末ウィンドウで **root** として接続し、次の操作を行います。

  - 次の方法でウォッチドッグプロセスを開始します。

    ```
    psadmin sra-watchdog -u uid -f password-filename -t instance-type on
    ```

    instance-type を `nlproxy` に置き換えて入力してください。このコマンドの詳細については、『Sun Java Portal Server Command Line Reference Guide』を参照してください。

    crontab ユーティリティーでエントリが作成され、ウォッチドッグプロセスが有効になります。ウォッチドッグはネットレット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、ネットレットプロキシを手動で起動します。

    ```
    psadmin start-sra-instance -u uid -f password-filename -N sra-instance-name
    -t instance-type
    ```

    instance-type を `nlproxy` に置き換えて入力してください。これは、目的のネットレットプロキシインスタンスに対応するプロファイル名です。このコマンドの詳細については、『Sun Java Portal Server Command Line Reference Guide』を参照してください。

▼ リライタプロキシインスタンスを作成する

1. インスタンスを作成する必要があるマシンに **telnet** で接続します。デフォルトのゲートウェイインスタンスはこのマシンで稼働しています。

2. `/opt/SUNWportal/template/sra/GWConfig.properties.template` ファイルを一時的な場所にコピーします。たとえば、`/tmp` などにコピーします。

3. 新しいプロファイル用のファイルで必要に応じて値を変更します。

4. 完了したら、次のコマンドを実行します。

   ```
   ./psadmin create-sra-instance -u amadmin -f <passwordfile> -S <template file
   location>.template -t rwproxy
   ```

5. 変更を有効にするには、目的のゲートウェイプロファイル名でリライタプロキシの新しいインスタンスを起動します。

   ```
   ./psadmin start-sra-instance –u amadmin – f <password file> –N <profile name> – t
   rwproxy
   ```

▼ リライタプロキシを再起動する

- 端末ウィンドウで **root** として接続し、次の操作を行います。

  - 次の方法でウォッチドッグプロセスを開始します。

    ```
    psadmin sra-watchdog -u uid -f password-filename -t instance-type on
    ```

    instance-type を `rwproxy` に置き換えて入力してください。このコマンドの詳細については、『Sun Java Portal Server Command Line Reference Guide』を参照してください。

    crontab ユーティリティーでエントリが作成され、ウォッチドッグプロセスが有効になります。ウォッチドッグはリライタ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、リライタプロキシを手動で起動します。

    ```
    start-sra-instance -u uid -f password-filename -N sra-instance-name -t
    instance-type
    ```

    instance-type を `rwproxy` に置き換えて入力してください。これは、目的のリライタプロキシインスタンスに対応するプロファイル名です。このコマンドの詳細については、『Sun Java Portal Server Command Line Reference Guide』を参照してください。

▼ 逆プロキシを有効化する

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/SUNWportal/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書き換え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書き換えるために使用されます。

3 ゲートウェイを再起動します。

`./psadmin start-sra-instance –u amadmin – f <password file> –N <profile name> – t <gateway>`

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。

▼ 既存の PDC インスタンスに認証モジュールを追加する

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」ドロップダウンボックスから「サービス」を選択します。
サービスが表示されます。

4 「認証設定」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 プロパティーを表示」ページが表示されます。

6 「編集」をクリックします。
「モジュールの追加」ページが表示されます。

7 「モジュール名」を選択し、「適用基準」を「必須」に設定します。

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 「gatewaypdc プロパティーの表示」ページをクリックします。

11 変更内容を有効にするために、ゲートウェイを再起動します。

gateway-install-location`/SUNWportal/bin/psadmin start-sra-instance –u amadmin – f <password file> –N <profile name>– t <gateway>`

▼ ブラウザキャッシングを無効にする

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/SUNWportal/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。

`./psadmin start-sra-instance –u amadmin – f <`*password file*`> –N <`*profile name*`> – t <`*gateway*`>`

▼ LDAP ディレクトリを共有する

1 `AMConfig.properties` の次の領域を変更して、最初にインストールした **Portal Server** および **Access Manager** サーバーのインスタンスと同期するようにします。

```
# The key that will be used to encrypt and decrypt passwords.
am.encryption.pwd=t/vnY9Uqjf12NbFywKuAaaHibwlDFNLO <== この文字列を最初のポータルのインストールの文字列に置き換えます

/* The following key is the shared secret for application auth module */
com.iplanet.am.service.secret=AQICxIPLNc0WWQRVlYZN0PnKgyvq3gTU8JA9 <== この文字列を最初のポータルのインストールの文字列に置き換えます
```

2 `/etc/opt/SUNWam/config/ums` にある `serverconfig.xml` の次の領域を変更して、最初にインストールした **Portal Server** および **Access Manager** サーバーのインスタンスと同期するようにします。

```
<DirDN>
    cn=puser,ou=DSAME Users,dc=sun,dc=net
</DirDN>
    <DirPassword>
        AQICxIPLNc0WWQT22gQnGgnCp9rUf+FuaqpY
        <==  この文字列を最初のポータルのインストールの文字列に置き換えます
  </DirPassword>

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

```
        AQICxIPLNc0WWQT22gQnGgnCp9rUf+FuaqpY
        <==  この文字列を最初のポータルのインストールの文字列に置き換えます
    </DirPassword>
```

3 **Access Manager** サービスを再起動します。

◆ ◆ ◆ 第 17 章

17 連携管理の例

この章では、次の内容について説明します。

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、ほか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リティー保護された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。Portal Server Secure Remote Access サーバーを使用してセキュリティー保護されたモードで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リティー保護された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ー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。詳細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と、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す。

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。

▼ 連携管理リソースを設定する

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。

▼ ゲートウェイでサービスプロバイダ (Portal Server) を設定する

1 **Portal Server** 管理コンソールに管理者としてログインします。

2 「**Secure Remote Access**」タブを選択し、属性を変更する適切な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。

3 「コア」タブを選択します。

4 「**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。

5 「セキュリティー」タブを選択します。

6 「非認証 **URL**」リストに含まれる `/amserver` や `/portal/dt` などの相対 **URL** を使用するには、「**Portal Server**」フィールドに **Portal Server** 名を入力します。次に例を示します。

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

`http://` *idp-host:por*`t/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`

7 「**Portal Server**」フィールドに、**Portal Server** 名を入力します。たとえば、`/amserver` と入力します。

8 「保存」をクリックします。

9 「セキュリティー」タブを選択します。

10 「非認証 **URL**」リストに、連携リソースを追加します。次に例を示します。

```
/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、「配備」タブを選択します。

14 「ドメインとサブドメインのプロキシ」フィールドに、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。

`./psadmin start-sra-instance –u amadmin – f` <*password file*> `–N` <*profile name*>`– t` <*gateway*>

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。

▼ ゲートウェイでサービスプロバイダ (Portal Server) を設定する

1 **Portal Server** 管理コンソールに管理者としてログインします。

2 「**Secure Remote Access**」タブを選択し、属性を変更する適切なゲートウェイプロファイルを選択します。

3 「コア」タブを選択します。

4 「**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。

5 「非認証 **URL**」リストに含まれる `/amserver` や `/portal/dt` などの相対 **URL** を使用するには、「**Portal Server**」フィールド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`

6 「保存」をクリックします。

7 「セキュリティー」タブをクリックします。

8 「非認証 **URL**」リストに、連携リソースを追加します。次に例を示します。

```
/amserver/config/federation

/amserver/IntersiteTransferService

/amserver/AssertionConsumerservice

/amserver/fed_images

/amserver/preLogin

/portal/dt
```

9 「追加」をクリックします。

10 「保存」をクリックします。

11 「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、「配備」タブを選択します。

12 「ドメインとサブドメインのプロキシ」フィールドに、**Web** プロキシに関する情報を入力します。

13 「追加」をクリックします。

14 「保存」をクリックします。

15 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。

```
./psadmin start-sra-instance –u amadmin – f <password file> –N <profile name>– t <gateway>
```

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。

▼ ゲートウェイで ID プロバイダ (Portal Server) を設定する

1 **Portal Server** 管理コンソールに管理者としてログインします。

2 「**Secure Remote Access**」タブを選択し、属性を変更する適切なゲートウェイプロファイルを選択します。

3 「コア」タブを選択します。

4 「**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。

5 「非認証 **URL**」リストに含まれる `/amserver` や `/portal/dt` などの相対 **URL** を使用するには、「**Portal Server**」フィールド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

6 「保存」をクリックします。

7 「セキュリティー」タブを選択します。

8 「非認証 **URL**」リストに、連携リソースを追加します。次に例を示します。

/amserver/config/federation

/amserver/IntersiteTransferService

/amserver/AssertionConsumerservice

/amserver/fed_images

/amserver/preLogin

/portal/dt

9 「追加」をクリックします。

10 「保存」をクリックします。

11 「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、「配備」タブを選択します。

12 「ドメインとサブドメインのプロキシ」フィールドに、**Web** プロキシに関する情報を入力します。

13 「追加」をクリックします。

14 「保存」をクリックします。

15 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。

./psadmin start-sra-instance –u amadmin – f <*password file*> –N <*profile name*>– t <*gateway*>

◆ ◆ ◆ 付 録 A

設定属性

この付録では、Portal Server 管理コンソールから Portal Server Secure Remote Access コンポーネントごとに設定できる、Sun Java System Portal Server Secure Remote Access の属性について説明します。

アクセス制御サービス

279 ページの「アクセス制御サービス」は、アクセス制御サービスの属性を示しています。

表 A–1 アクセス制御サービスの属性

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 A–1 アクセス制御サービスの属性 (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。

- 280 ページの「コア」
- 283 ページの「プロキシ」
- 283 ページの「セキュリティー」
- 285 ページの「リライタ」

コア

280 ページの「コア」は、ゲートウェイサービスのコア属性を示しています。

表 A–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 A–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ー保護できます。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、*host-name:port* の形式で個別に詳細を入力します。
ネットレットを有効	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ー保護できます。
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。
ネットレットプロキシを有効		クライアントからのセキュリティー保護されたトンネルを、ゲートウェイを経由してイントラネット内のネットレット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ネットレットトラフィックのセキュリティーを補強します。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。
ネットレットプロキシホスト		ネットレットプロキシホストを hostname:port 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 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。
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。
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。

表 A–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
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		サーブレットおよび 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		`http://`*portal server name:port -number* 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、またはネットレットプロキシがクラッシュや停止などで使用できなくなった場合に開始の試行を要求する時間間隔を指定します。
外部サーバーの Cookie を格納		ゲートウェイで、サードパーティー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。
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 の一部として符号化します。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

283 ページの「プロキシ」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 A–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。
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 プロキシ経由のネットレット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、セキュリティー保護されたトンネルを拡張します。

セキュリティー

283 ページの「セキュリティー」は、ゲートウェイサービスのセキュリティー属性を示しています。

表 A–4 ゲートウェイサービスのセキュリティー属性

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 サイトにふたた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。
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 Version 2.0 を有効	選択	SSL version 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、セキュリティーのレベルが格段に向上します。
SSL 暗号化方式選択を有効		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 暗号化方式		選択した SSL version 2 の暗号化方式をリストします。
SSL3 暗号化方式		選択した SSL version 3 の暗号化方式をリストします。
TLS 暗号化方式		TLS 暗号化方式をリストします。

表 A–4 ゲートウェイサービスのセキュリティー属性 (続き)

属性	デフォルト値	説明
SSL Version 3.0 を有効	選択	SSL version 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、セキュリティー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。
セキュリティー保護された Cookie としてマークする		セキュリティー保護された 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。

リライタ

「リライタ」タブは、さらに 2 つに分かれています。

- 285 ページの「基本」
- 286 ページの「詳細」

基本

285 ページの「基本」は、ゲートウェイサービスのリライタ基本属性を示しています。

表 A–5 ゲートウェイサービスのリライタ属性 - 基本

属性	デフォルト値	説明
すべての URI の書き換えを有効		「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 URI が書き換えられるように指定します。

表 A–5 ゲートウェイサービスのリライタ属性 - 基本 (続き)

属性	デフォルト値	説明
URI をルールセットにマップ	*://*.iportal.com*/portal/* |default_gateway_ruleset */portal/NetFileOpenFileServlet* |null_ruleset *|generic_ruleset REPLACE_WITH_IPLANET_MAIL_SERVER_NAME|iplanet_mail_ruleset REPLACE_WITH_EXCHANGE_SERVER_NAMEexchange_2000sp3_owa_ruleset *://*.iportal.com*/amconsole/* |default_gateway_ruleset REPLACE_WITH_INOTES_SERVER_NAME|inotes_ruleset http*://*/portal/NetFileController* |null_ruleset	「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。
パーサーを MIME タイプにマップ	JAVASCRIPT=application/x-java XML=text/xml HTML=text/html;text/htm;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、セミコロンまたはコンマで区切ります。
書き換えない URI		書き換えない URI をリストします。注: このリストに #* を追加すると、href ルールがルールセットの一部であっても、URI を書き換えできるようにできます。
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

286 ページの「詳細」は、ゲートウェイサービスのリライタ詳細属性を示しています。

表 A–6 ゲートウェイサービスのリライタ属性 - 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、リライタは拡張子が html、htm、Servlet のすべてのページのコンテンツを書き換えます。
マスキングを有効		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 を書き換えます。
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		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。 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。
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		HTML コンテンツ内で参照されるリソースへのアクセスに、リライタは同じプロトコルを使用できます。 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。

ネットファイルサービス

「ネットファイルサービス」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

「ホスト」タブは、さらに 2 つに分かれています。

設定

288 ページの「設定」は、ネットファイルサービスのホスト設定属性を示しています。

表 A–7　ネットファイル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		すべてのリモートネットファイルユーザーがネットファイル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		ネットファイル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 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。

表 A–7 ネットファイルサービスのホスト設定属性 (続き)

デフォルトの WINS/DNS サーバー		Windows ホストへのアクセスでネットファイルが使用する WINS/DNS サーバーを指定します。

アクセス

289 ページの「アクセス」は、ネットファイルサービスのホストアクセス属性を示しています。

表 A–8 ネットファイルサービスのホストアクセス属性

属性	デフォルト値	説明
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	*	ネットファイル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		ネットファイル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

ユーザーがネットファイルの使用を開始したあと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、ふたたびログインしたあとに変更内容が有効になります。

289 ページの「権限」は、ネットファイルサービスの権限属性を示しています。

表 A–9 ネットファイル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。

表 A–9 ネットファイル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。
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。
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

290 ページの「表示」は、ネットファイルサービスの表示属性を示しています。

表 A–10 NetFle 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ネットファイル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、ネットファイルはデフォルトの値を使用します。
ウィンドウの位置	100|50	ネットファイル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、ネットファイル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。

- 291 ページの「トラフィック」
- 291 ページの「検索」
- 291 ページの「圧縮」

トラフィック

291 ページの「トラフィック」は、ネットファイルサービスの操作トラフィック属性を示しています。

表 A–11 ネットファイル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	ネットファイル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。 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。 ネットファイル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。ネットファイルは一時ディレクトリの容量がなくなると機能しません。
ファイルのアップロード制限 (M バイト)	5	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、ネットファイル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。 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。

検索

291 ページの「検索」は、ネットファイルサービスの操作検索属性を示しています。

表 A–12 ネットファイル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

291 ページの「圧縮」は、ネットファイルサービスの操作圧縮属性を示しています。

表 A–13 ネットファイル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

292 ページの「一般」は、ネットファイルサービスの一般属性を示しています。

表 A–14 ネットファイル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/SUNWportal/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。

ネットレットサービス

292 ページの「ネットレットサービス」は、ネットレットサービスの属性を示しています。

表 A–15 ネットレットサービスの属性

属性	デフォルト値	説明
ネットレットルール		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 A–15 ネットレットサービスの属性 (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。 local-port:server-host:server-port
-- 拡張セッション		このルールに対応するネットレット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プ		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		ネットレット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。
-- 接続先ホスト		スタティックルールの場合は、ネットレット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」。
-- 接続先ポート		接続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		ネットレット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		ネットレット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	ネットレット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、ネットレットルール内で上書きできます。
接続の再認証		ネットレット接続を確立しようとするユーザーに、そのつどネットレットパスワードの入力を要求します。

表 A–15 ネットレットサービスの属性 (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	ユーザーがネットレット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	ネットレットがユーザーの標準ポータルデスクトップ上の接続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、アイドル状態のネットレット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ネットレットを終了	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。
ネットレットルールにアクセス	*	特定の組織、ロール、ユーザーに対して特定のネットレットルールへのアクセスを定義します。
ネットレットルールの拒否		特定の組織、ロール、ユーザーに対して特定のネットレット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

294 ページの「プロキシレットサービス」は、プロキシレットサービスの属性を示しています。

表 A–16 プロキシレットサービス属性

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	58081	プロキシレットが待機するポート。

◆ ◆ ◆ 付 録 B

ログファイル

次のファイルはデフォルトの /var/opt/SUNWportal/debug ディレクトリに格納されるログファイルで、デバック情報などの情報が記録されます。

ログファイルについて

表 B–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 /etc/opt/SUNWam/debug/ 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。Linux のパス名については、「Solaris と Linux のパス名の比較」を参照してください。	
amconsole	ネットファイル、ネットレット、および Gateway Admin ファイル
srapNetFile	ネットファイル情報ファイル
srapNetlet	ネットレット情報ファイル
srapProxylet	プロキシレットの情報ファイル

表 B–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
次のログファイルは、デフォルトディレクトリ `/etc/opt/SUNWportal` の `platform.conf`.gateway-profile-name ファイルのデバッグパラメータ `gateway.debug` によって制御されます。Linux のパス名については「Solaris と Linux のパス名の比較」を参照してください。	
srapGateway.gateway-profile-name	ゲートウェイ情報
Gateway_to_from_server.gateway-profile-name	
Gateway_to_from_browser.gateway-profile-name	
srapNetletProxy.gateway-profile-name	
srapRewriterProxy.gateway-profile-name	
rwproxy.log.rewriter-proxy-instance-name	リライタプロキシの開始時刻と停止時刻
nlproxy.log.netlet-proxy-instance-name	ネットレットプロキシの開始時刻と停止時刻
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、98 ページの「デバッグログを使用したトラブルシューティング」を参照してください。	
RuleSetInfo	書き換え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 (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。 このファイルには、解析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 モードでは書き換え時に発生した例外だけが記録されます。

表 B–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
Rewritten Pages	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、書き換えら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	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。 記録される詳細情報はページの URI、MIME、符号化データ、書き換え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。

◆ ◆ ◆ 付録 C

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。

国コードの一覧

表 C–1 2 文字の国コード

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

表 C–1　2 文字の国コード　*(続き)*

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
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

表 C–1　2 文字の国コード　(続き)

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
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

表 C–1　2 文字の国コード　*(続き)*

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
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

表 C–1　2 文字の国コード　(続き)

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
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

表 C–1　2 文字の国コード　*(続き)*

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
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	マラウィ
mx	メキシコ
my	マレーシア
mz	モザンビーク
na	ナミビア

表 C–1　2 文字の国コード　(続き)

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
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

表 C–1　2 文字の国コード　*(続き)*

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
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

表 C–1　2 文字の国コード　(続き)

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
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

表 C–1　2 文字の国コード　*(続き)*

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

索引

る

れ

ろ

わ