Sun Java System Web Proxy Server 4.0.4 管理ガイド

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

Part No: 820–0863
2007 年 3 月

070530@17466

目次

はじめに

このマニュアルでは、従来、Sun ONE™ Web Proxy Server および iPlanet™ Web Proxy Server と呼ばれていた、Sun Java ™ System Web Proxy Server 4 (以降、Sun Java System Web Proxy Server または単に Proxy Server と呼ぶ) の設定および管理の方法について説明します。

対象読者

このマニュアルは、運用環境の IT 管理者を対象にしています。このマニュアルでは、次の分野の知識があることを前提としています。

- 基本システムの管理タスクの実行
- ソフトウェアのインストール
- Web ブラウザの使用
- 端末ウィンドウでのコマンドの発行

このマニュアルをお読みになる前に

Sun Java System Web Proxy Server は単体でも、Sun Java Enterprise System のコンポーネントとしてもご購入頂けます。Sun Java Enterprise System は、ネットワークまたはインターネット環境へのエンタープライズアプリケーションの分散をサポートするソフトウェアインフラストラクチャです。Sun Java System Web Proxy Server を Java Enterprise System のコンポーネントとして購入した場合は、`http://docs.sun.com/coll/1286.2` にあるシステムマニュアルもよくお読みください。

内容の紹介

このマニュアルは、いくつかの部に分かれており、それぞれの部で特定の領域やタスクについて説明します。次の表は、マニュアルの各部とその内容を示しています。

表 P–1 マニュアルの構成

構成要素	説明
第 1 部サーバーの基本	Proxy Server とその管理の概要について説明します。 ■ 第 1 章 ■ 第 2 章
第 2 部管理サーバーの使用	管理サーバーの詳細設定、ユーザーとグループの管理、Proxy Server のセキュリティー保護、およびサーバー間で設定を共有するクラスタの使用の詳細について説明します。 ■ 第 3 章 ■ 第 4 章 ■ 第 5 章 ■ 第 6 章
第 3 部 Proxy Server の設定と監視	サーバーの詳細設定、アクセス制御の設定、およびサーバーアクティビティーの監視の詳細について説明します。 ■ 第 7 章 ■ 第 8 章 ■ 第 9 章 ■ 第 10 章
第 4 部 Proxy Server の管理	Proxy Server が要求を処理する方法に関連する概念とタスクの詳細について説明します。 ■ 第 11 章 ■ 第 12 章 ■ 第 13 章 ■ 第 14 章 ■ 第 15 章 ■ 第 16 章 ■ 第 17 章
第 5 部付録	アクセス制御リスト (Access Control List、ACL) ファイルの構文と、サーバーパフォーマンスの調整について説明します。 ■ 第 18 章 ■ 第 19 章

Proxy Server のマニュアルセット

マニュアルセットには、Proxy Server に関連のある Sun のドキュメントを記載しています。Proxy Server ドキュメントの URL は、`http://docs.sun.com/coll/1311.4` です。Proxy Server, の概要を理解するため、次の表に紹介されているマニュアルを、記載されている順番に参照してください。

表 P–2　Sun Java System Web Proxy Server のドキュメント

マニュアルタイトル	内容
『Sun Java System Web Proxy Server 4.0.4 リリースノート (UNIX 版)』	Proxy Server リリース ■ ソフトウェアとマニュアルの最新情報 ■ 新機能 ■ サポートされるプラットフォームと環境 ■ システム要件 ■ 既知の問題と回避策
『Sun Java System Web Proxy Server 4.0.4 Installation and Migration Guide 』	以下のインストールおよび移行作業の実行 ■ Sun Java System Web Proxy Server のインストール ■ バージョン 3.6 からバージョン 4 への移行
『Sun Java System Web Proxy Server 4.0.4 管理ガイド』	管理および管理タスクの実行 ■ 管理インタフェースおよびコマンド行インタフェースの使用 ■ サーバー環境の設定 ■ ユーザーとグループの管理 ■ サーバーアクティビティーの監視およびログ ■ サーバー保護のための証明書および公開鍵暗号の使用 ■ サーバーアクセスの制御 ■ URL のプロキシとルーティング ■ キャッシュ ■ コンテンツのフィルタリング ■ 逆プロキシの使用 ■ SOCKS の使用
『Sun Java System Web Proxy Server 4.0.4 Configuration File Reference 』	設定ファイルの編集
『Sun Java System Web Proxy Server 4.0.4 NSAPI Developer's Guide 』	カスタム NSAPI (Netscape Server Application Programmer's Interface) プラグインの作成

関連マニュアル

Sun Java Enterprise System (Java ES) とそのコンポーネントに関するすべてのドキュメントの URL は、`http://docs.sun.com/prod/entsys.5` です。

デフォルトのパスとファイル名

次の表は、このマニュアルで使用するデフォルトのパスやファイル名について説明したものです。

表 P–3 デフォルトのパスとファイル名

プレースホルダ	説明	デフォルト値
install-dir	Sun Java System Web Proxy Server のベースインストールディレクトリを示します。	Solaris と Linux のインストール先: /opt/sun/proxyserver40 Windows のインストール先: \Sun\ProxyServer40

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–4 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–4 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。

コマンド例のシェルプロンプト

次の表は、デフォルトのシステムプロンプトとスーパーユーザープロンプトを示しています。

表 P–5 シェルプロンプト

シェル	プロンプト
UNIX および Linux システムの C シェル	`machine_name%`
UNIX および Linux システムの C シェルのスーパーユーザー	`machine_name#`
UNIX および Linux システムの Bourne シェルおよび Korn シェル	`$`
UNIX および Linux システムの Bourne シェルおよび Korn シェルのスーパーユーザー	`#`
Microsoft Windows のコマンド行	`C:\`

記号の規則

次の表は、この用語集で使用される記号の一覧です。

表 P–6 記号の規則

記号	説明	例	意味
`[ ]`	省略可能な引数やコマンドオプションが含まれます。	`ls [-l]`	`-l` オプションは必須ではありません。
`{ | }`	必須のコマンドオプションの選択肢を囲みます。	`-d {y|n}`	`-d` オプションには `y` 引数か `n` 引数のいずれかを使用する必要があります。
`${ }`	変数参照を示します。	`${com.sun.javaRoot}`	`com.sun.javaRoot` 変数の値を参照します。
-	同時に押すキーを示します。	Control-A	Control キーを押しながら A キーを押します。
+	順番に押すキーを示します。	Ctrl+A+N	Control キーを押してから放し、それに続くキーを押します。
→	グラフィカルユーザーインタフェースでのメニュー項目の選択順序を示します。	「ファイル」→「新規」→「テンプレート」	「ファイル」メニューから「新規」を選択します。「新規」サブメニューから「テンプレート」を選択します。

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。

◆ ◆ ◆ 第 1 章

Sun Java System Web Proxy Server の概要

この章では、Sun Java System Web Proxy Server の概要について説明します。また、このリリースの新機能について簡単に説明し、Proxy Server の管理、設定、および管理で使用する Web ベースのユーザーインタフェースの概要について説明します。

この章の内容は次のとおりです。

Sun Java System Web Proxy Server について

Sun Java System Web Proxy Server とは、高パフォーマンスのインターネットおよびイントラネット環境に対する HTTP キャッシュおよびアクセラレーションの基盤です。Proxy Server は、Web コンテンツのキャッシュとフィルタ、およびネットワークパフォーマンスの向上のために使用されるシステムで、ネットワークインフラストラクチャーとの統合、クロスプラットフォームのサポート、および一元管理機能を備えています。ネットワークトラフィックマネージャーとして動作することで、情報を効率よく配布および管理するため、ネットワークトラフィックとユーザーの待ち時間が削減されます。また、Proxy Server を使用すると、コンテンツ配布用のセキュリティー保護されたゲートウェイとなり、インターネットトラフィックの制御点として機能するため、ユーザーは、ネットワークリソースに安全かつ生産的にアクセスできるようになります。

このリリースでの新機能

Sun Java System Web Proxy Server 4 では、次の機能が追加されています。

- 最新の HTTP コア
- Linux および Solaris™ x86 プラットフォームのサポート
- 全プラットフォームで最新の SSL (Secure Sockets Layer) をサポート
- 全プラットフォームでのマルチスレッドアーキテクチャー
- 管理機能、グラフィカルユーザーインタフェース、および管理しやすさの向上
- 新しい NSAPI (Netscape Server Application Programmer's Interface) フィルタ
- LDAP (Lightweight Directory Access Protocol) パフォーマンスの向上
- スケーラビリティとパフォーマンスの向上
- コンテンツフィルタリングの向上
- `server.xml` 設定ファイルの実装

新機能と機能追加については、次の Web サイトで入手できる『Proxy Server リリースノート』を参照してください。`http://docs.sun.com/app/docs/coll/1311.4`。

お読みになる前に

Sun Java System Web Proxy Server の管理と設定は、Web ベースのユーザーインタフェースである、管理サーバーとサーバーマネージャーを使用します。システムにインストールされたすべての Proxy Server インスタンスに共通の設定を管理するには管理サーバーを、個々のサーバーインスタンスを設定するにはサーバーマネージャーを使用します。

ここでは、次の内容について説明します。

- 30 ページの「管理サーバーの概要」
- 32 ページの「サーバーマネージャーの概要」
- 33 ページの「設定ファイル」
- 34 ページの「正規表現」

注 – サーバーの設定に必要な CGI プログラムを起動できるように、ブラウザの cookie を有効にする必要があります。

管理サーバーの概要

管理サーバーは Web ベースのユーザーインタフェースで、システムにインストールされたすべての Sun Java System Web Proxy Server インスタンスに共通の設定を管理するために使用します。

管理サーバーを起動したら、ブラウザを起動して URL を入力し、管理サーバーにアクセスします。URL は、インストール中に指定したホスト名とポート番号で決まります。たとえば `http://myserver.mycorp.com:1234` などです。

管理サーバーへのアクセス権は、複数の管理者に与えることができます。分散管理の詳細については、42 ページの「複数の管理者の許可」を参照してください。

管理サーバーの設定は、特定のタスクに対応するタブを使用して管理できます。次の表は、管理サーバーのタブと、各タブの使用目的についての簡単な説明を示しています。

- 「Servers」: Proxy Server の管理、追加、削除、および移行
- 「Preferences」: 管理サーバーのシャットダウン、待機ソケットの編集、スーパーユーザーのアクセスの設定、分散管理の設定 (複数の管理者の許可)、アクセスおよびエラーログのカスタマイズと表示
- 「Global Settings」: ディレクトリサービスの設定、アクセス制御の指定、SNMP マスターエージェントの設定
- 「Users and Groups」: ユーザー、グループ、組織単位の追加および管理
- 「Security」: 新しい信頼データベースの作成、VeriSign などの証明書の要求およびインストール、鍵ペアファイルパスワードの変更、インストール済み証明書の表示および管理、証明書の失効リスト (CRL) と危殆化鍵リスト (CKL) の追加および置換、CRL と CKL の管理、3.x 証明書の移行
- 「Cluster」: クラスタ内のリモートサーバーの制御、リモートサーバーの追加と削除、サーバー情報の変更

また、表示しているタブやページに関わらず、次のボタンも表示されます。

- 「Version」: Sun Java System Web Proxy Server のバージョン情報を表示します
- 「Refresh」: 現在のページを最新の情報に更新します
- 「Help」: 現在のページに対するオンラインヘルプを表示します

▼ 管理サーバーへアクセスするには

1. ブラウザを起動し、インストール時に管理サーバー用に指定したホスト名とポート番号を使用した **URL** を入力します。たとえば、`http://myserver.mycorp.com:1234` などです。

2. プロンプトが表示されたら、インストール時に指定したユーザー名とパスワードを入力します。

   管理サーバーのユーザーインタフェースが表示されます。

   管理サーバーの使用の詳細については、第 2 章を参照してください。管理サーバーのタブやページのオンラインヘルプも参照してください。

サーバーマネージャーの概要

サーバーマネージャーは、Web ベースのユーザーインタフェースで、これを使用して、Sun Java System Web Proxy Server の各インスタンスを起動、停止、および設定します。

サーバーマネージャーの設定は、特定のタスクに対応するタブを使用して管理できます。次の表は、サーバーマネージャーのタブと、各タブの使用目的についての簡単な説明を示しています。

- 「Preferences」: サーバーの起動と停止、サーバー設定の表示、設定情報の復元、システムの詳細設定、Proxy Server のパフォーマンスの調整、待機ソケットの追加と編集、MIME タイプの管理、アクセス制御の管理、ACL および DNS キャッシュの設定、DNS ローカルサブドメインの管理、HTTP キープアライブの設定、暗号化方式のサイズの設定
- 「Routing」: プロキシの有効化と無効化、ルーティングの詳細設定、クライアント資格の転送、Java IP アドレスチェックの有効化、自動設定ファイルの作成と編集、接続モードの設定、デフォルトの FTP 転送モードの変更、SOCKS ネームサーバーの IP アドレス設定、HTTP 要求のロードバランスの設定
- 「SOCKS」: SOCKS サーバーの起動と停止、SOCKS 認証、接続、ルーティングエントリの作成と管理
- 「URL」: URL マッピングとリダイレクトの表示、作成、および管理
- 「Caching」: キャッシュの詳細の設定、キャッシュパーティションの追加および変更、既存パーティション間でのセクションの移動、キャッシュ容量の設定、ガベージコレクションモードの設定、キャッシュの調整、ガベージコレクションのスケジュール、ガベージコレクション設定の調整、特定リソースに対するキャッシュの設定、ローカルホストのキャッシュの有効化、ファイルキャッシュ設定の変更、キャッシュのバッチ更新の設定、記録されたキャッシュ済み URL に関する情報の表示、隣接 ICP でのプロキシの設定、プロキシ配列メンバーリストの作成および更新、プロキシ配列メンバーの設定、PAT ファイル内の情報の表示
- 「Filters」: フィルタファイルの作成、コンテンツ URL のリライトの設定、user-agent 制限および要求のブロック処理の設定、送信されるヘッダーの抑止、MIME フィルタと HTML タグフィルタの設定、オンデマンドでのコンテンツの圧縮
- 「Server Status」: ログファイルの表示、ログのアーカイブ、ログの詳細設定、レポートの生成、現在のアクティビティーの監視、SNMP サブエージェントの設定および制御
- 「Security」: 新しい信頼データベースの作成、VeriSign などの証明書の要求およびインストール、鍵ペアファイルパスワードの変更、インストール済み証明書の表示および管理、証明書の失効リスト (CRL) と危殆化鍵リスト (CKL) の追加および置換、CRL と CKL の管理、3.x 証明書の移行
- 「Templates」: テンプレートの作成、削除、適用、表示、およびリソースの削除

また、表示しているタブやページに関わらず、次のボタンも表示されます。

- 「Version」: Sun Java System Web Proxy Server のバージョン情報を表示します
- 「Refresh」: 現在のページを最新の情報に更新します
- 「Help」: 現在のページに対するオンラインヘルプを表示します

「Refresh」ボタンの下に「Restart Required」リンクが表示されることがあります。このリンクは、再起動が必要なサーバーに変更が行われたことを示します。変更を適用するには、リンクをクリックし、目的のアクションを指定します。

サーバーマネージャーの使用については、このマニュアルに記載されている関連タスクを参照してください。サーバーマネージャーのタブやページのオンラインヘルプも参照してください。

▼ サーバーマネージャーへアクセスするには

1 30 ページの「管理サーバーの概要」の説明に従って、管理サーバーにアクセスします。

管理サーバーが「Server」タブに表示されます。

2 「**Manage Servers**」ページで、管理するサーバーインスタンスのリンクをクリックします。

サーバーマネージャーのユーザーインタフェースが表示されます。

設定ファイル

Sun Java System Web Proxy Server の設定と動作は、一連の設定ファイルによって決まります。管理インタフェースで設定された内容は、設定ファイルに反映されます。ファイルは手動で編集することもできます。

設定ファイルは、ディレクトリ *instance-dir* `/config` (ここで *instance-dir* はサーバーインスタンス) にあります。`config` ディレクトリには、各種のコンポーネントを制御するさまざまな設定ファイルが格納されています。設定ファイルの数や名前は、どのコンポーネントが有効化、またはロードされているかによって異なります。このディレクトリには、サーバーの動作に必須の 4 つの設定ファイルが常に格納されています。次の表は、それら 4 つの必須設定ファイルとその内容を示しています。

表 1–1 必須設定ファイル

ファイル	内容
`server.xml`	サーバー設定の大部分 (今回の Proxy Server のリリースで導入)
`magnus.conf`	サーバー初期化に関するグローバル情報

表 1–1 必須設定ファイル (続き)

ファイル	内容
obj.conf	クライアントからの要求処理に関する指示
mime.types	要求されたリソースのコンテンツタイプを決定するための情報

これらを始めとする設定ファイルについては、『Proxy Server 4.0.4 *Configuration File Reference*』を参照してください。

正規表現

正規表現を使用して、リソースを識別し、さまざまな URL から個別に送られてくる要求を処理できるように Proxy Server を設定できます。管理サーバーおよびサーバーマネージャーのユーザーインタフェースを使用して、さまざまなタスクを実行する場合にも、正規表現を指定できます。正規表現の使用の詳細については、第 16 章を参照してください。

◆ ◆ ◆ 第 2 章

Sun Java System Web Proxy Server の管理

この章では、管理サーバーを使用して Sun Java System Web Proxy Server 管理の基本について説明します。管理サーバーは Web ベースのユーザーインタフェースで、サーバーを管理、追加、削除、および移行するために使用します。

この章の内容は次のとおりです。

管理サーバーの詳細設定の設定については、第 3 章を参照してください。サーバークラスタを使用した複数の Proxy Server の管理については、第 6 章を参照してください。

管理サーバーの起動

この節では、さまざまなプラットフォームで管理サーバーを起動する方法について説明します。管理サーバーの停止については、36 ページの「管理サーバーの停止」を参照してください。

UNIX または Linux で管理サーバーを起動するには

1. コマンド行から *server-root*/`proxy-admserv` に移動します。
2. `./start` と入力して管理サーバーを起動するか、または `./restart` と入力して管理サーバーを再起動します。

Windows で管理サーバーを起動するには

Windows で管理サーバーを起動するには、次のいずれかの方法を使用します。

- 「スタート」 > 「プログラム」 > 「Sun Microsystems」 > 「Sun Java System Web Proxy Server *version*」 > 「Start Admin」を使用します。
- 「コントロール パネル」 > 「管理ツール」 > 「サービス」 > 「Sun Java System Web Proxy Server 4.0 Administration Server」 > 「開始」を使用します。
- コマンドプロンプトから *server-root* `\proxy-admserv` に移動し、`startsvr.bat` と入力して管理サーバーを起動するか、または `./restart` と入力して管理サーバーを再起動します。

管理サーバーが起動したら、ブラウザを起動し、インストール時に管理サーバー用に指定したホスト名とポート番号を使用した URL を入力すると、管理サーバーにアクセスできます (たとえば `http://myserver.mycorp.com:1234`)。プロンプトが表示されたら、インストール時に指定したユーザー名とパスワードを入力します。

管理サーバーへのアクセス権は、複数の管理者に与えることができます。分散管理については、42 ページの「複数の管理者の許可」を参照してください。

管理サーバーの停止

この節では、さまざまなプラットフォームで管理サーバーを停止する方法について説明します。管理サーバーの起動については、35 ページの「管理サーバーの起動」を参照してください。

UNIX または Linux で管理サーバーを停止するには

UNIX または Linux で管理サーバーを停止するには、次のいずれかの方法を使用します。

- 管理インタフェースを使用します。
  1. 管理サーバーへアクセスします。
  2. 「Preferences」タブをクリックします。
  3. 「Shutdown Server」リンクをクリックします。
  4. ［了解］をクリックします。
- コマンド行から *server-root*`/proxy-admserv/` に移動し、`./stop` と入力します。

Windows で管理サーバーを停止するには

Windows で管理サーバーを停止するには、次のいずれかの方法を使用します。

- 「サービス」ウィンドウの「Sun Java System Proxy Server 4.0 Administration Server」サービスを使用します。「コントロール パネル」 > 「管理ツール」 > 「サービス」 > 「Sun Java System Web Proxy Server 4.0 Administration Server」 > 「停止」を使用します。
- コマンドプロンプトから *server-root* `\proxy-admserv` に移動し、`stopsvr.bat` と入力します。

複数の **Proxy Server** の実行

システムで複数の Proxy Server を実行するには、複数のサーバーインスタンスをインストールして設定する必要があります。次の手順では、サーバーインスタンスを追加する方法について説明します。

▼ 複数のサーバーインスタンスをインストールするには

1. 管理サーバーへアクセスします。
2. 「**Servers**」タブで、「**Add Server**」をクリックします。
3. 要求された情報を入力し、「了解」をクリックします。
   特定のフィールドについては、オンラインヘルプを参照してください。
4. 必要に応じて、「**Success**」ページ上の「**Configure Your New Server**」リンクをクリックします。このページは、正常に新規サーバーインスタンスが追加されると表示されます。
   サーバーマネージャーのインタフェースが表示されます。これを使用して、サーバーインスタンスを設定します。

サーバーインスタンスの削除

管理サーバーを使用して、Proxy Server インスタンスを削除できます。このプロセスは元に戻すことができないため、次の手順を実行する前に、そのサーバーインスタンスを削除することを確認してください。

▼ サーバーインスタンスを削除するには

1. 管理サーバーへアクセスします。
2. 「**Servers**」タブで、「**Remove Server**」をクリックします。
3. 削除するサーバーインスタンスをドロップダウンリストから選択します。
4. 「**Confirming Server Removal**」チェックボックスを選択し、「了解」をクリックします。

Proxy Server 3.6 からの移行

Sun One Web Proxy Server 3.6 (iPlanet Web Proxy Server とも呼ばれる) は、Sun Java System Web Proxy Server 4 に移行できます。それまでの 3.6 サーバーは保持され、新たにバージョン 4 サーバーが同じ設定で作成されます。バージョン 3.6 からバージョン 4 への移行については、『Sun Java System Web Proxy Server 4.0.4 Installation and Migration Guide』を参照してください。Proxy Server ユーザーインタフェースから、移行関連ページのオンラインヘルプも参照してください。証明書の移行については、このマニュアルの 88 ページの「以前のバージョンからの証明書の移行」を参照してください。

◆ ◆ ◆ 第 3 章

3 管理の詳細設定の設定

この章では、管理サーバーを使用した管理の詳細設定方法について説明します。サーバーの設定に必要な CGI プログラムを起動できるように、ブラウザの cookie を有効にする必要があります。

この章の内容は次のとおりです。

待機ソケットの作成および管理

サーバーで要求を処理するには、待機ソケットを使用して要求を受け入れてから、適切なサーバーにその要求を送信する必要があります。Proxy Server をインストールすると、待機ソケット `ls1` が自動的に作成されます。この待機ソケットには、`0.0.0.0` の IP アドレスと、インストール時に管理サーバーのポート番号として指定したポート番号が割り当てられます。

待機ソケットの追加、編集、および削除は、管理サーバーの「Edit Listen Sockets」ページを使用して実行できます。サーバーにアクセスする待機ソケットは、少なくとも 1 つ必要です。待機ソケットが 1 つしかない場合は、削除できません。

この節では、待機ソケットの追加、編集、および削除について説明します。

▼ 待機ソケットを追加するには

1 管理サーバーにアクセスして、「**Preferences**」タブを選択します。

2 「**Edit Listen Sockets**」リンクをクリックします。

3 「新規」ボタンをクリックします。

4 設定を指定し、「了解」をクリックします。
特定のフィールドについては、オンラインヘルプを参照してください。

▼ 待機ソケットを編集するには

1 管理サーバーにアクセスして、「**Preferences**」タブを選択します。

2 「**Edit Listen Sockets**」リンクをクリックします。

3 編集する待機ソケットのリンクをクリックします。

4 変更を行い、「了解」をクリックします。

▼ 待機ソケットを削除するには

1 管理サーバーにアクセスして、「**Preferences**」タブを選択します。

2 「**Edit Listen Sockets**」リンクをクリックします。

3 削除する待機ソケットの隣のチェックボックスを選択し、「了解」をクリックします。

4 削除を確認するプロンプトが表示されたら「了解」をクリックします。
サーバーにアクセスする待機ソケットは、少なくとも 1 つ必要です。待機ソケットが 1 つしかない場合は、その待機ソケットを削除できません。

スーパーユーザー設定の変更

管理サーバーにスーパーユーザーのアクセスを設定できます。この設定は、スーパーユーザーアカウントにのみ影響します。管理サーバーが分散管理方式を採用している場合には、許可する管理者に対して、別のアクセス制御を設定する必要があります。

注意 – ユーザーやグループを管理するために Sun Java System Directory Server を使用する場合、スーパーユーザー名やパスワードを変更する前に、ディレクトリ内のスーパーユーザーエントリを更新する必要があります。先にディレクトリを更新しないと、管理サーバーの「Users and Groups」インタフェースにアクセスできません。これに対処するには、このディレクトリにアクセスできる管理者アカウントを使用して管理サーバーにアクセスするか、または Directory Server のコンソールや設定ファイルを使用してディレクトリを更新する必要があります。

▼ 管理サーバーのスーパーユーザーの設定を変更するには

1. 管理サーバーにアクセスして、「**Preferences**」タブを選択します。
2. 「**Control Superuser Access**」リンクをクリックします。
3. 変更を行い、「了解」をクリックします。
   特定のフィールドについては、オンラインヘルプを参照してください。

   スーパーユーザーのユーザー名とパスワードは、*server-root* `/proxy-admserv/config` 内の `admpw` ファイルに格納されています。このファイルは、*username* :*password* の書式になっています。このファイルを開いてユーザー名を確認できますが、パスワードは暗号化されているため読み取ることはできません。パスワードを忘れた場合は、新しいパスワードに変更できます。

▼ スーパーユーザーのパスワードを変更するには

1. `admpw` ファイルを編集して、暗号化されているパスワードを削除します。
2. ユーザー名を使用して、パスワードを指定せずに管理サーバーにアクセスします。
3. 「**Preferences**」タブをクリックします。
4. 「**Control Superuser Access**」リンクをクリックします。

5 新しいパスワードを入力し、「了解」をクリックします。

注意 – admpw ファイルは編集可能なため、サーバーコンピュータを安全な場所に保管し、そのファイルシステムへのアクセスを制限することが非常に重要です。

UNIX と Linux システムでは、ファイルの書き込みは root のみに限定するようにファイルの所有権を変更することを検討してください。そうしないと、どのシステムユーザーでも管理サーバーデーモンを実行できるようになってしまいます。Windows システムでは、ファイル所有権は、管理サーバーが使用するユーザーアカウントに限定します。

複数の管理者の許可

分散管理により、複数の管理者がサーバーの特定の部分を変更することができます。分散管理を有効にするには、事前にディレクトリサーバーをインストールしておく必要があります。デフォルトのディレクトリサービスは、LDAP ベースにする必要があります。

分散管理には、2 つのレベルのユーザーがあります。スーパーユーザーと管理者です。

- スーパーユーザーとは、*server-root* /proxy-admserv/config/admpw 内のリストにあるユーザーです。これは、インストール時に指定したユーザー名とパスワードになります。このユーザーは、管理サーバーのすべてのフォーム (ただし、「Users and Groups」フォームは除く) へのすべてのアクセス権を持っています。「Users and Groups」フォームへは、LDAP サーバーで有効なアカウントを持つスーパーユーザーがアクセスできます。
- 管理者は、管理サーバーを含む、特定のサーバーの「Server Manager」フォームへ直接、アクセスできます。フォームの内容は、設定されているアクセス制御規則 (通常はスーパーユーザーにより設定される) により変わります。管理者は、限定された管理業務を実行でき、また、ユーザーの追加、アクセス制御の変更などその他のユーザーに影響する項目を変更できます。

アクセス制御については、第 8 章を参照してください。

▼ 分散管理を有効にするには

1 ディレクトリサーバーがインストールされていることを確認します。

2 管理サーバーへアクセスします。

3. (オプション) ディレクトリサーバーをインストールした後で、管理グループをまだ作成していない場合は管理グループを作成する必要があります。グループを作成するには、次の手順を実行します。

   a. 「**Users and Groups**」タブをクリックします。

   b. 「**Create Group**」リンクをクリックします。

   c. **LDAP** ディレクトリに管理者グループを作成し、管理サーバー (または、サーバー **root** にインストールされたその他のサーバー) の設定アクセス権を付与するユーザーの名前を追加します。

      特定のフィールドについては、オンラインヘルプを参照してください。

      管理者グループ内のすべてのユーザーは、管理サーバーへのすべてのアクセス権を保持していますが、アクセス制御を使用して、それらのユーザーが設定できるサーバーやフォームを制限することもできます。

      アクセス制御リストを作成すると、このリストに分散管理グループが追加されます。管理者グループの名前を変更する場合は、参照先のグループを変更するため、アクセス制御リストを手動で編集する必要があります。

4. 「**Preferences**」タブをクリックします。

5. 「**Configure Distributed Administration**」リンクをクリックします。

6. 「**Yes**」を選択し、管理者グループを指定し、「了解」をクリックします。

ログファイルオプションの指定

管理サーバーのログファイルには、管理サーバーに関するデータが記録されます。これには、検出したエラーのタイプやサーバーアクセスに関する情報が記録されます。これらのログを確認することで、サーバーのアクティビティーを監視したり、障害追跡に役立てたりすることができます。「Log Preferences」ページのさまざまなオプションを使用して、管理サーバーログに記録されるデータのタイプや形式を指定できます。サーバーについて決まった量の情報を提供する共通ログファイル形式を選択したり、必要に応じてカスタムログファイル形式を作成したりすることもできます。

管理サーバーの「Log Preferences」ページにアクセスするには、「Preferences」タブをクリックし、次に「Access Log Preferences」または「Error Log Preferences」リンクをクリックします。ログファイルとログファイルオプションの設定については、第 9 章を参照してください。オンラインヘルプも参照してください。

ログファイルの表示

管理サーバーのログファイルは、*server-root* /proxy-admserv/logs にあります。エラーログとアクセスログについては、両方とも、Proxy Server の管理コンソールから表示したり、テキストエディタを使用して表示したりすることができます。

アクセスログファイル

アクセスログファイルには、サーバーへの要求やサーバーからの応答に関する情報が記録されます。

▼ アクセスログファイルを表示するには

1. 管理サーバーにアクセスして、「**Preferences**」タブをクリックします。
2. 「**View Access Log**」リンクをクリックします。
   特定のフィールドについては、オンラインヘルプを参照してください。第 9 章も参照してください。

エラーログファイル

エラーログには、ログファイル作成以降にサーバーが検出したエラーすべてが記録されます。このファイルには、サーバーの起動時刻や、ログインに失敗したユーザー名などのサーバーに関する情報メッセージも記録されます。

▼ エラーログファイルを表示するには

1. 管理サーバーにアクセスして、「**Preferences**」タブをクリックします。
2. 「**View Error Log**」リンクをクリックします。
   特定のフィールドについては、オンラインヘルプを参照してください。第 9 章も参照してください。

ディレクトリサービスの使用

ユーザーの名前やパスワードなどの情報は、LDAP を使用して 1 つのディレクトリサーバーで保管し、管理することができます。また、サーバーを設定して、簡単にアクセスできる複数のネットワークロケーションからユーザーがディレクトリ情報を引き出せるようにすることもできます。ディレクトリサービスの使用については、第 4 章を参照してください。

サーバーへのアクセスの制限

Proxy Server が受信した要求を評価する場合、アクセス制御エントリ (Access Control Entry、ACE) と呼ばれる規則の階層に基づいてアクセス権を決定し、一致するエントリを使用して、要求を許可するか、拒否するかを決定します。各 ACE は、サーバーが階層内の次の ACE に進むべきかどうかを指定します。ACE の集合は、アクセス制御リスト (ACL) と呼ばれます。

管理サーバーや、サーバーインスタンス内の特定リソース (ファイル、ディレクトリ、ファイルタイプなど) へのアクセスに対して、アクセス制御を設定できます。管理サーバーへのアクセス制御は、管理サーバーの「Global Settings」タブで設定します。サーバーインスタンス内のリソースへのアクセス制御は、サーバーマネージャーの「Preferences」タブで設定します。アクセス制御の設定については、第 8 章を参照してください。

注 – サーバーアクセスを制限する前に、分散管理を有効にする必要があります。詳細は、42 ページの「複数の管理者の許可」を参照してください。

SNMP マスターエージェントの設定

SNMP (Simple Network Management Protocol) は、ネットワークアクティビティーに関するデータをやり取りするために使用されるプロトコルです。この情報は、サブエージェントとマスターエージェントを使用して、ネットワーク管理ステーションとサーバーの間で転送されます。

SNMP マスターエージェントの設定は、管理サーバーの「Global Settings」タブで設定します。マスターエージェントは、管理サーバーとともにインストールされます。SNMP およびエージェントの設定については、第 10 章を参照してください。また、管理サーバーの「Global Settings」タブ上のマスターエージェントページと、サーバーマネージャーの「Server Status」タブ上のサブエージェントページについては、オンラインヘルプも参照してください。

◆ ◆ ◆ 第 4 章

4 ユーザーとグループの管理

この章では、Proxy Server にアクセスするユーザーとグループを追加、削除、変更、および管理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。

ユーザーとグループに関する情報へのアクセス

管理サーバーを使用して、ユーザーアカウント、グループリスト、アクセス特権、組織単位、およびその他のユーザーやグループに固有の情報に関するアプリケーションデータにアクセスできます。

ユーザーとグループの情報は、テキスト形式のフラットファイル、または LDAP (Lightweight Directory Access Protocol) をサポートする Sun Java System Directory Server などのディレクトリサーバーに格納されます。LDAP は、オープンディレクトリアクセスプロトコルで、TCP/IP (Transmission Control Protocol/Internet Protocol) 上で動作し、グローバルサイズに、また百万単位のエントリにまで拡張可能です。

ディレクトリサービスについて

ディレクトリサービスを使用すると、すべてのユーザー情報を 1 つのソースで管理できます。Proxy Server では、次の 3 種類のディレクトリサービスを設定できます。LDAP、鍵ファイル、およびダイジェストファイルです。

ディレクトリサービスが設定されていない場合、最初に作成される新しいディレクトリサービスには、種類に関係なく `default` という値が設定されます。ディレクトリサービスを作成すると、ディレクトリサービスの詳細によって *server-root* `/userdb/dbswitch.conf` ファイルが更新されます。

この節では、LDAP、鍵ファイル、およびダイジェストファイルのディレクトリサービスについて説明します。

LDAP ディレクトリサービス

LDAP ディレクトリサービスでは、ユーザーとグループの情報を LDAP ベースのディレクトリサーバーに格納します。

LDAP サービスがデフォルトサービスの場合、`dbswitch.conf` ファイルが次の例のように更新されます。

```
directory default
ldap://test22.india.sun.com:589/dc%3Dindia%2Cdc%3Dsun%2Cdc%3Dcomdefault:binddn
cn=Directory Managerdefault:encoded bindpw YWRtaW5hZG1pbg==
```

LDAP サービスがデフォルト以外のサービスの場合、`dbswitch.conf` ファイルが次の例のように更新されます。

```
directory ldap
ldap://test22.india.sun.com:589/dc%3Dindia%2Cdc%3Dsun%2Cdc%3Dcomldap:binddn
cn=Directory Managerldap:encoded bindpw YWRtaW5hZG1pbg==
```

鍵ファイルディレクトリサービス

鍵ファイルとは、ハッシュ形式のユーザーパスワード、およびそのユーザーが所属するグループのリストが含まれているテキストファイルです。鍵ファイル形式は、HTTP 基本認証の使用を目的としている場合にだけ使用できます。この認証方法については、169 ページの「ユーザーとグループの指定」を参照してください。

鍵ファイルベースのデータベースを作成すると、`dbswitch.conf` ファイルが次の例のように更新されます。

```
directory keyfile filekeyfile:syntax keyfilekeyfile:keyfile D:\\test22\
\keyfile\\keyfiledb
```

ダイジェストファイルディレクトリサービス

ダイジェストファイルでは、ユーザーとグループの情報を、暗号化されたユーザー名とパスワードに基づいて格納します。

ダイジェストファイル形式は、HTTP ダイジェスト認証の使用をサポートすることを目的としていますが、基本認証もサポートしているため、どちらの認証方法でも使用できます。これらの方法については、169 ページの「ユーザーとグループの指定」を参照してください。

ダイジェストベースのデータベースを作成すると、`dbswitch.conf` ファイルが次の例のように更新されます。

```
directory digest filedigest:syntax digestdigest:digestfile D:\\test22\\digest\
\digestdb
```

注 – 分散管理を設定するには、LDAP ベースのディレクトリサービスをデフォルトのディレクトリサービスにする必要があります。

ディレクトリサービスの設定

ディレクトリサービスは、管理サーバーの「Global Settings」タブで作成および設定します。その後、管理サーバーの「Users and Groups」タブで、ユーザー、グループ、および組織単位を作成および管理します。

この節では、ディレクトリサービスの作成および編集方法について説明します。

▼ ディレクトリサービスを作成するには

1. 管理サーバーにアクセスして、「**Global Settings**」タブをクリックします。
2. 「**Configure Directory Service**」リンクをクリックします。
3. 「**Create New Service of Type**」ドロップダウンリストから、作成するディレクトリサービスの種類を選択し、「**New**」をクリックします。
   そのディレクトリサービスの設定ページが表示されます。
4. 設定情報を入力し、「**Save Changes**」をクリックします。
   特定のフィールドについては、オンラインヘルプを参照してください。

注 – ディレクトリサービスが設定されていない場合、最初に作成される新しいディレクトリサービスには、種類に関係なく `default` という値が設定されます。

▼ ディレクトリサービスを編集するには

1. 管理サーバーにアクセスして、「**Global Settings**」タブをクリックします。
2. 「**Configure Directory Service**」リンクをクリックします。
3. 編集するディレクトリサービスのリンクをクリックします。
4. 変更を行い、「**Save Changes**」をクリックします。
   特定のフィールドについては、オンラインヘルプを参照してください。

識別名 (DN) について

管理サーバーの「Users and Groups」タブを使用して、ユーザー、グループ、および組織単位を作成または変更します。ユーザーとは、企業の社員などのような、LDAP データベース内の個人を意味します。グループとは、同じ属性を共有する複数のユーザーを意味します。組織単位とは、`organizationalUnit` オブジェクトクラスを使用する組織内の区分を意味します。ユーザー、グループ、および組織単位については、この章の最後に詳しく説明します。

企業内のユーザーやグループは、それぞれ、識別名 (Distinguished Name、DN) 属性で表されます。DN 属性は、関連するユーザー、グループ、またはオブジェクトを識別する情報が含まれているテキスト文字列です。ユーザーやグループのディレクトリエントリを変更する場合は、必ず DN を使用します。たとえば、ディレクトリエントリの作成または変更、アクセス制御の設定、メールまたはパブリッシングなどのアプリケーション用のユーザーアカウントの設定を行う場合は、そのたびに DN 情報を指定する必要があります。Proxy Server の「Users and Groups」インタフェースを使用して、DN を作成または変更します。

次の例は、Sun Microsystems の社員の一般的な DN を表しています。

```
uid=doe,e=doe@sun.com,cn=John Doe,o=Sun Microsystems Inc.,c=US
```

この例の略語には、それぞれ次のような意味があります。

- `uid` はユーザー ID
- `e` は電子メールアドレス
- `cn` はユーザーの共通名

- o は組織
- c は国名

DN には、さまざまな名前と値のペアを含めることができ、証明書の項目、および LDAP をサポートするディレクトリ内のエントリの両方を識別するために使用されます。

LDIF の使用

現在ディレクトリがない場合、または、既存のディレクトリに新規のサブツリーを追加する場合、ディレクトリサーバーの LDIF (Lightweight Directory Interchange Format) インポート機能を使用できます。この機能を使用すれば、LDIF を含むファイルを取り扱うことができ、ディレクトリを構築したり、LDIF エントリから新規のサブツリーを構築することが可能です。また、ディレクトリサーバーの LDIF エクスポート機能を使用して、現在のディレクトリを LDIF へエクスポートすることもできます。この機能は、該当するディレクトリを表す LDIF 形式のファイルを作成します。エントリは、`ldapmodify` コマンド行ユーティリティー (使用できる場合) を適切な LDIF 更新文とともに使用して追加または編集できます。

LDIF を使用してデータベースにエントリを追加するには、まず、LDIF ファイル内のエントリを定義し、次に、ディレクトリサーバーから LDIF ファイルをインポートします。

ユーザーの作成

管理サーバーの「Users and Groups」タブを使用して、ユーザーエントリを作成または変更します。ユーザーエントリには、データベース内の個人やオブジェクトに関する情報が含まれています。

注 – ユーザーがリソースに不正にアクセスできないようにして、サーバーのセキュリティーを保護してください。Proxy Server では、ACL ベースの承認および認証モデルを使用します。ACL ベースのセキュリティーについては、第 8 章を参照してください。その他のセキュリティー情報については、第 5 章を参照してください。

この節では、LDAP ベース認証データベース、鍵ファイル認証データベース、およびダイジェストファイル認証データベースでのユーザーの作成方法について説明します。

LDAP ベース認証データベースのユーザーの作成

LDAP ベースのディレクトリサービスにユーザーエントリを追加すると、ユーザーの認証と承認に、基本となる LDAP ベースのディレクトリサーバーのサービスが使用されます。この項では、LDAP ベースの認証データベースを使用する場合に考慮する必要のあるガイドラインを示し、Proxy Server の管理サーバーからユーザーを追加する方法について説明します。

LDAP ベースのユーザーエントリ作成のガイドライン

Proxy Server の管理コンソールを使用して LDAP ベースのディレクトリサービスに新しいユーザーエントリを作成するときには、次のガイドラインを考慮してください。

- 名 (ファーストネーム) および姓を入力すると、ユーザーのフルネームとユーザー ID が自動的に入力されます。ユーザー ID は、ユーザーのファーストネームの最初の 1 文字の後にユーザーのラストネームを組み合わせて生成されます。たとえば、ユーザーの名前が Billie Holiday の場合、ユーザー ID は、自動的に `bholiday` と設定されます。このユーザー ID は、必要に応じて、独自に作成する ID と置き換えることができます。
- ユーザー ID は一意である必要があります。管理サーバーは、検索ベース (ベース DN) からディレクトリ全体を検索し、同じユーザー ID が使われていないかを調べて、ユーザー ID が一意であることを確認します。ただし、ディレクトリサーバーの `ldapmodify` コマンド行ユーティリティー (使用できる場合) を使用してユーザーを作成する場合、ユーザー ID が一意であるかどうかは確認されないため注意が必要です。ディレクトリに重複したユーザー ID が存在する場合、該当するユーザーは、そのディレクトリでは認証されなくなります。
- ベース DN は識別名を特定します。それはディレクトリの検索がデフォルトで実行され、ディレクトリツリーにすべての Proxy Server 管理サーバーエントリが配置される場所です。 DN は、ディレクトリサーバーのエントリ名を表す文字列です。
- 新しいユーザーエントリを作成する場合、少なくとも次のユーザー情報を指定してください。
  - 姓 (ラストネーム)
  - 正式名
  - ユーザー ID

  組織単位がディレクトリに定義されている場合、管理サーバーの「Create User」ページにある「Add New User」リストを使用して、新規のユーザーを配置する場所を指定できます。デフォルトの場所は、ディレクトリのベース DN (またはルートポイント) になります。

ディレクトリサーバーのユーザーエントリ

ディレクトリサーバーのユーザーエントリについて、次の点に注意してください。

- ユーザーエントリは、`inetOrgPerson`、`organizationalPerson`、および `person` オブジェクトクラスを使用します。
- デフォルトでは、ユーザーの識別名の書式は次のとおりです。

  `cn=`*full name*`,ou=`*organization*` ,...,o=` *base organization*`,c=`*country*

  たとえば、Billie Holiday のユーザーエントリを Marketing という組織単位に作成し、ディレクトリのベース DN が `o=Ace Industry`、`c=US` の場合、DN は次のようになります。

  ```
  cn=Billie Holiday,ou=Marketing,o=Ace Industry,c=US
  ```

  この書式は、ユーザー ID (uid) ベースの識別名に変更することができます。
- ユーザーフォームフィールドの値は、LDAP 属性として格納されます。

  次の表は、Proxy Server インタフェースに新規ユーザーを作成または編集するときに表示されるフィールドおよび対応する LDAP 属性を示しています。

表 4–1　LDAP 属性 - ユーザーエントリの作成

ユーザーフィールド	LDAP 属性
Given Name	`givenName`
Surname	`sn`
Full Name	`cn`
User ID	`uid`
Password	`userPassword`
E-mail Address	`mail`
Title	`title`
Phone Number	`telephoneNumber`

次の表は、ユーザーエントリを編集するときに表示されるフィールドおよび対応する LDAP 属性を示しています。

LDAP ベースのユーザーエントリの作成

ユーザーエントリを作成するには、52 ページの「LDAP ベースのユーザーエントリ作成のガイドライン」に記載されているガイドラインを読み、そのあとで次の手順を実行します。

▼ LDAP ベースの認証データベースのユーザーを作成するには

1. 管理サーバーにアクセスして、「**Users and Groups**」タブをクリックします。
2. 「**Create User**」リンクをクリックします。
3. ドロップダウンリストから **LDAP** ディレクトリサービスを選択し、「**Select**」をクリックします。
4. 表示されるページに情報を入力します。
   特定のフィールドについては、オンラインヘルプを参照してください。
   52 ページの「ディレクトリサーバーのユーザーエントリ」も参照してください。
5. 「**Create**」をクリックして、ユーザーエントリを作成します。または、「**Create and Edit**」をクリックして、ユーザーエントリを作成してから、作成したエントリの編集ページを開きます。

鍵ファイル認証データベースのユーザーの作成

鍵ファイルとは、ハッシュ形式のユーザーパスワード、およびそのユーザーが所属するグループのリストが含まれているテキストファイルです。

▼ 鍵ファイル認証データベースのユーザーを作成するには

1. 管理サーバーにアクセスして、「**Users and Groups**」タブをクリックします。
2. 「**Create User**」リンクをクリックします。
3. ドロップダウンリストから鍵ファイルベースのディレクトリサービスを選択し、「**Select**」をクリックします。
4. 表示されるページに情報を入力し、「**Create User**」をクリックします。
   特定のフィールドについては、オンラインヘルプを参照してください。

ダイジェストファイル認証データベースのユーザーの作成

ダイジェストファイル認証データベースでは、ユーザーとグループの情報を暗号化された形式で格納します。

▼ ダイジェストファイル認証データベースのユーザーを作成するには

1. 管理サーバーにアクセスして、「**Users and Groups**」タブをクリックします。
2. 「**Create User**」リンクをクリックします。
3. ドロップダウンリストからダイジェストファイルベースのディレクトリサービスを選択し、「**Select**」をクリックします。
4. 表示されるページに情報を入力し、「**Create User**」をクリックします。
   特定のフィールドについては、オンラインヘルプを参照してください。

注 – Proxy Server ACL ユーザーインタフェースを使用してダイジェスト認証による ACL を作成するときは、同じレルム文字列を指定する必要があります。詳細は、164 ページの「アクセス制御の設定」を参照してください。

ユーザーの管理

ユーザー属性は、管理サーバーの「Users and Groups」タブにある「Manage Users」ページを使用して編集します。このページを使用して、ユーザーエントリを検索、変更、名前変更、および削除できます。

この節で説明する内容は、次のとおりです。

ユーザー情報の検索

ユーザーエントリを編集する前に、エントリを検索し、表示する必要があります。LDAP ベースのディレクトリサービスの場合、編集するエントリを説明する値を入力できます。

次の任意の情報を入力できます。

- 名前。フルネーム、または名前の一部を入力します。検索文字列と完全に一致するすべてのエントリが返されます。該当するエントリがない場合は、検索文字列を含むすべてのエントリが検索されます。該当するエントリがない場合は、検索文字列と類似したエントリが検索されます。
- ユーザー ID。ユーザー ID の一部だけを入力すると、その文字列を含むすべてのエントリが返されます。
- 電話番号。電話番号の一部だけを入力すると、最後の部分が検索番号に一致する電話番号を含むエントリがすべて返されます。
- 電子メールアドレス。アットマーク (@) 記号を含む検索文字列は、すべて、電子メールアドレスとして認識されます。完全に一致するエントリがない場合は、検索文字列で始まる電子メールアドレスがすべて検索されます。
- 任意の LDAP 検索フィルタ。等号 (=) を含む文字列はすべて、検索フィルタとして認識されます。
- アスタリスク (*) を入力すると、現在ディレクトリにあるエントリがすべて表示されます。検索フィールドに何も入力しないで検索しても、同じ結果が得られます。

カスタム検索クエリーの構築

LDAP サービスでは、「Find All Users Whose」セクションを使用して、カスタム検索フィルタを構築できます。これらのフィールドを使用して、「Find User」検索で返される検索結果を絞り込みます。

左のドロップダウンリストでは、検索の基準となる属性を指定します。次の表は、利用できる検索属性のオプションを示しています。

表 4–2 検索属性オプション

オプション	検索対象
Full name	各エントリのフルネーム
Last name	各エントリのラストネーム (姓)
User ID	各エントリのユーザー ID
Phone number	各エントリの電話番号

表 4–2 検索属性オプション (続き)

オプション	検索対象
E-mail address	各エントリの電子メールアドレス

中央のドロップダウンリストでは、実行する検索タイプを指定します。次の表は、利用できる検索タイプのオプションを示しています。

表 4–3 検索タイプのオプション

オプション	説明
Contains	部分文字列検索を実行します。指定した検索文字列を含む属性値のエントリを返します。たとえば、ユーザー名におそらく「Dylan」が含まれているということがわかっている場合、このオプションを使用して、検索文字列に「Dylan」と入力し、ユーザーエントリを検索します。
Is	正確に一致するものを検索します (等価検索を指定)。正確なユーザーの属性値がわかっているときには、このオプションを使用します。たとえば、ユーザー名の正確なスペルがわかっている場合などです。
Isn't	属性値が検索文字列と完全一致ではないエントリをすべて返します。ユーザー名が「John Smith」ではない、ディレクトリ内のすべてのユーザーを検索する場合、このオプションを使用します。このオプションを使用すると返されるエントリ数が膨大になるため、注意が必要です。
Sounds like	近似検索、つまり表音の似た検索を実行します。属性の値はわかっているが、スペルが正確にはわからない場合に、このオプションを使用します。たとえば、ユーザー名のスペルが、「Sarret」、「Sarette」、または「Sarett」か不確かな場合などです。
Starts with	部分文字列検索を実行します。属性値が指定した検索文字列で始まるエントリをすべて返します。たとえば、ユーザー名が「Miles」で始まるのはわかっているが、名前の残りの部分がわからない場合には、このオプションを使用します。
Ends with	部分文字列検索を実行します。属性値が指定した検索文字列で終わるエントリをすべて返します。たとえば、ユーザー名が「Dimaggio」で終わるのはわかっているが、名前の残りの部分がわからない場合には、このオプションを使用します。

右のテキストフィールドを使用して、検索文字列を入力します。「Look Within」フィールドで指定したディレクトリのユーザーエントリをすべて表示するには、このフィールドにアスタリスク (*) を入力するか、または何も入力せずに検索します。

▼ ユーザー情報を検索するには

1. 管理サーバーにアクセスして、「**Users and Groups**」タブをクリックします。
2. 「**Manage Users**」リンクをクリックします。
3. ドロップダウンリストからディレクトリサービスを選択し、「**Select**」をクリックします。
   鍵ファイルまたはダイジェストファイルディレクトリサービスの場合は、ユーザーのリストが表示されます。LDAP ベースのディレクトリサービスの場合は、検索フィールドが表示されます。
4. ユーザー情報を検索します。
   鍵ファイルまたはダイジェストファイルのディレクトリサービスの場合は、ユーザーのリンクをクリックし、編集ページを表示して変更を行います。特定のフィールドについては、オンラインヘルプを参照してください。

   LDAP ベースのディレクトリサービスの場合は、次の手順を実行します。

   a. 「**Find User**」フィールドに、編集するエントリを説明する値を入力します。
      ほかの方法としては、「Find All Users Whose」セクションのドロップダウンメニューを使用して、検索結果を絞り込む方法もあります。詳細は、56 ページの「カスタム検索クエリーの構築」を参照してください。
   b. 「**Look Within**」フィールドで、エントリの検索を行う組織単位を選択します。
      デフォルトは、ディレクトリのルートポイント、つまり最上位のエントリです。
   c. 「**Format**」フィールドで、画面に表示またはプリンタに出力する際の出力フォーマットを指定します。
   d. この処理の任意の段階で、「**Find**」ボタンをクリックします。
      検索条件に一致するユーザーがすべて表示されます。
   e. 表示するエントリのリンクをクリックします。

ユーザー情報の編集

▼ ユーザーエントリを編集するには

1. 管理サーバーにアクセスして、「**Users and Groups**」タブをクリックします。
2. 「**Manage Users**」リンクをクリックします。

3 56 ページの「ユーザー情報の検索」の説明に従って、ユーザーエントリを表示します。

4 必要に応じて設定を変更します。
特定のフィールドについては、オンラインヘルプを参照してください。

注 – ユーザーの編集ページに表示されていない属性値を変更するには、ディレクトリサーバーの `ldapmodify` コマンド行ユーティリティーが使用できる場合は、これを使用します。

ユーザー ID の変更については、59 ページの「ユーザー名の変更」を参照してください。

ユーザーのパスワードの管理

次の手順では、ユーザーパスワードを変更または作成する方法について説明します。

▼ ユーザーパスワードを変更または作成するには

1 管理サーバーにアクセスして、「**Users and Groups**」タブをクリックします。

2 「**Manage Users**」リンクをクリックします。

3 56 ページの「ユーザー情報の検索」の説明に従って、ユーザーエントリを表示します。

4 必要に応じて設定を変更します。
特定のフィールドについては、オンラインヘルプを参照してください。

LDAP データベースの場合、ユーザーのパスワードは、「Manage Users」ページからアクセスするユーザーパスワード情報の編集に使用するページの「Disable Password」ボタンをクリックして無効にすることができます。こうすることで、そのユーザーのディレクトリエントリを削除しなくても、そのユーザーがサーバーへログインできなくすることができます。このユーザーに再度アクセスを許可するには、新しいパスワードを提供します。

ユーザー名の変更

LDAP データベースでは、名前の変更機能は、ユーザー ID だけを変更します。ほかのフィールドは変更されません。名前の変更機能を使用して、エントリをある組織単位から別の組織単位に移動することはできません。

▼ ユーザーエントリの名前を変更するには

1. 管理サーバーにアクセスして、「**Users and Groups**」タブをクリックします。
2. 「**Manage Users**」リンクをクリックします。
3. 56 ページの「ユーザー情報の検索」の説明に従って、ユーザーエントリを表示します。
4. ユーザーの編集ページの「**Rename User**」ボタンをクリックします。
5. 表示されるページにユーザー **ID** を入力して、「**Save Changes**」をクリックします。

注 – エントリ名を変更するときに `keepOldValueWhenRenaming` パラメータを `false` (デフォルト) に設定すれば、古い値を今後保持しないよう管理サーバーに指示できます。このパラメータは、次のファイルにあります。

server-root`/proxy-admserv/config/dsgw-orgperson.conf`

ユーザーの削除

▼ ユーザーエントリを削除するには

1. 管理サーバーにアクセスして、「**Users and Groups**」タブをクリックします。
2. 「**Manage Users**」リンクをクリックします。
3. 56 ページの「ユーザー情報の検索」の説明に従って、ユーザーエントリを表示します。
4. 適切なボタンをクリックします。
   - LDAP サーバーの場合は、「Delete User」をクリックします。
   - 鍵ファイルデータベースとダイジェストファイルデータベースの場合は「Remove User」をクリックします。

グループの作成

グループとは、LDAP データベースにおいてオブジェクトのセットを表現するオブジェクトです。Sun Java System サーバーグループは、共通する属性を共有する複数のユーザーで構成されています。たとえば、会社のマーケティング部門で働く多数の従業員がオブジェクトのセットになります。この従業員たちは、「Marketing」というグループに属します。

LDAP サービスでは、グループのメンバーシップを定義するには、静的な方法 (スタティック) と動的な方法 (ダイナミック) の 2 つがあります。スタティックグループは、メンバーオブジェクトを明示的に列挙します。スタティックグループは 共通名 (Common Name、CN) であり、`uniqueMembers`、`memberURLs`、`memberCertDescriptions` のいずれかが含まれます。スタティックグループでは、メンバーは、`cn=groupname` 属性以外の共通属性は共有しません。

ダイナミックグループでは、LDAP URL を使用してグループメンバーにだけ一致する規則セットを定義できます。ダイナミックグループでは、メンバーは共通属性、または `memberURL` フィルタに定義される属性セットを共有します。たとえば、販売のすべての従業員を含むグループが必要で、全員がすでに `ou=Sales,o=Airius.com` の下の LDAP データベースに含まれている場合は、次のメンバー URL を使用してダイナミックグループを定義します。

```
ldap:///ou=Sales,o=sun??sub?(uid=*)
```

このグループは、`ou=Sales,o=sun` ポイントの下のツリーで、`uid` 属性を持つすべてのオブジェクトを持つことになります。

スタティックおよびダイナミックグループで、`memberCertDescription` を使用している場合は、メンバーは証明書から共通属性を共有できます。この共通属性の共有は、ACL が SSL メソッドを使用している場合にだけ当てはまります。

新規グループを作成したら、このグループにユーザー (メンバー) を追加することができます。

ここでは、次の内容について説明します。

- 61 ページの「スタティックグループについて」
- 62 ページの「ダイナミックグループについて」

スタティックグループについて

LDAP サービスでは、管理サーバーを使用し、任意のユーザー数の DN に同じグループ属性を指定して、スタティックグループを作成できます。スタティックグループは、ユーザーをグループに追加したり、グループから削除しないかぎり、変更されることはありません。

スタティックグループ作成のガイドライン

新規のスタティックグループを作成するために管理サーバーのインタフェースを使用するときには、次のガイドラインを考慮してください。

- スタティックグループには、その他のスタティックグループまたはダイナミックグループを含めることができます。
- 組織単位がディレクトリに定義されている場合、管理サーバーインタフェースの「Create Group」ページにある「Add New Group」リストを使用して、新規のグループを配置する場所を指定できます。デフォルトの場所は、ディレクトリのルートポイント、つまり最上位のエントリです。
- グループの編集については、68 ページの「グループエントリの編集」を参照してください。

▼ スタティックグループを作成するには

1. 管理サーバーにアクセスして、「**Users and Groups**」タブをクリックします。
2. 「**Create Group**」リンクをクリックします。
3. 「**Type of Group**」ドロップダウンリストから「**New Group**」を選択し、「**Go**」をクリックします。
4. 「**Create Group**」ページに情報を入力します。
   特定のフィールドについては、オンラインヘルプを参照してください。
5. 「**Create**」をクリックして、グループを作成します。または、「**Create and Edit**」をクリックして、グループを作成してから、作成したグループの編集ページを開きます。

ダイナミックグループについて

LDAP サービスの場合、Proxy Server では、任意の属性に基づいてユーザーを自動的にグループ化する場合、または一致する DN を含む特定のグループに ACL を適用する場合に、ダイナミックグループを作成できます。たとえば、`department=marketing` という属性を持つ任意の DN を自動的に含めるグループを作成することができます。`department=marketing` に検索フィルタを適用すると、検索結果は、`department=marketing` 属性を含むすべての DN からなるグループを返します。次に、このフィルタに基づいて検索結果からダイナミックグループを定義できます。さらに、結果として生成されるダイナミックグループの ACL を定義できます。

ダイナミックグループの実装方法

Proxy Server は、ダイナミックグループを `objectclass=groupOfURLs` として LDAP サーバースキーマ内に実装します。`groupOfURLs` クラスは、0 個以上の `memberURL` 属性を持つことができます。各属性はディレクトリ内のオブジェクトセットを記述する LDAP URL で構成されます。グループのメンバーは、これらのセットの組み合わせです。たとえば、次のグループには 1 つのメンバー URL だけが含まれます。

```
ldap:///o=mcom.com??sub?(department=marketing)
```

この例は、部署名が `marketing` である `o=mcom.com` の下のすべてのオブジェクトで構成されるセットを示しています。LDAP URL には、検索ベース DN、スコープ、およびフィルタを含むことができますが、ホスト名とポートを含むことはできません。このため、同じ LDAP サーバー上のオブジェクトだけを参照できます。すべてのスコープがサポートされます。LDAP URL については、64 ページの「ダイナミックグループ作成のガイドライン」を参照してください。

グループに DN を個別に追加しなくても、すべての DN が自動的に組み込まれます。Proxy Server は ACL 検証でグループ検索が必要になるたびに LDAP サーバー検索を行うため、グループは動的に変化します。ACL ファイルで使用されるユーザー名とグループ名は、LDAP データベース内のオブジェクトの `cn` 属性に対応します。

注 – Proxy Server は `cn` 属性を ACL のグループ名として使用します。

ACL から LDAP データベースへのマッピングは、`dbswitch.conf` ファイル (ACL データベース名と実際の LDAP データベース URL を関連付ける) と ACL ファイル (どの ACL でどのデータベースが使用されるかを定義する) の両方に定義されます。たとえば、`staff` というグループのメンバーシップに基本アクセス権を設定する場合、ACL コードは `groupOf`*anything* というオブジェクトクラスを持ち、CN が `staff` に設定されているオブジェクトを検索します。オブジェクトは、メンバー DN を明示的に列挙するか (スタティックグループの `groupOfUniqueNames` と同様)、または LDAP URL を指定することによって (たとえば `groupOfURLs`)、グループのメンバーを定義します。

注 – グループはスタティックおよびダイナミックの両方にすることができます。グループオブジェクトには、`objectclass=groupOfUniqueMembers` および `objectclass=groupOfURL s` の両方を設定することができます。そのため、`uniqueMember` および `memberURL` の両方の属性が有効になります。グループのメンバーシップには、スタティックなメンバーとダイナミックなメンバーが混在します。

ダイナミックグループがサーバーパフォーマンスに与える影響

ダイナミックグループを使用すると、サーバーパフォーマンスに影響を与えます。グループメンバーシップをテストするときに、DN がスタティックグループのメン

バーではない場合、Proxy Server はデータベースのベース DN に含まれるすべてのダイナミックグループをチェックします。Proxy Server は、ベース DN とスコープをユーザーの DN と比較して各 memberURL が一致するかどうかを識別します。次にユーザー DN をベース DN とし、memberURL のフィルタを使用してベース検索を実行します。この処理では、膨大な数の検索が行われることがあります。

ダイナミックグループ作成のガイドライン

新規のダイナミックグループを作成するために管理サーバーのインタフェースを使用するときには、次のガイドラインを考慮してください。

- ダイナミックグループにほかのグループを含めることはできません。
- LDAP URL は、次の書式を使用します。ホストとポートの情報は無視されるので、これらのパラメータは指定しません。

  ldap:/// *base-dn*?*attributes*? *scope*?(*filter*)

attributes、scope、および (filter) パラメータは、URL 内の位置で識別されます。どの属性も指定しない場合でも、疑問符 (?) を含めてそのフィールドを区切る必要があります。

- 組織単位がディレクトリに定義されている場合、管理サーバーインタフェースの「Create Group」ページにある「Add New Group」リストを使用して、新規のグループを配置する場所を指定できます。デフォルトの場所は、ディレクトリのルートポイント、つまり最上位のエントリです。

グループの編集については、68 ページの「グループエントリの編集」を参照してください。

次の表は、LDAP URL の必須パラメータを示しています。

表 4–4　LDAP URL の必須パラメータ

パラメータ名	説明
base_dn	検索ベースの DN、またはポイント。すべての検索は、LDAP ディレクトリ内のこの場所から実行されます。多くの場合、このパラメータは、o=mcom.com のようにディレクトリのサフィックスまたはルートに設定されます。
attributes	検索によって返される属性のリスト。複数の属性を指定するときは、属性をコンマで区切ります (たとえば、cn,mail,telephoneNumber)。属性を指定しない場合、すべての属性が返されます。このパラメータは、ダイナミックグループメンバーシップのチェックでは無視されます。

表 4–4　LDAP URL の必須パラメータ　(続き)

パラメータ名	説明
scope	これはパラメータは必須です。 検索のスコープ。次のいずれかの値を指定します。 ■ base は URL に指定された識別名 (base_dn) に関する情報だけを取得します。 ■ one は URL に指定された識別名 (base_dn) の 1 レベル下のエントリに関する情報を取得します。このスコープにはベースエントリは含まれません。 ■ sub は URL に指定された識別名 (base_dn) の下のすべてのレベルのエントリに関する情報を取得します。このスコープにはベースエントリが含まれます。
(filter)	このパラメータは必須です。 検索範囲内のエントリに適用される検索フィルタです。管理サーバーのインタフェースを使用する場合は、この属性を指定する必要があります。括弧で囲む必要があります。

ダイナミックグループの作成

▼ ダイナミックグループを作成するには

1. 管理サーバーにアクセスして、「**Users and Groups**」タブをクリックします。
2. 「**Create Group**」リンクをクリックします。
3. 「**Type of Group**」ドロップダウンリストから「**Dynamic Group**」を選択し、「**Go**」をクリックします。
4. 「**Create Group**」ページに情報を入力します。
   特定のフィールドについては、オンラインヘルプを参照してください。
5. 「**Create**」をクリックして、グループを作成します。または、「**Create and Edit**」をクリックして、グループを作成してから、作成したグループの編集ページを開きます。

グループの管理

LDAP サービスでは、管理サーバーを使用して、管理サーバーの「Users and Groups」タブの「Manage Groups」ページからグループを編集したり、グループのメンバーシップを管理できます。

この節では、次の作業について説明します。

グループエントリの検索

グループエントリを編集する前に、次の手順に従ってエントリを検索し、表示する必要があります。

▼ グループエントリを検索するには

1. 管理サーバーにアクセスして、「**Users and Groups**」タブをクリックします。
2. 「**Manage Groups**」リンクをクリックします。
3. 検索するグループ名を「**Find Group**」フィールドに入力します。

   次の任意の値を入力できます。

   - アスタリスク (*) を入力すると、現在ディレクトリにあるグループがすべて表示されます。検索フィールドに何も入力しないで検索しても、同じ結果が得られます。
   - 任意の LDAP 検索フィルタ。等号 (=) を含む文字列はすべて、検索フィルタとして認識されます。

     ほかの方法としては、「Find All Groups Whose」セクションを使用して、カスタム検索フィルタを構築し、検索結果を絞り込む方法もあります。詳細は、67 ページの「Find All Groups Whose」を参照してください。

- 名前。フルネーム、または名前の一部を入力します。検索文字列と完全に一致するすべてのエントリが返されます。該当するエントリがない場合は、検索文字列を含むすべてのエントリが検索されます。該当するエントリがない場合は、検索文字列と類似したエントリが検索されます。

4 「**Look Within**」フィールドで、エントリの検索を行う組織単位を選択します。
デフォルトは、ディレクトリのルートポイント、つまり最上位のエントリです。

5 「**Format**」フィールドで、画面に表示またはプリンタに出力する際の出力フォーマットを指定します。

6 検索条件に一致するグループをすべて表示するには、この処理の任意の段階で、「**Find**」ボタンをクリックします。

7 表示するエントリのリンクをクリックします。

Find All Groups Whose

LDAP サービスでは、「Find All Groups Whose」セクションを使用して、カスタム検索フィルタを構築できます。このセクション内のフィールドを使用して、「Find Group」で返される検索結果を絞り込みます。

左のドロップダウンリストでは、検索の基準となる属性を指定します。有効なオプションは次のとおりです。

- 「**Name**」: 各エントリのフルネームで一致しているものを検索します。
- 「**Description**」**:** 各グループエントリの記述で一致しているものを検索します。

中央のドロップダウンリストでは、実行する検索タイプを指定します。有効なオプションは次のとおりです。

- 「**Contains**」**:**部分文字列検索を実行します。指定した検索文字列を含む属性値のエントリを返します。たとえば、グループ名におそらく「Administrator」が含まれているということがわかっている場合、このオプションを使用して、検索文字列に「Administrator」と入力し、グループエントリを検索します。
- 「**Is**」**:**正確に一致するものを検索します。正確なグループの属性値がわかっているときには、このオプションを使用します。たとえば、グループ名の正確なスペルがわかっている場合などです。
- 「**Isn't**」**:**属性値が検索文字列と完全一致ではないエントリをすべて返します。グループ名に「administrator」が含まれない、ディレクトリ内のすべてのグループを検索する場合、このオプションを使用します。ただし、このオプションを使用すると返されるエントリ数が膨大になるため、注意が必要です。

- 「**Sounds like**」:近似検索、つまり表音の似た検索を実行します。属性の値はわかっているが、スペルが正確にはわからない場合に、このオプションを使用します。たとえば、グループ名のスペルが、「Sarret's list」、「Sarette's list」、または「Sarett's list」か不確かな場合などです。
- 「**Starts with**」:部分文字列検索を実行します。属性値が指定した検索文字列で始まるエントリをすべて返します。たとえば、グループ名が「Product」で始まるのはわかっているが、名前の残りの部分がわからない場合に、このオプションを使用します。
- 「**Ends with**」:部分文字列検索を実行します。属性値が指定した検索文字列で終わるエントリをすべて返します。たとえば、グループ名が「development」で終わるのはわかっているが、名前の残りの部分がわからない場合に、このオプションを使用します。

右のテキストフィールドに、検索文字列を入力します。「Look Within」ディレクトリのグループエントリをすべて表示するには、このフィールドにアスタリスク (*) を入力するか、または何も入力せずに検索します。

グループエントリの編集

▼ グループエントリを編集するには

この手順は、LDAP サービスにのみ該当します。

1. 管理サーバーにアクセスして、「**Users and Groups**」タブをクリックします。
2. 「**Manage Groups**」リンクをクリックします。
3. 66 ページの「グループエントリの検索」の説明に従って、編集するグループを特定します。
4. 必要に応じて設定を変更します。
   特定のフィールドとボタンについては、オンラインヘルプを参照してください。

注 – グループの編集ページで表示されていない属性値を変更する場合があります。この場合、ディレクトリサーバーの `ldapmodify` コマンド行ユーティリティーが使用できる場合は、これを使用します。

グループメンバーの追加

▼ グループにメンバーを追加するには

この手順は、LDAP サービスにのみ該当します。

1. 管理サーバーにアクセスして、「**Users and Groups**」タブをクリックします。
2. 「**Manage Groups**」リンクをクリックします。
3. 66 ページの「グループエントリの検索」の説明に従って、管理するグループを特定して表示し、「**Group Members**」の横の「**Edit**」ボタンをクリックします。

   表示されるページに、既存のグループメンバーが示されます。検索フィールドも表示されます。

   - メンバーのリストにユーザーエントリを追加する場合、「Users」が「Find」ドロップダウンリストで選択されている必要があります。
   - グループにグループエントリを追加する場合、「Groups」が選択されている必要があります。
4. 「**Matching**」テキストフィールドに、検索文字列を入力します。次のオプションのうち、いずれかを入力します。

   - 名前。フルネーム、または名前の一部を入力します。検索文字列と名前が一致するすべてのエントリが返されます。該当するエントリがない場合は、検索文字列を含むすべてのエントリが検索されます。該当するエントリがない場合は、検索文字列と類似したエントリが検索されます。
   - ユーザー ID。ユーザー ID の一部だけを入力すると、その文字列を含むすべてのエントリが返されます。
   - 電話番号。電話番号の一部だけを入力すると、最後の部分が検索番号に一致する電話番号を含むエントリがすべて返されます。
   - 電子メールアドレス。アットマーク (@) 記号を含む検索文字列は、すべて、電子メールアドレスとして認識されます。完全に一致するエントリがない場合は、検索文字列で始まる電子メールアドレスがすべて検索されます。
   - 現在ディレクトリ内にあるエントリまたはグループをすべて表示するには、このフィールドにアスタリスク (*) を入力するか、または何も入力せずに検索します。
   - 任意の LDAP 検索フィルタ。等号 (=) を含む文字列はすべて、検索フィルタとして認識されます。
5. 「**Add**」をクリックして、**LDAP** データベースで一致するすべてのエントリを検索し、グループにこのエントリを追加します。

6 (オプション) グループに追加する必要のないエントリが返された場合は、「**Remove From List**」列の対応するチェックボックスをクリックします。また、グループから削除するエントリに一致する検索フィルタを作成して、「**Remove**」をクリックすることもできます。詳細は、**70** ページの「グループメンバーリストからのエントリの削除」を参照してください。

7 グループメンバーのリストが完成したら、「**Save Changes**」をクリックします。エントリがグループメンバーリストに追加されます。

グループメンバーリストへのグループの追加

LDAP サービスでは、グループのメンバーリストに、個別のメンバーの代わりにグループを追加できます。追加されたグループに属するユーザーは、追加先のグループのメンバーになります。たとえば、Neil Armstrong が「Engineering Managers」グループのメンバーであり、この「Engineering Managers」グループを「Engineering Personnel」グループのメンバーにする場合、Neil Armstrong は、「Engineering Personnel」グループのメンバーにもなります。

グループを別のグループのメンバーリストへ追加するには、ユーザーエントリと同様に、グループを追加します。詳細は、69 ページの「グループメンバーの追加」を参照してください。

グループメンバーリストからのエントリの削除

この手順は、LDAP サービスにのみ該当します。

▼ グループメンバーリストからエントリを削除するには

1 管理サーバーにアクセスして、「**Users and Groups**」タブをクリックします。

2 「**Manage Groups**」リンクをクリックします。

3 管理するグループを特定します。

詳細は、66 ページの「グループエントリの検索」を参照してください。「Group Members」の横の「Edit」ボタンをクリックします。

4 削除するメンバーを指定します。

- 数人のメンバーのみを削除する場合は、「Remove From List」列の対応するチェックボックスをクリックします。
- 共通の条件に基づいてメンバーを削除するには、グループから削除するエントリに一致する検索フィルタを作成し、「Remove」をクリックします。

検索フィルタの作成については、69 ページの「グループメンバーの追加」を参照してください。

5 「**Save Changes**」をクリックします。
エントリが、グループメンバーリストから削除されます。

所有者の管理

LDAP サービスの場合、グループの所有者リストは、グループメンバーリストと同様の方法で管理します。

次の表は、このマニュアルでより詳細に説明しているトピックを示しています。

表 4–5 所有者の管理

目的	参照
グループに所有者を追加する	69 ページの「グループメンバーの追加」
所有者リストにグループを追加する	70 ページの「グループメンバーリストへのグループの追加」
所有者リストからエントリを削除する	70 ページの「グループメンバーリストからのエントリの削除」

See Alsos の管理

See Alsos は、現在のグループに関連のある、ほかのディレクトリのエントリへの参照です。See Alsos を使用して、現在のグループと関連のあるユーザーやほかのグループのエントリを簡単に見つけることができます。See Alsos は、グループメンバーリストと同様の方法で管理します。

次の表は、このマニュアルでより詳細に説明しているトピックを示しています。

表 4–6 See Alsos の管理

目的	参照
See Alsos にユーザーを追加する	69 ページの「グループメンバーの追加」
See Alsos にグループを追加する	70 ページの「グループメンバーリストへのグループの追加」
See Alsos からエントリを削除する	70 ページの「グループメンバーリストからのエントリの削除」

グループ名の変更

この手順は、LDAP サービスにのみ該当します。グループエントリの名前を変更する場合は、グループの名前だけが変更されます。グループ名の変更機能を使用して、エントリをある組織単位から別の組織単位に移動することはできません。たとえば、ある企業には次のような組織があるとします。

- Marketing および Product Management という組織単位
- Marketing という組織単位の下に Online Sales というグループ

この例では、Online Sales というグループ名を Internet Investments に変更することはできますが、Marketing という組織単位の下の Online Sales を、Product Management という組織単位の下の Online Sales にするようエントリの名前を変えることはできません。

▼ グループ名を変更するには

1. 管理サーバーにアクセスして、「**Users and Groups**」タブをクリックします。
2. 「**Manage Groups**」リンクをクリックして、66 ページの「グループエントリの検索」の説明に従って、変更するグループを特定します。
3. 「**Rename Group**」ボタンをクリックします。
4. 表示されるページに新しいグループ名を指定して、「**Save Changes**」をクリックします。

グループの削除

この手順は、LDAP サービスにのみ該当します。

▼ グループを削除するには

1. 管理サーバーにアクセスして、「**Users and Groups**」タブをクリックします。
2. 「**Manage Groups**」リンクをクリックします。
3. 66 ページの「グループエントリの検索」の説明に従って、管理するグループを特定し、「**Delete Group**」をクリックします。

注 – グループに属する個々のメンバーは削除されません。グループエントリだけが削除されます。

組織単位の作成

LDAP サービスでは、、組織単位には、複数のグループを含めることができ、それらは通常、部門、課などのエンティティーを表します。DN は、複数の組織単位に存在させることができます。

- 新規の組織単位は、`organizationalUnit` オブジェクトクラスを使用して作成します。
- 新規の組織単位の識別名の書式は次のとおりです。

  `ou=`*new organization* `,ou=`*parent organization*`,...,o=` *base organization*`,c=`*country*

▼ 組織単位を作成するには

1. 管理サーバーにアクセスして、「**Users and Groups**」タブをクリックします。
2. 「**Create Organizational Unit**」リンクをクリックします。
3. 情報を入力し、「**Create**」をクリックします。

   特定のフィールドについては、オンラインヘルプを参照してください。

   たとえば、Accounting という新規の組織を、組織単位 West Coast 内に作成する場合、ベース DN が `o=Ace Industry`、 `c=US` とすると、新規の組織単位の DN は、次のようになります。

   ```
   ou=Accounting,ou=West Coast,o=Ace Industry,c=US
   ```

組織単位の管理

LDAP サービスの場合、組織単位は、管理サーバーの「Users and Groups」タブにある「Manage Organizational Units」ページを使用して編集および管理します。

ここでは、次の内容について説明します。

- 74 ページの「組織単位の検索」
- 75 ページの「組織単位の属性の編集」
- 76 ページの「組織単位名の変更」
- 76 ページの「組織単位の削除」

組織単位の検索

この手順は、LDAP サービスにのみ該当します。

▼ 組織単位を検索するには

1 管理サーバーにアクセスして、「**Users and Groups**」タブをクリックします。

2 「**Manage Organizational Units**」リンクをクリックします。

3 検索する組織単位の名前を「**Find Organizational Unit**」フィールドに入力します。
次の任意の値を入力できます。

- 名前。フルネーム、または名前の一部を入力します。検索文字列と完全に一致するすべてのエントリが返されます。該当するエントリがない場合は、検索文字列を含むすべてのエントリが検索されます。該当するエントリがない場合は、検索文字列と類似したエントリが検索されます。
- アスタリスク (*) を入力すると、現在ディレクトリにあるグループがすべて表示されます。検索フィールドに何も入力しないで検索しても、同じ結果が得られます。
- 任意の LDAP 検索フィルタ。等号 (=) を含む文字列はすべて、検索フィルタとして認識されます。

  ほかの方法としては、「Find All Units Whose」セクションのドロップダウンメニューを使用して、検索結果を絞り込む方法もあります。詳細は、74 ページの「次の条件に一致するすべての単位を検索」を参照してください。

4 「**Look Within**」フィールドで、エントリの検索を行う組織単位を選択します。
デフォルトは、ディレクトリのルートポイント (最上位のエントリ) です。

5 「**Format**」フィールドで、画面に表示またはプリンタに出力する際の出力フォーマットを指定します。

6 この処理の任意の段階で、「**Find**」ボタンをクリックします。
検索条件に一致する組織単位がすべて表示されます。

7 表示するエントリのリンクをクリックします。

次の条件に一致するすべての単位を検索

LDAP サービスでは、「Find All Units Whose」セクションを使用して、カスタム検索フィルタを構築できます。このセクション内のフィールドを使用して、「Find Organizational Unit」で返される検索結果を絞り込みます。

左のドロップダウンリストでは、検索の基準となる属性を指定します。有効なオプションは次のとおりです。

- 「**Unit name**」:各エントリのフルネームで一致しているものを検索します。
- 「**Description**」: 各エントリの記述で一致している組織単位を検索します。

中央のドロップダウンリストでは、実行する検索タイプを指定します。有効なオプションは次のとおりです。

- 「**Contains**」:部分文字列検索を実行します。指定した検索文字列を含む属性値のエントリを返します。たとえば、組織単位名におそらく「Administrator」が含まれているということがわかっている場合、このオプションを使用して、検索文字列に「Administrator」と入力し、組織単位エントリを検索します。
- 「**Is**」:正確に一致するものを検索します。正確な組織単位の属性値がわかっているときには、このオプションを使用します。たとえば、組織単位名の正確なスペルがわかっている場合などです。
- 「**Isn't**」:属性値が検索文字列と完全一致ではないエントリをすべて返します。つまり、組織単位名に「administrator」が含まれない、ディレクトリ内のすべての組織単位を検索する場合、このオプションを使用します。ただし、このオプションを使用すると返されるエントリ数が膨大になるため、注意が必要です。
- 「**Sounds like**」:近似検索、つまり表音の似た検索を実行します。属性の値はわかっているが、スペルが正確にはわからない場合に、このオプションを使用します。たとえば、組織単位名のスペルが、「Sarret's list」、「Sarette's list」、または「Sarett's list」か不確かな場合などです。
- 「**Starts with**」:部分文字列検索を実行します。属性値が指定した検索文字列で始まるエントリをすべて返します。たとえば、組織単位名が「Product」で始まるのはわかっているが、名前の残りの部分がわからない場合に、このオプションを使用します。
- 「**Ends with**」:部分文字列検索を実行します。属性値が指定した検索文字列で終わるエントリをすべて返します。たとえば、組織単位名が「development」で終わるのはわかっているが、名前の残りの部分がわからない場合に、このオプションを使用します。

右のテキストフィールドに、検索文字列を入力します。「Look Within」ディレクトリの組織単位エントリをすべて表示するには、このフィールドにアスタリスク (*) を入力するか、または何も入力せずに検索します。

組織単位の属性の編集

この手順は、LDAP サービスにのみ該当します。

▼ 組織単位エントリを編集するには

1 管理サーバーにアクセスして、「**Users and Groups**」タブをクリックします。

2 「**Manage Organizational Units**」リンクをクリックします。

3 74 ページの「組織単位の検索」の説明に従って、編集する組織単位を特定します。

4 必要に応じて設定を変更します。
特定のフィールドについては、オンラインヘルプを参照してください。

注 – 組織単位の編集ページに表示されていない属性値を変更するには、ディレクトリサーバーの `ldapmodify` コマンド行ユーティリティーが使用できる場合は、これを使用します。

組織単位名の変更

この手順は、LDAP サービスにのみ該当します。組織単位エントリの名前を変更する場合は、組織単位の名前だけが変更されます。名前の変更機能を使用して、エントリをある組織単位から別の組織単位に移動することはできません。

▼ 組織単位名を変更するには

1 管理サーバーにアクセスして、「**Users and Groups**」タブをクリックします。

2 「**Manage Organizational Units**」リンクをクリックします。

3 74 ページの「組織単位の検索」の説明に従って、編集する組織単位を特定します。

4 「名前の変更」ボタンをクリックします。

5 表示されるページに新しい組織単位名を入力して、「**Save Changes**」をクリックします。

組織単位の削除

この手順は、LDAP サービスにのみ該当します。

▼ 組織単位を削除するには

1 管理サーバーにアクセスして、「**Users and Groups**」タブをクリックします。

2 「**Manage Organizational Units**」リンクをクリックします。

3 **74** ページの「組織単位の検索」の説明に従って、削除する組織単位を特定します。

4 「**Delete**」ボタンをクリックし、表示される確認ボックスで、「**OK**」をクリックします。

◆ ◆ ◆ 第 5 章

証明書と鍵の使用

この章では、証明書と鍵の認証を使用した、Sun Java System Web Proxy Server のセキュリティー保護について説明します。Proxy Server には、すべての Sun Java System サーバーのセキュリティーアーキテクチャーが統合されています。Proxy Server は、相互運用性と整合性を最大限確保するため、業界標準および標準プロトコルに基づいて構築されています。

この章では、暗号化と復号化、公開鍵と非公開鍵、デジタル証明書、暗号化プロトコルなど、公開鍵暗号方式に関する基本概念を理解していることを前提としています。

この章の内容は次のとおりです。

管理サーバーアクセスのセキュリティー確保

管理サーバーは Web ベースのユーザーインタフェースで、サーバーを管理、追加、削除、および移行するために使用し、セキュリティーを確保する必要があります。

デフォルトの「Administration Server」ページは HTTP モードで起動します。使用可能な Proxy Server インスタンスは、「Manage Servers」の見出しの下にリストとして表示されます。Proxy Server インスタンスを管理するには、リストの名前をクリックします。Proxy Server インスタンスの名前をクリックすると、そのインスタンスの「Server Manager」ページが表示されます。

「Server Manager」ページから、「Server Manager」ページの左上にある「Manage Servers」リンクをクリックして「Administration Server」ページに戻ることができます。

証明書ベースの認証、信頼データベースの作成、SSL の設定、証明書の要求およびインストール、セキュリティーに関する詳細設定などの、セキュリティー機能は、管理サーバーと個別の Proxy Server インスタンスの両方に適用されます。管理サーバーのセキュリティー関連の設定の場合、「Administration Server」ページに表示される「Preferences」タブと「Security」を使用します。Proxy Server インスタンスに関するセキュリティー設定の場合は、その Proxy インスタンスの「Server Manager」ページに表示される「Preferences」タブと「Security」を使用します。

セキュリティーモードで管理サーバーを起動するには、デフォルトの HTTP の代わりに HTTPS を使用してアクセスする必要があります。

セキュリティー機能については、次の節を参照してください。

証明書に基づく認証

認証とは、識別情報を確認するためのプロセスのことです。ネットワークを介した対話では、認証は、特定の対象を他と区別するための確実な手段です。証明書は、認証をサポートする方法の 1 つです。

証明書は、個人、企業、またはその他のエンティティーの名前を指定するデジタルデータで構成され、その証明書に含まれている公開鍵が、そのエンティティーに属していることを証明します。

クライアントとサーバーの両方が証明書を持つことができます。サーバー認証とは、クライアントによる、サーバーの確実な識別です。組織の識別は、特定のネットワークアドレスにあるサーバーに対して責任を持つとされています。クライアント認証とは、サーバーによる、クライアントの確実な識別、またはクライアントソフトウェアを使用していると見なされる人の識別です。クライアントは、複数の証明書を所有できます。これは、1 人の人が数種類の ID を所有しているのと同じことです。

証明書は、認証局 (Certificate Authority、CA) によって発行され、デジタル署名がなされます。CA は、証明書を販売する企業の場合も、企業で、イントラネットやエクストラネットの証明書の発行を担当する部門の場合もあります。他のユーザーの識別情報の検証手段としてどの CA を信頼するかは、ユーザー自身が決定します。

証明書には、次の情報が含まれます。

- 公開鍵
- 証明書によって識別されるエンティティーの名前
- 有効期限
- 証明書を発行した CA の名称
- 証明書を発行する CA のデジタル署名

注 – 暗号化機能を有効にするには、事前にサーバー証明書をインストールしておく必要があります。

信頼データベースの作成

サーバー証明書を要求する前に、信頼データベースを作成しておく必要があります。Proxy Server では、管理サーバーと各サーバーのインスタンスが、独自の信頼データベースを所有できます。信頼データベースは、ローカルコンピューター上にだけ作成できます。

信頼データベースを作成するときには、鍵ペアファイルに使用されるパスワードを指定します。このパスワードは、暗号化された通信を使用してサーバーを起動させるときにも必要です。パスワードを選択するときに考慮する必要のあるガイドラインについては、118 ページの「強固なパスワードの選択」を参照してください。

信頼データベースでは、鍵ペアファイルと呼ばれる公開鍵と非公開鍵を作成し、保存します。鍵ペアファイルは、SSL 暗号化に使用されます。サーバー証明書を要求し、インストールするときには、鍵ペアファイルを使用します。証明書は、インストールしたあとに信頼データベースに格納されます。

鍵ペアファイルは、次のディレクトリ内に暗号化されて保存されます。

server-root/alias/proxy-**serverid**-key3.db

管理サーバーは、信頼データベースを 1 つだけ所有できます。また、サーバーに含まれる各インスタンスは、それぞれ専用の信頼データベースを所有できます。

▼ 信頼データベースを作成するには

1. 管理サーバーまたはサーバーマネージャーにアクセスし、「**Security**」タブを選択します。
2. 「データベースを作成」リンクをクリックします。
3. 信頼データベースのパスワードを入力します。
4. もう一度パスワードを入力し、「了解」をクリックします。

password.conf の使用

デフォルトでは、Proxy Server を起動する前に、管理者に鍵データベースのパスワードを入力するよう求めるプロンプトが表示されます。Proxy Server を無人で再起動するには、`password.conf` ファイルにパスワードを保存する必要があります。このファイルと鍵データベースが危険にさらされないようにするために、これを行うのはシステムが充分にセキュリティー保護されている場合だけにしてください。

通常、サーバーは起動する前にパスワードを要求するため、`/etc/ rc.local` ファイルまたは `/etc/inittab` ファイルで、UNIX の SSL が有効なサーバーを起動することはできません。ファイル内にプレーンテキストでパスワードを保存しておくと SSL が有効なサーバーを自動的に起動することができますが、この方法は安全ではありません。サーバーの `password.conf` ファイルは、root またはサーバーをインストールしたユーザーが所有し、所有者だけが読み取りと書き込みのアクセス権を持つようにしてください。

UNIX で、`password.conf` ファイル内に SSL が有効なサーバーのパスワードを保存しておくと、セキュリティー上のリスクが大きくなります。ファイルにアクセス可能なユーザーは、SSL が有効なサーバーのパスワードにもアクセスできます。SSL が有効なサーバーのパスワードを `password.conf` ファイルに保存する前に、セキュリティー面のリスクを考慮してください。

Windows で、NTFS ファイルシステムを使用する場合は、`password.conf` ファイルを使用しなくても、アクセス制限によってこのファイルの保存されているディレクトリのセキュリティーを保護してください。ただしこのディレクトリには、管理サーバーのユーザーと Proxy Server のユーザーに対して読み取りおよび書き込み許可を持たせる必要があります。ディレクトリのセキュリティーを保護しておくと、他者が偽の `password.conf` ファイルを作成することを防げます。FAT ファイルシステム上では、ディレクトリやファイルへのアクセスを制限しても、ディレクトリやファイルのセキュリティーを保護することはできません。

SSL が有効なサーバーを自動的に起動

▼ SSL が有効なサーバーを自動的に起動するには

1. **SSL** が有効になっていることを確認します。
2. **Proxy Server** インスタンスの `config` サブディレクトリ内に、新規の `password.conf` ファイルを作成します。
   - Proxy Server に含まれている内部 PKCS #11 ソフトウェア暗号化モジュールを使用している場合には、次の情報を入力します。`internal:` *your-password*
   - 別の PKCS #11 モジュール (ハードウェアの暗号化またはハードウェアアクセラレータ用) を使用している場合は、PKCS #11 モジュールの名前を指定し、その後ろにパスワードを入力します。その例を次に示します。`nFast:`*your-password*

     `password.conf` ファイルを作成した後でも、Proxy Server を起動させるときには、毎回パスワードを入力するよう求めるプロンプトが表示されます。

VeriSign 証明書の要求およびインストール

VeriSign は、Proxy Server の推奨する認証局です。VeriSign のテクノロジによって、証明書リクエストプロセスはシンプルになります。VeriSign は、直接サーバーに対して証明書を返せるという利点があります。

サーバーに証明書信頼データベースを作成後、証明書を要求し、認証局 (CA) にこれを提出できます。会社に独自の内部 CA がある場合には、その部門から発行される証明書を要求します。商用 CA からの証明書購入を予定している場合には、CA を選定し、CA が必要とする情報の特定のフォーマットを入手してください。

管理サーバーは、サーバー証明書を 1 つしか所有できません。各サーバーのインスタンスは、専用のサーバー証明書を所有できます。

▼ VeriSign 証明書を要求するには

1. 管理サーバーまたはサーバーマネージャーにアクセスし、「**Security**」タブを選択します。
2. 「**Request VeriSign Certificate**」リンクをクリックします。
3. 表示されるページに記載されている手順を確認し、「了解」をクリックします。

   「VeriSign Enrollment Wizard」が表示され、手順が順番に指示されます。

▼ VeriSign 証明書をインストールするには

1. 管理サーバーまたはサーバーマネージャーにアクセスし、「**Security**」タブを選択します。
2. 「**Install VeriSign Certificate**」リンクをクリックします。
3. 外部の暗号化モジュールを使用する場合以外は、「**Cryptographic Module**」ドロップダウンリストから「**Internal**」を選択します。
4. 鍵ペアファイルパスワードまたは **PIN** を入力します。
5. ドロップダウンリストから取得する「**Transaction ID**」を選択し、「了解」をクリックします。

ほかのサーバー証明書の要求およびインストール

VeriSign のほかに、他の認証局からの証明書を要求し、インストールすることができます。会社または組織が独自の内部証明書を提供している場合もあります。この節では、ほかの種類のサーバー証明書を要求およびインストールする方法について説明します。

ここでは、次の内容について説明します。

必要な CA 情報

要求処理を始める前に、CA が必要とする情報を確認しておく必要があります。必要な情報のフォーマットは CA によって異なりますが、通常は、次に示される情報を設定するように指定されます。これらの情報のほとんどは、証明書の更新の場合には、通常必要ありません。

- 要求者の名前: 証明書が発行される対象の名前です。
- 電話番号:要求者の電話番号です。
- 共通名:DNS 検索で使用される完全修飾ホスト名である必要があります (たとえば、`www.example.com`)。
- 電子メールアドレス:ユーザーと CA との間の連絡に使用される、仕事の電子メールアドレスです。

- 組織:ユーザーの会社、教育機関、組織などの公式かつ法的な名前です。ほとんどの CA は、この情報をビジネスライセンスのコピーなどの法的文書で証明するように要求します。
- 組織単位:会社内の組織単位の説明です。
- 市区町村名:組織が所在する都市、郡、または国名の説明です。
- 都道府県名:会社が所在する都道府県名です。
- 国:国名の 2 文字の省略形 (ISOフォーマット) です。たとえば、米国の国コードは US になります。

すべての情報は、識別名 (DN) と呼ばれる、一連の属性と属性値のペアとして結合されており、これにより証明書の項目を一意に識別することができます。

商用の CA から証明書を購入する場合は、証明書が発行される前に、上記のほかにどんな情報が必要とされているのか知るために、事前に CA に確認しておく必要があります。ほとんどの CA では、識別情報の証明を要求します。たとえば、会社名や、会社によってサーバー管理者権限を与えられている人の名前を確認します。そして、場合によっては、提供した情報を使用する法的権利をユーザーが持っているかどうかを尋ねられることもあります。

一部の商用 CA では、さらに徹底した識別情報を提供した組織や個人に対して、さらに詳細で正確性の高い証明書を発行します。たとえば、個人が `www.example.com` というサイトのコンピュータの正当な管理者であるということを確認したことに加えて、企業が過去 3 年間にわたって運営されており、現在顧客と係争中の訴訟がないことを CA が確認したことが記述された証明書を購入することもできます。

ほかのサーバー証明書の要求

▼ ほかのサーバー証明書を要求するには

1. 管理サーバーまたはサーバーマネージャーにアクセスし、「**Security**」タブを選択します。
2. 「**Request Certificate**」リンクをクリックします。
3. 新しい証明書か証明書の更新かを選択します。

   多くの証明書は、6 か月や 1 年などの一定期間が経過すると、有効期限が切れます。自動的に更新した証明書を送信してくる CA もあります。
4. 証明書の要求を送信する方法を指定します。
   - 電子メールで要求を送信するには、「CA Email Address」を選択して、その要求に使用する適切な電子メールアドレスを入力します。

- CA の Web サイトから要求を送信するには、「CA URL」を選択して、その要求に使用する適切な URL を入力します。

5 「**Cryptographic Module**」ドロップダウンリストから、証明書を要求するときに使用する鍵ペアファイルの暗号化モジュールを選択します。

6 鍵ペアファイルのパスワードを入力します。

このパスワードは、「Internal」以外の暗号化モジュールを選択していないかぎり、信頼データベースを作成したときに指定したパスワードと同一です。サーバーは、このパスワードを使用して、ユーザーの非公開鍵を取得したり、CA に対するメッセージを暗号化したりします。そして、ユーザーの公開鍵と暗号化されたメッセージの両方を CA に送信します。CA は、公開鍵を使用してメッセージを復号化します。

7 名前や電話番号などの識別情報を入力します。

この情報のフォーマットは、CA によって異なります。これらの情報のほとんどは、証明書の更新の場合には、通常必要ありません。

8 正確に行うため、入力内容を見直して、「了解」をクリックします。

情報が正確であれば、証明書も早く承認されます。要求を証明書サーバーに送るとき、送信する前に、フォーム情報を確認するよう求めるプロンプトが表示されます。

サーバーは、入力した情報を含む証明書リクエストを作成します。要求には、ユーザーの非公開鍵を使用して作成されたデジタル署名が含まれます。CA は、デジタル署名を使用して、サーバーコンピュータから CA に送付されている間、その要求が不正に変更されていないことを確認します。まれに要求が不正に変更されたような場合には、通常 CA から電話で連絡があります。

要求を電子メールで送信する場合には、サーバーがその要求を含んだ電子メールメッセージを CA に送信します。通常、電子メールにより証明書が返されます。証明書サーバーに URL を指定した場合は、サーバーがその URL を使用して証明書サーバーにその要求を送信します。CA によって、電子メールで返信を受けるか、その他の手段になるかは異なります。

CA は、証明書を発行することに同意するかどうかを通知します。ほとんどの場合、CA は、電子メールで証明書を送信します。所属している組織が証明書サーバーを使用している場合には、証明書サーバーのフォームを使用して証明書を検索できます。

注 – 商用 CA に証明書を要求しても、必ず証明書が発行されるとは限りません。多くの CA では、証明書の発行前に、ユーザーの識別情報の証明を要求します。また、承認されるまでには、1 日～数週間かかることがあります。必要な情報をすべて迅速に CA に提供することが重要です。

証明書を受け取ったら、それをインストールします。それまでの間は、SSL を使用せずに Proxy Server を使用できます。

ほかのサーバー証明書のインストール

CA からの証明書は、ユーザーだけがこれを復号化できるように、公開鍵で暗号化されています。信頼データベースの正しいパスワードを入力しないと、証明書を復号化し、インストールすることはできません。

証明書には、次の 3 種類があります。

- クライアントに提示するための、ユーザーのサーバーの証明書
- 証明書チェーンで使用される、CA の独自の証明書
- 信頼された CA の証明書

証明書チェーンは、連続した認証局によって署名された、一連の階層的証明書です。CA 証明書は、認証局を識別し、その認証局によって発行される証明書に署名するために使用されます。認証局証明書は同様に、親 CA の認証局証明書によって署名されます。このプロセスは、ルート認証局までさかのぼって行われます。

注 – CA が CA の証明書を自動的にユーザーに送信しない場合には、証明書を要求してください。多くの CA は、ユーザーの証明書を電子メールで送信する際に CA 証明書も同時に送信してくるため、ユーザーのサーバーでは、両方の証明書が同時にインストールされます。

CA からの証明書は、ユーザーだけがこれを復号化できるように、公開鍵で暗号化されています。Proxy Server は、証明書をインストールする際、その証明書を復号するのに指定した鍵ペアファイルパスワードを使用します。サーバーがアクセス可能な場所にその電子メールを保存するか、またはその電子メールのテキストをコピーし、次の手順に説明する「Install Certificate」フォームにペーストできるようにします。

▼ 他のサーバー証明書をインストールするには

1 管理サーバーまたはサーバーマネージャーにアクセスし、「**Security**」タブを選択します。

2 「**Install Certificate**」リンクをクリックします。

3 「**Certificate**」の横のインストールする証明書の種類を選択します。

- This Server
  - Server Certificate Chain
  - Certification Authority

  特定の設定については、オンラインヘルプを参照してください。

4 ドロップダウンリストから、暗号化モジュールを選択します。

5 鍵ペアファイルパスワードを入力します。

6 手順 **3** で「**Server Certificate Chain**」または「**Certification Authority**」を選択した場合だけ、証明書の名前を入力します。

7 次のいずれかの方法で、証明書の情報を指定します。

- 「Message Is In This File」を選択して、CA 証明書を含むファイルのフルパス名を入力します。
  - 「Message Text (with headers)」を選択して、CA 証明書の内容をコピーしてペーストします。Begin Certificate および End Certificate ヘッダーを含むようにしてください。また、先頭と末尾のハイフンを忘れないでください。

8 「了解」をクリックします。

9 新しい証明書を追加するのか、既存の証明書を更新するのかを示します。

- 新しい証明書をインストールする場合は、「Add Certificate」。
  - 更新された証明書をインストールする場合は、「Replace Certificate」

  証明書は、サーバーの証明書データベースに保管されます。次に例を示します。

  server-root`/alias/ proxy-`*serverid*`-cert8.db`

以前のバージョンからの証明書の移行

Sun ONE Web Proxy Server 3.6 (iPlanet Web Proxy Server とも呼ばれる) から Sun Java System Web Proxy Server 4 へ移行するときは、信頼データベースや証明書データベースを始めとするファイルが自動的に更新されます。

Proxy Server 4 管理サーバーに、以前の 3.x データベースファイルに対する読み取り許可があることを確認します。これらのファイルは、*alias*`-cert.db` と *alias*`-key.db` で、*3.x-server-root*`/alias` ディレクトリにあります。

サーバーでセキュリティーが有効になっている場合だけ、鍵ペアファイルと証明書が移行されます。管理サーバーとサーバーマネージャーの「Security」タブにある「Migrate 3.x Certificates」オプションを使用して、鍵と証明書だけを移行させることもできます。特定の設定については、オンラインヘルプを参照してください。

以前のバージョンでは、証明書と鍵ペアファイルは、複数のサーバーインスタンスによって使用される可能性のあるエイリアスによって参照されていました。管理サーバーは、すべてのエイリアスとそれらの構成要素である証明書を管理していました。Sun Java System Web Proxy Server 4 では、管理サーバーと各サーバーインスタンスに独自の証明書と鍵ペアファイルがあり、エイリアスではなく信頼データベースとして参照されます。

管理サーバー自体については、信頼データベースとその構成要素である証明書の管理は管理サーバーで行い、サーバーインスタンスについては、サーバーマネージャーで行います。証明書および鍵ペアデータベースファイルは、それらを使用するサーバーインスタンス名をとって、名付けられます。以前のバージョンで複数のサーバーインスタンスが同じエイリアスを共用していた場合は、移行されると、証明書と鍵ペアファイルは新しいサーバーインスタンスの名前をとって名前変更されます。

サーバーインスタンスに関連のある信頼データベース全体が移行されます。以前のデータベースにリストされている CA はすべて、Proxy Server 4 データベースに移行されます。CA が重複している場合には、有効期限が切れるまで以前の CA を使用します。重複している CA は削除しないでください。

Proxy Server 3.x 証明書は、サポートされている Network Security Services (NSS) のフォーマットに移行されます。証明書は、アクセス元、つまり、管理サーバーの「Security」タブか、サーバーマネージャーの「Security」タブの「Proxy Server」ページに応じて名前が付けられます。

▼ 証明書を移行するには

1. ローカルコンピュータから管理サーバーまたはサーバーマネージャーにアクセスし、「**Security**」タブを選択します。
2. 「**Migrate 3.x Certificates**」リンクをクリックします。
3. **3.6** サーバーがインストールされているルートディレクトリを指定します。
4. このコンピュータのエイリアスを指定します。
5. 管理者のパスワードを入力し、「了解」をクリックします。

組み込みルート証明書モジュールの使用

動的に読み込み可能なルート証明書モジュールが、Proxy Server に含まれており、VeriSign を含む多数の CA のルート証明書が格納されています。ルート証明書モジュールを使用すると、より簡単な方法でルート証明書を新しいバージョンにアップグレードできます。以前のバージョンでは、古いルート証明書を 1 つずつ削除し、その後新しいルート証明書を 1 つずつインストールする必要がありました。現在は、ルート証明書モジュールファイルを Proxy Server の新しいバージョンへ更新するだけで、一般的な CA 証明書をインストールできます。この証明書モジュールファイルは将来の Proxy Server のバージョンでも継続して使用できます。

ルート証明書は PKCS #11 暗号化モジュールとして実装されているため、モジュールに含まれているルート証明書を削除することはできません。削除のオプションは、ルート証明書を管理するときには表示されません。サーバーインスタンスからルート証明書を削除する場合は、サーバーの `alias` ディレクトリ内で次のエントリを削除すれば、ルート証明書モジュールを無効にできます。

- `libnssckbi.so` (ほとんどの UNIX プラットフォーム)
- `nssckbi.dll` (Windows)

ルート証明書モジュールを復元する場合は、*server-root*`/bin/proxy/lib` (UNIX) または *server-root*`\\ bin\\proxy\\bin` (Windows) から、該当する拡張子を持つファイルを `alias` サブディレクトリにコピーできます。

ルート証明書の信頼情報は変更できます。信頼情報は、編集されるサーバーインスタンスの証明書データベースに書き込まれ、ルート証明書モジュールそのものには戻されません。

証明書の管理

ユーザーのサーバーにインストールされた CA からの証明書の信頼の設定値を表示、削除または編集できます。

▼ 証明書を管理するには

1. 管理サーバーまたはサーバーマネージャーにアクセスし、「**Security**」タブを選択します。
2. 「**Manage Certificates**」リンクをクリックします。
   - 内部暗号化モジュールを使用して、デフォルト設定の証明書を管理する場合には、インストールされているすべての証明書のリストがその種別および有効期限とともに表示されます。証明書はすべて、ディレクトリ *server-root*`/alias` に格納されます。

- ハードウェアアクセラレータなどの外部の暗号化モジュールを使用している場合には、各モジュールのパスワードを最初に入力し、「了解」をクリックします。モジュール内に証明書が組み込まれ、証明書リストが更新されます。

3 管理する証明書の名前をクリックします。

その種類の証明書に関する管理オプションのあるページが表示されます。クライアントの信頼情報を設定または設定解除できるのは、CA 証明書だけです。外部の暗号化モジュールの中には、証明書を削除できないものもあります。

4 必要な操作を行います。

有効なオプションは次のとおりです。

- 内部的に取得した証明書については、「Delete certificate」または「Quit」
- CA から発行された証明書については、「Set client trust」、「Unset server trust」、または「Quit」

  証明書情報には、所有者と発行者が含まれます。信頼の設定では、クライアントの信頼情報を設定したり、サーバーの信頼情報の設定を解除したりできます。LDAP サーバー証明書の場合は、サーバーが信頼されている必要があります。

CRL と CKL のインストールと管理

証明書の失効リスト (Certificate Revocation List、CRL) および危殆化鍵リスト (Compromised Key List、CKL) は、クライアントまたはサーバーのユーザーが信頼すべきでない証明書および鍵を知らせます。証明書の有効期限が切れる前にユーザーが事務所を変更したり、その組織を離れるような場合など、証明書のデータが変わった場合には、その証明書は無効になり、そのデータが CRL に表示されます。鍵が不正に変更されたり、その他不正に使用されたりした場合には、その鍵とそのデータが CKL に表示されます。CRL と CKL は、両方とも CA によって作成され、定期的に更新されます。これらのリストの取得については、特定の CA にお問い合わせください。

この節では、CRL と CKL のインストールと管理の方法について説明します。

▼ CRL または CKL をインストールするには

1 **CA** から **CRL** または **CKL** を取得して、ローカルディレクトリにダウンロードします。

2 管理サーバーまたはサーバーマネージャーにアクセスし、「**Security**」タブを選択します。

3 「**Install CRL/CKL**」リンクをクリックします。

4 次のいずれかを選択します。

- Certificate Revocation List
  - Compromised Key List

5 インストールするファイルへのフルパス名を入力して、「了解」をクリックします。

「Add Certificate Revocation List」ページまたは「Add Compromised Key List」ページが表示され、CRL または CKL の情報が一覧表示されます。データベースに CRL または CKL がすでにある場合には、「Replace Certificate Revocation List」ページまたは「Replace Compromised Key List」ページが表示されます。

6 **CRL** または **CKL** を追加または置換します。

▼ **CRL** と **CKL** を管理するには

1 管理サーバーまたはサーバーマネージャーにアクセスし、「**Security**」タブを選択します。

2 「**Manage CRL/CKL**」リンクをクリックします。

「Manage Certificate Revocation Lists/Compromised Key Lists」ページが表示されます。インストールされているすべての CRL と CKL が、有効期限とともに一覧表示されます。

3 「**Server CRLs**」または「**Server CKLs**」リストのどちらかから証明書を選択します。

4 「**Delete CRL**」または「**Delete CKL**」を選択して **CRL** または **CKL** を削除します。

5 「**Quit**」を選択して管理ページに戻ります。

セキュリティーに関する詳細設定

証明書を取得すると、サーバーのセキュリティー保護を開始できます。Sun Java System Web Proxy Server には、この節で説明するような、多くのセキュリティー機能が用意されています。

暗号化とは、情報を対象とした受信者以外の人が読めないような内容にするための、変換プロセスのことです。復号化とは、暗号化された情報を判読可能な状態に戻すための、変換プロセスのことです。Proxy Server は、SSL (Secure Sockets Layer) および TLS (Transport Layer Security) 暗号化プロトコルをサポートしています。

暗号化方式とは、暗号化または復号化に使用する暗号アルゴリズム (数学関数) のことです。SSL と TLS プロトコルには、多数の暗号化方式のセットが含まれています。安全度は、暗号化方式によって異なります。一般的に、暗号化方式で使用するビット数が多いほど、データの復号化は難しくなります。

双方向の暗号化プロセスでは、必ず、送信側と受信側の両方が同じ暗号化方式を使用する必要があります。多数の暗号化方式があるため、最も一般的に使用されている方式に対してサーバーを有効にしておく必要があります。

セキュリティー保護された接続時には、クライアントとサーバーは、通信に、その双方が持てる最も強力な暗号化方式を使用します。SSL 2.0、SSL 3.0、および TLS プロトコルから暗号化方式を選択できます。

注 – SSL 2.0 よりあとのバージョンで、セキュリティーとパフォーマンスが向上しています。システムに SSL 3.0 を使用できないクライアントが存在する場合を除き、SSL 2.0 を使用しないでください。クライアント証明書は、SSL 2.0 暗号化方式での動作が保証されていません。

暗号化プロセスだけでは、サーバーの機密情報のセキュリティー保護には十分ではありません。実際に暗号化結果を生成したり、すでに暗号化された情報を復号化するためには、暗号化方式と一緒に鍵を使用する必要があります。暗号化プロセスでは、この結果を出すために 2 つの鍵を使用します。これが、公開鍵と非公開鍵です。公開鍵を使用して暗号化された情報は、対応する非公開鍵を使用した場合にのみ復号化できます。公開鍵は、証明書の一部として発行されます。対応する非公開鍵だけがセキュリティー保護されます。

各種暗号化方式のセットについての説明と、鍵および証明書については、「Introduction to SSL」を参照してください。

サーバーが使用できる暗号化方式は指定できます。特定の暗号化方式を使わないようにする妥当な理由がないかぎり、すべての暗号化方式を選択する必要があります。ただし、最適でないと思われる暗号化方式を有効にする必要はありません。

注意 – 「Enable No Encryption, Only MD5 Authentication」は選択しないでください。クライアント側でその他の暗号化方式を利用できない場合には、サーバーがデフォルトによりこの設定を使用し、暗号化は行われません。

ここでは、次の内容について説明します。

- 97 ページの「待機ソケットのセキュリティーの有効化」
- 100 ページの「セキュリティーのグローバルな設定」

SSL と TLS プロトコル

Proxy Server では、暗号化通信として SSL と TLS プロトコルをサポートしています。SSL と TLS はアプリケーションには依存せず、ユーザーに意識させずに、より高レベルのプロトコルをこれらの上に階層化することができます。

SSL および TLS プロトコルは、サーバーとクライアントでお互いを認証するために使用される多くの暗号化方式のサポート、証明書の送信、およびセッション鍵の確立を行います。クライアントとサーバーは、サポートしているプロトコルや、暗号化の強度についての会社の方針および暗号化されたソフトウェアの輸出に対する行政上の制約条件などの要因に基づいて、別の暗号化方式セットをサポートすることができます。他の機能の中でも特に、SSL と TLS ハンドシェイクプロトコルは、どの暗号化方式のセットを通信に使用するかをサーバーとクライアントが交渉する方法を決定します。

SSL を使用した LDAP との通信

管理サーバーは SSL を使用して LDAP と通信するようにする必要があります。

注 – ここでは、Proxy Server は SSL クライアントとして動作し、SSL サーバー LDAP 証明書に署名したルート CA 証明書をインポートしている必要があります。LDAP の SSL 証明書が一般的な CA が発行したものでない場合、使用する CA ルート鍵を Proxy Server 鍵ストアにインポートする必要があります。

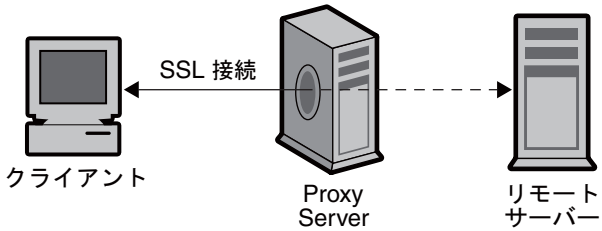
▼ 管理サーバーで SSL 接続を使用して LDAP を有効にするには

1. 管理サーバーにアクセスして、「**Global Settings**」タブをクリックします。
2. 「**Configure Directory Service**」リンクをクリックします。
3. 表示される表で、ディレクトリサービスのリンクをクリックします。

   「Configure Directory Service」ページが表示されます。LDAP ベースのディレクトリサービスが作成されていない場合は、「Create New Service of Type」ドロップダウンリストから、「LDAP Server」を選択し、「New」をクリックしてディレクトリサービスを設定します。LDAP ベースのディレクトリサービスで表示される特定のフィールドについては、オンラインヘルプを参照してください。
4. 接続に **SSL** を使用する場合は、「**Yes**」を選択して、「**Save Changes**」をクリックします。

Proxy Server を介した SSL のトンネリング

Proxy Server (プロキシ) を順方向に実行していて、クライアントがセキュリティー保護されたサーバーに対してプロキシ経由の SSL 接続を要求した場合、プロキシは、セキュリティー保護されたサーバーへの接続を開き、セキュリティー保護されたトランザクションには介入せず、データを双方向にコピーします。このプロセスは、SSL トンネリングとして知られています。概要を次の図に示します。



Proxy Server は、SSL トランザクションをトンネリングします。

図 5–1 SSL 接続

HTTPS URL で SSL トンネリングを使用するには、クライアントが SSL と HTTPS の両方をサポートしている必要があります。HTTPS は、SSL と通常の HTTP を使用して実装されます。HTTPS をサポートしないクライアントでも、Proxy Server の HTTPS プロキシ機能を使用して、HTTPS ドキュメントにアクセスできます。

SSL トンネリングは、アプリケーションレベル (HTTPS) には影響を及ぼさない、下位レベルのアクティビティーです。SSL トンネリングは、プロキシ機能のない SSL と同じぐらいセキュリティー保護されています。間に Proxy Server が存在することで、セキュリティーが危険にさらされたり、SSL の機能が低下したりすることはありません。

データストリームは SSL を使用して暗号化されるため、プロキシは、実際のトランザクションにはアクセスできません。そのため、アクセスログでは、リモートサーバーから受け取った状態コードや、ヘッダー長を表示することができません。また、このプロセスにより、プロキシやサードパーティーのプロキシは、トランザクションを傍受できません。

プロキシがデータを見ることはないため、クライアントとリモートサーバーの間で使用されているプロトコルが SSL であることは確認できません。つまり、プロキシは、その他のプロトコルでやり取りされることを防ぐことができない、ということでもあります。Internet Assigned Numbers Authority (IANA) によって割り当てられている一般的な SSL ポートだけ、つまり、HTTPS の場合はポート 443、SNEWS の場合はポート 563 だけに SSL 接続を制限することをお勧めします。セキュリティー保護されたサーバーがその他のポートで稼働しているサイトがある場合は、 `connect://.*` リソースを使用して、特定ホストのその他のポートに接続できるように、明示的な例外を設定できます。

SSL トンネリング機能は、SOCKS に似た汎用の機能で、プロトコルから独立しています。そのため、この機能をその他のサービスにも利用できます。Proxy Server は、HTTPS や SNEWS プロトコルだけでなく、SSL に対応した任意のアプリケーションの SSL トンネリングに対応できます。

SSL トンネリングの設定

次の手順では、SSL をトンネリングするように Proxy Server を設定する方法について説明します。

▼ SSL トンネリングを設定するには

1. サーバーマネージャーにアクセスしてサーバーインスタンスを選択し、「**Routing**」タブをクリックします。
2. 「**Enable/Disable Proxying**」リンクをクリックします。
3. ドロップダウンリストから `connect://.*.443` リソースを選択します。

   `connect://` 方式は、内部プロキシ表記であり、プロキシの外部には存在しません。`connect` については、96 ページの「SSL トンネリングの技術的詳細」を参照してください。

   その他のポートに接続を許可するには、テンプレートで類似の URL パターンを使用することができます。テンプレートについては、第 16 章を参照してください。
4. 「**Enable Proxying Of This Resource**」を選択して、「了解」をクリックします。

注意 – プロキシの設定が正しくない場合は、ほかのユーザーがプロキシを利用して、実際に接続しているホストからではなく、プロキシホストから telnet 接続が行われているかのように見せることができます。このため、本当に必要なポート以外は決して許可しないでください。また、クライアントホストを制限するために、プロキシでアクセス制御を使用してください。

SSL トンネリングの技術的詳細

内部的には、SSL トンネリングでは、次のように `CONNECT` 方式を宛先ホスト名およびポート番号とともにパラメータとして使用し、空行を続けます。

```
CONNECT energy.example.com:443 HTTP/1.0
```

Proxy Server からの正常な応答は次のようになり、その後に空行が続きます。

```
HTTP/1.0 200 Connection establishedProxy-agent:
Sun-Java-System-Web-Proxy-Server/4.0
```

その後、クライアントとリモートサーバーの間で接続が確立されます。どちらかが接続を閉じるまで、データを双方向に転送できます。

内部的には、URL パターンに基づいた典型的な設定機構を利用するために、ホスト名とポート番号は、自動的に次のような URL にマップされます。

```
connect://energy.example.com:443
```

`connect://` は、Proxy Server で使用される内部表記であり、設定を容易にし、ほかの URL パターンと統一するために使用されます。Proxy Server の外部では `connect` URL は存在しません。Proxy Server がこのような URL をネットワークから受け取ると、無効としてマークし、要求の処理を拒否します。

待機ソケットのセキュリティーの有効化

次の方法で、サーバーの待機ソケットをセキュリティー保護できます。

- セキュリティー機能を有効にする
- 待機ソケットのサーバー証明書を選択する
- 暗号化方式を選択する

注 – セキュリティーは、転送プロキシモードではなく、逆プロキシモードでのみ有効にできます。

セキュリティー機能の有効化

待機ソケット用に他のセキュリティー設定を行うには、セキュリティー機能を有効にしておく必要があります。新しい待機ソケットを作成したり、既存の待機ソケットを編集したりするときに、セキュリティー機能を有効にできます。

▼ 待機ソケットの作成時にセキュリティー機能を有効にするには

1. 管理サーバーまたはサーバーマネージャーにアクセスし、「**Preferences**」タブをクリックします。
2. 「**Add Listen Socket**」リンクをクリックします。
3. 必要な情報を入力します。

注 – 待機ソケットを作成したあとでセキュリティーの設定を行うには、「Edit Listen Sockets」リンクを使用します。

---

4 セキュリティーをオンにするには、「**Security**」ドロップダウンリストから「**Enabled**」を選択して、「了解」をクリックします。

サーバー証明書がインストールされていない場合は、「Disabled」だけが選択できます。特定の設定については、オンラインヘルプを参照してください。

▼ 待機ソケットの編集時にセキュリティー機能をオンにするには

1 管理サーバーまたはサーバーマネージャーにアクセスし、「**Preferences**」タブをクリックします。

2 「**Edit Listen Sockets**」リンクをクリックします。

3 編集する待機ソケットのリンクをクリックします。

4 「**Security**」ドロップダウンリストから「**Enabled**」を選択して、「了解」をクリックします。

サーバー証明書がインストールされていない場合は、「Disabled」だけが選択できます。

待機ソケットのサーバー証明書の選択

管理サーバーまたはサーバーマネージャーのどちらかで、ユーザーが要求し、インストールしたサーバー証明書を使用するように待機ソケットを設定できます。

注 – 少なくとも 1 つの証明書をインストールしておく必要があります。

▼ 待機ソケットのサーバー証明書を選択するには

1 管理サーバーまたはサーバーマネージャーにアクセスし、「**Preferences**」タブをクリックします。

2 「**Edit Listen Sockets**」リンクをクリックします。

3 編集する待機ソケットのリンクをクリックします。

4 「**Security**」ドロップダウンリストから「**Enabled**」を選択して、「了解」をクリックします。

サーバー証明書がインストールされていない場合は、「Disabled」だけが選択できます。

5 「**Server Certificate Name**」ドロップダウンリストから待機ソケットのサーバー証明書を選択して、「了解」をクリックします。

暗号化方式の選択

Proxy Server のセキュリティーを保護するには、SSL を有効にすることをお勧めします。SSL 2.0、SSL 3.0 および TLS 暗号化プロトコルを有効にして、各種の暗号化方式セットを選択することができます。管理サーバーの待機ソケットで、SSL および TLS プロトコルを有効にできます。サーバーマネージャーの待機ソケットで SSL と TLS を有効にすると、特定のサーバーインスタンスに対して、これらのセキュリティーの詳細が設定されます。少なくとも 1 つの証明書をインストールしておく必要があります。

注 – 待機ソケットでの SSL 有効化は、Proxy Server が逆プロキシを実行するように設定されている場合のみ適用されます。

デフォルトの設定では、最も一般的に使用されている暗号化方式が使用できます。特定の暗号化方式を使用してはならない理由がある場合を除き、すべてを選択するようにします。

「TLS Rollback」の、デフォルトで推奨される設定は「Enabled」です。「Enabled」に設定すると、「人が介在するバージョンロールバック」攻撃を検出するようにサーバーが設定されます。TLS 仕様を正しく実装できない一部のクライアントとの相互運用性を確保するために、「TLS Rollback」を「Disabled」に設定しなければならない場合があります。

「TLS Rollback」を無効にすると、通信がバージョンロールバック攻撃を受けやすくなります。バージョンロールバック攻撃は、第三者がクライアントおよびサーバーで SSL 2.0 などの古くてセキュリティー保護機能が低いプロトコルを使用して通信を行うようにするメカニズムです。SSL 2.0 プロトコルには既知の脆弱性があるため、「バージョンロールバック」攻撃を検出できない場合、第三者による暗号化された通信の傍受と復号化が容易になってしまいます。

▼ SSL や TLS を有効にするには

1. 管理サーバーまたはサーバーマネージャーにアクセスし、「**Preferences**」タブをクリックします。
2. 「**Edit Listen Sockets**」リンクをクリックして、編集する待機ソケットのリンクをクリックします。

   セキュリティー保護された待機ソケットの場合、利用できる暗号化方式の設定が表示されます。

   待機ソケットのセキュリティー機能が有効になっていない場合は、SSL と TLS の情報は一覧表示されません。暗号化方式を使用するには、選択している待機ソケットでセキュリティー機能が有効になっている必要があります。詳細は、97 ページの「待機ソケットのセキュリティーの有効化」を参照してください。

3 必要な暗号化方式セットに対応するチェックボックスを選択して、「了解」をクリックします。

4 **Netscape Navigator 6.0** では、**TLS** と **SSL 3.0** の両方を選択します。「**TLS Rollback**」の場合にも、**TLS** を選択して、**SSL 3.0** と **SSL 2.0** の両方が無効になっていることを確認してください。

サーバーで SSL が有効になったら、そのURL には `http` の代わりに `https` が使用されます。SSL 有効サーバー上のドキュメントを示す URL の書式は次のとおりです。
`https://`*servername*`.`*domain*`.`*dom*`:` *port*、例、`https://admin.example.com:443`

デフォルトのセキュリティー保護された HTTP ポート番号 (443) を使用する場合には、URL にポート番号を入力する必要はありません。

セキュリティーのグローバルな設定

SSL が有効なサーバーをインストールすると、グローバルセキュリティーパラメータのサーバーのメイン設定ファイルである `magnus.conf` ファイル内に指令エントリが作成されます。

SSLSessionTimeout

`SSLSessionTimeout` 指令は、SSL 2.0 セッションのキャッシュを制御します。構文は次のとおりです。

`SSLSessionTimeout` *seconds*

ここで *seconds* は、キャッシュされた SSL セッションが無効になるまでの秒数です。デフォルト値は 100 です。`SSLSessionTimeout` 指令が指定された場合には、秒数値は暗黙的に 5 〜 100 秒の間に制限されます。

SSLCacheEntries

キャッシュ可能な SSL セッションの数を指定します。

SSL3SessionTimeout

`SSL3SessionTimeout` 指令は、SSL 3.0 および TLS セッションのキャッシュを制御します。構文を次に示します。

`SSL3SessionTimeout` *seconds*

ここで *seconds* は、キャッシュされた SSL 3.0 セッションが無効になるまでの秒数です。デフォルト値は 86400 (24 時間) です。`SSL3SessionTimeout` 指令が指定された場合には、秒数値は暗黙的に 5 〜 86400 秒の間に制限されます。

▼ SSL 設定ファイル指令の値を設定するには

1. サーバーマネージャーから、サーバーインスタンスを選択します。
2. 設定する待機ソケットでセキュリティーが有効になっていることを確認してください。
   詳細は、97 ページの「待機ソケットのセキュリティーの有効化」を参照してください。
3. `magnus.conf` ファイルを手動で編集し、次の設定の値を入力します。
   - `SSLSessionTimeout`
   - `SSLCacheEntries`
   - `SSL3SessionTimeout`

   `magnus.conf` の詳細については、『Sun Java System Web Proxy Server 4.0.4 Configuration File Reference』を参照してください。

外部暗号化モジュールの使用

Proxy Server は、スマートカードやトークンリングなどの外部の暗号化モジュールとして、次の方法をサポートしています。

- PKCS #11
- FIPS-140

FIPS-140 暗号化標準を有効化する前に、PKCS #11 モジュールを追加しておく必要があります。

ここでは、次の内容について説明します。

- 101 ページの「PKCS #11 モジュールのインストール」
- 105 ページの「FIPS-140 標準」

PKCS #11 モジュールのインストール

Proxy Server は、Public Key Cryptography Standard (PKCS) #11 をサポートします。この標準は、SSL と PKCS #11 モジュール間の通信に使用されるインタフェースを定義します。PKCS #11 モジュールは、SSL ハードウェアアクセラレータへの標準ベースの接続に使用されます。外部のハードウェアアクセラレータにインポートされた証明書と鍵は、`secmod.db` ファイルに格納されます。このファイルは、PKCS #11 モジュールをインストールしたときに生成されます。このファイルは、*server-root*`/alias` ディレクトリに置かれます。

ツール modutil による PKCS #11 モジュールのインストール

PKCS #11 モジュールを、modutil ツールを使用して .jar ファイルまたはオブジェクトファイルの形式でインストールできます。

▼ ツール modutil を使用して PKCS #11 モジュールをインストールするには

1. 管理サーバーを含むすべてのサーバーが停止していることを確認します。
2. データベースが置かれている *server-root*/alias ディレクトリに移動します。
3. **PATH** に *server-root*/bin/proxy/admin/bin を追加します。
4. *server-root* /bin/proxy/admin/bin で modutil を見つけます。
5. 環境を設定します。
   - UNIX の場合: setenv

     LD_LIBRARY_PATH *server-root*/bin/proxy/lib:${LD_LIBRARY_PATH}
   - Windows の場合: PATH に次を追加します

     LD_LIBRARY_PATH *server-root*/bin/proxy/bin

     使用しているコンピュータの PATH は、次で確認できます。
     server-root/proxy-admserv/start
6. 端末ウィンドウに「modutil」と入力します。
   オプションの一覧が表示されます。
7. 必要な操作を行います。
   たとえば、UNIX に PCKS #11 モジュールを追加する場合には、次のように入力します。

   modutil -add (*PKCS#11* ファイルの名前) -libfile (*PKCS #11* 用の *libfile*) -nocertdb -dbdir (*db* ディレクトリ)

ツール pk12util によるエクスポート

pk12util を使用して、内部データベースから証明書と鍵をエクスポートしたり、内部または外部の PKCS #11 モジュールにこれらをインポートしたりすることができます。証明書と鍵は内部データベースにいつでもエクスポートできますが、ほとんどの外部トークンでは証明書と鍵のエクスポートは許可されません。デフォルトでは、pk12util は、cert8.db と key3.db という名前の証明書と鍵データベースを使用します。

▼ 内部データベースから証明書と鍵をエクスポートするには

1. データベースが置かれている *server-root*/alias ディレクトリに移動します。
2. **PATH** に *server-root*/bin/proxy/admin/bin を追加します。
3. *server-root* /bin/proxy/admin/bin で pk12util を見つけます。
4. 環境を設定します。
   - UNIX の場合:

     setenv LD_LIBRARY_PATH/ *server-root*/bin/proxy/lib:${LD_LIBRARY_PATH}
   - Windows の場合: PATH に次を追加します

     LD_LIBRARY_PATH *server-root*/bin/proxy/bin
     使用しているコンピュータの PATH は、次で確認できます。
     server-root/proxy-admserv/start
5. 端末ウィンドウに「pk12util」と入力します。
   オプションの一覧が表示されます。
6. 必要な操作を行います。
   たとえば、UNIX では次のように入力します。

   ```
   pk12util -o certpk12 -n Server-Cert [-d /server/alias] [-P https-test-host]
   ```
7. データベースパスワードを入力します。
8. pkcs12 パスワードを入力します。

▼ 内部または外部の **PKCS #11** モジュールに証明書と鍵をインポートするには

1. データベースが置かれている *server-root*/alias ディレクトリに移動します。
2. **PATH** に *server-root*/bin/proxy/admin/bin を追加します。
3. *server-root* /bin/proxy/admin/bin で pk12util を見つけます。
4. 環境を設定します。
   次に例を示します。
   - UNIX の場合:

```
setenv LD_LIBRARY_PATH/ server-root/bin/proxy/lib:${LD_LIBRARY_PATH}
```

- Windows の場合: PATH に次を追加します。

  ```
  LD_LIBRARY_PATH server-root/bin/proxy/bin
  ```

  使用しているコンピュータの PATH は、次で確認できます。

  ```
  server-root/proxy-admserv/start
  ```

5. 端末ウィンドウに「pk12util」と入力します。
   オプションの一覧が表示されます。

6. 必要な操作を行います。
   たとえば、UNIX では次のように入力します。

   ```
   pk12util -i pk12_sunspot [-d certdir][-h “nCipher”][-P
   https-jones.redplanet.com-jones-]
   ```

   -P は -h のあとに、最後の引数として使用します。

   引用符記号の中の大文字と空白文字を含む、正確なトークン名を入力します。

7. データベースパスワードを入力します。

8. pkcs12 パスワードを入力します。

外部証明書を使用したサーバーの起動

たとえば、ハードウェアアクセラレータなど、外部 PKCS #11 モジュールにサーバーの証明書をインストールする場合には、server.xml ファイルを編集するか、または次に説明するように、証明書名を指定するまで、サーバーはその証明書の使用を開始できません。

サーバーは常に、Server-Cert という名前の証明書を使用して起動しようとします。しかし、外部 PKCS #11 モジュール内の証明書には、識別子内にモジュールのトークン名のうちの 1 つが含まれています。たとえば、smartcard0 と呼ばれる外部スマートカードリーダー上にインストールされているサーバー証明書の名前が、smartcard0:Server-Cert となるなどです。

外部モジュールにインストールされている証明書を使用してサーバーを起動するには、稼動する待機ソケットの証明書名を指定する必要があります。

▼ 待機ソケットの証明書を選択するには

待機ソケットのセキュリティー機能が有効になっていない場合は、証明書情報は表示されません。待機ソケットの証明書名を選択するには、まず、その待機ソケットでセキュリティー機能が有効になっていることを確認する必要があります。詳細は、97 ページの「待機ソケットのセキュリティーの有効化」を参照してください。

1. 管理サーバーまたはサーバーマネージャーにアクセスし、「**Preferences**」タブをクリックします。
2. 「**Edit Listen Sockets**」リンクをクリックします。
3. 証明書と関連付ける待機ソケットのリンクをクリックします。
4. 「**Server Certificate Name**」ドロップダウンリストから待機ソケットのサーバー証明書を選択して、「了解」をクリックします。

   このリストには、インストールされているすべての内部および外部の証明書が記載されています。

   手動で `server.xml` ファイルを編集することにより、代わりにそのサーバー証明書を使用して起動することをサーバーに指示することもできます。`SSLPARAMS` の `servercertnickname` 属性を次のように変更します。

   ```
   $TOKENNAME:Server-Cert
   ```

   `$TOKENNAME` に使用する値を知るには、サーバーの「Security」タブに移動して、「Manage Certificates」リンクを選択します。Server-Cert の格納されている外部モジュールにログインすると、`$TOKENNAME:$NICKNAME` フォームのリストにその証明書が表示されます。

   信頼データベースを作成していない場合には、外部 PKCS #11 モジュールの証明書を要求するかまたはインストールすると、信頼データベースが 1 つ作成されます。作成されるデフォルトのデータベースには、パスワードがないためアクセスできません。外部モジュールは動作しますが、サーバー証明書を要求してインストールすることはできません。パスワードのないデフォルトのデータベースが作成された場合には、「Security」タブの「Create Database」ページを使用してパスワードを設定してください。

FIPS-140 標準

PKCS #11 API を使用すれば、暗号化操作を実行するソフトウェアまたはハードウェアモジュールとの通信が可能です。PKCS #11 を Proxy Server にインストールすると、 FIPS-140 に準拠するよう、Proxy Server を設定できます。FIPS は Federal Information Processing Standards の略です。これらのライブラリは、SSL 3.0 にのみ含まれています。

▼ FIPS-140 を有効にするには

1. **FIPS-140** の指示に従ってプラグインをインストールします。
2. 管理サーバーまたはサーバーマネージャーにアクセスし、「**Preferences**」タブをクリックします。

3 「**Edit Listen Sockets**」リンクをクリックします。

待機ソケットがセキュリティー保護されている場合は、「Edit Listen Sockets」ページに使用可能なセキュリティー設定が表示されます。

FIPS-140 を使用するには、選択している待機ソケットでセキュリティーが有効になっている必要があります。詳細は、97 ページの「待機ソケットのセキュリティーの有効化」を参照してください。

4 「**Enabled**」が選択されていない場合は、「**SSL Version 3**」ドロップダウンリストから選択します。

5 次のうち、適切な **FIPS-140** 暗号化方式のセットを選択して、「了解」をクリックします。

- Enable Triple DES with 168–bit encryption and SHA authentication (FIPS)
- Enable DES with 56–bit encryption and SHA authentication (FIPS)

クライアントセキュリティー要件の設定

サーバーをセキュリティー保護するためのすべての手順が終了したあと、クライアントに関するその他のセキュリティー要件を設定できます。

クライアント認証は、SSL 接続に必須ではありませんが、使用すると、暗号化された情報がさらに確実に適切な相手に送信されます。クライアント認証を逆プロキシで使用して、コンテンツサーバーが、承認されていないプロキシやクライアントと情報を共有しないようにすることができます。

ここでは、次の内容について説明します。

クライアント認証の要求

管理サーバーと各サーバーインスタンスの待機ソケットが、クライアント認証を要求できるようになります。クライアント認証を有効にすると、クエリーに対してサーバーが応答を送信する前に、クライアントの証明書が必要となります。

Proxy Server は、クライアント証明書に含まれる CA と、署名済みクライアント証明書の信頼された CA を照合することによってクライアント証明書の認証をサポートします。「Security」タブにある「Manage Certificates」ページで、署名済みクライアント証明書の信頼された CA のリストを表示できます。

信頼された CA からのクライアント証明書を持っていないクライアントを拒否するように Proxy Server を設定できます。信頼された CA を受け入れ、または拒否するには、その CA についてクライアントの信頼を設定する必要があります。詳細は、90 ページの「証明書の管理」を参照してください。

Proxy Server は、エラーの記録、証明書の拒否、および証明書が期限切れの場合にはクライアントに対してメッセージの返送を行います。また、「Manage Certificates」ページで、有効期限切れの証明書を表示できます。

クライアントの証明書から情報を収集し、これを LDAP ディレクトリ内のユーザーエントリと照合するようにサーバーを設定できます。このようにすると、確実に LDAP ディレクトリ内で有効な証明書とエントリをクライアントが持つようにできます。また、クライアント証明書が LDAP ディレクトリ内の証明書と確実に一致するようにできます。これを実行する方法については、110 ページの「LDAP へのクライアント証明書のマッピング」を参照してください。

証明書のあるユーザーは、信頼された CA だけでなく、アクセス制御の規則 (Access Control Rule、ACL) とも一致しなければならないように、クライアント証明書をアクセス制御と組み合わせることができます。詳細は、160 ページの「アクセス制御ファイルの使用」を参照してください。

▼ クライアントの認証を要求するには

1. 管理サーバーまたはサーバーマネージャーにアクセスし、「**Preferences**」タブをクリックします。
2. 「**Edit Listen Sockets**」リンクをクリックします。
3. クライアント認証を要求している待機ソケットのリンクをクリックします。
4. 「**Client Authentication**」ドロップダウンリストを使用し、待機ソケットのクライアント認証を要求して、「了解」をクリックします。

逆プロキシでのクライアント認証

逆プロキシでは、次のいずれかのシナリオに応じて、クライアント認証を設定できます。

- 「プロキシがクライアントを認証」:このシナリオでは、受け入れ可能な証明書を持つクライアントすべてにアクセスを許可するか、または受け入れ可能な証明書があり、かつ Proxy Server のアクセス制御リスト上で認識されたユーザーであるクライアントだけにアクセスを許可することができます。

注 – プロキシには、CA またはユーザー証明書に署名した自己署名アプリケーションのユーザールート鍵が必要です。ユーザーは、CA または Proxy Server 証明書に署名した自己署名アプリケーションの Proxy Server ルート鍵をロードしている必要があります。

- 「コンテンツサーバーがプロキシを認証」:このシナリオでは、コンテンツサーバーが Proxy Server に実際に接続していて、その他のサーバーには接続していないことを確認することができます。

注 – プロキシには、CA またはコンテンツサーバー証明書に署名した自己署名アプリケーションのコンテンツサーバールート鍵が必要です。コンテンツサーバーには、CA または Proxy Server 証明書に署名した自己署名アプリケーションの Proxy Server ルート鍵が必要です。

- 「プロキシがクライアントを認証、かつコンテンツサーバーがプロキシを認証」:このシナリオでは、逆プロキシでのセキュリティーと認証を最大限にします。

これらのシナリオの設定方法については、108 ページの「逆プロキシでのクライアント認証の設定」を参照してください。

逆プロキシでのクライアント認証の設定

セキュリティー保護された逆プロキシでクライアント認証を実行すると、接続のセキュリティー性がさらに保証されます。次の手順では、ユーザーが選択するシナリオに応じたクライアント認証の設定方法について説明します。

注 – 各シナリオでは、セキュリティー保護されたクライアント - プロキシ間接続と、セキュリティー保護されたプロキシ - コンテンツサーバー間接続の両方がなされていることを前提としています。

▼ 「プロキシがクライアントを認証」シナリオを設定するには

1. 第 14 章の「逆プロキシの設定」での指示に従って、セキュリティー保護されたクライアント - プロキシと、セキュリティー保護されたプロキシ - コンテンツサーバーのシナリオを設定します。
2. サーバーマネージャーにアクセスしてサーバーインスタンスを選択し、「**Preferences**」タブをクリックします。

3 「**Edit Listen Sockets**」リンクをクリックして、表示される表内にある目的の待機ソケットのリンクをクリックします。

「Add Listen Socket」リンクを使用して、待機ソケットを設定および追加します。

4 次のようにして、クライアント認証要件を指定します。

a. 有効な証明書を持つすべてのユーザーにアクセスを許可するには、次の手順に従います。

「Security」セクションで、「Client Authentication」設定を使用して、この待機ソケットのクライアント認証を要求します。サーバー証明書がインストールされていない場合は、この設定は表示されません。

b. 両方の有効な証明書を持ち、アクセス制御で受け入れ可能なユーザーとして指定されているユーザーだけにアクセスを許可するには、次の手順に従います。

i. 「**Security**」セクションで、「**Client Authentication**」設定をオフのままにします。サーバー証明書がインストールされていない場合は、この設定は表示されません。

ii. このサーバーインスタンスのサーバーマネージャーの「**Preferences**」タブで、「**Administer Access Control**」リンクをクリックします。

iii. **ACL** を選択し、「**Edit**」ボタンをクリックします。

「Access Control Rules For」ページが表示されます。プロンプトが表示された場合は、先に認証を行ってください。

iv. アクセス制御を有効にします。「**Access control Is On**」チェックボックスが選択されていない場合は選択します。

v. **Proxy Server** を逆プロキシとして認証するように設定します。

詳細は、327 ページの「逆プロキシの設定」を参照してください。

vi. 希望するアクセス制御ルールの「**Rights**」リンクをクリックして、下のフレームでアクセス権を指定し、「**Update**」をクリックしてこのエントリを更新します。

vii. 「**Users/Groups**」リンクをクリックします。下のフレームで、ユーザーとグループを指定し、認証方法として **SSL** を選択し、「**Update**」をクリックしてこのエントリを更新します。

viii. 上のフレームで「**Submit**」をクリックして、エントリを保存します。

アクセス制御の設定については、第 8 章を参照してください。

▼ 「コンテンツサーバーがプロキシを認証」シナリオを設定するには

1. 327 ページの「逆プロキシの設定」の指示に従って、セキュリティー保護されたクライアント - プロキシと、セキュリティー保護されたプロキシ - コンテンツサーバーのシナリオを設定します。
2. コンテンツサーバーで、クライアント認証を有効にします。

   このシナリオを変更して、Proxy Server に対してはセキュリティー保護されていないクライアント接続、コンテンツサーバーに対してはセキュリティー保護された接続を設定し、かつコンテンツサーバーが Proxy Server を認証するようにすることができます。これを行うには、次の手順で説明しているように、暗号化を無効にして、プロキシに証明書のみの初期化を指示する必要があります。

▼ 「プロキシがクライアントを認証、かつコンテンツサーバーがプロキシを認証」シナリオを設定するには

1. 108 ページの「「プロキシがクライアントを認証」シナリオを設定するには」での指示に従って、「プロキシがクライアントを認証」シナリオを設定します。
2. コンテンツサーバーで、クライアント認証を有効にします。

LDAP へのクライアント証明書のマッピング

この節では、Proxy Server がクライアント証明書を LDAP ディレクトリ内のエントリにマップするために使用するプロセスについて説明します。LDAP にクライアント証明書をマッピングする前に、必要な ACL も設定しておく必要があります。詳細は、第 8 章を参照してください。

サーバーがクライアントから要求を受信すると、処理を進める前にサーバーはクライアントの証明書を求めます。一部のクライアントは、要求と一緒にクライアント証明書をサーバーに送信します。

サーバーは、管理サーバーの信頼された CA リストとその証明書の発行元である CA を照合します。一致しない場合、Proxy Server は接続を終了します。一致した場合、サーバーは要求の処理を続行します。

証明書が信頼された CA からのものであることを確認したあと、サーバーは、次の方法で LDAP エントリにその証明書をマップします。

- クライアント証明書の発行者とサブジェクト DN を LDAP ディレクトリ内の分岐点にマップします。
- クライアント証明書のサブジェクト (エンドユーザー) に関する情報と一致するエントリがないか LDAP ディレクトリを検索します。

- (オプション) DN に対応する LDAP エントリ内のクライアント証明書とそのクライアント証明書を検証します。

サーバーは、certmap.conf と呼ばれる証明書マッピングファイルを使用して LDAP 検索の実行方法を決定します。このマッピングファイルは、クライアント証明書から取得する値 (エンドユーザー名、電子メールアドレスなど) をサーバーに指示します。サーバーは、これらの値を使用して LDAP ディレクトリ内にユーザーエントリがないか検索しますが、はじめに、LDAP ディレクトリ内のどこから検索を開始するかを決定する必要があります。証明書マッピングファイルは、開始する場所もサーバーに指示します。

サーバーに、検索を開始する場所および検索する内容が通知されると、LDAP ディレクトリ内で検索を実行します (2 つ目の項目)。一致するエントリがない、または一致するエントリがあってもマッピングが証明書を検証するように設定されていない場合、検索は失敗します。

次の表は、予期される検索結果の動作を示しています。予期される動作は、ACL で指定できます。たとえば、証明書の照合に失敗した場合、自分だけは Proxy Server に受け入れられるように指定できます。ACL の詳細設定については、160 ページの「アクセス制御ファイルの使用」を参照してください。

表 5–1 LDAP 検索結果

LDAP 検索結果	証明書の検証がオン	証明書の検証がオフ
検出されたエントリなし	認証失敗	認証失敗
検出されたエントリが 1 つのみ	認証失敗	認証成功
検出されたエントリが複数	認証失敗	認証失敗

サーバーが LDAP ディレクトリ内で一致するエントリと証明書を検出したあと、サーバーはその情報を使用してトランザクションを処理できます。たとえば、一部のサーバーでは、サーバーへのアクセスを判断するのに証明書 - LDAP 間マップを使用します。

certmap.conf ファイルの使用

証明書のマッピングは、LDAP ディレクトリ内のユーザーエントリをサーバーがどのように検索するかを決定します。certmap.conf ファイルを使用して、名前で指定された証明書を LDAP エントリにマップする方法を設定できます。このファイルを編集し、LDAP ディレクトリの組織と照合されるようにエントリを追加し、ユーザーに持たせる証明書のリストを表示するようにします。subjectDN 内で使用されているユーザー ID、電子メール、またはその他の値に基づいてユーザーを認証することができます。特に、マッピングファイルでは、次の情報を定義します。

- LDAP ツリー内でサーバーが検索を開始する場所
- LDAP ディレクトリ内のエントリを検索するときにサーバーが検索条件として使用する証明書の属性
- サーバーが追加の検証プロセスを実施するかどうか

証明書マッピングファイルは、次の場所にあります。

server-root`/userdb/certmap.conf`

このファイルには、それぞれが異なる CA に適用される、1 つ以上の名前付きマッピングが含まれています。マッピングの構文は、次のとおりです。

`certmap` *name issuerDNname* : *property* [*value*]

最初の行にはエントリの名前と、CA 証明書内に記載されている識別名を設定する属性を指定します。*name* は任意です。好きな名前に定義できます。ただし、*issuerDN* は、そのクライアント証明書を発行した CA の発行者 DN と完全に一致している必要があります。たとえば、次の 2 つの発行者 DN 行は、属性間に空白文字があるかどうかという点が異なるだけですが、サーバーは、これら 2 つのエントリを別のものとして取り扱います。

```
certmap sun1 ou=Sun Certificate Authority,o=Sun,c=UScertmap sun2 ou=Sun
Certificate Authority, o=Sun, c=US
```

注 – Sun Java System Directory Server を使用しているときに issuerDN の照合で問題があった場合は、Directory Server のエラーログを調べて有用な情報を探します。

名前付きマッピングの 2 行目以降の行は、プロパティーが値と照合されます。`certmap.conf` ファイルには、次に示す 6 つのデフォルトのプロパティーがあります。証明書 API を使用すると、ユーザー独自のプロパティーをカスタマイズできます。デフォルトのプロパティーは次のとおりです。

- `DNComps` はコンマで区切った属性のリストで、ユーザーの情報、つまりクライアント証明書の所有者と一致するエントリの検索を、サーバーが LDAP ディレクトリ内のどこから開始するかを判断するために使用されます。サーバーは、クライアント証明書からこれらの属性の値を収集し、LDAP DN を設定するためにその値を使用します。これが、LDAP ディレクトリ内でサーバーが検索を開始する場所を決定します。たとえば、DN の `o` 属性と `c` 属性を使用するよう `DNComps` を設定した場合、サーバーは、LDAP ディレクトリ内の `o=`*org*、`c=` *country* エントリから検索を開始します。ここで *org* と *country* は、証明書内の DN に記載されている値に置き換えられます。

  次のような場合には注意が必要です。

  - マッピング内に `DNComps` エントリがない場合、サーバーは `CmapLdapAttr` の設定、またはクライアント証明書内のサブジェクト DN 全体 (つまりエンドユーザーの情報) のいずれかを使用します。

- DNComps エントリはあるが値がないという場合、サーバーは LDAP ツリー全体を検索してフィルタに一致するエントリを探します。

FilterComps は、コンマで区切った属性のリストで、クライアント証明書内のユーザーの DN から情報を収集してフィルタを作成するために使用されます。サーバーは、これらの属性の値を使用して、LDAP ディレクトリ内でエントリを照合するために使用する検索条件を作成します。サーバーが LDAP ディレクトリ内で、証明書から収集したユーザー情報に一致する 1 つまたは複数のエントリを検出した場合、検索は成功し、オプションでサーバーが検証を行います。

たとえば、電子メール属性とユーザー ID 属性を使用するよう FilterComps を設定すると (FilterComps=e,uid)、サーバーは、電子メールとユーザー ID の値がクライアント証明書から収集したエンドユーザーの情報と一致するエントリをディレクトリから検索します。電子メールアドレスとユーザー ID は、通常、ディレクトリ内で一意のエントリであるため、フィルタとして適切です。フィルタは、LDAP データベース内で 1 つだけのエントリと一致するような特有のものである必要があります。

フィルタのための属性名は、LDAP ディレクトリではなく、証明書から取得した属性名である必要があります。たとえば、一部の証明書にはユーザーの電子メールアドレスの e 属性がありますが、LDAP では、この属性を mail と呼んでいます。

次の表は、x509v3 証明書の属性を示しています。

表 5–2 x509v3 証明書の属性

属性	説明
c	国
o	内容の紹介
cn	共通名
l	保存場所
st	状態
ou	組織単位
uid	UNIX/Linux ユーザー ID
email	電子メール アドレス

- verifycert は、LDAP 内にある証明書とクライアントの証明書を比較するかどうかをサーバーに指示します。プロパティーは次の 2 つの値を取ります。「on」と「off」です。ただし、このプロパティーは、LDAP ディレクトリに証明書があるときだけ使用してください。この機能は、エンドユーザーが、有効な、取り消されていない証明書を確実に所有できるようにするのに便利です。

- `CmapLdapAttr` は、LDAP ディレクトリ内の属性の名前で、対象のユーザーに属しているすべての証明書に記載されているサブジェクト DN が含まれています。このプロパティーのデフォルトは、`certSubjectDN` です。この属性は標準の LDAP 属性ではないため、このプロパティーを使用するには、LDAP スキーマを拡張する必要があります。詳細は、「Introduction to SSL」を参照してください。

  このプロパティーが `certmap.conf` ファイル内に存在する場合、サーバーは、このプロパティーの名前の付いた属性が、証明書から取得されたサブジェクトの完全な DN に一致しているエントリを LDAP ディレクトリ全体から検索します。エントリが検出されなかった場合、サーバーは `DNComps` マッピングと `FilterComps` マッピングを使用して、検索を再試行します。

  LDAP エントリと証明書を照合するためのこの方法は、`DNComps` と `FilterComps` を使用してエントリを照合することが難しい場合に便利です。
- `Library` は、共用ライブラリまたは DLL へのパス名です。証明書 API を使用して独自のプロパティーを作成する場合のみ、このプロパティーを使用してください。
- `InitFn` は、カスタムライブラリの `init` 関数の名前です。証明書 API を使用して独自のプロパティーを作成する場合のみ、このプロパティーを使用してください。

これらのプロパティーについては、114 ページの「マッピング例」に記載されている例を参照してください。

カスタムプロパティーの作成

クライアント証明書 API を使用して、独自のプロパティーを作成できます。カスタムマッピングを行ったら、次のようにマッピングを参照します。

name`:library` *path_to_shared_libraryname* `:InitFN` *name_of_ init_function*

次に例を示します。

```
certmap default1 o=Sun Microsystems, c=US default1:library
/usr/sun/userdb/plugin.so default1:InitFn plugin_init_fn default1:DNComps ou o
c default1:FilterComps l default1:verifycert on
```

マッピング例

`certmap.conf` ファイルには、少なくとも 1 つのエントリが必要です。次の例では、`certmap.conf` ファイルを使用できるいくつかの方法を示しています。

例 1 デフォルトのマッピングが 1 つだけある `certmap.conf` ファイル

```
certmap default defaultdefault:DNComps ou, o, cdefault:FilterComps e,
uiddefault:verifycert on
```

この例でサーバーは、ou=*orgunit*、 o=*org*、c=*country* エントリを格納している LDAP 分岐点から検索を開始します。ここで斜体のテキストは、クライアント証明書内のサブジェクト DN に記載されている値に置き換えられます。

次に、サーバーが証明書に記載されている電子メールアドレスとユーザー ID の値を使用して、LDAP ディレクトリ内で一致するエントリを検索します。エントリが検出されると、サーバーは、クライアントにより送信されたエントリをディレクトリ内に格納されているエントリと比較して、証明書を検証します。

例 2 つのマッピングが記述された certmap.conf ファイル

次のファイル例には、2 つのマッピングが記述されています。1 つはデフォルト用で、もう 1 つは US Postal Service 用です。

```
certmap default defaultdefault:DNCompsdefault:FilterComps e, uid

certmap usps ou=United States Postal Service, o=usps, c=USusps:DNComps
ou,o,cusps:FilterComps eusps:verifycert on
```

サーバーが US Postal Service 以外から証明書を取得する場合、サーバーはデフォルトのマッピングを使用します。これは、LDAP ツリーの最上部から、クライアントの電子メールアドレスとユーザー ID に一致するエントリの検索を開始します。証明書が US Postal Service からのものである場合、サーバーは、組織単位を格納している LDAP 分岐から検索を開始し、一致する電子メールアドレスを探します。サーバーは証明書の検証も行います。それ以外の証明書は検証されません。

注意 – 証明書内の発行者 DN (つまり CA の情報) は、マッピングの最初の行に記述されている発行者 DN と同じでなくてはなりません。前述の例では、o=United States Postal Service,c=US という発行者 DN からの証明書は、o 属性と c 属性の間に空白文字がないため一致しません。

例 3 LDAP データベースの検索

次の例では、CmapLdapAttr プロパティーを使用して、クライアント証明書から取得したサブジェクト DN 全体とぴったり一致する値を持つ certSubjectDN という属性を、LDAP データベースから検索します。この例では、LDAP ディレクトリに certSubjectDN 属性を持つエントリがあることを前提としています。

```
certmap myco ou=My Company Inc, o=myco, c=USmyco:CmapLdapAttr
certSubjectDNmyco:DNComps o, c myco:FilterComps mail, uid myco:verifycert on
```

次のようなクライアント証明書のサブジェクトを考えます。

```
uid=Walt Whitman, o=LeavesOfGrass Inc, c=US
```

サーバーは、はじめに次の情報を格納しているエントリを検索します。

```
certSubjectDN=uid=Walt Whitman, o=LeavesOfGrass Inc, c=US
```

1 つまたは複数の一致したエントリが検出された場合、サーバーはそのエントリの検証処理を進めます。一致するエントリが検出されなかった場合には、サーバーは、`DNComps` と `FilterComps` を使用して、一致するエントリを検索します。この例では、サーバーは、`o=LeavesOfGrass Inc, c=US` の下にあるすべてのエントリから `uid=Walt Whitman` を検索します。

強固な暗号化方式の設定

サーバーマネージャーの「Preferences」タブにある「Set Cipher Size」オプションでは、アクセスするための非公開鍵のサイズに 168、128、または 56 ビットのいずれか、または制限なしを選択できます。制限に適合しない場合に使用するファイルを指定することができます。ファイルが指定されていない場合、Proxy Server が、「Forbidden」ステータスを返します。

アクセスするための鍵サイズとして、「Security Preferences」での現在の暗号化方式の設定と整合しないサイズを選択すると、Proxy Server から、暗号化方式でより大きいサイズの非公開鍵を有効にする必要があることを知らせる警告が表示されます。

現在、鍵サイズ制限の実装は、`Service fn=key-toosmall` ではなく、`obj.conf` にある NSAPI `PathCheck` 指令に基づいています。この指令は次のとおりです。

```
PathCheck fn="ssl-check" [secret-keysize=nbits ] [bong-file=filename ]
```

ここで *nbits* は、非公開鍵で必要とされる最小ビット数で、*filename* は、制限に適合しない場合に使用されるファイルの名前です。

SSL が有効ではない場合、または `secret-keysize` パラメータが指定されていない場合、`PathCheck` は `REQ_NOACTION` を返します。現在のセッションの非公開鍵サイズが指定された `secret-keysize` より小さいとき、関数は、`bong-file` が指定されていない場合には `PROTOCOL_FORBIDDEN` のステータスとともに `REQ_ABORTED` を返します。`bong-file` が指定されている場合、関数は `REQ_PROCEED` を返して、path 変数が `bong-file` *filename* に設定されます。また、鍵のサイズ制限に適合しない場合は、現在のセッションの SSL セッションキャッシュエントリが無効になるため、次回、同じクライアントがサーバーに接続するとき完全な SSL ハンドシェイクが行われます。

注 – 「Set Cipher Size」フォームは、`PathCheck fn=ssl-check` を追加するときにオブジェクト内で検出された `Service fn=key-toosmall` 指令を削除します。

▼ 強固な暗号化方式を設定するには

1. サーバーマネージャーにアクセスしてサーバーインスタンスを選択し、「**Preferences**」タブをクリックします。
2. 「**Set Cipher Size**」リンクをクリックします。
3. ドロップダウンリストから、強固な暗号化方式を適用するリソースを選択して、「**Select**」をクリックします。正規表現を指定することもできます。
   詳細は、第 16 章を参照してください。
4. 非公開鍵サイズの制限を選択します。
   - 「168 bits or larger」
     - 「128 bits or larger」
     - 「56 bits or larger」
     - 「No restrictions」
5. アクセスを拒否するメッセージのファイルの場所を入力して、「了解」をクリックします。
   暗号化方式については、「Introduction to SSL」を参照してください。

その他のセキュリティーに関する注意事項

第三者が暗号を解読しようとする以外にも、セキュリティーに関するリスクがあります。ネットワークは常に、内側と外側から、ハッカーのリスクにさらされています。ハッカーはさまざまな手口を使って、サーバーやサーバーに格納されている情報にアクセスしようとしています。サーバーで暗号化を有効にするだけでなく、別のセキュリティー対策を立てる必要があります。たとえば、セキュリティー保護された部屋にサーバーコンピュータを設置し、信頼できない個人にサーバーへのプログラムのアップロードを許可しないようにするなどです。この節では、サーバーのセキュリティーをさらに強化するのに必要な、重要な事項についていくつか説明します。

ここでは、次の内容について説明します。

■ 121 ページの「サーバーの限界の把握」

物理的アクセスの制限

このシンプルなセキュリティー手段が、意外と見落とされがちです。承認された人だけが入室できる鍵の掛かった部屋にサーバーコンピュータを設置してください。このようにすると、サーバーコンピュータ自体へのハッキングを防げます。また、コンピュータの管理 (root) パスワードを所有している場合には、パスワードを保護しておく必要があります。

管理アクセスの制限

リモート構成を使用している場合、必ず、数人のユーザーと数台のコンピュータだけが管理作業を実行できるように、アクセス制御を設定します。管理サーバーからエンドユーザーの権限で LDAP サーバーやローカルディレクトリ情報へのアクセスを許可する場合、2 台の管理サーバーを維持し、クラスタ管理を使用することを検討してください。1 台を SSL を有効にしてマスターとなる管理サーバーにし、もう 1 台の管理サーバーをエンドユーザーからアクセスできるようにします。クラスタについては、第 6 章を参照してください。

管理サーバーの暗号化機能もオンにすることをお勧めします。管理に SSL 接続を使用しない場合、リモートサーバーの管理がセキュリティー保護されていないネットワークを介して行われるという点に注意してください。管理パスワードの傍受や、サーバーの再設定がだれにでも可能になってしまいます。

強固なパスワードの選択

サーバーでは多数のパスワードが使用されています。管理パスワード、非公開鍵パスワード、データベースパスワードなどです。この中でもっとも重要なパスワードは管理パスワードです。このパスワードを使えば、誰もがコンピュータ上のどのサーバーの設定でも行えるからです。次に重要なのは、非公開鍵パスワードです。非公開鍵と非公開鍵パスワードを持っていれば誰でも、ユーザーのサーバーであるように見せかけた偽のサーバーを作成したり、サーバーの通信を傍受して改ざんしたりすることができます。

優れたパスワードは、ユーザー自身が覚えやすく、第三者には推測できないようなパスワードです。たとえば、*MCi12!mo* は「My Child is 12 months old!」として覚えることができます。悪いパスワードの例としては、子供の名前や誕生日などが挙げられます。

推測しにくいパスワードの作成

これらのガイドラインに従って、安全性の高いパスワードを作成できます。1 つのパスワードに次の規則のすべてを取り入れる必要はありませんが、使用する規則が多ければ多いほど、パスワードは推測されにくくなります。いくつかヒントを示します。

- パスワードの長さは 6 ～ 14 文字にする
- 次のような正規以外の文字は使用しない: *、"、または空白文字
- 辞書に載っている語を使用しない (どの言語でも)
- E を 3 にする、L を 1 にする、などのよくある文字の置き換えは行わない
- 以下の種類の文字を、できるだけ多く混在させる
  - 大文字
  - 小文字
  - 数値
  - 記号

パスワードまたは PIN の変更

信頼データベースおよび鍵ペアファイルのパスワードまたは PIN は定期的に変更するようにしてください。管理サーバーで SSL を有効にしている場合、サーバーを起動するときにこのパスワードが必要です。パスワードを定期的に変更すると、サーバーのセキュリティー保護が強化されます。

このパスワードは、ローカルコンピュータでのみ変更することをお勧めします。パスワードを変更するときに考慮する必要のあるガイドラインについては、119 ページの「推測しにくいパスワードの作成」を参照してください。

▼ 信頼データベース/鍵ペアファイルのパスワードを変更するには

1. 管理サーバーまたはサーバーマネージャーにアクセスし、「**Security**」タブを選択します。
2. 「**Change Key Pair File Password**」リンクをクリックします。
3. 「**Cryptographic Module**」ドロップダウンリストから、パスワードを変更するセキュリティートークンを選択します。

   デフォルトでは、このトークンは内部鍵データベースの「Internal」になっています。PKCS #11 モジュールがインストールされている場合は、すべてのセキュリティートークンが一覧表示されます。
4. 現在のパスワードを入力します。

5 新しいパスワードを入力します。

6 もう一度新しいパスワードを入力し、「了解」をクリックします。

鍵ペアファイルは必ずセキュリティー保護するようにします。管理サーバーは、*server-root*/alias ディレクトリ内に鍵ペアファイルを格納します。

このファイルがバックアップテープに格納されるのかどうか、または第三者が傍受できるような状態かどうかを知っておくことも大切です。その場合には、バックアップをサーバーと同じように、完全にセキュリティー保護する必要があります。

サーバー上でのほかのアプリケーションの制限

サーバーと同じコンピュータで実行するすべてのアプリケーションを十分に検討します。サーバー上で実行するほかのアプリケーションのセキュリティーホールを利用して、サーバーのセキュリティーが危険にさらされる可能性があります。不必要なプログラムやサービスはすべて無効にしてください。たとえば、UNIX sendmail デーモンは、安全に設定することが難しく、悪影響を及ぼす可能性のあるほかのプログラムをサーバーコンピュータ上で実行するようにプログラムされる可能性があります。

UNIX と Linux

inittab スクリプトと rc スクリプトから開始するプロセスを注意して選択します。telnet または rlogin をサーバーコンピュータから実行しないでください。また、サーバーコンピュータに rdist を置かないでください。このプログラムを使用すると、ファイルを配布できますが、サーバーコンピュータ上のファイルの更新にも使用される可能性があります。

Windows

他のコンピュータと共有するドライブやディレクトリについて、十分に検討してください。また、どのユーザーがアカウントやゲストの特権を所有しているかについても検討してください。サーバー上に置いているプログラムや、ほかのユーザーにインストールを許可するプログラムについても注意してください。ほかのユーザーのプログラムにセキュリティーホールがある可能性もあります。最悪の場合、セキュリティーを侵害するために設計した悪意のあるプログラムを何者かがアップロードする可能性もあります。したがって、サーバー上にプログラムを置くことを許可する前に、そのプログラムを良く調べてください。

クライアントによる SSL ファイルキャッシングの防止

HTML ファイルの <HEAD> セクション内に次の行を追加することで、暗号化されているファイルがクライアントによりキャッシュに書き込まれるのを事前に防止することができます。

```
<meta http-equiv="pragma" content="no-cache">
```

ポートの制限

コンピュータ上で使用していないポートは、すべて無効にします。ルーターやファイアウォールの設定を使用して、最低限のポートセット以外には着信接続ができないように設定します。この保護は、すでに制限された領域内にあるサーバーコンピュータを物理的に使用することによってのみ、コンピュータ上でシェルを取得することができるということを意味します。

サーバーの限界の把握

サーバーは、サーバーとクライアントの間でセキュリティー保護された接続ができるようにします。サーバーは、情報がいったんクライアントに取得されると、情報のセキュリティーを制御することはできず、サーバーコンピュータ自体およびそのディレクトリやファイルに対するアクセスも制御できません。

このような限界を認識しておくことは、避けるべき状況を理解するのに役立ちます。たとえば、SSL 接続を介してクレジットカードの番号を取得するとします。しかしそれらの番号はサーバーコンピュータ上のセキュリティー保護されたファイルに格納されるでしょうか。SSL 接続が終了したあとでこれらの番号はどうなるのでしょうか。SSL を介してクライアントから送信されたすべての情報に対して、必ずセキュリティー保護するようにしてください。

◆ ◆ ◆ 第 6 章

サーバークラスタの管理

この章では、Sun Java System Web Proxy Server のクラスタ化の概念、およびクラスタを使用してサーバー間で設定を共有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。

サーバークラスタについて

クラスタとは、1 つの管理サーバーから管理することができる、複数の Sun Java System Web Proxy Server で構成されたグループのことです。各クラスタには、マスター管理サーバーとして指定された 1 つのサーバーを組み込む必要があります。

サーバーをクラスタに構成すると、次のことを実行できます。

- すべての Proxy Server を集中して管理する
- 1 つ、または複数の設定ファイルをサーバー間で共有する
- 1 つのマスター管理サーバーから、すべてのサーバーの起動または停止を行う
- 特定のサーバーのアクセスログやエラーログを表示する

クラスタの使用に関するガイドライン

次に、Proxy Server のグループで複数のクラスタを構成する際のガイドラインを示します。

- クラスタの作成に先立って、特定のクラスタに組み込むサーバーをすべてインストールしておく必要があります。
- クラスタ内のすべてのサーバーは、同じ種類 (UNIX または Windows) である必要があります。クラスタは、同じプラットフォームのものである必要があります。
- クラスタ内のすべてのサーバーは、Proxy Server バージョン 4 である必要があります。クラスタへの追加をサポートしているのは Proxy Server バージョン 4 だけです。
- すべての管理サーバーは、同じプロトコル (HTTP または HTTPS) を使用する必要があります。1 つのクラスタ内の 1 つの管理サーバーのプロトコルを変更する場合は、すべての残りの管理サーバーのプロトコルも同様に変更する必要があります。詳細については、126 ページの「サーバー情報の変更」を参照してください。
- すべてのクラスタに固有の管理サーバーは、マスター管理サーバーと同じユーザー名とパスワードを持っている必要があります。分散管理の機能を使用して、各管理サーバーに複数の管理者を設定できます。
- 1 つのクラスタに固有の管理サーバーをマスター管理サーバーとして指定する必要があります。どのサーバーを選択してもかまいません。
- マスター管理サーバーは、クラスタ固有の各管理サーバーにアクセスできる必要があります。マスター管理サーバーは、すべてのインストールされている Sun Java System Web Proxy Server に関する情報を取得します。

クラスタの設定

次に、Proxy Server クラスタを設定するための一般的な手順を示します。

1. クラスタに組み込む Proxy Server をインストールします。

   各クラスタの管理サーバーで、マスター管理サーバーが認証に使用できるユーザー名とパスワードを持っていることを確認します。これを行うには、デフォルトのユーザー名とパスワードを使用するか、または分散管理を設定します。
2. マスター管理サーバーに使用する Proxy Server をインストールします。ユーザー名とパスワードが、手順 1 で設定したものと一致していることを確認します。
3. サーバーをクラスタリストに追加します。

   詳細については、125 ページの「クラスタへのサーバーの追加」を参照してください。

4. リモートサーバーの管理は、「Control Cluster」ページからサーバーマネージャーのインタフェースにアクセスするか、または、同じクラスタ内のサーバーの設定ファイルを別のサーバーにコピーして行います。

クラスタへのサーバーの追加

クラスタに Proxy Server を追加する際は、管理サーバーとポート番号を指定します。追加する管理サーバーが複数のサーバー情報を持っている場合、すべてのサーバーが、そのクラスタに追加されます。後から個々のサーバーを削除できます。

リモート管理サーバーがクラスタの情報を持っている場合、このリモートクラスタの中のサーバーは追加されません。マスター管理サーバーに追加するサーバーは、リモートコンピュータに物理的にインストールされているサーバーだけです。

▼ クラスタにリモートサーバーを追加するには

1 マスター管理サーバーが起動していることを確認します。

2 マスター管理サーバーにアクセスして、「**Cluster**」タブをクリックします。

3 「**Add Server**」リンクをクリックします。

4 リモート管理サーバーが使用するプロトコルを選択します。

- 通常の管理サーバーの場合は HTTP
- セキュリティー保護された管理サーバーの場合は HTTPS

5 `magnus.conf` ファイルに表示されているように、リモート管理サーバーの完全修飾ホスト名を入力します (たとえば、`plaza.example.com`)。

6 リモート管理サーバーのポート番号を入力します。

7 リモート管理サーバーの管理者のユーザー名とパスワードを入力し、「了解」をクリックします。

マスター管理サーバーは、リモートサーバーへの通信を試みます。成功すると、クラスタにサーバーを追加することを確認するプロンプトが表示されます。

注– クラスタ制御を有効にすると、クラスタのマスターがクラスタ内のスレーブごとに、`proxy-`*serverid* `/config/cluster/`*server-name*`/proxy-` *serverid* ディレクトリ内に多数のファイルを作成します。これらのファイルが作成される場所は変更できません。

サーバー情報の変更

管理サーバーの「Cluster」タブにある「Modify Server」オプションは、スレーブサーバー上で、スレーブ管理ポート情報が変更されたあと、その情報を更新するときだけに使用します。クラスタ内のリモート管理サーバーのポート番号を変更したときは、そのクラスタに格納されている管理サーバーの情報も変更する必要があります。スレーブ管理サーバーに対するその他の変更の場合は、いったんそのサーバーを削除し、変更が終わったら、元のようにクラスタに追加する必要があります。

▼ クラスタ内のサーバーに関する情報を変更するには

1. マスター管理サーバーにアクセスして、「**Cluster**」タブをクリックします。
2. 「**Modify Server**」リンクをクリックします。一意のサーバー識別子で識別された、サーバーが表示されます。
3. 変更するサーバーを選択し、変更を行ってから、「了解」をクリックします。

クラスタからのサーバーの削除

▼ クラスタからサーバーを削除するには

1. マスター管理サーバーにアクセスして、「**Cluster**」タブをクリックします。
2. 「**Remove Server**」リンクをクリックします。
3. クラスタから削除するリモートサーバーを選択し、「了解」をクリックします。
   削除したサーバーは、クラスタからアクセスできなくなります。アクセスできるのは、そのサーバーの管理サーバーだけです。

サーバークラスタの制御

Proxy Server を使用すると、次のことを行うことによって、クラスタ内のリモートサーバーを制御することができます。

- サーバーの起動または停止
- アクセスログやエラーログの表示
- 設定ファイルの転送。マスター管理サーバーに Proxy Server のインスタンスが複数ある場合、ファイルは、それらの任意のサーバーから、クラスタに追加された任意のスレーブに転送できます。クラスタは、同じプラットフォームのものである必要があります。クラスタ内のすべてのサーバーは、同じ種類 (UNIX または Windows) である必要があります。設定ファイルを異なるプラットフォームから転送すると、サーバーがハングアップしたり、クラッシュしたりする可能性があります。次の設定ファイルがあります。
  - `server.xml`
  - `magnus.conf`
  - `obj.conf`
  - `mime.types`
  - `socks5.conf`
  - `bu.conf`
  - `icp.conf`
  - `parray.pat`
  - `parent.pat`

▼ クラスタ内のサーバーを制御するには

1. マスター管理サーバーにアクセスして、「**Cluster**」タブをクリックします。
2. 「**Control Cluster**」リンクをクリックします。
3. 制御するサーバーを選択し、必要な選択を行います。

   「Reset」ボタンをクリックすると、いつでも内容が変更前の値にリセットされます。

   - ドロップダウンリストから「Start」、「Stop」、または「Restart」を選択し、「Go」をクリックします。操作を確認するプロンプトが表示されます。
   - ドロップダウンリストから、「View Access」または「View Error」を選択し、ログファイル内で表示する最終行の番号を入力します。「Go」をクリックし、情報を表示します。表示される「Cluster Execution Report」で「View」ボタンをクリックします。
   - 設定ファイルを転送するには、次のことを行います。
     - 転送する設定ファイルを選択します

- ファイルのあるサーバーを選択します
- 「Go」をクリックして、情報を転送します

◆ ◆ ◆ 第 7 章

7 サーバーの詳細設定

この章では、プロキシサーバーのシステム設定と、その設定方法について説明します。システム設定は、プロキシサーバー全体に影響します。この設定には、プロキシサーバーが使用するユーザーアカウントや、待機するポートなどがあります。

この章の内容は次のとおりです。

プロキシサーバーの起動

この節では、さまざまなプラットフォームでプロキシサーバーを起動する方法について説明します。サーバーがインストールされると、サーバーは要求を待機し、受け取ります。

▼ 管理インタフェースからプロキシサーバーを起動するには

1 サーバーマネージャーにアクセスし、「**Preferences**」タブをクリックします。

2 「**Start/Stop Server**」リンクをクリックします。
「Start/Stop Server」ページが表示されます。

3 「**On**」ボタンをクリックします。
サーバーの状態が「Start/Stop Server」ページに表示されます。

UNIX または Linux でプロキシサーバーを起動するには

UNIX または Linux 上では、次のいずれかの方法でプロキシーサーバーを起動できます。

- コマンド行から *server-root* `/proxy-server`*id* に移動し、`./start` と入力してプロキシーサーバーを起動します。
- `start` を使用します。このスクリプトを `init` とともに使用する場合、`/etc/inittab` に起動コマンド `prxy:2: respawn:`*server-root* `/proxy-server`*id*`/start -start -i` を記述する必要があります。

Windows でプロキシサーバーを起動するには

Windows 上では、次のいずれかの方法でプロキシーサーバーを起動できます。

- 「スタート」>「プログラム」>「Sun Microsystems」>「Sun Java System Web Proxy Server *version*」>「Start Proxy Server」を使用します。
- 「コントロール パネル」>「管理ツール」>「サービス」>「Sun Java System Web Proxy Server 4.0 (*id*)」>「開始」を使用します。
- コマンドプロンプトから *server-root* `\proxy-server`*id* に移動し、`startsvr.bat` と入力してプロキシサーバーを起動します。

SSL が有効なサーバーの起動

SSL が有効なサーバーを起動するには、パスワードが必要です。ファイル内にプレーンテキストでパスワードを保存しておくと SSL 有効サーバーを自動的に起動することができますが、これには大きなセキュリティー上のリスクがあります。ファイル

にアクセス可能なユーザーは、SSL が有効なサーバーのパスワードにもアクセスできます。SSL が有効なサーバーのパスワードをプレーンテキストで保存する前に、セキュリティー上のリスクを考慮してください。

サーバーの起動スクリプト、鍵ペアファイル、および鍵パスワードは、ルートが所有しており、または、ルートではないユーザーがサーバーをインストールした場合は、そのユーザーアカウントが所有しています。その所有者のみがそれらに対する読み取りおよび書き込みアクセス権を持ちます。

プロキシサーバーの停止

この節では、さまざまなプラットフォームでプロキシサーバーを停止するいくつかの方法について説明します。

▼ 管理インタフェースからプロキシサーバーを停止するには

1 サーバーマネージャーにアクセスし、「**Preferences**」タブをクリックします。

2 「**Start/Stop Server**」リンクをクリックします。
「Start/Stop Server」ページが表示されます。

3 「**Off**」ボタンをクリックします。
サーバーの状態が「Start/Stop Server」ページに表示されます。

UNIX または Linux でプロキシサーバーを停止するには

UNIX または Linux 上では、次のいずれかの方法でプロキシーサーバーを停止できます。

- コマンド行から *server-root* `/proxy-server`*id* に移動し、`./stop` と入力します。

注 – `etc/inittab` ファイルを使用してサーバーを再起動した場合は、サーバーの停止を試行する前に、`/etc/inittab` からサーバーを起動するための行を削除し、`kill -1 1` を入力する必要があります。そうしないと、サーバーは停止した後で自動的に再起動してしまいます。

- `stop` を使用して、サーバーを完全にシャットダウンします。サービスは、サーバーが再起動されるまで中断されます。`etc/inittab` ファイルで「`respawn`」を使用して自動的に再起動するように設定している場合は、サーバーをシャットダウンする前にプロキシサーバーに関する行を `etc/inittab` から削除する必要があります。この処理を行わない場合、サーバーは自動的に再起動されます。

サーバーをシャットダウンしたあと、シャットダウンプロセスが完了し、状態が「Off」に変更されるまでに数秒かかる場合があります。

システムに障害が発生した場合やオフラインになっている場合、サーバーは停止し、処理中の要求が失われる可能性があります。

注 – サーバーにセキュリティーモジュールがインストールされている場合、サーバーを起動したり、停止したりする前に、適切なパスワードを入力するように要求されます。

Windows でプロキシサーバーを停止するには

Windows 上では、次のいずれかの方法でプロキシーサーバーを停止できます。

- 「スタート」 > 「プログラム」 > 「Sun Microsystems」 > 「Sun Java System Web Proxy Server *version*」 > 「Stop Proxy Server」を使用します。
- コマンドプロンプトから *server-root* `\proxy-server`*id* に移動し、`stopsvr.bat` と入力してプロキシサーバーを停止します。
- 「サービス」ウィンドウの「Sun Java System Proxy Server 4.0 (proxy-server *id*)」サービスを使用します。「コントロール パネル」 > 「管理ツール」 > 「サービス」

プロキシサーバーの再起動

この節では、さまざまなプラットフォームでプロキシサーバーを再起動するいくつかの方法について説明します。

UNIX または Linux でのサーバーの再起動

以下のいずれかの方法で、サーバーを再起動できます。

- サーバーを手動で再起動します。
- `inittab` ファイルからサーバーを自動的に再起動します。

  System V から派生したものではないバージョン (SunOS™ 4.1.3 など) の UNIX または Linux を使用している場合は、`inittab` ファイルを使用できません。

- システムの再起動時に、/etc/rc2.d 内のデーモンでサーバーを自動的に再起動します。

インストールスクリプトでは /etc/rc.local ファイルや /etc/inittab ファイルを編集できないため、テキストエディタでそれらのファイルを編集する必要があります。これらのファイルの編集方法がわからない場合は、システム管理者に問い合わせるか、ご使用のシステムのマニュアルを参照してください。

▼ コマンド行からプロキシサーバーを再起動するには

1. **1024** より小さい番号のポートでサーバーを実行している場合は、**root** としてログインします。**1024** 以上の番号の場合は、**root** として、またはそのサーバーのユーザーアカウントを使用してログインします。

2. コマンド行プロンプトで、以下の行を入力し、**Enter** キーを押します。

   server-root/proxy-server *id*/restart

   ここで *server-root* はサーバーをインストールしたディレクトリです。

   - 行の最後にオプションのパラメータ -i を使用できます。-i オプションは、サーバーを inittab モードで実行します。このモードでは、サーバーのプロセスが強制終了されたかクラッシュした場合に、 inittab がサーバーを再起動します。また、このオプションは、サーバーがバックグラウンド処理に切り替わることを防止します。

inittab を使用してサーバーを再起動するには

/etc/inittab ファイル内に以下のテキストを 1 行で挿入します。

```
prxy:23:respawn:server-root /proxy-serverid/start -start -i
```

ここで *server-root* はサーバーをインストールしたディレクトリ、proxy-server*id* はサーバーのディレクトリです。

-i オプションは、サーバーがバックグラウンド処理に切り替わることを防止します。

この行は、サーバーを停止する前に削除する必要があります。

システムの rc (実行制御) スクリプトを使用してサーバーを再起動するには

/etc/rc.local、または使用しているシステムのそれに相当するスクリプトを使用する場合は、/etc/rc.local 内に以下の行を追加します。

server-root/proxy-server *id*/start

server-root を、サーバーがインストールされているディレクトリに変更します。

Windows でのサーバーの再起動

「サービス」コントロールパネルを使用するか、次の作業を実行してサーバーを再起動できます。

▼ Windows でサーバーを再起動するには

1. 「コントロール パネル」>「管理ツール」>「サービス」 を使用します。
2. サービスのリストから「**Sun Java System Web Proxy Server 4.0 (proxy-server *id*)**」を選択します。
3. 「**Properties**」ウィンドウで「**Startup**」タイプを「**Automatic**」に変更します。コンピュータが起動やリブートするたびにシステムがサーバーを起動するようになります。
4. 「了解」をクリックします。

終了タイムアウトの設定

サーバーをオフにすると、新しい接続の受け入れは停止します。その後、サーバーはすべての未処理の接続処理が完了するまで待ちます。タイムアウトになるまでサーバーが待機する時間は、`magnus.conf` ファイルで設定できます。デフォルトでは、この値は 30 秒に設定されています。この値を変更するには、次の行を `magnus.conf` ファイルに追加します。

`TerminateTimeout` *seconds*

seconds は、タイムアウトになるまでサーバーが待機する秒数を表します。

この値を変更することによる利点は、接続の処理が完了するまでサーバーが待機する時間が、より長くなることです。ただし、サーバーは応答していないクライアントに接続されていることがあるため、終了タイムアウト値を大きくすると、サーバーのシャットダウンにかかる時間が長くなる可能性があります。

サーバー設定の表示

インストール中に、プロキシサーバーの一部の設定を行います。これらを始めとするシステム設定は、サーバーマネージャーで表示できます。「View Server Settings」ページには、プロキシサーバーのすべての設定が一覧表示されます。このページには、保存または適用していない変更があるかどうかも表示されます。適用していない変更がある場合は、変更を保存し、新しい設定を使用できるようにプロキシサーバーを再起動してください。

サーバーの設定には、テクニカルとコンテンツの 2 つがあります。サーバーのコンテンツの設定は、サーバーの設定内容によって異なります。通常、プロキシの場合は、すべてのテンプレート、URL マッピング、およびアクセス制御が一覧表示されます。個々のテンプレートについては、テンプレート名、その正規表現、およびキャッシュ設定などのテンプレートの設定が「View Server Settings」ページに一覧表示されます。

プロキシサーバーのテクニカルの設定は `magnus.conf` ファイルと `server.xml` ファイル、コンテンツの設定は `obj.conf` ファイルで行います。これらのファイルは、サーバーのルートディレクトリのサブディレクトリ `proxy-`*id*`/config` にあります。

▼ プロキシサーバーの設定を表示するには

1. サーバーマネージャーにアクセスし、「**Preferences**」タブをクリックします。
2. 「**View Server Settings**」リンクをクリックします。
   「View Server Settings」ページが表示されます。

設定ファイルのバックアップの表示と復元

次の設定ファイルのバックアップコピーを表示または復元することができます。`server.xml`、 `magnus.conf`、 `obj.conf`、 `mime types`、 `server.xml.clfilter`、 `magnus.conf.clfilter`、 `obj.conf.clfilter`、 `socks5.conf`、 `bu.conf`、 `icp.conf`、 `parray.pat`、 `parent.pat`、 `proxy-id.acl`。この機能を使用すると、現在の設定で問題が発生したときに以前の設定に戻すことができます。たとえば、プロキシの設定に複数の変更を行ったところ、プロキシが予期したとおりに動作しない場合 (たとえば、ユーザーの URL へのアクセスは拒否されるが、プロキシは要求を処理するなど)、以前の設定に戻して、設定の変更をやり直すことができます。

▼ 以前の設定を表示するには

1 サーバーマネージャーにアクセスし、「**Preferences**」タブをクリックします。

2 「**Restore Configuration**」リンクをクリックします。
「Restore Configuration」ページが表示されます。このページには、日時順に以前の設定がすべて一覧表示されます。

3 「**View**」リンクをクリックして、特定バージョンのテクニカルおよびコンテンツの設定に関するリストを表示します。

▼ 設定ファイルのバックアップコピーを復元するには

1 サーバーマネージャーにアクセスし、「**Preferences**」タブをクリックします。

2 「**Restore Configuration**」リンクをクリックします。
「Restore Configuration」ページが表示されます。このページには、日時順に以前の設定がすべて一覧表示されます。

3 復元するバージョンの「**Restore**」リンクをクリックします。
すべてのファイルを特定時の状態に復元する場合は、表の左の列にある「Restore to*time*」リンクをクリックします。*time* は、復元する日時を示します。

▼ 表示されるバックアップの数を設定するには

1 サーバーマネージャーにアクセスし、「**Preferences**」タブをクリックします。

2 「**Restore Configuration**」リンクをクリックします。
「Restore Configuration」ページが表示されます。

3 「**Set Number Of Sets Of Backups**」フィールドに、表示するバックアップの数を入力します。

4 「変更」ボタンをクリックします。

システムの詳細設定

「Configure System Preferences」ページを使用すると、サーバーの基本部分を設定または変更できます。このページでは次を実行できます。

- サーバーユーザー、プロセス数、待機キューサイズ、プロキシのタイムアウト、プロキシサーバーの中断後のタイムアウトなどの変更
- DNS、ICP、プロキシ配列、および親配列の有効化

詳細設定のオプションは次のとおりです。

- **Server User**。サーバーユーザーは、プロキシが使用するユーザーアカウントです。プロキシサーバーのユーザーとして入力したユーザー名は、すでに、通常のユーザーアカウントとして存在しているはずです。サーバーの起動時は、このユーザーによって起動されたものとして稼動します。

  ユーザーアカウントを新規に作成しない場合は、同じホスト上で稼動中の別のサーバーで使用されているアカウントを選択するか、または UNIX プロキシを実行しているときはユーザー `nobody` を選択することができます。ただし、システムによっては、ユーザー `nobody` はファイルを所有できてもプログラムを実行できないため、プロキシユーザー名として適していない場合があります。

  UNIX システムでは、プロキシによって開始されるすべてのプロセスが、サーバーユーザーアカウントに割り当てられます。
- **Processes**。「Processes」フィールドには、要求を処理するために利用できるプロセス数が表示されます。デフォルトでは、この値は 1 です。必要のないかぎり、この設定は変更しないでください。
- **Listen Queue Size**。「Listen Queue Size」フィールドでは、待機ソケットで保留にする接続の最大数を指定します。
- **DNS**。ドメイン名サービス (Domain Name Service、DNS) は、IP アドレスをホスト名に復元します。Web ブラウザがサーバーに接続するとき、サーバーはクライアントの IP アドレス (たとえば `198.18.251.30`) だけを取得します。サーバーは、`www.example.com` などのホスト名情報を保持していません。アクセスログやアクセス制御の場合は、サーバーは IP アドレスをホスト名に解決できます。「System Preferences」ページでは、IP アドレスをホスト名に解決するかどうかをサーバーに知らせることができます。
- **ICP**。Internet Cache Protocol (ICP) は、メッセージの送受に使用するプロトコルで、キャッシュ間の対話を可能にします。キャッシュでは ICP を使用して、キャッシュされた URL の存在と、それらの URL を取得するための最適な場所について、クエリーの送信と応答を行います。ICP は、「Configure System Preferences」ページで有効にできます。ICP については、285 ページの「隣接 ICP を経由したルーティング」を参照してください。
- **Proxy Array**。プロキシ配列とは、分散キャッシュ処理を行うために 1 つのキャッシュとして機能する、プロキシの配列です。「Configure System Preferences」ページでプロキシ配列オプションを有効にすると、設定中のプロキ

シサーバーは、プロキシ配列のメンバーになり、配列内のその他すべてのメンバーは、その兄弟関係となります。プロキシ配列の使用については、293 ページの「プロキシ配列を経由したルーティング」を参照してください。

- **Parent Array**。親配列とは、プロキシまたはプロキシ配列メンバーがルーティングに使用するプロキシ配列のことです。そのため、プロキシが上流のプロキシ配列をルーティングに使用してリモートサーバーにアクセスする場合、その上流のプロキシ配列は親配列と見なされます。プロキシサーバーでの親配列の使用については、306 ページの「親配列を経由したルーティング」を参照してください。
- **Proxy Timeout**。プロキシタイムアウトは、プロキシサーバーが要求をタイムアウトする前に、リモートサーバーからネットワークデータパケットを連続して受信する間隔の最大時間です。プロキシタイムアウトのデフォルト値は 5 分です。

注 – リモートサーバーがサーバープッシュを使用し、ページ間の遅延がプロキシタイムアウトよりも長くなると、転送完了前に接続が終了する可能性があります。代わりに、クライアントプルを使用してください。クライアントプルでは、複数の要求をプロキシに送信します。

▼ システムの詳細設定を変更するには

1. サーバーマネージャーにアクセスし、「**Preferences**」タブをクリックします。
2. 「**Configure System Preferences**」リンクをクリックします。
   「Configure System Preferences」ページが表示されます。
3. **optionsd** を変更して、「了解」をクリックします。
4. 「**Restart Required**」をクリックします。
   「Apply Changes」ページが表示されます。
5. 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

プロキシサーバーの調整

「Tune Proxy」ページでは、デフォルトパラメータを変更し、プロキシサーバーのパフォーマンスを調整することができます。

▼ デフォルトの調整パラメータを変更するには

1. サーバーマネージャーにアクセスし、「**Preferences**」タブをクリックします。
2. 「**Tune Proxy**」リンクをクリックします。
   「Tune Proxy」ページが表示されます。
3. **(**オプション**)** 長いファイル名が表示でき、ファイル名の切り詰めが少なくなるよう、**FTP** リストの幅を変更することもできます。
   デフォルトの幅は、80 文字です。
4. 「了解」をクリックします。
5. 「**Restart Required**」をクリックします。
   「Apply Changes」ページが表示されます。
6. 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

待機ソケットの追加と編集

サーバーで要求を処理するには、待機ソケットを介して要求を受け入れてから、適切なサーバーにその要求を送信する必要があります。プロキシサーバーをインストールすると、ls1 という待機ソケットが自動的に作成されます。この待機ソケットには、`0.0.0.0` の IP アドレスと、インストール時にプロキシサーバーのポート番号として指定したポート番号が割り当てられます。デフォルトの待機ソケットは削除できません。

- 「**General**」
  - 「**Listen Socket ID**」**:**待機ソケットの内部名。待機ソケットの作成後は、この名前を変更できません。
  - 「**IP Address**」**:**待機ソケットの IP アドレス。このアドレスはドットで区切り、または IPv6 の表記法で指定できます。`0.0.0.0`、`any`、`ANY` 、または `INADDR_ANY` (すべての IP アドレス) を指定することもできます。
  - 「**Port**」**:**待機ソケットを作成するポート番号。有効な値は 1 ～ 65535 です。UNIX では、ポート 1 ～ 1024 で待機するソケットを作成するには、スーパーユーザー権限が必要です。SSL 待機ソケットを設定し、ポート 443 で待機するようにします。
  - 「**Server Name**」**:**この待機ソケットのデフォルトサーバー。

「**Security**」

セキュリティーが無効になっている場合は、次のパラメータのみが表示されます。

- 「**Security**」:選択した待機ソケットのセキュリティーを有効または無効にします。

セキュリティーが有効になっている場合は、次のパラメータが表示されます。

- 「**Security**」:選択した待機ソケットのセキュリティーを有効または無効にします。
  - 「**Server Certificate Name**」: この待機ソケットに使用するインストール済みの証明書をドロップダウンリストから選択します。
  - 「**Client Authentication**」:この待機ソケットにクライアント認証が必要かどうかを指定します。このパラメータはデフォルトで「Optional」になっています。
  - 「**SSL Version 2**」: SSL バージョン 2 を有効または無効にします。この設定は、デフォルトで有効になっています。
  - 「**SSL Version 2 Ciphers**」:この方式群に含まれるすべての暗号化を一覧表示します。チェックボックスを選択または選択解除して、編集中の待機ソケットに対して有効にする暗号化を選択します。デフォルトバージョンでは、選択解除されています。
  - 「**SSL Version 3**」: SSL バージョン 3 を有効または無効にします。この設定は、デフォルトで有効になっています。
  - 「**TLS**」:暗号化通信で TLS (Transport Layer Security) プロトコルを有効または無効にします。デフォルトでは、有効になっています。
  - 「**TLS Rollback**」: TLS ロールバックを有効または無効にします。「TLS Rollback」を無効にすると、通信がバージョンロールバック攻撃を受けやすくなるので注意してください。デフォルトでは、有効になっています。
  - 「**SSL Version 3 and TLS Ciphers**」:この方式群に含まれるすべての暗号化を一覧表示します。チェックボックスを選択または選択解除して、編集中の待機ソケットに対して有効にする暗号化を選択します。デフォルトバージョンでは、選択されています。

「**Advanced**」

- 「**Number Of Acceptor Threads**」: 待機ソケットの Acceptor Thread の数。推奨される値は、マシンのプロセッサの数です。デフォルト値は 1 で、値は 1 〜 1024 です。

  「**Protocol Family**」: ソケットのファミリ・タイプ。有効な値は `inet`、`inet6` 、および `nca` です。IPv6 待機ソケットを使用する場合は、`inet6` の値を使用します。Solaris Network Cache and Accelerator を使用する場合は、`nca` を指定します。

待機ソケットの追加、編集、および削除は、サーバーマネージャーの「Add Listen Socket」および「Edit Listen Sockets」ページを使用して実行できます。

待機ソケットのセキュリティーは、必要な証明書がインストールされた後、ドロップダウンボックスに「Disabled」のみが表示されるまで、オプションとして「Enabled」にできます。

ここでは、次の内容について説明します。

- 108 ページの「「プロキシがクライアントを認証」シナリオを設定するには」
- 110 ページの「「コンテンツサーバーがプロキシを認証」シナリオを設定するには」
- 110 ページの「「プロキシがクライアントを認証、かつコンテンツサーバーがプロキシを認証」シナリオを設定するには」

▼ 待機ソケットを追加するには

1. サーバーマネージャーにアクセスし、「**Preferences**」タブをクリックします。
2. 「**Add Listen Socket**」リンクをクリックします。
   「Add Listen Socket」ページが表示されます。
3. 待機ソケットの内部名を入力します。
   待機ソケットの作成後は、この名前を変更できません。
4. 待機ソケットの **IP** アドレスを指定します。
   IP アドレスはドットで区切り、または IPv6 の表記法で指定できます。`0.0.0.0`、`any`、`ANY` 、または `INADDR_ANY` (すべての IP アドレス) を指定することもできます。
5. 待機ソケットを作成するポート番号を指定します。有効な値は 1 ～ 65535 です。
   UNIX の場合、ポート 1 ～ 1024 で待機するソケットを作成するには、スーパーユーザー特権が必要です。SSL 待機ソケットを設定し、ポート 443 で待機するようにします。
6. サーバーがクライアントに送信するすべての **URL** のホスト名セクションで使用されるサーバー名を指定します。
   この設定は、サーバーが自動生成する URL には影響しますが、サーバーに格納されているディレクトリおよびファイルの URL には影響しません。サーバーがエイリアスを使用する場合は、この名前もエイリアス名にする必要があります。
7. ドロップダウンリストから、待機ソケットでセキュリティーを有効と無効のどちらにするかを指定します。
8. 「了解」をクリックします。

9 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

10 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

▼ 待機ソケットを編集するには

1 サーバーマネージャーにアクセスし、「**Preferences**」タブをクリックします。

2 「**Edit Listen Sockets**」リンクをクリックします。
「Edit Listen Sockets」ページが表示されます。

3 「**Configured Sockets**」表で、編集する待機ソケットのリンクをクリックします。
「Edit Listen Sockets」ページが表示されます。

4 必要に応じて、オプションを変更します。
オプションの説明は、この節の最初を参照してください。

5 「了解」をクリックします。

6 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

7 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

▼ 待機ソケットを削除するには

1 サーバーマネージャーにアクセスし、「**Preferences**」タブをクリックします。

2 「**Edit Listen Sockets**」リンクをクリックします。

3 削除する待機ソケットの隣のチェックボックスを選択し、「了解」をクリックします。
削除を確認するプロンプトが表示されます。任意の待機ソケットを削除できますが、そのインスタンスに待機ソケットが 1 つしかない場合は削除できません。

4 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

5 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

ディレクトリサービスの選択

「Select Directory Services」ページには、指定されたプロキシサーバーインスタンスのディレクトリサービスがすべて一覧表示されます。このページでは、特定のプロキシサーバーインスタンスに使用できるディレクトリサービスを選択できます。詳細は、49 ページの「ディレクトリサービスの設定」を参照してください。

▼ ディレクトリサービスを選択するには

1. サーバーマネージャーにアクセスし、「**Preferences**」タブをクリックします。
2. 「**Select Directory Services**」リンクをクリックします。
   「Select Directory Services」ページに、指定されたプロキシサーバーインスタンスのディレクトリサービスがすべて表示されます。
3. リストからディレクトリサービスを選択します。
4. 「了解」をクリックします。
5. 「**Restart Required**」をクリックします。
   「Apply Changes」ページが表示されます。

MIME タイプ

Multi-purpose Internet Mail Extension (MIME) タイプは、マルチメディア電子メールおよびメッセージングの標準です。MIME タイプに応じてファイルをフィルタできるようにするため、プロキシサーバーでは、サーバーで使用する新しい MIME タイプを作成するページを用意しています。プロキシでは、新しいタイプを `mime.types` ファイルに追加します。MIME タイプに基づいたファイルのブロックについては、317 ページの「MIME タイプによるフィルタリング」を参照してください。

この節では、MIME タイプを作成、編集、または削除する方法について説明します。

MIME タイプの作成

▼ MIME タイプを作成するには

1 サーバーマネージャーにアクセスし、「**Preferences**」タブをクリックします。

2 「**Create/Edit MIME Types**」リンクをクリックします。
「Create/Edit MIME Types」ページが表示され、プロキシの `mime.types` ファイルに記載された MIME タイプがすべて表示されます。

3 ドロップダウンリストから、**MIME** タイプのカテゴリを指定します。`type`、**enc**、または`lang`を指定できます。`type` はファイルまたはアプリケーションタイプ、`enc` は圧縮に使用するエンコーディング、`lang` は言語のエンコーディングです。
カテゴリについては、オンラインヘルプを参照してください。

4 **HTTP** ヘッダーに表示されるコンテンツタイプを指定します。

5 ファイルサフィックスを指定します。
ファイルサフィックスとは、MIME タイプにマップされるファイル拡張子です。複数の拡張子を指定するには、エントリをコンマで区切ります。ファイル拡張子は、一意である必要があります。つまり、1 つのファイル拡張子を 2 つの MIME タイプにマップしないでください。

6 「**New**」ボタンをクリックして、**MIME** タイプを追加します。

▼ MIME タイプを編集するには

1 サーバーマネージャーにアクセスし、「**Preferences**」タブをクリックします。

2 「**Create/Edit MIME Types**」リンクをクリックします。
「Create/Edit MIME Types」ページが表示され、プロキシの `mime.types` ファイルに記載された MIME タイプがすべて表示されます。

3 編集する **MIME** タイプの「**Edit**」リンクをクリックします。

4 必要に応じて設定を変更します。「**Change MIME Type**」ボタンをクリックします。

▼ MIME タイプを削除するには

1 サーバーマネージャーにアクセスし、「**Preferences**」タブをクリックします。

2 「**Create/Edit MIME Types**」リンクをクリックします。
「Create/Edit MIME Types」ページが表示され、プロキシの `mime.types` ファイルに記載された MIME タイプがすべて表示されます。

3 削除する **MIME** タイプの「**Remove**」リンクをクリックします。

アクセス制御の管理

「Administer Access Control」ページを使用すると、アクセス制御リスト (Access Control List、ACL) を管理することができます。ACL では、サーバーにアクセスできるクライアントを制御できます。ACL を利用すると、サーバーの一部に対するアクセスを許可または拒否する特定のユーザー、グループ、またはホストを選別できます。また、有効なユーザーまたはグループだけがサーバーの一部にアクセスできるように認証を設定することができます。アクセス制御については、第 8 章を参照してください。

▼ アクセス制御リストを管理するには

1 サーバーマネージャーにアクセスし、「**Preferences**」タブをクリックします。

2 「**Administer Access Control**」リンクをクリックします。
「Administer Access Control」ページが表示されます。

3 リソースや既存の **ACL** を選択するか、**ACL** 名を入力し、「**Edit**」ボタンをクリックします。
「Access Control Rules for」ページが表示されます。

4 変更を行い、「**Submit**」をクリックします。
アクセス制御の設定については、第 8 章の「サーバーインスタンスに対するアクセス制御の設定」を参照してください。

ACL キャッシュの設定

「Configure ACL Cache」ページを使用して、プロキシ認証キャッシュの有効化または無効化、プロキシ認証キャッシュディレクトリの設定、キャッシュテーブルサイズの設定、およびエントリの有効期限の設定を行います。

▼ ACL キャッシュを設定するには

1. サーバーマネージャーにアクセスし、「**Preferences**」タブをクリックします。
2. 「**Configure ACL Cache**」リンクをクリックします。
   「Configure ACL Cache」ページが表示されます。
3. プロキシ認証キャッシュを有効または無効にします。
4. 「**Proxy Auth User Cache Size**」ドロップダウンリストから、ユーザーキャッシュ内のユーザー数を選択します。
   デフォルトサイズは 200 です。
5. 「**Proxy Auth Group Cache Size**」ドロップダウンリストから、**1** つの **UID** またはキャッシュエントリでキャッシュできるグループ **ID** の数を選択します。
   デフォルトサイズは 4 です。
6. キャッシュエントリの有効期限が切れるまでの秒数を選択します。
   キャッシュのエントリが参照されるたびにその経過時間が計算され、この値と照合されます。経過時間が「Proxy Auth Cache Expiration」と同じか、超えている場合、このエントリは使用されません。この値が 0 に設定されている場合、キャッシュはオフです。

   この値に大きな値を使用する場合、LDAP エントリを変更したときにプロキシサーバーの再起動が必要となる可能性があります。たとえば、この値を 120 秒に設定した場合は、プロキシサーバーと LDAP サーバーの同期が 2 分間にわたって取られない可能性があります。LDAP エントリが頻繁に変更されないようであれば、大きな値を使用します。デフォルトの有効期限値は 2 分です。
7. 「了解」をクリックします。
8. 「**Restart Required**」をクリックします。
   「Apply Changes」ページが表示されます。
9. 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

DNS キャッシュについて

プロキシサーバーでは、DNS ホスト名を IP アドレスに解決する際にプロキシサーバーが実行する DNS 検索の数を減らすため、DNS キャッシュをサポートしています。

DNS キャッシュの設定

「Configure DNS Cache」ページを使用して、DNS キャッシュの有効化または無効化、DNS キャッシュのサイズの設定、DNS キャッシュエントリの有効期限の設定、ネガティブ DNS キャッシュの有効化または無効化を行います。

▼ DNS キャッシュを設定するには

1. サーバーマネージャーにアクセスし、「**Preferences**」タブをクリックします。
2. 「**Configure DNS Cache**」リンクをクリックします。
   「Configure DNS Cache」ページが表示されます。
3. **DNS** キャッシュを有効または無効にします。
4. 「**DNS Cache Size**」ドロップダウンリストから、**DNS** キャッシュ内に格納できるエントリ数を選択します。
   デフォルトサイズは 1024 です。
5. **DNS** キャッシュの有効期限を設定します。
   プロキシサーバーでは、DNS キャッシュエントリがあらかじめ設定された有効期限に達すると、キャッシュからパージします。デフォルトの DNS 有効期限値は 20 分です。
6. ホスト名が見つからないときのエラーのキャッシュを有効または無効にします。
7. 「了解」をクリックします。
8. 「**Restart Required**」をクリックします。
   「Apply Changes」ページが表示されます。
9. 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

DNS サブドメインの設定

URL には、多くのレベルのサブドメインを持つホスト名を含むものがあります。この場合、最初の DNS サーバーがホスト名を解決できない場合に、プロキシサーバーでは DNS チェックに時間がかかることがあります。「host not found」というメッセージをクライアントに返す前に、プロキシサーバーがチェックするレベル数を設定できます。

たとえば、クライアントが `http://www.sj.ca.example.com/index.html` を要求した場合、プロキシでこのホストを IP アドレスに解決するのに時間がかかる可能性があります。なぜなら、このホストコンピューターの IP アドレスを取得するには、4 つの DNS サーバーにアクセスしなければならないからです。これらの検索には時間がかかることがあるため、プロキシで一定数以上の DNS サーバーを使用しなければならない場合に、プロキシサーバーが IP アドレスの検索を終了するように設定することができます。

▼ プロキシが検索するサブドメインのレベル数を設定するには

1. サーバーマネージャーにアクセスし、「**Preferences**」タブをクリックします。
2. 「**Configure DNS Subdomains**」リンクをクリックします。
   「Configure DNS Subdomains」ページが表示されます。
3. ドロップダウンリストからリソースを選択するか、正規表現を指定します。
4. 「**Local Subdomain Depth**」ドロップダウンリストから、レベル数を選択します。
5. 「了解」をクリックします。
6. 「**Restart Required**」をクリックします。
   「Apply Changes」ページが表示されます。
7. 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

HTTP キープアライブの設定

「Configure HTTP Client」ページを使用して、プロキシサーバーのキープアライブを有効にします。

キープアライブとは、クライアントがオープン接続をすぐに再利用できるように、要求の完了後も接続を開いたままにしておく、TCP/IP の機能です。デフォルトでは、プロキシはキープアライブ接続を使用しませんが、システムによっては、キープアライブ機能を使用して、プロキシのパフォーマンスを向上させることができるものがあります。

Web 上での通常のクライアントサーバートランザクションでは、クライアントは、サーバーに対して複数の接続を行い、複数のドキュメントを要求することができます。たとえば、クライアントが複数のグラフィック画像のある Web ページを要求する場合、そのクライアントは、各グラフィックファイルに対して個別の要求を行う必要があります。接続の再確立には、時間がかかります。このため、キープアライブパケットが役立ちます。

▼ HTTP キープアライブを設定するには

1. サーバーマネージャーにアクセスし、「**Preferences**」タブをクリックします。
2. 「**Configure HTTP Client**」リンクをクリックします。
   「Configure HTTP Client」ページを表示します。
3. ドロップダウンリストからリソースを選択します。
   プロキシサーバーでキープアライブを設定する HTTP または HTTPS リソースを選択するか、正規表現を指定します。
4. **HTTP** クライアントが持続的接続を使用するかどうかについて、適切な「**Keep Alive**」オプションを選択して指定します。
5. 「**Keep Alive Timeout**」フィールドに、持続的接続を開いたままにしておく最大の秒数を指定します。
   デフォルト値は 29 です。
6. 適切な「**Persistent Connection Reuse**」オプションを選択することで、すべての種類の要求に対して、**HTTP** クライアントが既存の持続的接続を再利用できるようにするかどうかを指定します。
   デフォルト値はオフで、GET 以外の要求やボディーを含む要求に対しては、持続的接続を再利用できません。

7 「**HTTP Version String**」フィールドに、**HTTP** プロトコルバージョン文字列を指定します。
プロトコルの相互運用について特定の問題が発生しないかぎり、このパラメータは指定しないでください。

8 「**Proxy Agent Header**」フィールドに、プロキシサーバーの製品名とバージョンを入力します。

9 「**SSL Client Certificate Nickname**」フィールドに、リモートサーバーに表示するクライアント証明書のニックネームを入力します。

10 適切な「**SSL Server Certificate Validation**」オプションを選択し、プロキシサーバーがリモートサーバーに表示された証明書を検証する必要があるかどうかを示します。

11 「了解」をクリックします。

12 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

13 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

◆ ◆ ◆ 第 8 章

サーバーへのアクセス制御

この章では、管理サーバーおよび Proxy Server によって処理されるデータへのアクセス制御の方法について説明します。サーバーにより処理されるすべてのデータ、またはサーバーがサービスを提供する特定の URL のアクセスを制限できます。たとえば、特定の URL へのアクセスを特定のユーザーに限定したり、特定のユーザー以外のすべてのユーザーがファイルを表示できるように指定できます。すべてのクライアントに HTTP の URL へのアクセスを許可し、FTP へは限定されたアクセスのみを許可することもできます。また、複数の内部 Web サーバーにサービスを提供している Proxy Server があり、それらのサーバーのいずれかに格納された極秘調査プロジェクトへは特定のユーザーのみがアクセスできるようにしたい場合などに、ホスト名またはドメイン名に基づいて URL を制限することもできます。

管理サーバーでアクセス制御を使用するには、分散管理を有効にして、LDAP データベースに管理グループを設定する必要があります。この章の内容は、それらの作業が済んでいることを前提としています。

この章の内容は次のとおりです。

アクセス制御とは

アクセス制御により、Proxy Server にアクセスできるユーザー、およびアクセス可能なサーバーの部分を指定することができます。サーバーへのアクセスの制御は、サーバー全体に対して、またはディレクトリ、ファイル、ファイルタイプなどのサーバーの一部に対して行うことができます。受信した要求が評価される場合、アクセス制御エントリ (Access Control Entry、ACE) と呼ばれる規則の階層に基づいてア

クセスが決定されます。Proxy Server は一致するエントリを検索して、アクセスを許可するか、拒否するかを決定します。各 ACE は、サーバーが階層内の次のエントリに進むべきかどうかを指定します。ACE の集合は、アクセス制御リスト (ACL) と呼ばれます。要求を受信すると、`obj.conf` ファイルでアクセスの決定に使用される ACL の参照がチェックされます。デフォルトでは、サーバーには複数の ACL が含まれる 1 つの ACL ファイルがあります。

アクセスは次の内容に基づいて許可または拒否されます。

- 要求を送信したユーザー (ユーザー - グループ)
- 要求の送信元 (ホスト - IP)
- 要求が発生した日時 (時刻など)
- 使用されている接続のタイプ (SSL)

ここでは、次の内容について説明します。

- 152 ページの「ユーザー - グループのアクセス制御」
- 159 ページの「ホスト - IP のアクセス制御」
- 160 ページの「アクセス制御ファイルの使用」
- 161 ページの「ACL ユーザーキャッシュの設定」
- 161 ページの「クライアント証明書によるアクセス制御」

ユーザー - グループのアクセス制御

特定のユーザーまたはグループに対して、サーバーへのアクセスを制限することができます。ユーザー - グループのアクセス制御を設定した場合、サーバーにアクセスする前に、ユーザーはユーザー名とパスワードの入力を求められます。サーバーは、クライアント証明書の情報、またはクライアント証明書自体を、ディレクトリサーバーのエントリと比較します。

管理サーバーでは、基本認証だけを使用します。管理サーバーでクライアント認証を要求する場合、`obj.conf` の ACL ファイルを手動で編集し、認証方法を SSL に変更する必要があります。

ユーザー - グループ認証は、サーバーに設定されているディレクトリサービスによって実行されます。詳細は、49 ページの「ディレクトリサービスの設定」を参照してください。

ディレクトリサービスがアクセス制御の実装に使用する情報は、次のソースのいずれかから入手できます。

- 内部フラットファイルタイプのデータベース
- 外部 LDAP データベース

サーバーは、外部 LDAP ベースのディレクトリサービスを使用するとき、サーバーインスタンスに対して次のタイプのユーザーグループ認証方法をサポートします。

- デフォルト
- 基本
- SSL
- ダイジェスト
- その他

サーバーが内部ファイルベースのディレクトリサービスを使用するときに、サーバーインスタンスに対してサポートされるユーザー - グループ認証方法には、次のものがあります。

- デフォルト
- 基本
- ダイジェスト

ユーザー - グループ認証を設定した場合、ユーザーはアクセスする前に、ユーザー自身の認証を求められます。認証の際、ユーザーはユーザー名とパスワードの入力、クライアント証明書の使用、またはダイジェスト認証プラグインを使用することによってユーザー自身の識別情報を証明します。クライアント証明書を使用する場合、暗号化が必要になります。

デフォルト認証

デフォルト認証は、推奨される方法です。デフォルト設定では、`obj.conf` ファイルで指定したデフォルトの方法を使用します。`obj.conf` で方法が設定されていない場合は、「Basic」を使用します。「Default」を選択した場合、ACL 規則によって ACL ファイル内の方法が指定されることはありません。「Default」を選択すると、`obj.conf` ファイル内の 1 行を編集することによって、すべての ACL の方法を簡単に変更できます。

基本認証

基本認証を選択した場合、サーバーにアクセスするためにユーザーはユーザー名とパスワードの入力を求められます。基本認証がデフォルトの設定です。Sun Java System Directory Server などの LDAP データベース、またはファイルにユーザーとグループのリストを作成して格納する必要があります。Proxy Server とは別のサーバールートにインストールされたディレクトリサーバー、またはリモートコンピュータにインストールされたディレクトリサーバーを使用する必要があります。

ユーザーがユーザー - グループ認証を行うリソースへアクセスしようとすると、ユーザー名とパスワードの入力を求められます。サーバーで暗号化が設定されている (SSL が有効) かどうかに応じて、サーバーはこの情報を暗号化された状態、または暗号化されていない状態で受信します。

注 – SSL 暗号化なしで基本認証を使用する場合、暗号化されていないユーザー名とパスワードがネットワークを経由して送信されます。ネットワークパケットは傍受される可能性があり、ユーザー名とパスワードが不正に知られてしまう可能性があります。基本認証は、SSL 暗号化とホスト - IP 認証のどちらか、またはその両方と組み合わせた場合にもっとも効果的です。ダイジェスト認証を使用しても、この問題を回避できます。

認証に成功した場合、ユーザーには要求されたリソースが表示されます。ユーザー名またはパスワードが無効な場合は、アクセスが拒否されたことを示すメッセージが発行されます。

承認されていないユーザーから受信するメッセージはカスタマイズできます。詳細は、174 ページの「アクセスが拒否された場合の応答」を参照してください。

SSL 認証

サーバーは、次の 2 つの方法で、セキュリティー証明書付きのユーザーの識別情報を確認できます。

- クライアント証明書の情報を識別情報の証明として使用する
- LDAP ディレクトリで発行されたクライアント証明書を確認する (追加)

クライアントの認証で証明書の情報を使用するようにサーバーを設定した場合、サーバーは次の処理を実行します。

- 信頼できる CA (認証局) から発行された証明書かどうかを確認します。そうでない場合、認証は失敗し、トランザクションが終了します。クライアント認証を有効にする方法については、92 ページの「セキュリティーに関する詳細設定」を参照してください。
- 証明書が信頼できる CA から発行されたものである場合は、`certmap.conf` ファイルを使用して、証明書をユーザーのエントリにマップします。証明書マッピングファイルの設定方法については、111 ページの「`certmap.conf` ファイルの使用」を参照してください。
- 証明書が正しくマップされている場合は、そのユーザーに対して指定されている ACL 規則を確認します。証明書が正しくマップされている場合でも、ACL 規則によってユーザーのアクセスが拒否される可能性もあります。

特定のリソースへのアクセスを制御するためにクライアント認証を要求することは、サーバーへの接続のすべてに対してクライアント認証を要求することとは異なります。すべての接続に対してクライアント認証を要求するようにサーバーを設定した場合、クライアントは信頼できる CA によって発行された有効な証明書のみを提示する必要があります。ユーザーとグループの認証に SSL メソッドを使用するようにサーバーを設定した場合、次のことが行われます。

- クライアントは信頼できる CA によって発行された有効な証明書を提示する

- 証明書はLDAP内の有効なユーザーにマッピングされている
- アクセス制御リストで、適切に評価される

アクセス制御を使用してクライアント認証を要求する場合、Proxy ServerでSSL暗号化方式を有効にする必要があります。SSLの有効化については、第5章を参照してください。

SSLで認証されるリソースにアクセスするには、Proxy Serverにより信頼できるCAから、クライアント証明書が発行されている必要があります。ブラウザのクライアント証明書とディレクトリサーバーのクライアント証明書を比較するようにProxy Serverの`certmap.conf`ファイルが設定されている場合、クライアント証明書はディレクトリサーバーで発行されている必要があります。ただし、証明書から選択した情報とディレクトリサーバーのエントリだけを比較するように、`certmap.conf`ファイルを設定することもできます。たとえば、ブラウザ証明書のユーザーIDおよび電子メールアドレスとディレクトリサーバーのエントリだけを比較するように、`certmap.conf`ファイルを設定することができます。`certmap.conf`と証明書マッピングについては、第5章を参照してください。『Sun Java System Web Proxy Server 4.0.4 Configuration File Reference』も参照してください。

ダイジェスト認証

Proxy Serverは、LDAPベースまたはファイルベースのいずれかのディレクトリサービスを使用して、ダイジェスト認証を行うように設定できます。

ダイジェスト認証では、ユーザーがユーザー名とパスワードをクリアテキストとして送信することなく、ユーザー名とパスワードに基づいた認証を行うことができます。ブラウザはMD5アルゴリズムを使用して、ユーザーのパスワードとProxy Serverによって提供される情報の一部を使用するダイジェスト値を作成します。

サーバーがLDAPベースのディレクトリサービスを使用してダイジェスト認証を行うとき、このダイジェスト値は、ダイジェスト認証プラグインを使用してサーバー側で計算され、クライアントによって提示されたダイジェスト値と比較されます。ダイジェスト値が一致した場合、ユーザーは認証されます。これが機能するには、ディレクトリサーバーがクリアテキスト形式のユーザーのパスワードにアクセスできる必要があります。Sun Java System Directory Serverにはリバーシブルパスワードプラグインが組み込まれています。このプラグインは、対称暗号化アルゴリズムを使用し、暗号化された形式にしてデータを格納し、あとで元の形式に復号化することができます。データへの鍵を持っているのはDirectory Serverだけです。

LDAPベースのダイジェスト認証の場合、リバーシブルパスワードプラグインと、Proxy Serverに組み込まれているダイジェスト認証専用のプラグインを有効にする必要があります。ダイジェスト認証を処理するようにProxy Serverを設定するには、*server-root* `/userdb/`内の`dbswitch.conf`ファイルで、データベース定義の`digestauth`プロパティーを設定します。

dbswitch.confファイルの例を次に示します。

```
directory default ldap://<host_name>:<port>
default:binddn cn=Directory Manager
default:encoded bindpw ***********
default:digestauth on
```

または、

```
directory default ldap://<host_name>:<port>/
default:binddn cn=Directory Manager
default:encoded bindpw ***********
default:digestauthstate on
```

サーバーは、指定されている ACL 方式に基づいて、LDAP データベースに対して認証を試みます。155 ページの「ダイジェスト認証」を参照してください。ACL 方式を指定しない場合、認証が必要であればダイジェスト認証または基本認証のいずれかが使用され、認証が必要ない場合は基本認証が使用されます。

次の表は、認証データベースでサポートされる、あるいはサポートされないダイジェスト認証を示しています。

表 8–1 ダイジェスト認証チャレンジの生成

ACL 方式	認証データベースでサポートされる方式	認証データベースでサポートされない方式
デフォルト 指定されていない	ダイジェストと基本	基本
基本	基本	基本
ダイジェスト	ダイジェスト	エラー

`method=digest` を指定して ACL を処理する場合、サーバーは次のことを実行して認証を試みます。

- 認証要求のヘッダーを確認します。ヘッダーが見つからない場合は、ダイジェストチャレンジに対して 401 の応答が生成され、プロセスが停止します。
- 認証のタイプを確認します。認証のタイプがダイジェストの場合、サーバーは次のことを実行します。
  - nonce を確認します。このサーバーによって生成された、有効で新しい nonce がない場合は、401 の応答が生成されてプロセスが停止します。nonce が無効な場合は、`stale=true` として 401 の応答が生成され、プロセスが停止します。

    server-root `/proxy-server_name/config/` にある `magnus.conf` ファイルの `DigestStaleTimeout` パラメータ値を変更することで、nonce の有効期限を設定できます。この値を設定するには、次の行を `magnus.conf` に追加します。

    `DigestStaleTimeout` *seconds*

ここで *seconds* は、nonce の有効期限 (秒) です。指定されている秒数が経過すると、nonce の有効期限は切れ、ユーザーからの新しい認証が必要になります。

- レルムを確認します。一致するものが見つからない場合は、401 の応答が生成されてプロセスが停止します。
- 認証ディレクトリが LDAP ベースの場合は LDAP ディレクトリにユーザーが存在するかどうかを確認します。認証ディレクトリがファイルベースの場合はファイルデータベースにユーザーが存在するかどうかを確認します。ユーザーが見つからない場合は、401 の応答が生成されてプロセスが停止します。
- ディレクトリサーバーまたはファイルデータベースから `request-digest` 値を取得し、クライアントの `request-digest` 値と一致する値があるかを調べます。一致するものが見つからない場合は、401 の応答が生成されてプロセスが停止します。
- `Authorization-Info` ヘッダーを構築し、サーバーヘッダーに挿入します。

ダイジェスト認証プラグインのインストール

LDAP ベースのディレクトリサービスを使用するダイジェスト認証の場合、ダイジェスト認証プラグインをインストールする必要があります。このプラグインは、サーバー側でダイジェスト値を計算し、この値をクライアントによって提供されるダイジェスト値と比較します。ダイジェスト値が一致した場合、ユーザーは認証されます。

ファイルベースの認証データベースを使用する場合、ダイジェスト認証プラグインをインストールする必要はありません。

UNIX でのダイジェスト認証プラグインのインストール

ダイジェスト認証プラグインは、共用ライブラリと ldif ファイルで構成されています。

- `libdigest-plugin.lib`
- `libdigest-plugin.ldif`

▼ UNIX でダイジェスト認証プラグインをインストールするには

始める前に
- この共用ライブラリが、Sun Java System Directory Server がインストールされているのと同じサーバーコンピュータにあることを確認します。
- ディレクトリマネージャーのパスワードを確認します。
- `/path/to` へのすべての参照が、ダイジェストプラグインの共用ライブラリのインストール先に変更されるように、`libdigest-plugin.ldif` ファイルを修正します。

● プラグインをインストールするには、次のコマンドを入力します。

```
% ldapmodify -D "cn=Directory Manager" -w password -a < libdigest-plugin.ldif
```

Windows でのダイジェスト認証プラグインのインストール

ダイジェストプラグインとともに Sun Java System Directory Server コンピュータが正しく起動できるようにするには、Proxy Server のいくつかの `.dll` ファイルを Directory Server コンピュータにコピーする必要があります。

▼ Windows でダイジェスト認証プラグインをインストールするには

1. *server-root* `\bin\proxy\bin` にある **Proxy Server** の共用ライブラリにアクセスします。
2. ファイル `nsldap32v50.dll`、`libspnr4.dll`、および `libplds4.dll` を適切なディレクトリにコピーします。
3. これらのファイルを次の両方の場所にペーストします。
   - `\Winnt\system32`
   - Sun Java System Directory Server インストールディレクトリ: *server-root*`\bin\sldap\server`

DES アルゴリズムを使用するための Sun Java System Directory Server の設定

DES アルゴリズムは、ダイジェストパスワードが格納されている属性を暗号化するために必要です。

▼ DES アルゴリズムを使用するように Directory Server を設定するには

1. **Sun Java System Directory Server** コンソールを起動します。
2. **Sun ONE Directory Server 5.1 SP1** (またはそれ以降のバージョン) のインスタンスを開きます。
3. **Configuration** タブを選択します。
4. プラグインの横の + 記号をクリックします。
5. **DES** プラグインを選択します。
6. 「**Add**」を選択して新しい属性を追加します。
7. `iplanetReversiblePassword` と入力します。
8. 「**Save**」をクリックします。

9 ダイジェスト認証のパスワードを設定します。

注 – サーバーは、オブジェクトクラス iplanetReversiblePassword 内にある iplanetReversiblePassword 属性を使用します。`iplanetReversiblePassword` 属性でユーザーのダイジェスト認証のパスワードを使用するには、エントリに `iplanetReversiblePasswordobject` オブジェクトを含める必要があります。

これを行なうには、`ldapmodify` を使用するか、Directory Server 管理インタフェースを使用します。

`ldapmodify` を使用する場合—

`digest.ldif` ファイルを作成して、LDAP のコマンドを格納します。パスワードを追加するには、手順が 2 つあります。

a. オブジェクトクラスを `digest.ldif` に追加します。

このファイルは、次のようになっています (Directory Server ユーザーと ACL に応じて、複数の `ldif` ファイルがあることがあります)。

```
dn:uid=user1,dc=india,dc=sun,dc=com
changetype:modify
add:objectclass
objectclass:iplanetReversiblePasswordobject

dn:uid=user1,dc=india,dc=india,dc=sun,dc=com
changetype:modify
add:iplanetReversiblePassword
iplanetReversiblePassword:user1
```

b. `# ldapmodify -D “cn={CN_Value}” -w <password> -a <ldif_file_name>`

10 **Sun Java System Directory Server** インスタンスを再起動し、ユーザーの属性が **Directory Server** データベースに追加されたことを確認します。

その他の認証

アクセス制御 API を使用すると、カスタムの認証方法を作成できます。

ホスト - IP のアクセス制御

管理サーバーおよびそのファイルとディレクトリに対して、特定のコンピュータを使用しているクライアントだけが利用できるように、アクセスを制限できます。そのためには、アクセスの許可または拒否を行うコンピュータのホスト名または IP アドレスを指定します。ホスト - IP 認証を使用したファイルまたはディレクトリへの

アクセスは、ユーザーにはシームレスに見えます。このため、ユーザーは、ユーザー名やパスワードを入力することなく、すぐにファイルやディレクトリにアクセスできます。

特定のコンピュータを複数のユーザーが使用していることもあるため、ホスト - IP 認証は、ユーザー - グループ認証と組み合わせるとより効果的です。両方の認証方法を使用する場合は、アクセスするときにユーザー名とパスワードを求められます。

ホスト - IP 認証では、サーバーで DNS (ドメインネームサービス) を設定する必要はありません。ホスト - IP 認証を使用する場合は、ネットワークで DNS が稼働しており、この認証を使用するようにサーバーが設定されている必要があります。DNS を有効にするには、サーバーのサーバーマネージャーにアクセスし、「Preferences」タブをクリックし、「Configure System Preferences」をクリックします。DNS の設定が表示されます。

DNS を有効にすると、サーバーで DNS 検索が強制的に実行されるため、Proxy Server のパフォーマンスが低下します。サーバーパフォーマンスへの DNS 検索の影響を小さくするには、すべての要求に対して IP アドレスを解決する代わりに、アクセス制御と CGI に対してだけ IP アドレスを解決します。この制限を設定するには、`obj.conf` ファイルで次の内容を指定します。

```
AddLog fn="flex-log" name="access" iponly=1
```

アクセス制御ファイルの使用

管理サーバーまたはサーバー上のファイルやディレクトリに対してアクセス制御を使用する場合、拡張子が `.acl` のファイルに設定が格納されます。アクセス制御ファイルは *server-root* `/httpacl` ディレクトリに置かれます。*server-root* はサーバーがインストールされている場所です。たとえば、`/usr/Sun/Servers` にサーバーをインストールした場合、管理サーバーとサーバーに設定されている各サーバーインスタンスの両方の ACL ファイルが、`/usr/Sun/Servers/httpacl/` に格納されます。

主要な ACL ファイルは `generated-proxy-`*serverid* `.acl` です。一時的な作業ファイルは `genwork-proxy-`*serverid* `.acl` です。管理サーバーを使用してアクセスを設定する場合は、これらの 2 つのファイルが作成されます。ただし、複雑な制約が必要な場合は、複数のファイルを作成し、`server.xml` ファイルから参照することができます。また、時刻や曜日を基準にしたサーバーへのアクセス制限など、ファイルを編集するだけで利用できる機能もいくつかあります。

アクセス制御ファイルとその構文については、第 18 章を参照してください。`server.xml` については、『Sun Java System Web Proxy Server 4.0.4 Configuration File Reference』を参照してください。

ACLユーザーキャッシュの設定

デフォルトでは、Proxy Serverによるユーザーとグループの認証の結果が、ACLユーザーキャッシュに保存されます。magnus.conf ファイルの ACLCacheLifetime 指令を使用して、ACLユーザーキャッシュを有効にする期間を制御することができます。キャッシュのエントリが参照されるたびにその経過時間が計算され、ACLCacheLifetime と照合されます。経過時間が ACLCacheLifetime と同じか、それよりも長い場合、このエントリは使用されません。デフォルト値は120秒です。値を0(ゼロ)に設定すると、キャッシュが無効になります。この値に大きな値を使用する場合、LDAPエントリを変更するたびにProxy Serverの起動が必要となる可能性があります。たとえば、この値を120秒に設定した場合は、Proxy ServerとLDAPディレクトリの同期が2分間にわたって取られない可能性があります。LDAPディレクトリが頻繁に変更される可能性が低い場合にだけ、大きな値を設定します。

magnus.conf の ACLUserCacheSize パラメータを使用すると、キャッシュ内に保存できるエントリの最大数を設定できます。このパラメータのデフォルト値は200です。新しいエントリがリストの先頭に追加され、キャッシュが最大サイズに達すると、新しいエントリを作成するために、このリストの最後のエントリが再利用されます。

また、magnus.conf に含まれるパラメータである ACLGroupCacheSize を使用して、ユーザーエントリごとにキャッシュできるグループメンバーシップの最大数を設定することもできます。このパラメータのデフォルト値は4です。ただし、グループのメンバーではないユーザーはキャッシュされず、要求ごとに何回かLDAPディレクトリにアクセスすることになります。

クライアント証明書によるアクセス制御

サーバー上でSSLが有効になっている場合、アクセス制御とともにクライアント証明書を使用できます。特定のリソースへのアクセスにクライアント証明書が必要であることを指定する必要があります。サーバーでこの機能を有効にした場合、証明書を持つユーザーは、制限されたリソースに初めてアクセスを試みる場合にのみ、名前とパスワードを入力します。ユーザーの識別情報がいったん確立されると、サーバーはログイン名とパスワードを個々の証明書にマップします。その後、ユーザーはクライアント認証の必要なリソースにアクセスする場合、ログイン名とパスワードを入力する必要がなくなります。

ユーザーが制限されたリソースへのアクセスを試みた場合、ユーザーのクライアントはクライアント証明書をサーバーに送信し、サーバーはこれをマッピングリストと照合します。証明書がアクセス権を付与したユーザーのものである場合、リソースが提供されます。

特定のリソースへのアクセスを制御するためにクライアント認証を要求することは、サーバーへの接続のすべてに対してクライアント認証を要求することとは異なります。また、すべてのSSL接続にクライアント証明書を要求しても、証明書が自

動的にデータベース内のユーザーにマップされないことに注意してください。このマッピングを設定するには、特定のリソースへのアクセスにクライアント証明書が必要であることを指定する必要があります。

アクセス制御のしくみ

サーバーはページの要求を受け取ると、ACL ファイル内の規則を使用して、アクセスを許可するか拒否するか判断します。規則は、要求を送信しているコンピュータのホスト名や IP アドレスを参照できます。また、規則が LDAP ディレクトリに格納されているユーザーやグループを参照するように設定することもできます。

次の例では、ACL ファイルのコンテンツと、アクセス制御規則を示しています。

```
version 3.0;
# The following "es-internal" rules protect files such
# as icons and images related to Sun Java System Web Proxy Server.
# These "es-internal" rules should not be modified.
  acl "es-internal";
  allow (read, list, execute,info) user = "anyone";
  deny (write, delete) user = "anyone";

# The following rules deny access to the directory "web"
# to everyone not in the directory server and deny everyone
# in the directory server who is not in GroupB.
# Only the users in GroupB are allowed read, execute, list,
# and info permissions. GroupA cannot gain access to the
# directory "web" even though (in the ACL rule below) they
# can access the directory “my_stuff”. Furthermore, members
# of GroupB cannot write or delete files.
  acl "path=/export/user/990628.1/docs/my_stuff/web/";
  authenticate (user,group) {
     database = "default";
     method = "basic";
  };
  deny (all)
  (user = "anyone");

  allow (read,execute,list,info)
  (group = "GroupB");

# The following rule denies everyone not in the directory
# server and denies everyone in the directory server except
# users with the ID of "SpecificMemberOfGroupB". The ACL rule
# in this setting also has a requirement that the user
# connect from a specific IP address. The IP address setting
# in the rule is optional, and has been added for extra
```

```
# security. Also, this ACL rule has a Customized prompt
# of "Presentation Owner". This Customized prompt appears
# in the username and password dialog box in the client's
# browser.

  acl "path=/export/user/990628.1/docs/my_stuff/web/presentation.html";
  authenticate (user,group) {
     database = "default";
     method = "basic";
     prompt = "Presentation Owner";
  };
  deny (all)
  (user = "anyone" or group = "my_group");
  allow (all)
  (user = "SpecificMemberOfGroupB") and
  (ip = "208.12.54.76");

# The following ACL rule denies everyone not in the directory
# server and everyone in the directory server except for
# GroupA and GroupB access to the directory "my_stuff"
  acl "path=/export/user/990628.1/docs/my_stuff/";
  authenticate (user,group) {
     database = "default";
     method = "basic";
  };
  deny (all)
  (user = "anyone");
  allow (read,execute,list,info)
  (group = "GroupA,GroupB");
```

たとえば、ユーザーが `http://server_name/my_stuff/web/presentation.html` という URL を要求した場合、Proxy Server は、まずサーバー全体のアクセス制御を確認します。サーバー全体への ACL で続行するように設定されている場合、サーバーは `my_stuff` ディレクトリの ACL を確認します。ACL が存在する場合、サーバーは ACL 内の ACE を確認し、次のディレクトリに移動します。このプロセスは、アクセスを拒否する ACL が見つかるまで、または要求された URL の最後の ACL (この場合は、ファイル `presentation.html`) に到達するまで続行されます。

サーバーマネージャーを使用してこの例のアクセス制御を設定するには、このファイルだけを対象とした ACL の作成のほか、ファイルへ誘導する各リソースの ACL を作成することができます。つまり、1 つはサーバー全体用、1 つは `my_stuff` ディレクトリ用、1 つは `my_stuff/web` ディレクトリ用、1 つはファイル用です。

一致する ACL が複数ある場合、サーバーは最後に一致した ACL 文を使用します。

アクセス制御の設定

この節では、アクセスの制限プロセスについて説明します。すべてのサーバーに対してグローバルアクセス制御規則を設定することも、特定のサーバーに対して個別に設定することもできます。たとえば、人事部門では、認証されたすべてのユーザーに各自の給与データを見ることを許可し、データを更新するのは人事部門の給与担当者だけに制限する ACL を作成することもできます。

ここでは、次の内容について説明します。

注 – グローバルアクセス制御を設定する前に、分散管理を設定して有効にしておく必要があります。

グローバルなアクセス制御の設定

▼ すべてのサーバーにアクセス制御を設定するには

1. 管理サーバーにアクセスして、「**Global Settings**」タブをクリックします。
2. 「**Administer Access Control**」リンクをクリックします。
3. ドロップダウンリストから管理サーバー (`proxy-admserv`) を選択し、「**Go**」をクリックしてデータをロードし、「**New ACL**」(または「**Edit ACL**」) をクリックします。
4. プロンプトが表示されたら認証します。

   「Access Control Rules」ページが表示されます。管理サーバーには、編集できないデフォルトアクセス制御規則が 2 行あります。
5. チェックマークが付いていない場合は、「**Access Control Is On**」チェックボックスにチェックマークを付けます。
6. テーブルの最下行にデフォルトの **ACL** 規則を追加するには、「**New Line**」ボタンをクリックします。

   アクセス制御の制限位置を変更するには、上向きまたは下向き矢印をクリックします。
7. 「**Users/Groups**」列の「**Anyone**」をクリックします。

   下のフレームに「Users/Groups」ページが表示されます。

8. アクセスを許可するユーザーやグループを選択し、「**Update**」をクリックします。
   グループまたはユーザーの「List」ボタンをクリックすると、選択肢のリストが表示されます。設定については、オンラインヘルプを参照してください。169 ページの「ユーザーとグループの指定」も参照してください。

9. 「**From Host**」列の「**Anyplace**」をクリックします。
   下のフレームに「From Host」ページが表示されます。

10. アクセスを許可するホスト名と **IP** アドレスを入力し、「**Update**」をクリックします。
    設定については、オンラインヘルプを参照してください。171 ページの「「From Host」の指定」も参照してください。

11. 「**Programs**」列の「**All**」をクリックします。
    下のフレームに「Programs」ページが表示されます。

12. 「**Program Groups**」を選択するか、または「**Program Items**」フィールドにアクセスを許可する特定のファイル名を入力し、「**Update**」をクリックします。
    設定については、オンラインヘルプを参照してください。172 ページの「プログラムへのアクセス制限」も参照してください。

13. (オプション)「**Extra**」列の「**X**」をクリックして、カスタマイズした **ACL** 式を追加します。
    下のフレームに「Customized Expressions」ページが表示されます。詳細は、174 ページの「カスタマイズされた式の作成」を参照してください。

14. 「継続」列のチェックボックスをまだ選択していない場合は、選択します。
    サーバーは次の行を評価してから、ユーザーがアクセスを許可されているかどうかを判断します。複数の行を作成する場合は、もっとも一般的な制限からより特殊な制限に移るようにしてください。

15. (オプション) ごみ箱アイコンをクリックして、アクセス制御規則から対応する行を削除します。

16. (オプション)「**Response When Denied**」リンクをクリックし、ユーザーがアクセスを拒否されたときに受け取る応答を指定します。
    下のフレームに「Access Deny Response」ページが表示されます。

    a. 希望する応答を選択します。

    b. 必要に応じて追加情報を指定します。

c. 「**Update**」をクリックします。

設定については、174 ページの「アクセスが拒否された場合の応答」を参照してください。

17 「**Submit**」をクリックして新しいアクセス制御規則を **ACL** ファイルに保存するか、「**Revert**」をクリックして、ページ内の要素を変更前の値にリセットします。

サーバーインスタンスに対するアクセス制御の設定

サーバーマネージャーを使用すると、特定のサーバーインスタンスに対するアクセス制御の作成、編集、または削除を実行できます。削除する場合、ACL ファイルからすべての ACL 規則を削除しないでください。サーバーを起動するには、少なくとも 1 つの ACL 規則が含まれる ACL ファイルが 1 つ以上必要です。すべての ACL 規則を削除してサーバーを再起動すると、構文エラーが発生します。

▼ サーバーインスタンスにアクセス制御を設定するには

1 サーバーマネージャーにアクセスしてサーバーインスタンスを選択し、「**Preferences**」タブをクリックします。

2 「**Administer Access Control**」リンクをクリックします。

3 次の方法のいずれかを使用して **ACL** を選択します。

- 「Select A Resource」: ACL を使用するリソースを表示して、アクセスを制限します。ドロップダウンリストからリソースを選択するか、「Regular Expression」をクリックして正規表現を指定します。詳細は、『Proxy Server 管理ガイド』の第 16 章を参照してください。
- 「An Existing ACL」: 有効なすべての ACL を表示します。

  このリストには、有効になっていない既存の ACL は表示されません。ドロップダウンリストから ACL を選択します。
- 「Type In The ACL Name」: 名前付き ACL を作成します。このオプションは、ACL ファイルについてよく理解している場合にだけ使用してください。名前付き ACL をリソースに適用する場合は、`obj.conf` ファイルを手動で編集する必要があります。詳細は、第 18 章を参照してください。

4 対応する「**Edit**」ボタンをクリックします。

「Access Control Rules」ページが表示されます。

5 チェックマークが付いていない場合は、「**Access Control Is On**」チェックボックスにチェックマークを付けます。

6 テーブルの最下行にデフォルトの **ACL** 規則を追加するには、「**New Line**」ボタンをクリックします。
アクセス制御の制限位置を変更するには、上向きまたは下向き矢印をクリックします。

7 このサーバーインスタンス用の **ACL** を編集するには、「**Action**」列でアクションをクリックします。
下のフレームに「Allow/Deny」ページが表示されます。

8 デフォルトとして選択されていない場合は「**Allow**」を選択し、「**Update**」をクリックします。
「Allow」または「Deny」については169 ページの「アクションの設定」を参照してください。

9 「**Users/Groups**」列の「**Anyone**」をクリックします。下のフレームに「**User/Group**」ページが表示されます。

10 アクセスを許可するユーザーやグループを選択し、認証情報を指定して、「**Update**」をクリックします。
グループまたはユーザーの「List」ボタンをクリックすると、選択肢のリストが表示されます。設定については、オンラインヘルプを参照してください。169 ページの「ユーザーとグループの指定」も参照してください。

11 「**From Host**」列の「**Anyplace**」をクリックします。
下のフレームに「From Host」ページが表示されます。

12 アクセスを許可するホスト名と **IP** アドレスを入力し、「**Update**」をクリックします。
設定については、オンラインヘルプを参照してください。171 ページの「「From Host」の指定」も参照してください。

13 「**Rights**」列の「**All**」をクリックします。
下のフレームに「Access Rights」ページが表示されます。

14 該当ユーザーのアクセス権限を指定し、「**Update**」をクリックします。
詳細は、172 ページの「プログラムへのアクセス制限」を参照してください。

15 (オプション)「**Extra**」列の「**X**」をクリックして、カスタマイズした **ACL** 式を追加します。

下のフレームに「Customized Expressions」ページが表示されます。詳細は、174 ページの「カスタマイズされた式の作成」を参照してください。

16 「継続」列のチェックボックスをまだ選択していない場合は、選択します。

サーバーは次の行を評価してから、ユーザーがアクセスを許可されているかどうかを判断します。複数の行を作成する場合は、もっとも一般的な制限からより特殊な制限に移るようにしてください。

17 (オプション) ごみ箱アイコンをクリックして、アクセス制御規則から対応する行を削除します。

ACL ファイルからすべての ACL 規則を削除しないでください。サーバーを起動するには、1 つ以上の ACL 規則が含まれる ACL ファイルが少なくとも 1 つ必要です。ACL ファイルのすべての ACL 規則を削除し、サーバーを再起動しようとすると、構文エラーになります。

18 (オプション)「**Response When Denied**」リンクをクリックし、ユーザーがアクセスを拒否されたときに受け取る応答を指定します。

下のフレームに「Access Deny Response」ページが表示されます。希望する応答を選択し、必要に応じて追加情報を指定し、「Update」をクリックします。設定については、174 ページの「アクセスが拒否された場合の応答」を参照してください。

19 「**Submit**」をクリックして新しいアクセス制御規則を **ACL** ファイルに保存するか、「**Revert**」をクリックして、ページ内の要素を変更前の値にリセットします。

アクセス制御オプションの選択

次のトピックでは、アクセス制御を設定するときに選択できるさまざまなオプションについて説明します。管理サーバーの場合は、最初の 2 行はデフォルトとして設定されていて、編集できません。

ここでは、次の内容について説明します。

アクションの設定

要求がアクション制御規則と一致する場合にサーバーが実行するアクションを指定できます。

- 「Allow」は、ユーザーまたはシステムが、要求されたリソースにアクセスできることを意味します。
- 「Deny」は、ユーザーまたはシステムが、要求されたリソースにアクセスできないことを意味します。

サーバーはアクセス制御エントリ (Access Control Entry、ACE) のリストを参照して、アクセス権を決定します。たとえば、最初の ACE は通常、すべてのユーザーを拒否します。最初の ACE に「継続」が設定されている場合、サーバーはリストの 2 番目の ACE を確認し、一致している場合は、次の ACE を使用します。「継続」チェックボックスが選択されていない場合は、すべてのユーザーがリソースへのアクセスを拒否されます。サーバーは、一致しない ACE か、一致しているが継続しない ACE のどちらかに到達するまでリストを参照し続けます。一致する最後の ACE によって、アクセスが許可されるか拒否されるかが決まります。

ユーザーとグループの指定

ユーザーとグループの認証が行われる場合、ユーザーがアクセス制御規則で指定されているリソースにアクセスするには、ユーザー名とパスワードを入力する必要があります。

Proxy Server は、Sun Java System Directory Server などの LDAP サーバー、または内部ファイルベースの認証データベースのいずれかに格納されているユーザーとグループのリストを確認します。

データベース内のすべてのユーザーに対してアクセスを許可または拒否することも、ワイルドカードパターンを使用して特定のユーザーに対してアクセスを許可または拒否することも、アクセスを許可または拒否する対象をユーザーとグループのリストから選択することもできます。

ユーザーインタフェースの「Access Control Rules」ページにある「Users/Groups」には、次の要素が表示されます。

- 「Anyone (No Authentication)」はデフォルト値で、すべてのユーザーがユーザー名とパスワードを入力しなくても、リソースにアクセスできることを意味します。ただし、ホスト名や IP アドレスなど、その他の設定に基づいてユーザーのアクセスが拒否される場合もあります。管理サーバーの場合、この設定は、分散管理のために指定した管理者グループ内のすべてのユーザーがページにアクセスできることを意味します。
- 「Authenticated People Only」

- 「All In The Authentication Database」は、データベースにエントリがあるユーザーに一致します。
- 「Only The following People」では、一致するユーザーとグループを指定します。エントリをコンマ (,) で区切るか、またはワイルドカードパターンを使用すると、ユーザーとユーザーグループの任意のリストを作成することができます。また、データベースに格納されているユーザーやグループのリストから選択することもできます。「Group」は、指定したグループ内のすべてのユーザーに一致します。「User」は、指定した個々のユーザーに一致します。管理サーバーでは、ユーザーを分散管理のために指定した管理者グループのメンバーにする必要があります。

「Prompt For Authentication」では、認証ダイアログボックスに表示されるメッセージテキストを入力します。このテキストを使用して、ユーザーが入力する必要のある項目について説明することができます。オペレーティングシステムによっては、最初の 40 文字程度しか表示されません。ほどんどのブラウザでは、ユーザー名とパスワードをキャッシュし、それらをプロンプトのテキストと関連付けます。つまり、ユーザーが同じプロンプトが表示されるサーバーの領域 (ファイルやディレクトリ) にアクセスする場合、ユーザー名とパスワードを再入力する必要がありません。逆に、さまざまな領域でユーザーに対して再認証を求めるようにする場合、そのリソースに対する ACL のプロンプトを変更する必要があります。

- 「認証メソッド」では、クライアントから認証情報を取得するためにサーバーが使用する方法を指定します。管理サーバーで使用できるのは、認証の基本メソッドだけです。サーバーマネージャーは、次の方法を提供します。
  - 「デフォルト」では、`obj.conf` ファイルで指定されているデフォルトメソッドを使用します。`obj.conf` ファイルに設定されていない場合は「Basic」を使用します。「Default」を選択した場合、ACL 規則によって ACL ファイル内のメソッドが指定されることはありません。「Default」を選択すると、`obj.conf` ファイル内の 1 行を編集することによって、すべての ACL の方法を簡単に変更できます。
  - 「Basic」では、HTTP メソッドを使用して、クライアントから認証情報を取得します。ユーザー名とパスワードは、サーバーで暗号化が設定されている (SSL が有効) 場合にのみ暗号化されます。それ以外の場合、ユーザー名とパスワードはクリアテキスト形式で送信され、傍受された場合は読み取られる可能性があります。
  - 「SSL」では、クライアント証明書を使用してユーザーの認証を行います。このメソッドを使用するには、サーバーの SSL を有効にする必要があります。暗号化が有効になっている場合、基本メソッドと SSL メソッドを組み合わせて使用することができます。

注 – セキュリティーを有効にできるのは、逆プロキシモードのときに限られ、順プロキシモードのときは有効にできません。

- 「Digest」では、ユーザー名とパスワードをクリアテキストとして送信することなく、ブラウザでユーザー名とパスワードに基づいて認証を行えるようにする認証メカニズムを使用します。ブラウザは MD5 アルゴリズムを使用して、ユーザーのパスワードと Proxy Server によって提供される情報の一部を使用するダイジェスト値を作成します。このダイジェスト値は、サーバー側でダイジェスト認証プラグインを使用して計算され、クライアントによって提示されたダイジェスト値と比較されます。

  注 – ダイジェスト認証では、「Prompt For Authentication」が必須のパラメータです。レルムに一致するように値を変更します (ダイジェストファイルの場合は必須)。たとえば、ダイジェストファイルですべてのユーザーがレルム *test* 内に存在するように設定した場合、「Prompt For Authentication」フィールドにはテキスト *test* が含まれるようにします。

- 「Other」では、アクセス制御 API を使用して作成するカスタムのメソッドを使用します。

「Authentication Database」では、サーバーでユーザーの認証に使用するデータベースを指定します。このオプションは、サーバーマネージャーからしか利用できません。「Default」を選択した場合、サーバーはデフォルトとして設定されているディレクトリサービス内のユーザーとグループを検索します。異なるデータベースを使用するように個々の ACL を設定する場合は、「Other」を選択し、データベースを指定します。 *server-root*`/userdb/dbswitch.conf` でデフォルト以外のデータベースと LDAP ディレクトリを指定する必要があります。カスタムデータベースにアクセス制御 API を使用する場合は、「Other」を選択し、データベース名を入力します。

「From Host」の指定

どのコンピュータから要求が送られたかに基づいて、管理サーバーへのアクセスを制限できます。

ユーザーインタフェースの「Access Control Rules」ページにある「Users and Groups」には、次の要素が表示されます。

- 「Anyplace」では、すべてのユーザーとシステムに対してアクセスを許可します。
- 「Only From」では、特定のホスト名または IP アドレスへのアクセスが制限できます。

「Only From」オプションを選択する場合は、「Host Names」フィールドまたは「IP Addresses」フィールドに、ワイルドカードパターンまたはコンマで区切ったリストを入力します。IP アドレスよりホスト名で制限する方が、より柔軟にできます。ユーザーの IP アドレスが変更された場合でも、このリストを更新する必要がありません。ただし、IP アドレスで制限する方が、より確実です。接続したクライアントの DNS 検索が失敗した場合、ホスト名による制限が使用できないためです。

コンピュータのホスト名または IP アドレスと一致するワイルドカードパターンとして使用できるのは、* というワイルドカード表記だけです。たとえば、指定ドメインのすべてのコンピュータに対してアクセスを許可または拒否する場合、`*.example.com` のように、特定ドメイン内のすべてのホストと一致するワイルドカードパターンを指定します。管理サーバーにアクセスするスーパーユーザーに対しては、その他のユーザーとは異なるホスト名と IP アドレスを設定することができます。

ホスト名の場合、* は名前の構成要素全体を表している必要があります。つまり、`*.example.com` は許容されますが、 `*users.example.com` は許容されません。また、ホスト名で * を使用する場合、この記号は文字列の一番左に使用する必要があります。たとえば、`*.example.com` は許容されますが、`users.*.com` は許容されません。

IP アドレスの場合、* はアドレスのバイト全体を表している必要があります。たとえば、`198.95.251.*` は許容されますが、`198.95.251.3*` は許容されません。IP アドレスで * を使用する場合、この記号は文字列の一番右に使用する必要があります。たとえば、`198.*` は許容されますが、`198.*.251.30` は許容されません。

プログラムへのアクセス制限

プログラムへのアクセスを制限できるのは、管理サーバーだけです。プログラムへのアクセス制限を適用すると、特定のユーザーだけがサーバーマネージャーのページを参照し、そのサーバーを設定できるように制限できます。たとえば、一部の管理者に対して管理サーバーの「Users and Groups」セクションを設定することを許可するものの、「Global Settings」セクションへのアクセスは拒否するように制限することができます。

異なるユーザーが異なる機能ドメインにアクセスするように設定することもできます。選択したいくつかの機能ドメインへのアクセス権をユーザーに与えると、そのユーザーのログイン後、アクセス権を設定した機能ドメインの管理サーバーページのみがユーザーに表示されます。

ユーザーインタフェースの「Access Control Rules」ページの「Programs」には、次の要素が表示されます。

- 「All Programs」では、すべてのプログラムへのアクセスを許可または拒否できます。デフォルトでは、管理者はサーバーのすべてのプログラムにアクセスできます。

- 「Only The Following」では、ユーザーがアクセスできるプログラムを指定できます。
  - 「Program Groups」は、「Preferences」や「Global Settings」などの、管理サーバーのタブに反映され、各ページへのアクセスを表します。管理者が管理サーバーにアクセスする場合、サーバーは管理者のユーザー名、ホスト、および IP アドレスを使用して、表示できるページを決定します。
  - 「Program Items」では、フィールドにページ名を入力して、プログラム内の特定のページへのアクセスを制御できます。

アクセス権の設定

サーバーインスタンスに対するアクセス権を設定できるのは、サーバーマネージャーを使用した場合だけです。アクセス権は、サーバーのファイルやディレクトリへのアクセスを制限します。すべてのアクセス権の許可または拒否に加えて、一部のアクセス権の許可または拒否を行うための規則を指定することもできます。たとえば、ユーザーに対してファイルへの読み取り専用アクセスを許可することができます。この設定では、ユーザーは情報を表示することはできますが、ファイルを変更することはできません。

ユーザーインタフェースの「Access Control Rules For」ページの「Rights」には、次の要素が表示されます。

- 「All Access Rights」はデフォルトで、すべてのアクセス権を許可または拒否します。
- 「Only The following Rights」では、許可または拒否するアクセス権の組み合わせを選択できます。
  - 「Read」は、HTTP メソッドの GET、HEAD、POST、INDEX を含むファイルの表示をユーザーに許可します。
  - 「Write」は、HTTP メソッドの PUT、DELETE、MKDIR、RMDIR、MOVE を含むファイルの変更または削除をユーザーに許可します。ファイルを削除するには、書き込み権と削除権の両方が必要です。
  - 「Execute」は、ユーザーに CGI プログラム、Java アプレット、エージェントなどのサーバー側アプリケーションの実行を許可します。
  - 「Delete」は、書き込み特権を持つユーザーにファイルやディレクトリの削除を行う権限を与えます。
  - 「List」は、ユーザーに `index.html` ファイルが存在しないディレクトリ内のファイルリストへのアクセスを許可します。
  - 「Info」は、ユーザーに URI (たとえば `http_head`) についての情報の取得を許可します。

カスタマイズされた式の作成

ACL には、カスタマイズされた式を入力できます。このオプションは、ACL ファイルの構文や構造をよく理解している場合にだけ選択してください。ACL ファイルを編集するか、カスタマイズされた式を作成する場合にだけ使用できる機能がいくつかあります。たとえば、時刻、曜日、またはその両方を基準として、サーバーへのアクセスを制限することができます。

次のカスタマイズされた式で、時刻や曜日によってアクセスを制限する方法を示します。この例では、LDAP ディレクトリに 2 つのグループが存在していることを前提としています。「regular」グループは、月曜から金曜の午前 8 時から午後 5 時までアクセスできます。「critical」グループはいつでもアクセスできます。

```
allow (read){(group=regular and dayofweek=" mon,tue,wed,thu,fri" );
(group=regular and (timeofday>=0800 and timeofday<=1700));(group=critical)}
```

有効な構文と ACL ファイルについては、第 18 章を参照してください。

アクセス制御の解除

「Access Control Rules」ページのオプション「Access Control Is On」の選択を解除した場合、ACL のレコードの消去について確認するプロンプトが表示されます。「了解」をクリックすると、ACL ファイルから該当するリソースの ACL エントリが削除されます。

ACL を無効にする場合、`generated-proxy-` *`serverid`*`.acl` の各行の先頭に # 記号を使用することにより、ACL が記述された行をコメントにすることができます。

管理サーバーからアクセス制御を作成し、特定のサーバーインスタンスに対して有効に設定し、ほかのサーバーに対しては無効 (デフォルト) のままにしておくことができます。たとえば、管理サーバーからサーバーマネージャーページへのアクセスをすべて拒否することができます。デフォルトでは、ほかのサーバーに対して、分散管理は有効に、アクセス制御は無効に設定されます。管理者は管理サーバーを設定することはできませんが、ほかのサーバーにアクセスして設定することはできます。

アクセスが拒否された場合の応答

Proxy Server は、アクセスが拒否された場合、デフォルトメッセージを表示しますが、必要に応じて応答をカスタマイズできます。また、アクセス制御オブジェクトごとに異なるメッセージを作成することもできます。

管理サーバーの場合、デフォルトではユーザーは *`server-root`*`/httpacl/admin-denymsg.html` ファイルの「Permission Denied」メッセージを受け取ります。

▼ アクセス拒否メッセージを変更するには

1. 「**Access Control Rules For**」ページの「**Response When Denied**」リンクをクリックします。
2. 希望する応答を選択し、必要に応じて追加情報を指定し、「**Update**」をクリックします。ユーザーがリダイレクトされた応答にアクセスできることを確認してください。
3. 「**Submit**」をクリックして変更内容を保存するか、「**Revert**」をクリックして、ページ内の要素の値を変更前の値にリセットします。

サーバーの一部へのアクセス制御

この節では、サーバーとその内容に対して一般的に使用されている制限について説明します。各手順のステップごとに、必要な操作を詳細に説明します。ただし、166 ページの「サーバーインスタンスに対するアクセス制御の設定」で説明するステップは実行しておく必要があります。

ここでは、次の内容について説明します。

サーバー全体へのアクセス制限

グループ内のユーザーに対して、サブドメイン内のコンピュータからサーバーへのアクセスを許可したい場合があります。たとえば、ある会社のある部署のサーバーで、ネットワークの特定のサブドメインにあるコンピュータからのアクセスだけをユーザーに対して許可するような場合です。

▼ サーバー全体へのアクセスを制限するには

1. サーバーインスタンスのサーバーマネージャーにアクセスします。
2. 「**Preferences**」タブで、「**Administer Access Control**」リンクをクリックします。

3 ドロップダウンリストからサーバー全体を選択し、「**Select**」をクリックして、対応する「**Edit**」ボタンをクリックします。
「Access Control Rules」ページが表示されます。

4 すべてへのアクセスを拒否する規則を追加します。

5 特定のグループへのアクセスを許可する別の規則を追加します。

6 「**From Host**」を使用して、制限するホスト名および**IP**アドレスを指定します。

7 「**Submit**」をクリックして変更内容を保存します。

ディレクトリへのアクセス制限

グループの所有者によって制御されている、ディレクトリ内のアプリケーションやサブディレクトリおよびファイルの読み取りまたは実行をグループ内のユーザーに許可することができます。たとえば、プロジェクトマネージャは、参照するプロジェクトチームのステータス情報を更新できます。

▼ ディレクトリへのアクセスを制限するには

サーバーインスタンスに対するアクセス制御の設定の節で説明した手順を使用して(166 ページの「サーバーインスタンスに対するアクセス制御の設定」)、次の操作を実行します。

1 サーバーインスタンスのサーバーマネージャーにアクセスします。

2 「**Preferences**」タブで、「**Administer Access Control**」リンクをクリックします。

3 ドロップダウンリストから必要なリソースを選択し、「**Edit**」をクリックします。

4 すべての場所からのすべてのアクセスを拒否する、デフォルト値を使用した規則を作成します。

5 特定のグループのユーザーに対して、読み取り権と実行権だけを許可する別の規則を作成します。

6 特定のユーザーに対してすべてのアクセス権を許可する**3**つ目の規則を作成します。

7 最後の**2**つの規則の「継続」の選択を解除します。

8 「**Submit**」をクリックして変更内容を保存します。

ファイルタイプへのアクセス制限

ファイルタイプに対してアクセスを制限できます。たとえば、特定のユーザーだけに、サーバーで実行されるプログラムの作成を許可することができます。すべてのユーザーがプログラムを実行できますが、作成や削除を実行できるのはグループ内の指定されたユーザーだけです。

▼ ファイルタイプに対してアクセスを制限するには

1. サーバーインスタンスのサーバーマネージャーにアクセスします。
2. 「**Preferences**」タブで、「**Administer Access Control**」リンクをクリックします。
3. 「**Select A Resource**」セクションで「**Regular Expression**」をクリックし、正規表現 (たとえば `*.cgi`) を指定します。
4. ［編集］をクリックします。
5. すべてのユーザーに対して読み取りアクセスを許可する規則を作成します。
6. 指定されたグループだけに読み取りアクセスと削除アクセスを許可する、別の規則を作成します。
7. 「**Submit**」をクリックして変更内容を保存します。

   ファイルタイプの制限については、両方の「継続」チェックボックスのチェックマークを付けたままにします。ファイルが要求されると、サーバーはまず ACL のファイルタイプを確認します。

   `Pathcheck` 関数は `obj.conf` 内に作成されます。この関数には、ファイルまたはディレクトリのワイルドカードパターンが含まれている場合があります。ACL ファイルのエントリは、次のようになります。`acl"*.cgi";`

時刻に基づくアクセス制限

指定した時間または指定した日に、サーバーに対する読み取りアクセスと削除アクセスを制限することができます。

▼ 時刻に基づきアクセスを制限するには

1. サーバーインスタンスのサーバーマネージャーにアクセスします。
2. 「**Preferences**」タブで、「**Administer Access Control**」リンクをクリックします。

3 「**Select A Resource**」セクションのドロップダウンリストからサーバー全体を選択し、「**Edit**」をクリックします。

4 すべてのユーザーに対して読み取り権と実行権を許可する規則を作成します。
つまり、ユーザーがファイルやディレクトリの追加、更新、または削除を行う場合にこの規則が適用されず、サーバーは一致する別の規則を検索します。

5 すべてのユーザーに対して書き込み権と削除権を拒否する、別の規則を作成します。

6 「**X**」リンクをクリックして、カスタマイズされた式を作成します。

7 アクセスを許可する曜日と時刻を入力します。その例を次に示します。

```
user = "anyone" anddayofweek = "sat,sun" or(timeofday >= 1800
andtimeofday <= 600)
```

8 「**Submit**」をクリックして変更内容を保存します。
カスタマイズされた式にエラーがあると、エラーメッセージが生成されます。修正してから、もう一度送信してください。

セキュリティーに基づくアクセス制限

同じサーバーインスタンスに対して、SSL を使用する待機ソケットと SSL を使用しない待機ソケットを設定することができます。セキュリティーに基づく制限を使用すると、セキュリティー保護されたチャネルを経由して送信する必要のあるリソースを保護できます。

▼ セキュリティーに基づきアクセスを制限するには

1 サーバーインスタンスのサーバーマネージャーにアクセスします。

2 「**Preferences**」タブで、「**Administer Access Control**」リンクをクリックします。

3 「**Select A Resource**」セクションのドロップダウンリストからサーバー全体を選択し、「**Edit**」をクリックします。

4 すべてのユーザーに対して読み取り権と実行権を許可する規則を作成します。
つまり、ユーザーがファイルやディレクトリの追加、更新、または削除を行う場合にこの規則が適用されず、サーバーは一致する別の規則を検索します。

5 すべてのユーザーに対して書き込み権と削除権を拒否する、別の規則を作成します。

6 「**X**」リンクをクリックして、カスタマイズされた式を作成します。

7 `ssl="on"` と入力します。次に例を示します。

```
user = "anyone" and ssl="on"
```

8 「**Submit**」をクリックして変更内容を保存します。

カスタマイズされた式にエラーがあると、エラーメッセージが生成されます。修正してから、もう一度送信してください。

リソースへのアクセスのセキュリティー保護

この節では、分散管理を有効にした後、Proxy Server でアクセス制御をセキュリティー保護するために必要な追加作業について説明します。

サーバーインスタンスへのアクセスのセキュリティー保護

サーバーインスタンスへのアクセスを制御するように Proxy Server を設定するには、*server-root*`/httpacl/*.proxy-admserv.acl` ファイルを編集して、アクセス制御権限を与えるユーザーを指定します。次に例を示します。

```
acl "proxy-server_instance "; authenticate (user,group) { database = "default";
method = "basic"; }; deny absolute (all) user != "UserA";
```

IP ベースのアクセス制御の有効化

`ip` 属性を参照するアクセス制御エントリが、管理サーバーに関連する ACL ファイル (`gen*.https-admserv.acl`) にある場合は、次の手順 1、2 を実行してください。

`ip` 属性を参照するアクセス制御エントリが、サーバーインスタンスに関連する ACL ファイルにある場合は、その特定の ACL について次の手順 1 だけを実行してください。

▼ IP ベースのアクセス制御を有効にするには

1 *server-root*`/httpacl/gen*.proxy-admserv.acl` ファイルを次のように編集し、`user` と `group` のほかに、`ip` を認証リストに追加します。

```
acl "proxy-admserv"; authenticate (user,group,ip) { database = "default"; method
= "basic"; };
```

2 次のアクセス制御エントリを追加します。

```
deny absolute (all) ip !="ip_for_which_access_is_allowed ";
```

次に例を示します。

```
acl "proxy-admserv"; authenticate (user,group,ip) { database = "default"; method
= "basic"; }; deny absolute (all) ip !="205.217.243.119";
```

ファイルベースの認証用 ACL の作成

Proxy Server は、ファイルベースの認証データベースの使用をサポートしています。このデータベースには、ユーザーとグループに関する情報がテキスト形式でフラットファイルに格納されます。ACL のフレームワークは、ファイル認証データベースを利用できるように設計されています。

注 – Proxy Server は動的フラットファイルをサポートしていません。フラットファイルデータベースは、サーバーの起動時に読み込まれます。ファイルへの変更は、サーバーを再起動した場合にだけ適用されます。

この節では、ファイル認証とダイジェスト認証に基づくディレクトリサービス用 ACL を作成する方法について説明します。

ACL エントリは、`database` キーワードを使用してユーザーデータベースを参照できます。次に例を示します。

```
acl "default";    authenticate (user) {...    database="myfile";...};
```

server-root`/userdb/dbswitch.conf` ファイルには、ファイル認証データベースとその設定を定義するエントリが含まれています。次に例を示します。

```
directory myfiledb filemyfiledb:syntax keyfilemyfiledb:keyfile
/path/to/config/keyfile
```

次の表は、ファイル認証データベースでサポートされるパラメータを示しています。

表 8–2 ファイル認証データベースがサポートするパラメータ

パラメータ	説明
`syntax`	(オプション) 値は `keyfile` または `digest` のいずれか。省略した場合のデフォルト値は `keyfile`。

表 8–2 ファイル認証データベースがサポートするパラメータ (続き)

パラメータ	説明
keyfile	(syntax=keyfile の場合は必須) ユーザーデータを含むファイルへのパス。
digestfile	(syntax=digest の場合は必須) ダイジェスト認証用のユーザーデータを含むファイルへのパス。

注意 – ファイル認証データベースファイルに指定可能な最大行数は 255 です。この制限を超えた場合、サーバーは起動に失敗し、エラーがログファイルに記録されます。

ファイルベースの認証データベースを使用して ACL を設定する前に、ファイルベースの認証ディレクトリサービスがすでに設定されていることを確認します。詳細は、49 ページの「ディレクトリサービスの設定」を参照してください。

ファイル認証に基づくディレクトリサービス用 ACL の作成

▼ ファイル認証に基づいてディレクトリサービス用 ACL を作成するには

1. サーバーインスタンスのサーバーマネージャーにアクセスします。
2. 「**Preferences**」タブで、「**Administer Access Control**」リンクをクリックします。
3. ドロップダウンリストから **ACL** ファイルを選択し、「**Edit**」をクリックします。
4. 「**Access Control Rules For**」ページで、編集する **ACL** エントリの「**Users/Groups**」リンクをクリックします。
   下のフレームに「User/Group」ページが表示されます。
5. 認証データベースのドロップダウンリストから、鍵ファイルデータベースを指定します。

6 「**Update**」をクリックし、「**Submit**」をクリックして変更内容を保存します。

鍵ファイルベースのファイル認証データベースに対して ACL を設定すると、次のエントリ例のように、`dbswitch.conf` ファイルが ACL エントリで更新されます。

```
version 3.0;acl "default";authenticate (user) {prompt =
"Sun Java System Proxy Server 4.0";database = "mykeyfile";
method = "basic";};deny (all) user = "anyone";
allow (all) user = "all";
```

ダイジェスト認証に基づくディレクトリサービス用 **ACL** の作成

ファイル認証データベースは、RFC 2617 に規定されているダイジェスト認証での使用に適したファイル形式もサポートしています。パスワードとレルムに基づくハッシュが格納されます。クリアテキストのパスワードは保存されません。

▼ ダイジェスト認証に基づいてディレクトリサービス用 **ACL** を作成するには

1 サーバーインスタンスのサーバーマネージャーにアクセスします。

2 「**Preferences**」タブで、「**Administer Access Control**」リンクをクリックします。

3 ドロップダウンリストから **ACL** ファイルを選択し、「**Edit**」をクリックします。

4 「**Access Control Rules For**」ページで、編集する **ACL** の「**Users/Groups**」リンクをクリックします。

下のフレームに「User/Group」ページが表示されます。

5 認証データベースのドロップダウンリストから、ダイジェストデータベースを指定します。

6 「**Update**」をクリックし、「**Submit**」をクリックして変更内容を保存します。

ダイジェスト認証ベースのファイル認証データベースに対して ACL を設定すると、次のエントリ例のように、`dbswitch.conf` ファイルが ACL エントリで更新されます。

```
version 3.0;acl "default";authenticate (user) {prompt = "filerealm";
database = "mydigestfile";method = "digest";}; deny (all) user = "anyone";
allow (all) user = "all";
```

◆ ◆ ◆ 第 9 章

ログファイルの使用

複数の方法で、サーバーのアクティビティーを監視することができます。この章では、ログファイルを記録して参照することによって、サーバーを監視する方法について説明します。組み込みのパフォーマンス監視サービス、すなわち SNMP の使用については、第 10 章を参照してください。

この章の内容は次のとおりです。

ログファイルについて

サーバーのログファイルには、サーバーのアクティビティーが記録されます。このようなログを使用してサーバーを監視すると、障害追跡時に役立ちます。サーバールートディレクトリの `proxy-server_name/logs/errors` にあるエラーログファイルには、サーバーで検出されたすべてのエラーが一覧表示されます。サーバーのルートディレクトリの `proxy-server_name/logs/access` にあるアクセスログには、サーバーに対する要求とサーバーの応答に関する情報が記録されます。Proxy Server の `access` ログファイルに記録される情報を設定することができます。サーバーの統計情報を生成するには、ログアナライザを使用します。サーバーのエラーログファイルとアクセスログファイルをアーカイブすることによって、バックアップすることができます。

注 – オペレーティングシステムの制限により、Linux 上で稼動している Proxy Server では 2G バイトを超えるログファイルを処理できません。最大サイズに達すると、ロギングが終了します。

UNIX および Windows プラットフォームへのログオン

この節では、ログファイルがどのように作成されるのかについて説明します。さらに、この節では、次の項目についても説明します。

- 184 ページの「デフォルトのエラーログ」
- 184 ページの「syslog を利用したログ」

注 – Windows オペレーティング環境で使用されるイベントログメカニズムの詳細を参照するには、Windows のヘルプシステムの索引から「イベントログ」を参照してください。

デフォルトのエラーログ

UNIX と Windows のどちらのプラットフォームでも、管理サーバーからのログは、`proxy-admserv/logs/` 管理ディレクトリに収集されます。サーバーインスタンスからのログは、`proxy-server_name/logs/` ディレクトリに収集されます。

サーバー全体のデフォルトのログレベルを設定することができます。 `stdout` と `stderr` をサーバーのイベントログにリダイレクトし、ログをオペレーティングシステムのシステムログに出力するようにできます。さらに、stdout と stderr の内容をサーバーのイベントログに出力することもできます。デフォルトでは、ログメッセージは stderr と、指定のサーバーログファイルに送信されます。

syslog を利用したログ

一元的なログ記録が必要となる安定した運用環境では、`syslog` が適しています。診断とデバッグのためにログ出力が頻繁に必要となる環境では、サーバーインスタンス別のログのほうが管理が容易です。

サーバーインスタンスおよび管理サーバーのすべてのログデータが 1 つのファイルに記録される場合、内容を解釈したり、デバッグに利用することが難しくなります。`syslog` マスターログファイルは、円滑に稼動している配備済みアプリケーションだけで使用してください。

ログに記録されるメッセージには、Solaris デーモンアプリケーションからのその他のログも含まれています。

syslog ログファイルを syslogd およびシステムログデーモンと組み合わせて使用することで、syslog.conf ファイルを設定して次の処理を行うことができます。

- 適切なシステムログにメッセージを記録する
- システムコンソールにメッセージを出力する
- ログに記録されたメッセージをユーザーリストに転送する、または、ログに記録されたメッセージをネットワーク経由で別のホストの syslogd に転送する

syslog へのログ記録では、Proxy Server およびその他のデーモンアプリケーションからのログが同じファイルに収集されるため、Proxy Server に固有のメッセージと、特定のサーバーインスタンスのメッセージを区別するために、ログメッセージに次の情報が追加されます。

- 一意のメッセージ ID
- タイムスタンプ
- インスタンス名
- プログラム名 (proxyd または proxyd-wdog)
- プロセス ID (proxyd プロセスの PID)
- スレッド ID (オプション)
- サーバー ID

server.xml ファイルでは、管理サーバーとサーバーインスタンスの両方に対してLOG 要素を設定することができます。

UNIX オペレーティング環境で使用される syslog のロギングメカニズムの詳細を参照するには、端末のプロンプトで次の man コマンドを使用します。

```
man syslog
man syslogd
man syslog.conf
```

ログレベル

次の表は、Proxy Server のログレベルとメッセージを重要度の低い順から示しています。

表 9–1 ログレベル

ログレベル	説明
finest finer fine	メッセージはデバッグメッセージの詳細度を示します。finest が最も詳細な情報を示します。

表 9–1 ログレベル (続き)

ログレベル	説明
info	通常は、サーバーの設定やサーバーの状態に関する情報を示します。これらのメッセージは、すぐに対応が必要なエラーを示すものではありません。
warning	警告を示します。このメッセージには、例外も含まれます。
failure	アプリケーションの通常の実行の妨げとなる、深刻な障害を示します。
config	特定の設定に関連する問題のデバッグに役立つ設定の各種統計情報に関連しています。
security	セキュリティーに関する問題を示します。
catastrophe	致命的なエラーを示します。

ログファイルのアーカイブ

アクセスログファイルとエラーログファイルが自動的にアーカイブされるように設定することができます。指定の時刻、または指定の間隔で、ログがローテーションされます。Proxy Server は、古いログファイルを保存し、そのファイルに保存日時を含む名前を付けます。

たとえば、アクセスログファイルを毎時間ローテーションするように設定することができます。Proxy Server は「access.200505160000」という名前を付けてファイルを保存します。ここで、ログファイル名、年、月、日、24 時間形式の時刻は 1 つの文字列で表わされます。ログアーカイブファイルの形式は、設定したログローテーションのタイプによって異なります。

Proxy Server では、次の 2 つのタイプのアーカイブファイルのログローテーションを使用できます。内部デーモンログローテーションと Cron ベースのログローテーション。

内部デーモンログローテーション

内部デーモンログローテーションは HTTP デーモン内で行われ、起動時にだけ設定を変更できます。サーバーの再起動を必要とせず、サーバーで内部的にログをローテーションできます。この方法でローテーションされるログは、次の形式で保存されます。

```
access.<YYYY><MM><DD><HHMM>

errors.<YYYY><MM><DD><HHMM>
```

ログファイルをローテーションし、新しいログファイルでの記録を開始する間隔として使用する時間を指定できます。たとえば、ローテーションの開始時刻が午前 0 時であれば、ローテーション間隔は 1440 分 (1 日) となり、変更を保存して適用すると、現在の時刻に関係なく新しいログファイルが直ちに作成されます。ログファイルは毎日午前 0 時にローテーションされ、アクセスログのタイムスタンプは午前 0 時になり、`access.200505172400` という名前で保存されます。同様に、間隔を 240 分 (4 時間) に設定した場合、午前 0 時から 4 時間おきにログがローテーションされます。アクセスログファイルには、午前 0 時から午前 4 時まで、午前 4 時から午前 8 時まで、などというように収集された情報が保存されます。

ログローテーションが有効になっている場合、サーバーの起動時にログファイルのローテーションが開始されます。ローテーションされる最初のログファイルでは、現在時刻から次のローテーション時刻までの間の情報が収集されます。前の例を使用して、開始時刻を午前 0 時に設定し、ローテーションの間隔を 240 分に設定した場合、現在時刻が午前 6 時とすると、ローテーションされる最初のログファイルには午前 6 時から午前 8 時の間に収集された情報、次のログファイルには午前 8 時から午後 12 時 (正午) までの間に収集された情報、などというように保存されます。

スケジューラベースのログローテーション

スケジューラベースのログローテーションは、*server-root*`/proxy-` *server_name*`/config/` ディレクトリの `server.xml` ファイルに記録される時刻と日付を基準にします。この方法では、すぐにログファイルをアーカイブすることも、サーバーで特定の日の特定の時間にログファイルをアーカイブするように設定することもできます。サーバーのスケジューラ設定オプションは、`server-root/proxy-` *server_name*`/config/` ディレクトリの `server.xml` ファイルに保存されます。スケジューラベースの方法でローテーションされるログは、次の形式で保存されます。

<original_filename>.<YYYY><MM><DD><HHMM>

たとえば、午後 4 時 30 分にローテーションされる `access` は `access.200505171630` という名前になります。

ログローテーションは、サーバーの起動時に初期化されます。ローテーションを有効にすると、Proxy Server はタイムスタンプの付いたアクセスログファイルを作成し、ローテーションがサーバーの起動時に開始されます。

ローテーションが開始されると、アクセスログファイルまたはエラーログファイルに記録する必要のある要求やエラーが、事前にスケジュールされている「次のローテーション時」の後に発生した場合、Proxy Server で新しいタイムスタンプの付いたログファイルが作成されます。

注 – ログアナライザを実行する前に、サーバーログをアーカイブしてください。

ログファイルをアーカイブし、内部デーモンの方法とスケジュールベースの方法のどちらを使用するかを指定するには、サーバーマネージャーで「Archive Log」ページを使用します。

アクセスログの詳細設定

インストール中、サーバーに access という名前のアクセスログファイルが作成されます。アクセスをログに記録するかどうか、ログの記録に使用する形式、クライアントがリソースにアクセスした場合にサーバーでそのクライアントのドメイン名を検索する必要があるかどうかを指定することによって、リソースへのアクセスログをカスタマイズできます。

ログの詳細設定を指定するには、サーバーマネージャーの「Set Access Log Preferences」ページを使用するか、または obj.conf ファイルで指令を手動で設定します。obj.conf では、サーバーは関数 flex-init を呼び出してフレキシブルロギングシステムを初期化し、関数 flex-log を呼び出して要求された特定のデータをフレキシブルログ形式で記録します。共通ログファイル形式を使用して要求をログに記録するには、サーバーはinit-clf を呼び出してobj.conf で使用される共通ログのサブシステムを初期化し、common-log を呼び出して要求された特定のデータを、ほとんどの HTTP サーバーで使用される共通ログ形式で記録します。

リソースのアクセスログが作成されると、そのログをアーカイブする場合や、同じリソースに対して新しいアクセスログファイルを作成する場合を除いて、アクセスログの形式を変更することはできません。

表 9–2 管理サーバーのログファイル形式

ログファイル項目	説明
「**Client Hostname**」:	アクセスを要求するクライアントのホスト名 (DNS が無効の場合は IP アドレス)。
「**Authenticate User Name**」:	認証が必要な場合、認証されたユーザー名をアクセスログに記録することができます。
「システム日付」:	クライアント要求の日時。
「**Full Request**」:	クライアントからの完全な要求。
「状態」:	サーバーからクライアントに返された状態コード。

表 9–2 管理サーバーのログファイル形式 *(続き)*

ログファイル項目	説明
「**Content Length**」:	クライアントに送信されるドキュメントのコンテンツ長 (バイト数)。
「**HTTP Header, "referer"**」:	referer はクライアントが現在のページにアクセスを行ったページを特定します。たとえば、ユーザーがテキスト検索クエリーの結果を調べている場合、referer にはユーザーがテキスト検索エンジンにアクセスしたページが入ります。referer により、サーバーは逆走査したリンクのリストを作成できます。
「**HTTP Header, "user-agent"**」:	クライアントが使用しているブラウザの種類とそのバージョン、およびブラウザが実行されているオペレーティングシステムが含まれるユーザーエージェント情報。この情報は、クライアントがサーバーに送信する HTTP ヘッダー情報のユーザーエージェントフィールドから取得されます。
「**Method**」:	使用される HTTP 要求メソッド。GET、PUT、POST など。
「**URI**」:	Universal Resource Identifier の略。サーバー上のリソースの場所。たとえば、`http://www.a.com:8080/special/docs` の場合、URI は `special/docs` になります。
「**Query String Of The URI**」	URI の疑問符に続く文字列。たとえば、`http://www.a.com:8080/special/docs?find_this` の場合、URI のクエリー文字列は `find_this` になります。
「プロトコル」:	使用される転送プロトコルとバージョン。

既存のログファイルの形式を変更する場合は、最初に既存のログファイルを削除するか、名前を変更します。あるいは、別のファイル名を使用します。

▼ 管理サーバーのアクセスログ詳細設定を設定するには

1 管理サーバーにアクセスして、「**Preferences**」タブをクリックします。

2 「**Set Access Log Preferences**」リンクをクリックします。
「Set Access Log Preferences」ページが表示されます。

3. ドロップダウンリストからリソースを選択するか、「**Regular Expression**」ボタンをクリックして正規表現を入力し、「了解」をクリックします。

4. クライアントアクセスを記録するかどうかを指定します。
   この指定では、ドメインネームサービス (DNS) を有効にする必要があります。

5. アクセスログファイルの絶対パスを指定します。
   デフォルトで、ログファイルはサーバールートの `logs` ディレクトリに保存されます。部分パスを指定した場合、サーバーは、パスがサーバールートの `logs` ディレクトリを基準にしているものと見なします。

   サーバー全体を編集する場合、このフィールドのデフォルト値は、設定ファイルでサーバーのアクセスログファイルを表す変数、`$accesslog` です。

6. サーバーにアクセスするシステムのドメイン名または **IP** アドレスをアクセスログに記録するかどうかを指定します。

7. アクセスログで使用するログファイル形式の種類を選択します。
   有効なオプションは次のとおりです。

   - 「**Use Common** *LogFile Format*」: クライアントのホスト名、認証されたユーザー名、要求日時、HTTP ヘッダー、クライアントに返される状態コード、クライアントに送信されるドキュメントのコンテンツ長などが含まれています。
   - 「**Only Log**」: ログに記録する情報を選択できます。表 9–2に挙げられたフレキシブルログ形式の項目から選択できます。
   - カスタム形式を選択する場合、「Custom Format」フィールドに入力します。

8. 「了解」をクリックします。

9. 「**Restart Required**」をクリックします。
   「Apply Changes」ページが表示されます。

10. 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

サーバーインスタンスのアクセスログの詳細設定の設定

サーバーインスタンスのアクセスログの詳細設定の設定に使用できるフレキシブルログ形式を次の表に示します。

表 9–3 サーバーインスタンスのログファイル形式

ログファイル項目	説明
「**Client Hostname**」:	アクセスを要求するクライアントのホスト名 (DNS が無効の場合は IP アドレス)。
「**Authenticate User Name**」:	認証が必要だった場合、認証されたユーザー名をアクセスログに記録することができます。
「**System Date**」:	クライアント要求の日時。
「**Full Request**」:	クライアントからの完全な要求。
「状態」	サーバーからクライアントに返された状態コード。
「**Content Length**」:	クライアントに送信されるドキュメントのコンテンツ長 (バイト数)。
「**HTTP Header, "referer"**」:	referer はクライアントが現在のページにアクセスを行ったページを特定します。たとえば、ユーザーがテキスト検索クエリーの結果を調べている場合、referer にはユーザーがテキスト検索エンジンにアクセスしたページが入ります。referer により、サーバーは逆走査したリンクのリストを作成できます。
「**HTTP Header, "user-agent"**」:	クライアントが使用しているブラウザの種類とそのバージョン、およびブラウザが実行されているオペレーティングシステムが含まれるユーザーエージェント情報。この情報は、クライアントがサーバーに送信する HTTP ヘッダー情報のユーザーエージェントフィールドから取得されます。
「**Method**」:	使用される HTTP 要求メソッド。GET 、PUT 、POST など。
「**URI**」:	Universal Resource Identifier の略。サーバー上のリソースの場所。たとえば、`http://www.a.com:8080/special/docs` の場合、URI は `special/docs` になります。
「**Query String Of The URI**」:	URI の疑問符に続く文字列。たとえば、`http://www.a.com:8080/special/docs?find_this` の場合、URI のクエリー文字列は `find_this` になります。
「プロトコル」:	使用される転送プロトコルとバージョン。

表 9–3 サーバーインスタンスのログファイル形式 (続き)

ログファイル項目	説明
「**Cache Finish Status**」	このフィールドは、「キャッシュが書き込まれた」、「キャッシュが更新された」、「最新状態チェックを行なった結果キャッシュの内容が返された」のうち、どの状態であるかを示します。
「**Remote Server Finish Status**」	このフィールドは、リモートサーバーへの要求が正常に完了したか、クライアントがブラウザで「停止」ボタンをクリックしたことにより中断されたか、またはエラー条件により中止されたかどうかを示します。
「**Status Code From Server**」:	サーバーから返された状態コード。
「**Route To Proxy (PROXY, SOCKS, DIRECT)**」:	リソースの取得に使用される経路。ドキュメントは直接、プロキシ経由、または SOCKS サーバー経由で取得できます。
「**Transfer Time**」:	転送時間の長さ (秒またはミリ秒)。
「**Header-length From Server Response**」:	サーバー応答のヘッダーの長さ。
「**Request Header Size From Proxy To Server**」	プロキシからサーバーへの要求ヘッダーのサイズ。
「**Response Header Size Sent To Client**」:	クライアントに送信される応答ヘッダーのサイズ。
「**Request Header Size Received From Client**」:	クライアントから受信する要求ヘッダーのサイズ。
「**Content-length From Proxy To Server Request**」:	プロキシからサーバーに送信されるドキュメントの長さ (バイト)。
「**Content-length Received From Client**」:	クライアントからのドキュメントの長さ (バイト)。
「**Content-length From Server Response**」:	サーバーからのドキュメントの長さ (バイト)。
「**Unverified User From Client**」	認証時にリモートサーバーに渡されるユーザー名。

▼ サーバーインスタンスのアクセスログ詳細設定を設定するには

1 サーバーマネージャーにアクセスし、「**Server Status**」タブをクリックします。

2 「**Set Access Log Preferences**」リンクをクリックします。
「Set Access Log Preferences」 ページが表示されます。

3 ドロップダウンリストからリソースを選択するか、「**Regular Expression**」ボタンをクリックして正規表現を入力し、「了解」をクリックします。

4 クライアントアクセスを記録するかどうかを指定します。
この指定では、ドメインネームサービス (DNS) を有効にする必要があります。

5 アクセスログファイルの絶対パスを指定します。
デフォルトで、ログファイルはサーバールートの `logs` ディレクトリに保存されます。部分パスを指定した場合、サーバーは、パスがサーバールートの `logs` ディレクトリを基準にしているものと見なします。

サーバー全体を編集する場合、このフィールドのデフォルト値は、設定ファイルでサーバーのアクセスログファイルを表す変数、`$accesslog` です。

6 サーバーにアクセスするシステムのドメイン名または **IP** アドレスをアクセスログに記録するかどうかを指定します。

7 ログファイルの形式を、共通、拡張、拡張 **2**、指定した情報のみ (「**Only log**」ラジオボタン)、またはカスタムから選択します。
「Only log」をクリックした場合、次のフレキシブルログ形式の項目から選択できます。

8 アクセスログで使用するログファイル形式の種類を選択します。
サーバーアクセスログは、共通ログファイル形式、拡張ログファイル形式、拡張 2 ログファイル形式、フレキシブルログ形式、または独自のカスタマイズ可能な形式にすることができます。共通ログファイル形式は一般的にサポートされている形式で、サーバーに関する一定量の情報が提供されます。フレキシブルログ形式では、(Proxy Server から) ログに記録するコンテンツを選択できます。カスタマイズ可能な形式では、パラメータブロックを指定してログのコンテンツを制御します。

- 「**Use Common** *LogFile Format*」: クライアントのホスト名、認証されたユーザー名、要求日時、HTTP ヘッダー、クライアントに返される状態コード、クライアントに送信されるドキュメントのコンテンツ長などが含まれています。
- 「**Use Extended** *LogFile Format*」: 共通ログファイル形式のすべてのフィールドだけでなく、リモート状態、プロキシからクライアントまでのコンテンツ長、リモートからプロキシまでのコンテンツ長、プロキシからリモートまでのコンテンツ長、クライアントからプロキシまでのヘッダー長、プロキシからクライアントまでのヘッダー長、プロキシからリモートまでのヘッダー長、リモートからプロキシまでのヘッダー長、転送時間などのいくつかの追加フィールドが含まれています。
- 「**Use Extended2** *LogFile Format*」: 拡張ログファイル形式のすべてのフィールドだけでなく、クライアント状態、サーバー状態、リモート状態、キャッシュ完了状態、実際の経路などのいくつかの追加フィールドが含まれています。

- 「**Only Log**」: ログに記録する情報を選択できます。表 9–3に示されたフレキシブルログ形式の項目から選択できます。
- カスタム形式を選択する場合、「Custom Format」フィールドに入力します。

9 特定のホスト名または **IP** アドレスからのクライアントアクセスを記録しない場合は、ホスト名と **IP** アドレスのフィールドにそれぞれ入力します。

アクセス記録を残さないサーバーのホストのワイルドカードパターンを入力します。たとえば、`*.example.com` と入力した場合、ドメイン example.com のユーザーからのアクセスはログに記録されません。ワイルドカードパターンは、ホスト名、IP アドレス、または両方について入力できます。

10 ログファイルに書式文字列を含めるかどうかを選択します。

Proxy Server のログアナライザを使用する場合、書式文字列を含める必要があります。サードパーティー製のアナライザを使用する場合、ログファイルに書式文字列を含める必要はありません。

11 「了解」をクリックします。

12 「**Restart Required**」をクリックします。

「Apply Changes」ページが表示されます。

13 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

Cookie を使用した簡易ロギング

Proxy Server には、flexlog 機能を使用して簡単に特定の cookie のログを記録する方法があります。`obj.conf` 設定ファイル内の `flex-log` サブシステムを初期化する行に `Req-headers.cookie.cookie_name` を追加します。この命令によって、要求のヘッダーに cookie 変数がある場合は cookie 変数 `cookie_name` の値がログに記録され、cookie 変数がない場合は - が記録されます。

エラーロギングオプションの設定

サーバーのエラーログに記録される情報を設定することができます。

▼ エラーロギングオプションを設定するには

1. エラーロギングオプションを管理サーバーから設定するには、「**Preferences**」タブを選択し、「**Set Error Log Preferences**」リンクをクリックします。
   サーバーインスタンスのエラーロギングオプションをサーバーマネージャーから設定するには、「Server Status」タブを選択し、「Set Error Log Preferences」リンクをクリックします。
2. 「**Error Log File Name**」フィールドに、サーバーからのメッセージを保存するファイルを指定します。
3. 「**Log Level**」ドロップダウンリストから、エラーログに記録する必要のある情報量を指定します。有効なオプションは次のとおりです。
4. **stdout** の出力がエラーログにリダイレクトされるようにする場合は、「**Log Stdout**」チェックボックスを選択します。
5. **stderr** の出力がエラーログにリダイレクトされるようにする場合は、「**Log Stderr**」チェックボックスを選択します。
6. コンソールにログメッセージをリダイレクトする場合は、「**Log To Console**」チェックボックスを選択します。
7. **UNIX** の **syslog** サービスまたは **Windows** のイベントログを使用してログを生成および管理する場合は、「**Use System Logging**」チェックボックスを選択します。
8. 「了解」をクリックします。
9. 「**Restart Required**」をクリックします。
   「Apply Changes」ページが表示されます。
10. 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

LOG 要素の設定

次の表は`server.xml` ファイル内で設定できる LOG 要素の属性を示しています。

表 9–4 LOG 属性

属性	デフォルト	説明
file	errors	サーバーからのメッセージを格納するファイルを指定します。
loglevel	info	他の要素によってエラーログに記録されるメッセージのデフォルトの種類を制御します。使用できる値は、最高から最低まで次のようになります: finest 、fine、fine、info、 warning、failure、config、security、および catastrophe。
logstdout	true	(オプション) true に設定した場合、stdout の出力がエラーログにリダイレクトされます。有効な値は、on、off、 yes、no、1、0、 true、false です。
logstderr	true	(オプション) true に設定した場合、stderr の出力がエラーログにリダイレクトされます。有効な値は、on、off、 yes、no、1、0、 true、false です。
logtoconsole	true	(オプション、UNIX のみ) true に設定した場合、ログメッセージがコンソールにリダイレクトされます。
createconsole	false	(オプション、Windows のみ) true に設定した場合、stderr 出力用の Windows コンソールが作成されます。有効な値は、on、off、yes、no、1、 0、true、false です。
usesyslog	false	(オプション) true に設定した場合、ログの生成と管理に UNIX の syslog サービス、または Windows のイベントログが使用されます。有効な値は、on 、off、yes、no、 1、0、true、false です。

アクセスログファイルの表示

サーバーで使用中のアクセスログファイル、およびアーカイブされたアクセスログファイルを参照できます。

管理サーバーのアクセスログを管理サーバーから表示するには、「Preferences」タブを選択し、「View Access Log」リンクをクリックします。

サーバーインスタンスのアクセスログをサーバーマネージャーから表示するには、「Server Status」タブを選択し、「View Access」ページを選択します。

次の例は、共通ログファイル形式のアクセスログを示します。

```
198.18.17.222 - - [20/May/2005:14:15:49 +0530]
"GET http://www.example.com/ HTTP/1.1" 504 622 198.18.17.222 - abc
[20/May/2005:14:16:09 +0530] "GET http://www.test.com/report.zip HTTP/1.1"
504 630
```

次の表に、アクセスログの最後の行について説明します。

アクセスログフィールド	例
クライアントのホスト名または IP アドレス	198.18.17.222。この例では、Proxy Server の DNS 検索の設定が無効になっているため、クライアントの IP アドレスが表示されています。DNS 検索が有効に設定されている場合、クライアントのホスト名が表示されます。
RFC 931 情報	- (RFC 931 の識別情報は表示されない)
ユーザー名	*abc* (認証のためにクライアントによって入力されたユーザー名)
要求の日時	20/May/2005:14:16:09 +0530
要求	GET
プロトコル	HTTP/1.1
状態コード	504
転送されたバイト	630

エラーログファイルの表示

エラーログファイルには、ログファイルが作成されてからサーバーで検出されたエラーが記録されます。また、このログファイルにはサーバーの起動時などのサーバーに関する情報メッセージも記録されます。エラーログには、失敗したユーザー認証も記録されます。エラーログを使用して、誤った URL パスや不足しているファイルを見つけることもできます。

管理サーバーのエラーログを管理サーバーから表示するには、「Preferences」タブを選択し、「View Error Log」リンクをクリックします。

サーバーインスタンスのエラーログファイルを管理サーバーから表示するには、「Server Status」タブを選択し、「View Error Log」リンクをクリックします。

次に、エラーログの 3 つのエントリ例を示します。

```
20/May/2005:14:08:37] info ( 6141): CORE1116: Sun Java System Web Proxy
Server 4.0 B05/10/2005 01:26 20/May/2005:14:08:37] info ( 6142): CORE3274:
```

```
successful server startup 20/May/2005:14:08:37] security (23246):
for host 198.18.148.89 trying to GET /, deny-service reports:
 denying service of /
```

ログアナライザの使用

server-root/extras/log_anly ディレクトリには、サーバーマネージャーのユーザーインタフェースから実行するログ分析ツールがあります。このログアナライザは、共通ログ形式のファイルだけを分析します。log_anly ディレクトリにある HTML ドキュメントに、このツールのパラメータが説明されています。
server-install/extras/ flexanlg ディレクトリには、フレキシブルログファイル形式用のコマンド行ログアナライザがあります。ただし、サーバーマネージャーのデフォルト設定では、選択したログファイル形式に関係なく、フレキシブルログファイルレポートツールを使用するように設定されています。

ログアナライザを使用して、アクティビティーの要約、もっとも頻繁にアクセスされる URL、サーバーがもっとも頻繁にアクセスされる時間など、デフォルトサーバーの統計情報を生成します。ログアナライザは、Proxy Server から実行することも、コマンド行から実行することもできます。

flexanlg コマンド行ユーティリティーを実行する前に、ライブラリパスを設定する必要があります。各種プラットフォームでの設定は、次のとおりです。

Solaris および Linux:

```
LD_LIBRARY_PATH=server-root/bin/proxy/lib:$LD_LIBRARY_PATH
```

AIX:

```
LIBPATH=server-root/bin/proxy/lib:$LIBPATH
```

HP-UX:

```
SHLIB_PATH=server-root/bin/proxy/lib:$SHLIB_PATH
```

Windows:

```
path=server-root\\bin\\proxy\\bin;%path%
```

注 – ログアナライザを実行する前に、サーバーログをアーカイブする必要があります。サーバーログのアーカイブについては、186 ページの「ログファイルのアーカイブ」を参照してください。

ライブラリパスを設定する代わりに、`server-root/proxy-server`*id* ディレクトリに変更を行った後、コマンドプロンプトで `./start -shell` と入力することもできます。

拡張または拡張 2 ロギング形式を使用する場合、ログアナライザは報告用に指定した情報のほかに、出力ファイルに複数のレポートを生成します。次の節では、これらのレポートについて説明します。

転送時間分散レポート

転送時間分散レポートには、Proxy Server が要求の転送に要する時間が示されます。このレポートには、情報がサービス時間別、および完了率別に分類されて表示されます。次の例は転送時間分散レポートの例です。

サービス時間カテゴリ別:

```
< 1 sec [644%] ....................................

< 2 sec [33.3%] ...................

< 3 sec [ 2.7%] .

< 4 sec [ 1.7%] .

< 5 sec [ 0.6%]

< 6 sec [ 0.4%]

< 7 sec [ 0.2%]

< 8 sec [ 0.0%]

< 9 sec [ 0.0%]
```

完了率別:

```
< 1 sec [64.4%] ....................................

< 2 sec [97.7%] ................................

< 3 sec [100.4%]........................................
```

データフローレポート

データフローレポートには、クライアントからプロキシ、プロキシからクライアント、プロキシからリモートサーバー、およびリモートサーバーからプロキシへのデータフロー (転送バイト数) が示されます。これらの各シナリオについて、レポートには転送されたデータの量がヘッダーおよびコンテンツの形式で示されます。データフローレポートには、キャッシュからクライアントへのデータフローも示されます。次はデータフローレポートの例です。

	Headers	**Content**	**Total**
- Client -> Proxy.....	0 MB	0 MB	0 MB
- Proxy -> Client.....	0 MB	2 MB	3 MB
- Proxy -> Remote.....	0 MB	0 MB	0 MB
- Remote -> Proxy.....	0 MB	2 MB	2 MB
Approx:			
- Cache -> Client.....	0 MB	0 MB	0 MB

状態コードレポート

状態コードレポートには、Proxy Server がリモートサーバーから受信した状態コード、およびクライアントに送信した状態コードの内容と数が示されます。この状態コードレポートには、これらのすべての状態コードの説明も表示されます。次の例は状態コードレポートの例です。

Code	**-From remote-**	**-To client-**	**-Explanation-**
200	338 [70.7%]	352 [73.6%]	OK
302	33 [6.9%]	36 [7.5%]	Redirect
304	90 [18.8%]	99 [20.7%]	Not modified
404	3 [0.6%]	3 [0.6%]	Not found
407		5 [1.0%]	Proxy authorization required
500		2 [0.4%]	Internal server error

Code	-From remote-	-To client-	-Explanation-
504		6 [1.3%]	Gateway timeout

要求と接続レポート

要求と接続レポートには、Proxy Server がクライアントから受信した要求数、プロキシからリモートサーバーに設定した接続数 (初期取得、最新状態チェック、更新)、Proxy Server がキャッシュされたドキュメントを使用することにより回避したリモート接続数が示されます。次の例は要求と接続レポートの例です。

```
- Total requests.............     478
- Remote connections.........     439
- Avoided remote connects....      39 [ 8.2%]
```

キャッシュパフォーマンスレポート

キャッシュパフォーマンスレポートには、クライアントのキャッシュ、Proxy Server のキャッシュ、直接の接続のパフォーマンスが示されます。

クライアントのキャッシュ

クライアントのキャッシュのヒットは、クライアントがドキュメントで最新状態チェックを実行し、リモートサーバーが 304 メッセージ、すなわちクライアントにドキュメントが変更されなかったことを伝えるメッセージが返された場合に起こります。クライアントにより開始される最新状態チェックから、クライアントがドキュメントの独自のコピーがキャッシュにあることが示されます。

クライアントのキャッシュについては、レポートに次の内容が示されます。

- クライアントおよびプロキシのキャッシュのヒット: このクライアントのキャッシュのヒットでは、Proxy Server とクライアントのいずれにも要求されたドキュメントのコピーがあり、リモートサーバーにプロキシのコピーについての最新状態チェックが照会され、そのあと、プロキシのコピーについてクライアントの要求が評価されます。キャッシュパフォーマンスレポートには、プロキシにより処理されたこのタイプの要求数と、これらの要求の処理にプロキシが要した平均時間が示されます。
- プロキシのショートカットチェックなし:このクライアントのキャッシュのヒットでは、Proxy Server とクライアントのいずれにも要求されたドキュメントのコピーがあり、Proxy Server はクライアントのキャッシュのドキュメントが最新であることをクライアントに伝えます (リモートサーバーでのチェックは行われない)。キャッシュパフォーマンスレポートには、プロキシにより処理されたこのタイプの要求数と、これらの要求の処理にプロキシが要した平均時間が示されます。

- クライアントのキャッシュのヒットのみ**:** このクライアントのキャッシュのヒットでは、クライアントのキャッシュのみに要求されたドキュメントのコピーがあります。このタイプの要求では、Proxy Server がクライアントの If-modified-since GET ヘッダーに直接トンネリングします。キャッシュパフォーマンスレポートには、プロキシにより処理されたこのタイプの要求数と、これらの要求の処理にプロキシが要した平均時間が示されます。
- クライアントのキャッシュの合計ヒット数**:**クライアントのキャッシュの合計ヒット数と、これらの要求の処理にクライアントが要した平均時間。

プロキシのキャッシュ

プロキシのキャッシュのヒットは、クライアントが Proxy Server からドキュメントを要求し、Proxy Server のキャッシュにすでにドキュメントがある場合に起こります。Proxy Server のキャッシュのヒットの場合、レポートに次の内容が示されます。

- チェックによるプロキシのキャッシュのヒット**:** このプロキシのキャッシュのヒットでは、Proxy Server がドキュメントの最新状態チェックをリモートサーバーに照会します。キャッシュパフォーマンスレポートには、プロキシにより処理されたこのタイプの要求数と、これらの要求の処理にプロキシが要した平均時間が示されます。
- チェックなしのプロキシのキャッシュのヒット**:**このプロキシのキャッシュのヒットでは、Proxy Server がドキュメントの最新状態チェックをリモートサーバーに照会しません。キャッシュパフォーマンスレポートには、プロキシにより処理されたこのタイプの要求数と、これらの要求の処理にプロキシが要した平均時間が示されます。
- 純粋なプロキシのキャッシュのヒット**:** このプロキシのキャッシュのヒットでは、クライアントに要求されたドキュメントのコピーがキャッシュにありません。キャッシュパフォーマンスレポートには、プロキシにより処理されたこのタイプの要求数と、これらの要求の処理にプロキシが要した平均時間が示されます。

組み合わせたプロキシのキャッシュのヒット

組み合わせたプロキシのキャッシュのヒットの場合、レポートに Proxy Server のキャッシュに対するヒットの合計数と、これらの要求の処理に Proxy Server が要した平均時間が示されます。

直接のトランザクション

直接のトランザクションとは、キャッシュにヒットせずに、直接リモートサーバーから Proxy Server、そしてクライアントに送られるトランザクションです。直接のトランザクションの場合、レポートに次の内容が示されます。

- 取得したドキュメント数:リモートサーバーから直接取得したドキュメントの数。キャッシュパフォーマンスレポートには、プロキシが処理したこのタイプの要求数と、これらの要求の処理にプロキシが要した平均時間、および合計トランザクションに対する割合が示されます。
- その他のトランザクション:200 または 304 以外の状態コードが返されるトランザクション。キャッシュパフォーマンスレポートには、プロキシが処理したこのタイプの要求数と、これらの要求の処理にプロキシが要した平均時間が示されます。
- 直接のトラフィックの合計数: クライアントからリモートサーバーに直接送られた要求数 (失敗した要求や正常に取得されたドキュメントを含む)。キャッシュパフォーマンスレポートには、プロキシが処理したこのタイプの要求数と、これらの要求の処理にプロキシが要した平均時間、および合計トランザクションに対する割合が示されます。

次の例はキャッシュパフォーマンスレポートの例です。

```
                    CLIENT CACHE:
- Client & proxy cache hits... 86 reqs [18.0%] 0.21 sec/req- Proxy shortcut
no-check........ 13 reqs [ 2.7%] 0.00 sec/req- Client cache hits only.....
- TOTAL client cache hits.......... 99 reqs [20.7%] 0.18 sec/req

                    PROXY CACHE:
- Proxy cache hits w/check........ 4 reqs [ 0.8%] 0.50 sec/req- Proxy cache
hits w/o check.. 10 reqs [ 2.1%] 0.00 sec/req- Pure proxy cache hits......
14 reqs [ 2.9%] 0.14 sec/req

                    PROXY CACHE HITS COMBINED:
- TOTAL proxy cache hits....... 113 reqs [23.6%] 0.18 sec/req

                    DIRECT TRANSACTIONS:
- Retrieved documents..313 reqs [65.5%]  0.90 sec/req 2 MB- Other
transactions.. 52 reqs [10.9%] 7.79 sec/req- TOTAL direct traffic..
365 reqs [76.4%] 1.88 sec/req 2 MB
```

転送時間レポート

転送時間レポートには、Proxy Server がトランザクションの処理に要した時間に関する情報が示されます。このレポートには、次のカテゴリの値が示されます。

平均トランザクション時間:ログに記録されたすべての転送時間の平均。

キャッシュがない場合の転送時間の平均:キャッシュから返されないトランザクション (リモートサーバーからの 200 の応答) の転送時間の平均。

エラーがない場合のキャッシュの平均:エラーがないトランザクション (状態コードが 2xx と 3xx) の転送時間の平均。

平均転送時間の向上:平均トランザクション時間から、エラーがない場合のキャッシュの平均転送時間を引いた時間。

次の例は転送時間レポートの例です。

```
- Average transaction time... 1.48 sec/req- Ave xfer time w/o caching..
    0.90 sec/req- Ave w/caching, w/o errors.. 0.71 sec/req - Ave xfer
    time improvement.. 0.19 sec/req
```

毎時アクティビティーレポート

分析された各時間について、毎時アクティビティーレポートに次の内容が示されます。

- 読み込み平均
- リモートサーバーへの最新状態チェックを実行しない場合のキャッシュのヒット数
- リモートサーバーに対して最新状態チェックを実行し、ドキュメントが最新であり、クライアントのキャッシュ内にドキュメントがあることが証明された Proxy Server のキャッシュのヒット数
- リモートサーバーに対して最新状態チェックを実行し、ドキュメントが最新であり、クライアントのキャッシュ内にドキュメントがないことが証明された Proxy Server のキャッシュのヒット数
- リモートサーバーに対して最新状態チェックを実行し、ドキュメントの一部が更新された Proxy Server のキャッシュのヒット数
- リモートサーバーに対して最新状態チェックを実行し、要求されたドキュメントの新しいコピーを状態コード 200 で返した Proxy Server のキャッシュのヒット数
- Proxy Server のキャッシュにヒットせず、リモートサーバーから直接ドキュメントを取得した要求数

▼ サーバーマネージャーからログアナライザを実行するには

1 サーバーマネージャーにアクセスし、「**Server Status**」タブをクリックします。

2 「**Generate Report**」リンクをクリックします。
「Generate Report」ページが表示されます。

3 サーバー名を入力します。この名前は生成されるレポートに表示されます。

4 レポートを **HTML** 形式または **ASCII** 形式で表示するかどうかを指定します。

5 分析するログファイルを選択します。

6 結果をファイルに保存する場合は、「**Output File**」フィールドに出力ファイル名を入力します。

このフィールドを空白のままにすると、レポート結果は画面に出力されます。大容量ログファイルの場合、画面への出力に時間がかかる場合があるため、結果をファイルに保存するようにしてください。

7 特定のサーバー統計用に合計を生成するかどうかを指定します。

次の合計を生成できます。

- 合計ヒット数: アクセスログが有効になってから、サーバーが受信した合計ヒット数。
- **304** (変更なし) 状態コード: サーバーによりページが返されるのではなく、要求したドキュメントのローカルコピーが使用された回数。
- **302** (リダイレクト) 状態コード: 元の URL が移動したため、サーバーが新しいURL にリダイレクトした回数。
- **404** (見つかりません) 状態コード: サーバーが要求されたドキュメントを見つけられなかった回数、またはクライアントが承認されたユーザーではなかったためサーバーがドキュメントを提供しなかった回数。
- **500** (サーバーエラー) 状態コード: サーバー関連のエラーが発生した回数。
- 固有の **URL** の合計数 : アクセスログが有効になってからアクセスされた固有の URL の数。
- 固有のホストの合計数: アクセスログが有効になってからサーバーにアクセスした固有のホスト数。
- 合計転送量 (**K** バイト): アクセスログが有効になってからサーバーが転送した K バイト数。

8 一般統計を生成するかどうかを選択します。統計の生成を選択する場合、次の項目から選択します。

- 「**Find Top***Number* **Seconds Of Log**」: 最新の統計対象期間 (秒) からの情報に基づいて、統計情報を生成します。
- 「**Find Top***Number* **Minutes Of Log**」:
- 最新の統計対象期間 (分) からの情報に基づいて、統計情報を生成します。
- 「**Find Top***Number* **Hours Of Log**」: 最新の統計対象期間 (時間) からの情報に基づいて、統計情報を生成します。

- 「**Find** *Number***Users (If Logged)**」: ユーザーの数からの情報に基づいて情報を生成します。
- 「**Find Top** *Number***Referers (If Logged)**」: referrer の数からの情報に基づいて情報を生成します。
- 「**Find Top** *Number***User Agents (If Logged)**」: ブラウザの種類、ブラウザのバージョン、オペレーティングシステムなど、ユーザーエージェントに関する情報に基づいて統計を生成します。
- 「**Find Top** *Number***Miscellaneous Logged Items (If Logged)**」: ユーザーの数からの情報に基づいて統計を生成します。

9 リストを生成するかどうかを選択します。

リストの生成を選択する場合、リストを生成する項目を指定します。

- 「**URLs Accessed**」: アクセスされた URL を表示します。
- 「*Number***Most Commonly Accessed URL**」: 最も頻繁にアクセスされた URL、またはアクセス回数が指定数を超えた URL を表示します。
- 「**URLs That Were Accessed More Than** *Number***Times**」: 指定された回数よりも多くアクセスされた URL を表示します。
- 「**Hosts Accessing Your Server**」: Proxy Server にアクセスしたホストを表示します。
- 「*Number***Hosts Most Often Accessing Your Server**」: 最も頻繁にサーバーにアクセスするホスト、または指定された回数を超えてサーバーにアクセスしたホストを表示します。
- 「**Hosts That Accessed Your Server More Than** *Number***Times**」: 指定された回数よりも多くサーバーにアクセスしたホストを表示します。

10 結果を表示する順序を指定します。

レポートの各セクションで表示する次の項目の順序を 1 ～ 3 番目まで指定します。項目をいずれも生成しない場合、そのセクションは自動的に省略されます。セクションには次のものがあります。

- Find Totals
- General Statistics
- Make Lists

11 「了解」をクリックします。

新しいウィンドウにレポートが表示されます。

コマンド行からログアナライザを実行するには

コマンド行からアクセスログファイルを分析するには、flexanlg ツールを実行します。このツールは server-install/extras/flexanlg ディレクトリにあります。

flexanlg を実行するには、コマンドプロンプトに以下のコマンドとオプションを入力します。

```
./flexanlg [-n name] [-x] [-r] [-p order] [-i file]* [ -m metafile ]* [-o file]
[-c opts] [-t opts] [-l opts]
```

* の付いているオプションは繰り返すことができます。

./flexanlg -h と入力すると、この情報をオンラインで入手できます。

```
-P: proxy log format                                    Default: no
-n servername: The name of the server
-x : Output in HTML                                     Default: no
-r : Resolve IP addresses to hostnames                  Default: no
-p [c,t,l]: Output order (counts, time stats, lists)  Default: ctl
-i filename: Input log file                             Default: none
-o filename: Output log file                            Default: stdout
-m filename: Meta file                                  Default: none
-c [h,n,r,f,e,u,o,k,c,z]: Count these item(s) -       Default: hnreuokc
    h: total hits
    n: 304 Not Modified status codes (Use Local Copy)
    r: 302 Found status codes (Redirects)
    f: 404 Not Found status codes (Document Not Found)
    e: 500 Server Error status codes (Misconfiguration)
    u: total unique URL's
    o: total unique hosts
    k: total kilobytes transferred
    c: total kilobytes saved by caches
    z: Do not count any items.
-t [sx,mx,hx, xx,z]: Find time stats -      Default:s5m5h10u10a10r10x10
    s(number): Find top (number) seconds of log
    m(number): Find top (number) minutes of log
    h(number): Find top (number) hours of log
    u(number): Find top (number) users of log
    a(number): Find top (number) user agents of log
    r(number): Find top (number) referers of log
    x(number): Find top (number) for miscellaneous keywords
    z: Do not find any time stats.
-l [cx,hx]: Make a list of -                            Default: c+3h5
    c(x,+x): Most commonly accessed URL's
             (x: Only list x entries)
             (+x: Only list if accessed more than x times)
    h(x,+x): Hosts (or IP addresses) most often accessing your server
```

```
            (x: Only list x entries)
            (+x: Only list if accessed more than x times)
    z: Do not make any lists.
```

イベントの表示 (Windows)

Proxy Server は、エラーをサーバーエラーログに記録しますが、深刻なシステムエラーはイベントビューアにもログ記録します。イベントビューアでは、システム上のイベントを監視できます。イベントビューアを使用して、基本的な設定の問題によって発生したエラーを参照します。この問題は、エラーログが開けるようになる前に発生する可能性があります。

▼ イベントビューアを使用するには

1. 「スタート」メニューから、「プログラム」、「管理ツール」を順に選択します。
   「管理ツール」プログラムグループで「イベントビューア」を選択します。

2. 「ログ」メニューの「アプリケーション」を選択します。
   「イベントビューア」に「アプリケーションログ」が表示されます。Proxy Server のエラーには、`proxy-server`*id* というソースラベルが付いています。

3. 「表示」メニューの「検索」を選択すると、ログ内でこのようなラベルを検索できます。
   「表示」メニューの「最新の情報に更新」を選択すると、更新されたログエントリを表示できます。

   イベントビューアについては、使用しているシステムのマニュアルを参照してください。

◆ ◆ ◆ 第 10 章

10 サーバーの監視

この章では、組み込みの監視ツールや SNMP (Simple Network Management Protocol) を含む、サーバーの監視方法について説明します。

Sun Java System の Management Information Base (MIB) や、HP OpenView のようなネットワーク管理ソフトウェアとともに SNMP を使用して、ネットワーク内のほかのデバイスを監視するのと同じように、リアルタイムでサーバーを監視できます。

注 – Windows では、Proxy Server 4 をインストールする前に、Windows SNMP コンポーネントがすでにシステムにインストールされていることを確認してください。

統計機能または SNMP を使用することによって、サーバーの状態をリアルタイムで表示できます。UNIX または Linux を使用している場合に、SNMP を使用するときは、Proxy Server を SNMP 用に設定する必要があります。

この章の内容は次のとおりです。

統計情報によるサーバーの監視

統計機能を使用して、サーバーの現在の稼動状況を監視できます。統計情報は、サーバーが処理している要求数と、それらの要求の処理状況を示します。対話型サーバーモニターを通してサーバーが多数の要求を処理していることがわかる場合、要求数に合わせてサーバー設定またはシステムのネットワークカーネルを調整する必要がある場合もあります。統計情報の収集により Proxy Server へのオーバーヘッドが増えるため、デフォルトでは、統計情報は無効になっています。統計情報を有効にすると、サーバーは統計情報の収集および保存を開始します。

統計情報を使用可能にすると、次の分野の統計情報を表示できます。

- 接続
- DNS *
- KeepAlive
- キャッシュ
- サーバー要求

対話型サーバーモニターで総計をレポートするサーバーの各種統計情報については、オンラインヘルプの「Monitor Current Activity」ページを参照してください。

Proxy Server の統計情報の処理

Proxy Server の統計情報の収集には、`stats-xml` と呼ばれる組み込み関数が使用されます。この関数はサーバーマネージャーから統計情報を表示する場合、または `perfdump` 関数を使用してレポートを生成する場合に有効にする必要があります。また、カスタム NSAPI 関数を使用して統計情報を監視するための条件であるプロファイリングを使用可能にするために、`stats-xml` 関数も使用します。サーバーで統計情報とプロファイリングを有効にすると、`obj.conf` ファイルの `stats-init` と呼ばれるサーバー関数が初期化され、統計の収集が開始されます。

```
Init profiling="on" fn="stats-init"
```

また、この命令により、ブラウザウィンドウから統計情報にアクセスできるようにする `NameTrans` 指令が作成されます。

```
NameTrans fn="assign-name" name="stats-xml" from="(/stats-xml|/stats-xml/.*)
```

最後に、統計情報の有効化により、`NameTrans` 指令が選択された場合に `stats-xml` 関数を処理する `Service` 指令が追加されます。

```
<Object name="stats-xml">

Service fn="stats-xml"

</Object>
```

統計情報が収集されると、obj.conf ファイルの Init 関数が更新されます。そのため、サーバーを停止および再起動して、これらの変更を有効にする必要があります。

次の例に、obj.conf ファイルの stats-init を示します。

```
Init profiling="on" fn="stats-init" update-interval="5"
```

また、次の値も指定できます。

- **update-interval**: 統計の更新間隔 (秒)。値を高くすると (頻度を低くする)、パフォーマンスが向上します。最小値は 1、デフォルト値は 5 です。
- **profiling**: NSAPI パフォーマンスのプロファイリングを有効にします。デフォルトは *no* です。no にすると、サーバーのパフォーマンスが多少向上します。ただし、ユーザーインタフェースから統計情報を有効にする場合、プロファイリングはデフォルトで有効になっています。

次の URL から stats-xml 出力を取得できます。

```
http://computer_name:proxyport /stats-xml/proxystats.xml
```

この要求は Proxy Server の統計情報を含む XML ページを返します。ブラウザによってはブラウザウィンドウにデータを表示できますが、データを外部ファイルに保存して、外部ビューアでデータを表示する必要のあるブラウザもあります。データ分析のためにさまざまなビューで表示する統計情報の分析機能がなければ、この情報がどれだけ役に立つかが、はっきりとはわかりません。サードパーティー製ツールを使うと、このプロセスをサポートできます。解析ツールを使用しない場合、サーバーマネージャーまたは perfdump SAF から stats-xml 出力を監視することをお勧めします。

stats-xml 出力へのアクセスの制限

サーバーの stats-xml 統計情報をブラウザに表示できるユーザーを制限する場合、/stats-xml URI 用の ACL を作成する必要があります。

ACL ファイルは、obj.conf ファイルの stats-xml オブジェクト定義でも参照されるようにする必要があります。たとえば、/stats-xml URI に名前付きの ACL を作成した場合、次のようにして、オブジェクト定義の PathCheck 文の中で ACL ファイルを参照するようにする必要があります。

```
<Object name="stats-xml">

PathCheck fn="check-acl" acl="stats.acl"

Service fn="stats-xml"

</Object>
```

統計情報の有効化

パフォーマンスを監視する前に、Proxy Server で統計情報を有効にする必要があります。統計情報はサーバーマネージャーから、または obj.conf ファイルおよび magnus.conf ファイルを編集することにより行います。自動化されたツールを作成するユーザー、または監視と調整用にカスタマイズしたプログラムを作成するユーザーは、直接 stats-xml で作業する方がよい場合があります。

注意 – 統計情報およびプロファイリングを有効にすると、サーバーのすべてのユーザーが統計情報を使用できるようになります。

▼ サーバーマネージャーから統計情報を有効にするには

1. サーバーマネージャーにアクセスし、「**Server Status**」タブをクリックします。
2. 「**Monitor Current Activity**」をクリックします。
   「Monitor Current Activity」ページが表示されます。
3. 「**Activate Statistics/Profiling**」の「**Yes**」オプションを選択して、統計情報を有効にします。
4. 「了解」をクリックします。
5. 「**Restart Required**」をクリックします。
   「Apply Changes」ページが表示されます。
6. 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

▼ stats-xml を使用して統計情報を有効にするには

1. obj.conf ファイルのデフォルトオブジェクトの下に、次の行を追加します。

```
NameTrans fn="assign-name" name="stats-xml" from="
(/stats-xml|/stats-xml/.*)"
```

2. 次の **Service** 関数を obj.conf に追加します。

```
<Object name="stats-xml">

Service fn="stats-xml"

</Object>
```

3. stats-init **SAF** を obj.conf に追加します。

統計情報の使用法

統計情報を有効にすると、サーバーインスタンスの稼動状況に関するさまざまな情報を得ることができます。統計情報は、機能別に分類されます。

サーバーマネージャーでの統計情報の表示

この節では `proxystats.xml` データのサブセットをサーバーマネージャーで表示する方法について説明します。

合計、最大値、ピーク回数、Proxy Server との接続に関する情報の棒グラフ、DNS 処理、キープアライブ値、キャッシュ、サーバー要求を表示できます。

次の節では、各項目について取得できる情報の種類について説明します。

接続統計情報

サーバーマネージャーから次の接続統計情報を利用できます。

- 接続の合計数
- キューに入る接続の最大数
- キューに入る接続のピーク数
- キューに入っている現在の接続の数
- プロセス数

DNS 統計情報

サーバーマネージャーから次の DNS 統計情報を利用できます。

- 最大 DNS キャッシュエントリ
- プロセス数
- DNS キャッシュのヒット数 (棒グラフとしても表示)
- DNS キャッシュの欠落数 (棒グラフとしても表示)

キープアライブ統計情報

サーバーマネージャーから次のキープアライブ統計情報を利用できます。

- 最大キープアライブ接続
- キープアライブタイムアウト
- プロセス数
- キープアライブのヒット数 (棒グラフとしても表示)
- キープアライブのフラッシュ数 (棒グラフとしても表示)
- キープアライブの拒否数 (棒グラフとしても表示)
- キープアライブのタイムアウト数 (棒グラフとしても表示)

サーバー要求統計情報

サーバーマネージャーから次のサーバー統計情報を利用できます。

- 要求の合計数
- 受信したバイト数
- 送信したバイト数
- プロセス数
- 要求の HTTP サーバーコードによる分類 (棒グラフとしても表示)。たとえば、HTTP サーバーコード 200 は要求が遂行されたことを示します。

▼ 統計情報にアクセスするには

1. サーバーマネージャーにアクセスし、「**Server Status**」タブをクリックします。
2. 「**Monitor Current Activity**」をクリックします。
3. 「**Select Refresh Interval**」ドロップダウンリストから、更新間隔を選択します。
   更新間隔は、表示される統計情報の更新間隔を示す秒数です。
4. 「**Select Statistics To Be Displayed**」ドロップダウンリストから、表示する統計情報の種類を選択します。
   統計情報の種類については、213 ページの「サーバーマネージャーでの統計情報の表示」を参照してください。
5. 「送信」をクリックします。
   サーバーインスタンスが稼動中で、統計情報およびプロファイリングを有効にしている場合、選択した統計情報の種類を示すページが表示されます。このページは、更新間隔の値に応じて、5 ～ 15 秒ごとに更新されます。
6. ドロップダウンリストからプロセス **ID** を選択します。
   現在のアクティビティーはサーバーマネージャーから表示できますが、表示されるカテゴリはサーバーの調整に完全には関連していません。サーバーの調整には `perfdump` 統計情報をお勧めします。詳細については、次の節を参照してください。

perfdump ユーティリティーを使用した現在のアクティビティーの監視

`perfdump` ユーティリティーは Proxy Server に組み込まれた Server Application Function (SAF) であり、Proxy Server の内部統計情報からさまざまなパフォーマンスデータを収集し、ASCII テキストで表示します。`perfdump` ユーティリティーを使用することで、サーバーマネージャーから行うよりも幅広い統計情報を監視することができます。

perfdump ユーティリティーでは、統計情報が統合されます。単一のプロセスを監視するのではなく、統計情報をプロセス数で乗算するため、サーバーの全体像をより正確に把握することができます。

perfdump ユーティリティーの有効化

perfdump SAF は、 stats-xml 関数を有効にしてからのみ、有効にすることができます。

▼ perfdump SAF を有効にするには

1. obj.conf ファイルのデフォルトオブジェクトの後に、次のオブジェクトを追加します。

   ```
   <Object name="perf">

   Service fn="service-dump"

   </Object>
   ```

2. デフォルトオブジェクトに次の行を追加します。

   ```
   NameTrans fn=assign-name from="/.perf" name="perf"
   ```

3. サーバーソフトウェアを再起動します。

4. http:// *computer_name:proxyport*/.perf に移動して、perfdump にアクセスします。

   perfdump 統計情報を要求し、統計情報がブラウザで自動的に更新される頻度 (秒) を指定できます。次の例では、更新が 5 秒ごとに設定されています。

   ```
   http:// computer_name:proxyport/.perf?refresh=5
   ```

perfdump の出力例

次の例に、perfdump の出力例を示します。

```
proxyd pid: 6751

Sun Java System Web Proxy Server 4.0 B05/02/2005 15:32 (SunOS DOMESTIC)

Server started Thu May 19 13:15:14 2005
Process 6751 started Thu May 19 13:15:14 2005

ConnectionQueue:
----------------------------------------
Current/Peak/Limit Queue Length            0/1/4096
Total Connections Queued                   1
```

```
Average Queue Length (1, 5, 15 minutes)    0.00, 0.00, 0.00
Average Queueing Delay                     0.09 milliseconds

ListenSocket ls1:
-----------------------
Address                   http://0.0.0.0:8081
Acceptor Threads          1

KeepAliveInfo:
--------------------
KeepAliveCount        0/256
KeepAliveHits         0
KeepAliveFlushes      0
KeepAliveRefusals     0
KeepAliveTimeouts     0
KeepAliveTimeout      30 seconds

SessionCreationInfo:
------------------------
Active Sessions           1
Keep-Alive Sessions       0
Total Sessions Created    48/128

DiskCacheInfo:
------------------
Hit Ratio               0/0 (  0.00%)
Misses                  0
Cache files at startup  0
Cache files created     0
Cache files cleaned up  0

Native pools:
---------------------------
NativePool:
Idle/Peak/Limit               1/1/128
Work Queue Length/Peak/Limit  0/0/0

Server DNS cache disabled

Async DNS disabled

Performance Counters:
-----------------------------------------------
 ......................Average         Total      Percent

Total number of requests:                     1
Request processing time:    0.2559        0.2559
```

```
default-bucket (Default bucket)
Number of Requests:                             1    (100.00%)
Number of Invocations:                          7    (100.00%)
Latency:                    0.2483        0.2483    ( 97.04%)
Function Processing Time:   0.0076        0.0076    (  2.96%)
Total Response Time:        0.2559        0.2559    (100.00%)

Sessions:
---------------------------
Process  Status     Function
6751     response   service-dump
```

これらのパラメータの詳細については、『Sun ONE Web Server 6.1 *Performance Tuning, Sizing, and Scaling Guide*』の第 2 章の「Using Statistics to Tune Your Server」を参照してください。

perfdump 出力へのアクセスの制限

サーバーの perfdump 統計情報をブラウザに表示できるユーザーを制限する場合、/.perf URI 用の ACL を作成する必要があります。

ACL ファイルは、obj.conf ファイルの perf オブジェクト定義でも参照される必要があります。たとえば、/.perf URI に名前付きの ACL を作成した場合、次のようにして、オブジェクト定義の PathCheck 文の中で ACL ファイルを参照するようにする必要があります。

```
<Object name="perf">

PathCheck fn="check-acl" acl="perf.acl"

Service fn="service-dump"

</Object>
```

パフォーマンスバケットの使用

パフォーマンスバケットを使用すると、バケットを定義し、さまざまなサーバー関数にリンクすることができます。これらの関数のいずれかを呼び出すごとに、サーバーは統計データを収集し、それをバケットに追加します。たとえば、send-cgi と NSServletService は、それぞれ CGI と Java サーブレットの要求に使用される関数です。2 つのバケットを定義して CGI とサーブレットの要求に対して別々のカウンタを保持するか、または両方のタイプの動的コンテンツに対する要求をカウントするバケットを 1 つ作成することができます。この情報を収集するためにかかる負担はほとんどなく、サーバーパフォーマンスへの影響も通常はわずかです。この情報へはあとで perfdump ユーティリティーを使用してアクセスできます。

バケットには次の情報が格納されます。

- バケットの名前: この名前はバケットを関数に関連付ける場合に使用します。
- 説明: バケットが関連付けられている関数の説明。
- この関数の要求数: この関数を呼び出した要求の合計数。
- 関数が呼び出された回数: この数字は関数の要求数と一致しない場合があります。関数の中には 1 つの要求に対して複数回実行されることがあるためです。
- 関数の遅延またはディスパッチ時間: サーバーが関数の呼び出しに要した時間。
- 関数時間: 関数自体に費やされた時間。

`default-bucket` はサーバーにより事前に定義されています。default-bucket は、ユーザー定義のバケットに関連付けられていない関数の統計情報を記録します。

構成

パフォーマンスバケットのすべての設定情報を、`magnus.conf` ファイルと `obj.conf` ファイルで指定する必要があります。デフォルトのバケットのみが、自動的に有効に設定されています。

まず、214 ページの「perfdump ユーティリティーを使用した現在のアクティビティーの監視」の説明に従って、パフォーマンス測定を有効にする必要があります。

次の例は、`magnus.conf` ファイルで新しいバケットを定義する方法を示しています。

```
Init fn="define-perf-bucket" name="acl-bucket" description="ACL bucket"

Init fn="define-perf-bucket" name="file-bucket" description="Non-cached
responses"

Init fn="define-perf-bucket" name="cgi-bucket" description="CGI Stats"
```

上の例では、`acl-bucket`、 `file-bucket`、および `cgi-bucket` の 3 つのバケットが作成されます。これらのバケットを関数に関連付けるには、パフォーマンスを測定する `obj.conf` に `bucket=`*bucket-name* を追加します。

例

```
PathCheck fn="check-acl" acl="default" bucket="acl-bucket"

...

Service method="(GET|HEAD|POST)" type="*~magnus-internal/*" fn="send-file"
bucket="file-bucket"

...
```

```
<Object name="cgi">

ObjectType fn="force-type" type="magnus-internal/cgi"

Service fn="send-cgi" bucket="cgi-bucket"

</Object>
```

パフォーマンスレポート

バケットのサーバー統計情報には、perfdump ユーティリティーを使用してアクセスできます。パフォーマンスバケット情報は、perfdump によって返されるレポートの最後のセクションにあります。

レポートには次の情報が含まれています。

- 平均 (Average)、合計 (Total)、パーセント (Percent) の各列には、要求された各統計情報のデータが表示されます。
- 要求処理時間 (Request Processing Time) はサーバーがそれまでに受信したすべての要求を処理するために要した合計時間です。
- 要求数 (Number of Requests) は関数の要求の合計数です。
- 呼び出し数 (Number of Invocations) は、関数が呼び出された合計回数です。1 つの要求の処理中に、関数が何度も呼び出される場合があるため、この値は要求の数とは異なります。この行のパーセント列は、すべてのバケットの呼び出し回数の合計に基づいて計算されます。
- 遅延 (Latency) は Proxy Server が関数の呼び出しに要した時間 (秒) です。
- 関数処理時間 (Function Processing Time) は Proxy Server が関数の処理に費やした時間 (秒) です。Function Processing Time と Total Response Time のパーセンテージは、Request Processing Time の合計に基づいて計算されます。
- 合計応答時間 (Total Response Time) は Function Processing Time と Latency の合計 (秒) です。

次の例に、perfdump を使用して入手できるパフォーマンスバケット情報の例を示します。

```
Performance Counters:
----------------------------------------------
                           Average         Total      Percent

Total number of requests:                      1
Request processing time:    0.2559        0.2559

default-bucket (Default bucket)
Number of Requests:                            1    (100.00%)
Number of Invocations:                         7    (100.00%)
```

```
Latency:                    0.2483        0.2483    ( 97.04%)
Function Processing Time:   0.0076        0.0076    (  2.96%)
Total Response Time:        0.2559        0.2559    (100.00%)
```

SNMP の基本

SNMP は、ネットワークアクティビティーに関するデータをやり取りするために使用されるプロトコルです。SNMP では、管理対象デバイスとネットワーク管理ステーション (NMS) の間をデータが移動します。管理対象デバイスは、SNMP を使用するすべてのデバイス、つまり、ネットワーク上のホスト、ルーター、Proxy Server、その他のサーバーなどです。NMS は、そのネットワークをリモートで管理するために使用するシステムです。一般に、NMS ソフトウェアでは、収集されたデータをグラフに表示したり、そのデータを使用してサーバーが特定の許容範囲内で動作していることを確認したりします。

NMS は通常、1 つ以上のネットワーク管理アプリケーションがインストールされた強力なワークステーションです。HP OpenView のようなネットワーク管理アプリケーションでは、Web サーバーなどの管理対象デバイスに関する情報がグラフィカルに表示されます。この情報には、社内のどのサーバーが稼動またはダウンしているかを表示したり、受け取ったエラーメッセージの数と種類が含まれます。Proxy Server で SNMP を使用する場合、この情報は、サブエージェントとマスターエージェントという 2 種類のエージェントを使用して、NMS とサーバーの間で転送されます。

サブエージェントは、サーバーに関する情報を収集し、その情報をサーバーのマスターエージェントに渡します。管理サーバー以外のすべてのサーバーには、サブエージェントがあります。

注 – SNMP の設定を変更したあとは、「Apply Required」ボタンをクリックし、SNMP サブエージェントを再起動する必要があります。

マスターエージェントは、NMS と通信します。マスターエージェントは、Administration Server と一緒にインストールされます。

1 つのホストコンピュータに複数のサブエージェントをインストールできますが、マスターエージェントは 1 つしかインストールできません。たとえば、Directory Server、Proxy Server、および Messaging Server を同じホストにインストールしている場合、各サーバーのサブエージェントは、同じマスターエージェントと通信します。

Management Information Base

Proxy Server にはネットワーク管理に関する変数が格納されます。マスターエージェントがアクセスできる変数は、管理対象オブジェクトと呼ばれます。これらのオブジェクトは、Management Information Base (MIB) と呼ばれるツリー構造で定義されます。MIB により、Proxy Server のネットワーク設定、状態、および統計情報にアクセスすることができます。SNMP を使用すると、この情報を NMS から見ることができます。

MIB ツリーのトップレベルには、次の 4 つのサブツリーを持つインターネットオブジェクト識別子が表示されます。directory、mgmt、experimental、private。private サブツリーには、enterprises ノードが含まれています。enterprises ノードの各サブツリーは、個別の企業に割り当てられます。この企業は、独自の MIB 拡張機能を登録している組織です。企業は、自社のサブツリーの下に製品別のサブツリーを作成できます。企業が作成した MIB は、enterprises ノードの下に置かれます。また、Sun Java System サーバーの MIB が、enterprises ノードの下に置かれます。各 Sun Java Ssytem サーバーのサブエージェントには、SNMP 通信で使用する MIB が用意されています。サーバーは、これらの変数を含むメッセージまたはトラップを送信することによって、重大なイベントを NMS に報告します。NMS では、サーバーの MIB にデータを照会したり、MIB の変数をリモートで変更することもできます。各 Sun Java System サーバーには専用の MIB があります。すべての Sun Java System サーバーの MIB は次の場所にあります。

```
server-root/plugins/snmp
```

Proxy Server の MIB は、`proxyserv40.mib` という名前のファイルです。この MIB には、Proxy Server のネットワーク管理に関する各種変数の定義が格納されています。Proxy Server に関する管理情報を表示し、Proxy Server の MIB を使用してサーバーをリアルタイムで監視できます。

SNMP の設定

SNMP を使用する場合は、システムにマスターエージェントと 1 つ以上のサブエージェントがインストールされ、実行されている必要があります。サブエージェントを有効にする前に、マスターエージェントをインストールする必要があります。

SNMP の設定手順は、システムによって異なります。

設定を開始する前に、次の 2 つの点を確認する必要があります。

- SNMP エージェント (使用するオペレーティングシステムのネイティブエージェント) がシステムですでに稼動しているかどうか
- その場合、ネイティブ SNMP エージェントが SMUX 通信をサポートしていること (AIX プラットフォームを使用している場合、システムは SMUX をサポートしている)

この情報を確認する方法については、使用しているシステムのマニュアルを参照してください。

注 – 管理サーバーの SNMP の設定を変更したあと、新しいサーバーをインストールしたあと、または既存のサーバーを削除したあとは、次の手順を実行する必要があります。

- (Windows) Windows SNMP サービスを再起動するか、システムを再起動します。
- (UNIX) 管理サーバーを使用して SNMP マスターエージェントを再起動します。

表 10–1　SNMP のマスターエージェントおよびサブエージェントを有効にする手順の概要

サーバーが満たしている条件	...実行する手順。これらの手順については、次の項で説明。
■ ネイティブエージェントが現在実行されていない	1. マスターエージェントを起動します。 2. システムにインストールされている各サーバーのサブエージェントを有効にします。
■ ネイティブエージェントが現在実行されている ■ SMUX をサポートしていない ■ ネイティブエージェントの使用を継続する必要がない	1. 管理サーバーのマスターエージェントをインストールする場合は、ネイティブエージェントを停止します。 2. マスターエージェントを起動します。 3. システムにインストールされている各サーバーのサブエージェントを有効にします。
■ ネイティブエージェントが現在実行されている ■ SMUX をサポートしていない ■ ネイティブエージェントの使用を継続する必要がある	1. プロキシ SNMP エージェントをインストールします。 2. マスターエージェントを起動します。 3. プロキシ SNMP エージェントを起動します。 4. マスターエージェントのポート番号以外のポート番号を使用して、ネイティブエージェントを再起動します。 5. システムにインストールされている各サーバーのサブエージェントを有効にします。
■ ネイティブエージェントが現在実行されている ■ SMUX をサポートしている	1. SNMP ネイティブエージェントを再設定します。 2. システムにインストールされている各サーバーのサブエージェントを有効にします。

プロキシ SNMP エージェントの使用 (UNIX)

ネイティブエージェントがすでに実行されていて、今後も Proxy Server のマスターエージェントとともに使用し続ける場合には、プロキシ SNMP エージェントを使用する必要があります。ここでの手順を始める前に、ネイティブのマスターエージェントを停止してください。詳細については、使用しているシステムのマニュアルを参照してください。

注 – プロキシエージェントを使用するには、このエージェントをインストールして起動する必要があります。さらに、Proxy Server のマスターエージェントが実行されているポート番号以外のポート番号を使用して、ネイティブ SNMP エージェントを再起動する必要があります。

この節では、次の内容について説明します。

プロキシ SNMP エージェントのインストール

SNMP がシステムで稼動中で、ネイティブ SNMP デーモンの使用を継続する必要がある場合は、次の手順に従います。

▼ プロキシ SNMP エージェントをインストールするには

1. **SNMP** マスターエージェントをインストールします。
   225 ページの「SNMP マスターエージェントのインストール」を参照してください。
2. プロキシ **SNMP** エージェントをインストールして起動し、ネイティブ **SNMP** デーモンを再起動します。
   223 ページの「プロキシ SNMP エージェントの使用 (UNIX)」を参照してください。
3. **SNMP** マスターエージェントを起動します。
   226 ページの「SNMP マスターエージェントの有効化と起動」を参照してください。
4. サブエージェントを有効にします。
   231 ページの「サブエージェントの有効化」を参照してください。

   SNMP プロキシエージェントをインストールするには、サーバーのルートディレクトリの `plugins/snmp/sagt` にある `CONFIG` ファイルを編集します。SNMP デーモンが待

機するポートを追加します。また、このファイルには、プロキシ SNMP エージェントが転送する MIB ツリーおよびトラップも指定します。

次に CONFIG ファイルの例を示します。

```
AGENT AT PORT 1161 WITH COMMUNITY public
SUBTREES       1.3.6.1.2.1.1,
               3.6.1.2.1.2,
               1.3.6.1.2.1.3,
               1.3.6.1.2.1.4,
               1.3.6.1.2.1.5,
               1.3.6.1.2.1.6,
               1.3.6.1.2.1.7,
               1.3.6.1.2.1.8
FORWARD ALL TRAPS;
```

プロキシ SNMP エージェントの起動

プロキシ SNMP エージェントを起動するには、コマンドプロンプトで次のように入力します。

```
# sagt -c CONFIG&
```

ネイティブ SNMP デーモンの再起動

プロキシ SNMP エージェントを起動したあと、CONFIG ファイルで指定したポートでネイティブ SNMP デーモンを再起動します。ネイティブ SNMP デーモンを再起動するには、コマンドプロンプトで次のように入力します。

```
# snmpd -P port-number
```

port-number は CONFIG ファイルで指定したポート番号を示します。たとえば、Solaris プラットフォームで、前に示した CONFIG ファイル例のポート番号を使用する場合は、次のように入力します。

```
# snmpd -P 1161
```

SNMP ネイティブエージェントの再設定

SNMP デーモンが AIX で稼動している場合は、SMUX がサポートされています。このため、マスターエージェントをインストールする必要はありません。ただし、AIX の SNMP デーモンの設定を変更する必要があります。

AIX では、いくつかの設定ファイルを使用して通信内容を制限しています。snmpd.conf を編集して、SNMP デーモンが SMUX サブエージェントからの受信

メッセージを受け入れるようにする必要があります。詳細は、オンラインマニュアルの snmpd.conf のページを参照してください。このファイルに、各サブエージェントを定義する行を追加します。

たとえば、次の行を snmpd.conf に追加します。

smux 1.3.6.1.4.1.1.1450.1 “ “ *IP-address net-mask*

IP_address は、そのサブエージェントを実行するホストの IP アドレス、net_mask は、そのホストのネットワークマスクを示します。

注 – ループバックアドレスの 127.0.0.1 は使用できません。実 IP アドレスを使用してください。

SNMP マスターエージェントのインストール

SNMP マスターエージェントを設定するには、root ユーザーとして管理サーバーインスタンスをインストールする必要があります。ただし、root 以外のユーザーでも、マスターエージェントとやり取りを行う SNMP サブエージェントを設定することで、MIB のブラウズなどの基本的な SNMP タスクを Web サーバーインスタンスから実行できます。

▼ マスター SNMP エージェントをインストールするには

1. **root** としてログインします。
2. **SNMP** デーモン (snmpd) がポート **161** で実行されているかどうか確認します。
   - **SNMP** デーモンが実行されていない場合は、**225** ページの「**SNMP** マスターエージェントのインストール」に進みます。
   - **SNMP** デーモンが実行されている場合は、その再起動方法と、どの **MIB** ツリーをサポートしているかを確認します。次に、そのプロセスを終了します。
3. 管理サーバーで、「**Global Settings**」タブから「**Set SNMP Master Agent Trap**」リンクを選択します。
4. ネットワーク管理ソフトウェアを実行するシステムの名前を入力します。

5 ネットワーク管理システムがトラップを待機するポート番号を入力します。一般的なポートは **162** です。
トラップについては、231 ページの「トラップ送信先の設定」を参照してください。

6 トラップで使用するコミュニティー文字列を入力します。
コミュニティー文字列については、231 ページの「コミュニティー文字列の設定」を参照してください。

7 「了解」をクリックします。

8 管理サーバーで、「**Global Settings**」タブから「**Set SNMP Master Agent Community**」リンクを選択します。

9 マスターエージェントのコミュニティー文字列を入力します。

10 コミュニティーの動作を選択します。

11 「新規」をクリックします。

SNMP マスターエージェントの有効化と起動

マスターエージェントの動作は、`CONFIG` という名前のエージェント設定ファイルに定義されています。サーバーマネージャーを使用して `CONFIG` ファイルを編集できます。また、手動でこのファイルを編集することもできます。SNMP サブエージェントを有効にするためには、マスター SNMP エージェントをインストールする必要があります。

マスターエージェントを再起動しようとしたときに、`System Error:Could not bind to port` のようなバインドエラーメッセージが発生する場合は、 `ps -ef | grep snmp` を使用して、`magt` が実行されているかどうかを確認します。実行されている場合は、`kill -9 pid` コマンドを使用して、そのプロセスを終了します。SNMP 用の CGI がふたたび機能し始めます。

この節では、次の内容について説明します。

別のポートでのマスターエージェントの起動

管理インタフェースでは、161 以外のポートで SNMP マスターエージェントを起動することはありません。

▼ 別のポートでマスターエージェントを手動で起動するには

1. /*server-root* /plugins/snmp/magt/CONFIG ファイルに目的のポートを指定します。
2. 次のように起動スクリプトを実行します。

   cd / *server-root*/proxy-admserv

   ./start -shell /*server-root*/plugins/snmp/magt/magt

   /*server-root* /plugins/snmp/magt/CONFIG

   / *server-root*/plugins/snmp/magt/INIT

   これで、マスターエージェントが目的のポートで起動します。マスターエージェントが動作していることは、ユーザーインタフェースから検出できます。

手動による SNMP マスターエージェントの設定

▼ SNMP マスターエージェントを手動で設定するには

1. スーパーユーザーとしてログインします。
2. **SNMP** デーモン (snmpd) がポート **161** で実行されているかどうか確認します。

   SNMP デーモンが実行されている場合は、その再起動方法と、どの MIB ツリーをサポートしているかを確認します。次に、そのプロセスを終了します。
3. サーバーのルートディレクトリの plugins/snmp/magt にある CONFIG ファイルを編集します。
4. (オプション) CONFIG ファイルに sysContact 変数と sysLocation 変数を定義します。

マスターエージェントの CONFIG ファイルの編集

▼ マスターエージェントの CONFIG ファイルを編集するには

1 スーパーユーザーとしてログインします。

2 ポート 161 上で実行されている SNMP デーモン (snmpd) があるかどうかを確認します。
SNMP デーモンが実行されている場合は、その再起動方法と、どの MIB ツリーをサポートしているかを確認します。次に、そのプロセスを終了します。

3 サーバーのルートディレクトリの plugins/snmp/magt にある CONFIG ファイルを編集します。

4 (オプション) CONFIG ファイルに sysContact 変数と sysLocation 変数を定義します。

sysContact 変数と sysLocation 変数の定義

CONFIG ファイルの sysContact エントリと sysLocation エントリは sysContact 変数と sysLocation MIB-II 変数を指定します。この例では、sysContact および sysLocation に指定する文字列が引用符で囲まれています。空白文字、改行、タブなどを含む文字列は、引用符で囲む必要があります。また、16 進法表記で値を指定することもできます。

sysContract 変数および sysLocation 変数が定義された CONFIG ファイルの例を次に示します。

```
COMMUNITY public

ALLOW ALL OPERATIONS

MANAGER nms2

SEND ALL TRAPS TO PORT 162

WITH COMMUNITY public

INITIAL sysLocation "Server room

987 East Cannon RoadMountain View, CA 94043 USA" INITIAL sysContact "Jill Dawson
email: jdawson@example.com"
```

SNMP サブエージェントの設定

SNMP サブエージェントを設定してサーバーを監視することができます。

▼ SNMP サブエージェントを設定するには

1 サーバーマネージャーにアクセスし、「**Server Status**」タブをクリックします。

2 「**Configure SNMP SubagentSNMP**」リンクをクリックします。
「Configure SNMP Subagent」 ページが表示されます。

3 「**Master Host**」フィールドにサーバーの名前とドメインを入力します。

4 「**Description**」にサーバーの説明 (オペレーティングシステム情報を含む) を指定します。

5 「**Organization**」にサーバーを管理する組織を指定します。

6 「場所」フィールドにサーバーの絶対パスを指定します。

7 「**Contact**」フィールドにサーバーの担当者名と、担当者の連絡先情報を指定します。

8 「**Enable the SNMP Statistics Collection**」の「**On**」を選択します。

9 「了解」をクリックします。

10 「**Restart Required**」をクリックします。
「Apply Changes」 ページが表示されます。

11 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

SNMP マスターエージェントの起動

SNMP マスターエージェントはインストールしたあと、手動で、または管理サーバーを使用して起動できます。

手動による SNMP マスターエージェントを起動するには

マスターエージェントを手動で起動するには、コマンドプロンプトで次のコマンドを入力します。

```
# magt CONFIG INIT&
```

`INIT` ファイルは、システムの場所や連絡先情報など、MIB-II システムグループからの情報が格納された不揮発性ファイルです。`INIT` ファイルがない場合は、マスター

エージェントを最初に起動した時に作成されます。無効なマネージャー名が CONFIG ファイルに指定されていると、マスターエージェントの起動が失敗する原因になります。

マスターエージェントを標準以外のポートで起動するには、次の 2 つの方法のどちらかを使用します。

方法 **1:** CONFIG ファイルに、マスターエージェントがマネージャーからの SNMP 要求を待機する各インタフェースのトランスポートマッピングを指定します。トランスポートマッピングを指定することで、マスターエージェントは標準ポートと標準以外のポートで接続を受け入れることができます。また、マスターエージェントは、標準以外のポートで SNMP トラフィックを受け入れることもできます。同時に使用可能な SNMP の最大数は、プロセス当たりのオープンソケット数またはファイル記述子数に関するシステムの制限値によって制限されます。トランスポートマッピングのエントリ例を次に示します。

```
TRANSPORT extraordinary SNMP

OVER UDP SOCKET

AT PORT 11161
```

CONFIG ファイルを手動で編集したあと、コマンドプロンプトで次のように入力して、手動でマスターエージェントを起動する必要があります。

```
# magt CONFIG INIT&
```

方法 **2:** /etc/services ファイルを編集して、マスターエージェントが標準ポートと標準以外のポートで接続を受け入れられるようにします。

▼ 管理サーバーを使用して **SNMP** マスターエージェントを起動するには

1. 管理サーバーにログインします。
2. 管理サーバーで、「**Global Settings**」タブから「**Control SNMP Master Agent**」ページを選択します。
3. 「開始」をクリックします。

   「Control SNMP Master Agent」ページから、SNMP エージェントの停止および再起動も実行できます。

SNMP マスターエージェントの設定

マスターエージェントを有効にし、ホストコンピュータのサブエージェントを有効にしたあと、ホストの管理サーバーを設定する必要があります。この設定では、コミュニティー文字列とトラップ送信先を指定します。

コミュニティー文字列の設定

コミュニティー文字列は、SNMP エージェントが認証に使用するテキスト文字列です。ネットワーク管理ステーションは、エージェントに送信する各メッセージと一緒にコミュニティー文字列を送信します。この結果、エージェントは、そのネットワーク管理ステーションが情報の取得を承認されているかどうかを確認できます。コミュニティー文字列は、SNMP パケットでの送信時に秘匿されません。文字列は ASCII テキストで送信されます。

SNMP マスターエージェントのコミュニティー文字列は、管理サーバーの「Set SNMP Master Agent Community」ページから設定できます。また、特定のコミュニティーで実行できる SNMP 関連オペレーションを定義することもできます。管理サーバーから、設定済みのコミュニティーの表示、編集、および削除を行うこともできます。

トラップ送信先の設定

SNMP トラップとは、SNMP エージェントがネットワーク管理ステーションに送信するメッセージのことです。たとえば、SNMP エージェントは、インタフェースの状態が稼動から停止に変わった時にトラップを送信します。SNMP エージェントにトラップの送信先がわかるように、ネットワーク管理ステーションのアドレスを設定する必要があります。SNMP マスターエージェントのトラップ送信先は、Proxy Server から設定できます。また、設定済みのトラップ送信先の表示、編集、および削除を行うこともできます。Proxy Server を使用してトラップ送信先を設定する場合、実際には、`CONFIG` ファイルを編集することになります。

サブエージェントの有効化

管理サーバーに付属するマスターエージェントをインストールしたあと、そのマスターエージェントを起動する前に、サーバーインスタンスのサブエージェントを有効にする必要があります。詳細については、225 ページの「SNMP マスターエージェントのインストール」を参照してください。サーバーマネージャーを使用してサブエージェントを有効にできます。

UNIX プラットフォームまたは Linux プラットフォームで SNMP 機能を停止する場合は、サブエージェントを先に停止し、その後でマスターエージェントを停止する必

要があります。マスターエージェントを先に停止すると、サブエージェントを停止できなくなることがあります。そうなった場合は、マスターエージェントを再起動し、サブエージェントを停止し、次にマスターエージェントを停止します。

SNMP サブエージェントを有効にするには、サーバーマネージャーの「Configure SNMP SubagentSNMP」ページを使用して、「Control SNMP Subagent」ページからサブエージェントを起動します。詳細については、オンラインヘルプの対応する項目を参照してください。

サブエージェントを有効にすると、「*Control SNMP Subagent*」ページ、または Windows のコントロールパネルの「サービス」からそのサブエージェントを起動、停止、または再起動できます。

注 – SNMP の設定を変更したあとは、「Apply Required」ボタンをクリックし、SNMP サブエージェントを再起動する必要があります。

SNMP メッセージについて

GET および SET は、SNMP で定義されている 2 種類のメッセージです。GET メッセージと SET メッセージは、ネットワーク管理ステーション (NMS) によってマスターエージェントに送信されます。管理サーバーで、これらのメッセージを使用できます。

SNMP は、プロトコルデータユニット (PDU) の形式でネットワーク情報をやり取りします。このユニットには、Web サーバーなどの管理対象デバイスに保存された変数に関する情報が格納されます。これらの変数は管理対象オブジェクトとも呼ばれ、必要に応じて NMS に報告される値と名前を保持しています。サーバーから NMS に送信されるプロトコルデータユニットはトラップと呼ばれます。次の例では、NMS またはサーバーによって開始された通信での GET 、SET、およびトラップの各メッセージの使用法を示します。

NMS 主導の通信: NMS は、サーバーからの情報を要求するか、サーバーの MIB 内に格納されている変数の値を変更します。次に例を示します。

1. NMS は、管理サーバーのマスターエージェントにメッセージを送信します。このメッセージは、データの要求 (GET メッセージ) の場合と、MIB の変数を設定する命令 (SET メッセージ) の場合があります。
2. マスターエージェントは、そのメッセージを適切なサブエージェントに転送します。
3. サブエージェントは、データを取り出すか、または MIB 内の変数を変更します。
4. サブエージェントは、マスターエージェントにデータまたは状態を報告します。次に、マスターエージェントは GET メッセージを NMS に返送します。

5. NMS は、ネットワーク管理アプリケーションを通して、そのデータを文字またはグラフィックで表示します。

   サーバー主導の通信: サーバーのサブエージェントは、重大なイベントが発生した時に、メッセージまたはトラップを NMS を送信します。次に例を示します。

6. サブエージェントは、サーバーが停止したことをマスターエージェントに通知します。
7. マスターエージェントは、イベントを報告するメッセージまたはトラップを NMS に送信します。
8. NMS は、ネットワーク管理アプリケーションを通して、その情報を文字またはグラフィックで表示します。

◆ ◆ ◆ 第 11 章

11 URLのプロキシ設定とルーティング

この章では、Proxy Server が要求を処理する方法について説明します。また、特定のリソースにプロキシを有効にする方法についても説明します。この章では、URL を異なる URL またはサーバーにルーティングするように Proxy Server を設定する方法についても説明します。

この章の内容は次のとおりです。

リソースに対するプロキシの有効化/無効化

リソースに対してプロキシを有効または無効にできます。リソースは個々の URL、共通の要素を含む URL のグループ、またはプロトコル全体を指します。プロキシをサーバー全体、さまざまなリソース、またはテンプレートファイルに指定されたリソースに対して有効にするかどうかを制御できます。そのリソースに対するプロキシを無効にすることで、1 つ以上の URL へのアクセスを拒否できます。この設定はリソースに対するすべてのアクセスを拒否または許可するグローバルな方法です。また、URL フィルタを使用してリソースへのアクセスを許可または拒否することもできます。URL フィルタについては、310 ページの「URL のフィルタリング」を参照してください。

▼ リソースに対するプロキシを有効にするには

1. サーバーマネージャーにアクセスし、「**Routing**」タブをクリックします。
2. 「**Enable/Disable Proxying**」リンクをクリックします。
   「Enable/Disable Proxying」ページが表示されます。
3. ドロップダウンリストからリソースを選択するか、「**Regular Expression**」ボタンをクリックして正規表現を入力し、「了解」をクリックします。
4. 指定したリソースに対してデフォルト設定を選択できます。
   - 「**Use Default Setting Derived From A More General Resource**」**:**このリソースを含むより一般的なリソースに対する設定が、このリソースに使用されます。
   - 「**Do Not Proxy This Resource**」**:**プロキシを通じてこのリソースに到達できなくなります。
   - 「**Enable Proxying Of This Resource**」**:**クライアントはプロキシを通じてこのリソースにアクセスできるようになります (クライアントがその他のセキュリティーおよび認証チェックをパスした場合)。リソースに対するプロキシを有効にした場合、すべてのメソッドが有効になります。GET、HEAD、INDEX、POST、SSL トンネリングのための CONNECT を含む読み込みメソッド、および PUT、MKDIR、RMDIR、MOVE、DELETE を含む書き込みメソッドはすべて、そのリソースに対して有効になります。その他のセキュリティーチェックを禁止し、すべてのクライアントは読み込みおよび書き込みアクセス権を持ちます。
5. 「了解」をクリックします。
6. 「**Restart Required**」をクリックします。
   「Apply Changes」ページが表示されます。
7. 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

別のプロキシを経由したルーティング

「Set Routing Preferences」ページは、派生したデフォルト設定または直接の接続を使用して、あるいはプロキシ配列、隣接 ICP、別のプロキシサーバー、または SOCKS サーバーを通じて特定のリソースをルーティングするように Proxy Server を設定する場合に使用します。

リソースのルーティング設定

▼ リソースのルーティングを設定するには

1 サーバーマネージャーにアクセスし、「**Routing**」タブをクリックします。

2 「**Set Routing Preferences**」リンクをクリックします。
「Set Routing Preferences」ページが表示されます。

3 ドロップダウンリストからリソースを選択するか、「**Regular Expression**」ボタンをクリックして正規表現を入力し、「了解」をクリックします。

4 設定を行うリソースに使用するルーティングタイプを選択します。
利用できるオプションは次のとおりです。

- 「**Derived Default Configuration**」:Proxy Server は、リモートサーバーまたは別のプロキシを使用すべきかどうかを判断する場合、より汎用的なテンプレートつまり、正規表現に一致する短いテンプレートを使用します。たとえば、プロキシが `http://.*` の要求をすべて別のプロキシサーバーにルーティングし、`http://www.*` の要求をすべてリモートサーバーにルーティングする場合、`http://www.example.*` の要求に対して派生するデフォルト設定のルーティングを作成できます。この要求は `http://www.*` テンプレートの設定に基づき、直接リモートサーバーに移動します。
- 「**Direct Connections**」:要求はプロキシを経由せず、常に直接リモートサーバーに移動します。
- 「**Route Through A SOCKS Server**」:指定されたリソースの要求は、SOCKS サーバーを経由してルーティングされます。このオプションを選択した場合、プロキシサーバーが経由する SOCKS サーバーの名前または IP アドレスとポート番号を指定します。
- 「**Route Through**」:プロキシ配列、隣接 ICP、親配列、またはプロキシサーバーを経由してルーティングするかどうかを指定できます。ここで複数のルーティング方法を選択した場合、プロキシはフォームに示される階層 (すなわち、プロキシ配列、リダイレクト、ICP、親配列、別のプロキシ) に従って移動します。プロキシサーバーを経由したルーティングについては、238 ページの「プロキシサーバーの連鎖」を参照してください。

  SOCKS サーバーを経由したルーティングについては、239 ページの「SOCKS サーバーを経由したルーティング」を参照してください。プロキシ配列、親配列、または隣接 ICP を経由したルーティングについては、第 12 章を参照してください。

注 – 443 以外のポート上での接続要求のルーティングを有効にするには、`obj.conf` ファイルの `ppath` パラメータを `connect://.*` に変更する必要があります。

5 「了解」をクリックします。

6 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

7 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

プロキシサーバーの連鎖

リモートサーバーにアクセスする代わりに、プロキシから一部のリソース用の別のプロキシにアクセスするように設定できます。連鎖はファイアウォールの背後で複数のプロキシを構成する便利な方法です。また、連鎖により階層キャッシュを構築することもできます。

▼ 別のプロキシサーバーを経由してルーティングするには

1 サーバーマネージャーにアクセスし、「**Routing**」タブをクリックします。

2 「**Set Routing Preferences**」リンクをクリックします。
「Set Routing Preferences」ページが表示されます。

3 ドロップダウンリストからリソースを選択するか、「**Regular Expression**」ボタンをクリックして正規表現を入力し、「了解」をクリックします。

4 同じページの「**Routing Through Another Proxy**」セクションの「**Route Through**」オプションを選択します。

5 「**Another Proxy**」チェックボックスを選択します。

6 「**Another Proxy**」フィールドに、経由するプロキシサーバーの名前とポート番号を入力できます。
次の形式でサーバーの名前とポート番号を入力します。servername:port

7 「了解」をクリックします。

8 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

9 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

SOCKS サーバーを経由したルーティング

ネットワーク上ですでにリモート SOCKS サーバーが稼働している場合、このリモート SOCKS サーバーに接続して特定のリソースにアクセスするようにプロキシを設定できます。

▼ SOCKS サーバーを経由してルーティングするには

1 サーバーマネージャーにアクセスし、「**Routing**」タブをクリックします。

2 「**Set Routing Preferences**」リンクをクリックします。
「Set Routing Preferences」ページが表示されます。

3 ドロップダウンリストからリソースを選択するか、「**Regular Expression**」ボタンをクリックして正規表現を入力し、「了解」をクリックします。

4 同じページの「**Routing Through Another Proxy**」セクションの「**Route Through**」オプションを選択します。

5 「**Route Through SOCKS Server**」オプションを選択します。

6 プロキシサーバーが経由する **SOCKS** サーバーの名前または **IP** アドレスとポート番号を指定します。

7 「了解」をクリックします。

8 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

9 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

次の手順

SOCKS サーバーを経由したルーティングを有効にした場合、「SOCKS v5 Routing」ページを使用してプロキシの経路を作成する必要があります。プロキシの経路から、プロキシが経由する SOCKS サーバーからアクセス可能な IP アドレスが特定されます。またプロキシの経路では、SOCKS サーバーが直接ホストに接続するかどうかも指定されます。

クライアント IP アドレスのサーバーへの転送

「Forward Client Credentials」ページは、クライアント証明をリモートサーバーに送信するようにプロキシを設定する場合に使用します。

▼ クライアント IP アドレスを送信するようにプロキシを設定するには

1. サーバーマネージャーにアクセスし、「**Routing**」タブをクリックします。
2. 「**Forward Client Credentials**」リンクをクリックします。
   「Forward Client Credentials」ページが表示されます。
3. ドロップダウンリストからリソースを選択するか、「**Regular Expression**」ボタンをクリックして正規表現を入力し、「了解」をクリックします。
4. 転送オプションを設定します。
   - 「**Client IP Addressing Forwarding**」**:**Proxy Server は、ドキュメントを要求する場合、クライアントの IP アドレスをリモートサーバーに送信しません。その代わり、プロキシがクライアントとして機能し、IP アドレスをリモートサーバーに送信します。ただし、次の場合には、ユーザーがクライアントの IP アドレスを渡した方がよい場合があります。
     - プロキシが内部プロキシの連鎖の 1 つである場合。
     - クライアントが、クライアントの IP アドレスの取得に依存するサーバーにアクセスする必要がある場合。クライアントの IP アドレスを特定のサーバーにのみ送信する場合は、テンプレートを使用できます。

     オプションを設定し、クライアントの IP アドレスを送信するようにプロキシを設定します。
     - 「**Default**」**:**Proxy Server がクライアントの IP アドレスを転送できるようにします。
     - 「**Blocked**」**:**プロキシによるクライアントの IP アドレスの転送を許可しません。
     - 「**Enabled Using HTTP Header**」**:**IP アドレスを転送する場合に使用するプロキシの HTTP ヘッダーを指定できます。デフォルトの HTTP ヘッダーの名前は `Client-ip` ですが、選択したどのヘッダーでも IP アドレスを送信できます。
   - 「**Client Proxy Authentication Forwarding**」**:**オプションを設定し、クライアント認証の詳細を送信するようにプロキシを設定します。
     - 「**Default**」**:**Proxy Server がクライアント認証の詳細を転送できるようにします。

- 「**Blocked**」**:**プロキシによるクライアント認証の詳細の転送を許可しません。
- 「**Enabled Using HTTP Header**」**:**認証の詳細を転送する場合に使用するプロキシの HTTP ヘッダーを指定できます。

- 「**Client Cipher Forwarding**」**:**オプションを設定し、リモートサーバーにクライアントの SSL/TLS 暗号化方式群の名前を送信するようにプロキシを設定します。
  - 「**Default**」**:**Proxy Server が、リモートサーバーにクライアントの SSL/TLS 暗号化方式群の名前を転送できるようにします。
  - 「**Blocked**」**:**プロキシが、リモートサーバーにクライアントの SSL/TLS 暗号化方式群の名前を転送するのを許可しません。
  - 「**Enabled Using HTTP Header**」**:**リモートサーバーにクライアントの SSL/TLS 暗号化方式群の名前を転送する場合に使用するプロキシの HTTP ヘッダーを指定できます。デフォルトの HTTP ヘッダーの名前は `Proxy-cipher` ですが、選択したどのヘッダーでもクライアントの SSL/TLS 暗号化方式群の名前を送信できます。
- 「**Client Keysize Forwarding**」**:**オプションを設定し、リモートサーバーにクライアントの SSL/TLS キーのサイズを送信するようにプロキシを設定します。
  - 「**Default**」**:**Proxy Server が、リモートサーバーにクライアントの SSL/TLS キーのサイズを転送できるようにします。
  - 「**Blocked**」**:**プロキシが、リモートサーバーにクライアントの SSL/TLS キーのサイズを転送するのを許可しません。
  - 「**Enabled Using HTTP Header**」**:**リモートサーバーにクライアントの SSL/TLS キーのサイズを転送する場合に使用するプロキシの HTTP ヘッダーを指定できます。デフォルトの HTTP ヘッダーの名前は `Proxy-keysize` ですが、選択したどのヘッダーでもクライアントのSSL/TLS キーのサイズを送信できます。
- 「**Client Secret Keysize Forwarding**」**:**オプションを設定し、リモートサーバーにクライアントの SSL/TLS 秘密鍵のサイズを送信するようにプロキシを設定します。
  - 「**Default**」**:**Proxy Server が、リモートサーバーにクライアントの SSL/TLS 秘密鍵のサイズを転送できるようにします。
  - 「**Blocked**」**:**プロキシが、リモートサーバーにクライアントの SSL/TLS 秘密鍵のサイズを転送するのを許可しません。
  - 「**Enabled Using HTTP Header**」**:**リモートサーバーにクライアントの SSL/TLS 秘密鍵のサイズを転送する場合に使用するプロキシの HTTP ヘッダーを指定できます。デフォルトの HTTP ヘッダーの名前は `Proxy-secret-keysize` ですが、選択したどのヘッダーでもクライアントのSSL/TLS 秘密鍵のサイズを送信できます。
- 「**Client SSL Session ID Forwarding**」**:**オプションを設定し、リモートサーバーにクライアントの SSL/TLS セッション ID を送信するようにプロキシを設定します。
  - 「**Default**」**:**Proxy Server が、リモートサーバーにクライアントの SSL/TLS セッション ID を転送できるようにします。

- 「**Blocked**」:プロキシが、リモートサーバーにクライアントのSSL/TLSセッションIDを転送するのを許可しません。
- 「**Enabled Using HTTP Header**」:リモートサーバーにクライアントのSSL/TLSセッションIDを転送する場合に使用するプロキシのHTTPヘッダーを指定できます。デフォルトのHTTPヘッダーの名前は`Proxy-ssl-id`ですが、選択したどのヘッダーでもクライアントのSSL/TLSセッションIDを送信できます。

- 「**Client Issuer DN Forwarding**」:オプションを設定し、リモートサーバーにクライアントのSSL/TLS証明書の発行者の識別名を送信するようにプロキシを設定します。
  - 「**Default**」:Proxy Serverが、リモートサーバーにクライアントのSSL/TLS証明書の発行者の識別名を転送できるようにします。
  - 「**Blocked**」:プロキシが、リモートサーバーにクライアントのSSL/TLS証明書の発行者の識別名を転送するのを許可しません。
  - 「**Enabled Using HTTP Header**」:リモートサーバーにクライアントのSSL/TLS証明書の発行者の識別名を転送する場合に使用するプロキシのHTTPヘッダーを指定できます。デフォルトのHTTPヘッダーの名前は`Proxy-issuer-dn`ですが、選択したどのヘッダーでもクライアントのSSL/TLS証明書の発行者の名前を送信できます。
- 「**Client User DN Forwarding**」:オプションを設定し、リモートサーバーにクライアントのSSL/TLS証明書のサブジェクト識別名を送信するようにプロキシを設定します。
  - 「**Default**」:Proxy Serverが、リモートサーバーにクライアントのSSL/TLS証明書のサブジェクトの識別名を転送できるようにします。
  - 「**Blocked**」:プロキシが、リモートサーバーにクライアントのSSL/TLS証明書のサブジェクト識別名を転送するのを許可しません。
  - 「**Enabled Using HTTP Header**」:リモートサーバーにクライアントのSSL/TLS証明書のサブジェクトの識別名を転送する場合に使用するプロキシのHTTPヘッダーを指定できます。デフォルトのHTTPヘッダーの名前は`Proxy-user-dn`ですが、選択したどのヘッダーでもクライアントのSSL/TLS証明書のサブジェクトの名前を送信できます。
- 「**Client SSL/TLS Certificate Forwarding**」:オプションを設定し、リモートサーバーにクライアントのSSL/TLS証明書を送信するようにプロキシを設定します。
  - 「**Default**」:Proxy Serverが、リモートサーバーにクライアントのSSL/TLS証明書を転送できるようにします。
  - 「**Blocked**」:プロキシが、リモートサーバーにクライアントのSSL/TLS証明書を転送するのを許可しません。
  - 「**Enabled Using HTTP Header**」:リモートサーバーにクライアントのSSL/TLS証明書を転送する場合に使用するプロキシのHTTPヘッダーを指定できます。デフォルトのHTTPヘッダーの名前は`Proxy-auth-cert`ですが、選択したどのヘッダーでもクライアントのSSL/TLS証明書を送信できます。

- 「**Client Cache Information Forwarding**」**:**いずれかのオプションを選択し、ローカルのキャッシュのヒットに関する情報を送信するようにプロキシを設定します。
  - 「**Default**」**:**Proxy Server がリモートサーバーにローカルのキャッシュのヒットに関する情報を転送できるようにします。
  - 「**Blocked**」**:**プロキシがリモートサーバーにローカルのキャッシュのヒットに関する情報を転送するのを許可しません。
  - 「**Enabled Using HTTP Header**」**:**リモートサーバーにローカルのキャッシュのヒットに関する情報を転送する場合に使用するプロキシの HTTP ヘッダーを指定できます。デフォルトの HTTP ヘッダーの名前は `Cache-info` ですが、選択したどのヘッダーでもローカルのキャッシュのヒットに関する情報を送信できます。
- 「**Set Basic Authentication Credentials**」**:**オプションを設定し、HTTP 要求を送信するようにプロキシを設定します。
  - 「**User**」**:**認証するユーザーを指定します。
  - 「**Password**」**:** ユーザーのパスワードを指定します。
  - 「**Using HTTP Header**」**:**資格の伝達に使用するプロキシの HTTP ヘッダーを指定できます。

5 「了解」をクリックします。

6 「**Restart Required**」をクリックします。「**Apply Changes**」ページが表示されます。

7 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

クライアントによる **IP** アドレスの確認の許可

ネットワークのセキュリティーを維持するために、特定の IP アドレスに対してアクセスを制限する機能がクライアント側に設定されている場合があります。クライアントがこの機能を使用できるように、Proxy Server では Java IP アドレスの確認をサポートしています。このサポートにより、クライアントはリソースの取得に使用される IP アドレスを Proxy Server に照会できます。この機能が有効になっている場合、クライアントは Proxy Server による配信元サーバーの IP アドレスの送信を要求できます。Proxy Server はヘッダーに IP アドレスを添付します。クライアントが配信元サーバーの IP アドレスを取得すると、クライアントはそのあとの接続に同じ IP アドレスを使用するように明示的に指定できます。

▼ Java IP アドレスを確認するには

1 サーバーマネージャーにアクセスし、「**Routing**」タブをクリックします。

2 「**Check Java IP Address**」リンクをクリックします。
「Check Java IP Address」ページが表示されます。

3 ドロップダウンリストからリソースを選択するか、「**Regular Expression**」ボタンをクリックして正規表現を入力し、「了解」をクリックします。

4 **Java IP** アドレスの確認を有効にするか、無効にするか、またはデフォルト設定を使用します。

注 – デフォルトオプションでは、より汎用的なテンプレートから、派生するデフォルト設定を使用します。つまり、正規表現に一致する短いテンプレートを使用して Java IP アドレスの確認を有効にするか、無効にするかを決定します。

---

5 「了解」をクリックします。

6 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

7 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

クライアントの自動設定

Proxy Server が多くのクライアントをサポートする場合、クライアントの自動設定ファイルを使用して、すべてのブラウザクライアントを設定できます。自動設定ファイルには、さまざまな URL にアクセスする場合、ブラウザが使用するプロキシがあれば、どのプロキシを使用するかを決定する JavaScript™ 関数が含まれています。この機能については、第 17 章を参照してください。

ネットワーク接続モードの設定

Proxy Server のコンピュータは、ネットワークから接続または切断できます。この機能により、デモンストレーションに使用するポータブルコンピュータに簡単にプロキシをインストールできます。

プロキシがネットワークから切断されると、ドキュメントは直接キャッシュから返されます。プロキシは最新状態チェックを行わないため、ドキュメントを高速に取得できます。ただし、ドキュメントは最新の状態ではない可能性があります。
キャッシュについては、第 12 章を参照してください。

ネットワークに接続していない場合、Proxy Server がネットワークからの切断を認識し、リモートサーバーへの接続も試みないため、接続がハングすることはありません。このネットワークなしの設定は、ネットワークが停止していて、Proxy Server コンピュータは稼働している場合に使用できます。ネットワークから切断された状態でプロキシを実行した場合、結果的にアクセスするのはキャッシュ内の古いデータになります。また、ネットワークから切断されたプロキシを実行する場合、プロキシのセキュリティー機能は不要になります。

Proxy Server では、次の 4 つのネットワーク接続モードを用意しています。

- デフォルトモードは、最も一般的な一致オブジェクトの設定から派生します。
- 通常モードは、プロキシの通常の動作モードです。プロキシはドキュメントがまだキャッシュにない場合、コンテンツサーバーから取得します。ドキュメントがキャッシュにある場合、ドキュメントはコンテンツサーバーに照合され、最新のドキュメントかどうかを判断します。キャッシュ内のファイルが変更された場合、そのファイルは最新のコピーに置き換えられます。
- Fast-demo モードでは、ネットワークが利用できる場合に、デモンストレーションをスムーズにすることができます。キャッシュ内にドキュメントがある場合、コンテンツサーバーへの接続は行われず、ドキュメントが変更されたかどうかも確認されません。このモードでは、コンテンツサーバーからの応答を待機することにより生じる遅延を回避できます。キャッシュ内にドキュメントがない場合、ドキュメントはコンテンツサーバーから取得され、キャッシュされます。Fast-demo モードは、通常モードよりも遅延が少なくなりますが、古いデータを返す場合があります。これはドキュメントのコピーが一度作成されると、そのドキュメントの最新状態チェックが行われないためです。
- No-network モードは、ポータブルコンピュータが、ネットワークに接続されていない間を意図したモードです。プロキシは、ドキュメントがキャッシュにある場合はドキュメントを返し、キャッシュにない場合はエラーを返します。プロキシはコンテンツサーバーとの接続を試みません。このため、プロキシが存在しない接続の取得を試みる間にタイムアウトになることがありません。

▼ Proxy Server の実行モードを変更するには

1. サーバーマネージャーにアクセスし、「**Routing**」タブをクリックします。
2. 「**Set Connectivity Mode**」リンクをクリックします。「**Set Connectivity Mode**」ページが表示されます。
3. ドロップダウンリストからリソースを選択するか、「**Regular Expression**」ボタンをクリックして正規表現を入力し、「了解」をクリックします。
4. 必要なモードを選択します。

5 「了解」をクリックします。

6 「**Restart Required**」をクリックします。「**Apply Changes**」ページが表示されます。

7 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

デフォルト **FTP** 転送モードの変更

FTP には FTP サーバーとクライアント間でデータ接続を確立する 2 つの方法があります。プロキシはクライアントとして機能します。この 2 つのモードは、PASV モード FTP と PORT モード FTP と呼ばれます。

- パッシブモード *(PASV):*データ接続は Proxy Server により開始され、FTP サーバーが接続を受け入れます。このモードは、インバウンド接続を受け入れる必要がないため、Proxy Server を実行するサイトではセキュリティーが向上します。
- アクティブモード *(PORT):*データ接続はリモート FTP サーバーにより開始され、プロキシが受信接続を受け入れます。Proxy Server がファイアウォール内にある場合、ファイアウォールは FTP サーバーから送信される FTP データ接続をブロックする可能性があります。これは PORT モードが機能していない可能性があることを意味します。

FTP サイトの中にはファイアウォールを実行しているものがあり、その場合、PASV モードはプロキシサーバーに対して機能しなくなります。このため、Proxy Server では PORT モード FTP を使用するように設定することができます。サーバー全体に対して PORT モードをオンにするか、特定の FTP サーバーに対してのみ PORT モードをオンにできます。

PASV モードがオンの場合でも、リモート FTP サーバーが PASV モードをサポートしていなければ、Proxy Server は PORT モードを使用します。

Proxy Server がファイアウォールの背後にあり、PORT モード FTP が機能しない場合、PORT モードを有効にすることはできません。リソースにデフォルトが選択されている場合、Proxy Server はより汎用的なリソースのモードを使用します。何も指定されていない場合、PASV モードが使用されます。

▼ **FTP** モードを設定するには

1 サーバーマネージャーにアクセスし、「**Routing**」タブをクリックします。

2 「**Set FTP Mode**」リンクをクリックします。「**Set FTP Mode**」ページが表示されます。

3 ドロップダウンリストからリソースを選択するか、「**Regular Expression**」ボタンをクリックして正規表現を入力し、「了解」をクリックします。

4 **FTP** 転送モードを選択します。

5 「了解」をクリックします。

6 「**Restart Required**」をクリックします。「**Apply Changes**」ページが表示されます。

7 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

SOCK ネームサーバーの IP アドレスの指定

アウトバウンド接続が SOCKS サーバーを経由するようにプロキシを設定している場合、SOCKS で使用されるネームサーバーの IP アドレスを明示的に指定する必要があります。

ネームサーバーの IP アドレスを指定する必要があるのは、ファイアウォール内の内部 DNS サービス以外の DNS サーバーで、外部ホスト名を解決する場合です。

▼ SOCKS ネームサーバーの IP アドレスを指定するには

1 サーバーマネージャーにアクセスし、「**Routing**」タブをクリックします。

2 「**Set SOCKS Name Server**」リンクをクリックします。
「Set SOCKS Name Server」ページが表示されます。

3 フィールドに **DNS** ネームサーバーの **IP** アドレスを入力します。

4 「了解」をクリックします。

注 – SOCKS ネームサーバーの IP アドレスを指定するための機能は、かつて SOCKS_NS 環境変数からのみアクセスできました。環境変数を設定し、「SOCKS Name Server Setting」フォームからネームサーバーの IP アドレスを指定する場合、プロキシは環境変数ではなく、フォームに指定された IP アドレスを使用します。

---

5 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

6 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

HTTP 要求のロードバランスの設定

「Configure HTTP Request Load Balancing」ページは、指定された配信元サーバー間で負荷を分散するために使用します。

▼ HTTP 要求のロードバランスを設定するには

1. サーバーマネージャーにアクセスし、「**Routing**」タブをクリックします。
2. 「**Configure HTTP Request Load Balancing**」リンクをクリックします。
   「Configure HTTP Request Load Balancing」ページが表示されます。
3. ドロップダウンリストからリソースを選択するか、「**Regular Expression**」ボタンをクリックして正規表現を入力し、「了解」をクリックします。
4. 「**Server**」フィールドで、配信元サーバーの **URL** を指定します。複数のサーバーパラメータが指定されている場合、**Proxy Server** は指定された配信元サーバー間で負荷を分散します。
5. 「**Sticky Cookie**」フィールドに、**Cookie** 名を指定します。この **Cookie** 名が応答内に表示されると、以降の要求からはその配信元サーバーに接続されます。デフォルト値は `JSESSIONID` です。
6. 「**Sticky Parameter**」フィールドに、経路情報を検査するための **URI** パラメータ名を指定します。**URI** パラメータが要求 **URI** に表示され、その値にコロンが含まれ、そのあとに経路 **ID** が続く場合、要求はその経路 **ID** で特定される配信元サーバーに「固定」されます。デフォルト値は `jsessionid.` です。
7. 「**Route Header**」フィールドに、経路 **ID** が配信元サーバーと通信するために使用される **HTTP** 要求のヘッダー名を指定します。デフォルト値は `proxy-jroute` です。
8. 「**Route Cookie**」フィールドに、応答でスティッキー **Cookie** があったときに **Proxy Server** により生成されるクッキーの名前を指定します。
   デフォルト値は `JROUTE` です。
9. 「**Rewrite Host**」オプションを設定し、サーバーパラメータにより指定されるホストと一致するようにホスト **HTTP** 要求のヘッダーを書き換えるかどうかを指定します。
10. 「**Rewrite Location**」オプションを設定して、サーバーパラメータに一致する位置 **HTTP** 応答ヘッダーを書き換えるかどうかを指定します。

11 「**Rewrite Content Location**」オプションを設定して、サーバーパラメータに一致する **Content-location HTTP** 応答ヘッダーを書き換えるかどうかを指定します。

12 サーバーパラメータに一致する *headername* **HTTP** 応答ヘッダーを書き換えるかどうかを指定します。ここで *headername* はユーザー定義のヘッダー名です。
「**Headername**」フィールドにヘッダー名を指定します。

13 「了解」をクリックします。

14 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

15 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

URL と URL のマッピングの管理

サーバーマネージャーでは、時にミラーサーバーと呼ばれる別のサーバーに URL をマッピングできます。クライアントがミラー化された URL でプロキシにアクセスする場合、プロキシは要求されたドキュメントを URL に指定されたサーバーからではなく、ミラー化されたサーバーから取得します。クライアントは、要求が別のサーバーに送られることを認識しません。また URL をリダイレクトすることも可能です。この場合、プロキシはリダイレクトされた URL のみをクライアントに返し、ドキュメントを返さないため、クライアントは新しいドキュメントを要求できます。またマッピングを使用すると、PAC マッピングや PAT マッピングと同様に、ファイルに URL をマップできます。

URL のマッピングの作成と変更

URL をマッピングするには、URL プレフィックスとマッピング先を指定します。次の節では、さまざまな種類の URL のマッピングについて説明します。次の種類の URL のマッピングを作成できます。

- 通常のマッピングは、URL プレフィックスを別の URL プレフィックスにマップします。 たとえば、プロキシが `http://www.example.com` で始まる要求を取得する場合は常に、特定の URL に移動するようにプロキシを設定できます。
- 逆マッピングは、リダイレクトされる URL プレフィックスを別の URL プレフィックスにマップします。逆マッピングは、内部サーバーがドキュメントをプロキシに送信するのではなく、リダイレクトされた応答を送信する場合に、逆プロキシとともに使用されます。詳細は、第 14 章を参照してください。

- 正規表現は、式に一致するすべての URL を 1 つの URL にマップします。たとえば、.*job.* に一致するすべての URL を特定の URL、たとえば、ユーザーが ProxyServer を使用して特定の URL を表示できない理由を説明する URL にマップします。
- クライアントの自動設定は、URL を Proxy Server に格納された特定の .pac ファイルにマップします。自動設定ファイルについては、第 17 章を参照してください。
- プロキシ配列テーブル (PAT) は、URL を Proxy Server に格納された特定の .pat ファイルにマップします。このタイプのマッピングは、マスタープロキシからのみ作成することをお勧めします。PAT ファイルとプロキシ配列については、293 ページの「プロキシ配列を経由したルーティング」を参照してください。

URL にアクセスするクライアントは、同じサーバー上または別のサーバー上の異なる場所に送られます。リソースを移動した場合、または末尾にスラッシュを付けずにディレクトリにアクセスするとき、関連するリンクの整合性を維持する必要がある場合に、この機能が役立ちます。

たとえば、hi.load.com と呼ばれる負荷の高い Web サーバーを、mirror.load.com と呼ばれる別のサーバーにミラー化する場合を考えてみます。hi.load.com コンピュータに移動する URL については、mirror.load.com コンピュータを使用するように Proxy Server を設定できます。

ソース URL プレフィックスのエスケープを解除する必要がありますが、宛先 (ミラー) URL では、HTTP 要求内の禁止文字のみエスケープする必要があります。

プレフィックスでは、決して末尾にスラッシュを使用しないでください。

▼ URL のマッピングを作成するには

1 サーバーマネージャーにアクセスし、「**URLs**」タブをクリックします。

2 「**Create Mapping**」リンクをクリックします。
「Create Mapping」ページが表示されます。

3 作成するマッピングの種類を選択します。

- 「**Regular Mappings**」:このオプションを選択した場合、ページ下部に次のオプションが表示されます。
  - 「*Rewrite Host*」:to パラメータで指定されたホストに一致するように Host HTTP ヘッダーを書き換えるかどうかを指定します。
  - 「**Reverse Mappings**」:リダイレクトされる URL プレフィックスを別の URL プレフィックスにマップします。このオプションを選択した場合、ページ下部に次のオプションが表示されます。
    - 「*Rewrite Location*」:位置 HTTP 応答ヘッダーを書き換えるかどうかを指定します。

- 「*Rewrite Content Location*」:コンテンツ位置 HTTP 応答ヘッダーを書き換えるかどうかを指定します。
- 「*Rewrite Headername*」:*headername* HTTP 応答ヘッダーを書き換えるかどうかを指定するチェックボックスを選択します。*headername* はユーザー定義のヘッダー名。

「**Regular Expressions**」:式に一致するすべての URL を 1 つの URL にマップします。正規表現については、第 16 章を参照してください。

- 「**Client Autoconfiguration**」:URL を Proxy Server に格納された特定の `.pac` ファイルにマップします。自動設定ファイルについては、第 17 章を参照してください。
- 「**Proxy Array Table (PAT)**」: URL を Proxy Server に格納された特定の `.pat` ファイルにマップします。このタイプのマッピングは、マスタープロキシからのみ作成することをお勧めします。PAT ファイルとプロキシ配列については、第 12 章の「プロキシ配列を経由したルーティング」を参照してください。

4 マップソースのプレフィックスを入力します。

通常マッピングと逆マッピングの場合、このプレフィックスは置き換える URL の一部になります。

正規表現マッピングの場合、URL プレフィックスは一致させるすべての URL を表す正規表現である必要があります。また、マッピングにテンプレートを選択する場合、正規表現はテンプレートの正規表現内の URL についてのみ機能します。

クライアントの自動設定マッピングとプロキシ配列テーブルマッピングの場合、URL プレフィックスは、クライアントがアクセスする完全な URL になります。

5 マップ先を入力します。

クライアントの自動設定およびプロキシ配列テーブルを除くすべてのマッピングの場合、この宣言はマップ先の完全な URL になります。クライアントの自動設定マッピングの場合、この値は Proxy Server のハードディスク上にある `.pac` ファイルへの絶対パスになります。プロキシ配列テーブルマッピングの場合、この値はマスタープロキシのローカルディスク上にある `.pat` ファイルへの絶対パスになります。

6 ドロップダウンリストからテンプレート名を選択します。あるいはテンプレートを適用しない場合は、値を「**NONE**」のままにします。

7 「了解」をクリックし、マッピングを作成します。

8 「**Restart Required**」をクリックします。

「Apply Changes」ページが表示されます。

9 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

▼ 既存のマッピングを変更するには

1 サーバーマネージャーにアクセスし、「**URLs**」タブをクリックします。

2 「**View/Edit Mappings**」リンクをクリックします。
「View/Edit Mappings」ページが表示されます。

3 マッピングを編集するには、マッピングの横に表示される「**Edit**」リンクをクリックします。プレフィックス、マップされた **URL**、およびマッピングの影響を受けるテンプレートを編集できます。「了解」をクリックして変更を確認します。

4 「**Restart Required**」をクリックします。「**Apply Changes**」ページが表示されます。

5 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

▼ マッピングを削除するには

1 サーバーマネージャーにアクセスし、「**URLs**」タブをクリックします。

2 「**View/Edit Mappings**」リンクをクリックします。
「View/Edit Mappings」ページが表示されます。

3 削除するマッピングを選択して、マッピングの横に表示される「**Remove**」リンクをクリックします。

4 「**Restart Required**」をクリックします。「**Apply Changes**」ページが表示されます。

5 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

URL のリダイレクト

ドキュメントを取得して返すのではなく、リダイレクトされた URL をクライアントに返すように Proxy Server を設定できます。リダイレクトにより、クライアントは最初に要求された URL が別の URL にリダイレクトされたことを認識します。クライアントは通常は、すぐにリダイレクトされた URL を要求します。Netscape Navigator は自動的にリダイレクトされた URL を要求します。ユーザーが再度、明示的にドキュメントを要求する必要はありません。

URL のリダイレクトは、ユーザーをアクセスが拒否された理由が説明されている URL にリダイレクトできるため、ある領域へのアクセスを拒否する場合に便利です。

▼ 1 つまたは複数の URL にリダイレクトするには

1. サーバーマネージャーにアクセスし、「**URLs**」タブをクリックします。
2. 「**Redirect URLs**」リンクをクリックします。「**Redirect URLs**」ページが表示されます。
3. ソース **URL**、つまり **URL** プレフィックスを入力します。
4. リダイレクト先の **URL** を入力します。この **URL** は **URL** プレフィックスか固定 **URL** のいずれかになります。
   - リダイレクト先 URL として URL プレフィックスを使用する場合、URL プレフィックスフィールドの横のラジオボタンを選択して、URL プレフィックスを入力します。
   - 固定 URL を使用する場合、「Fixed URL」フィールドの横のラジオボタンを選択して、固定 URL を入力します。
5. 「了解」をクリックします。
6. 「**Restart Required**」をクリックします。
   「Apply Changes」ページが表示されます。
7. 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

◆ ◆ ◆ 第 12 章

12 キャッシュ

この章では Sun Java System Web Proxy Server がドキュメントをキャッシュするしくみを説明します。また、オンラインページを使用してキャッシュを設定する方法についても説明します。

この章の内容は次のとおりです。

キャッシュのしくみ

キャッシュを利用すると、ネットワークのトラフィックが減少し、プロキシサーバーを使用するクライアントへの応答時間は、リモートサーバーに直接アクセスしないため短縮されます。

クライアントがプロキシサーバーから Web ページまたはドキュメントを要求する場合、プロキシサーバーはクライアントにドキュメントを送信する間に、リモートサーバーのドキュメントをローカルのキャッシュディレクトリ構造にコピーします。

次の図に示すように、以前に要求したことがあり、プロキシのキャッシュにコピーされているドキュメントをクライアントが要求した場合、プロキシはドキュメントを再度リモートサーバーから取得するのではなく、キャッシュからドキュメントを返します。プロキシで、ファイルが最新の状態ではないと判断すると、リモートサーバーのドキュメントを更新し、キャッシュを更新してからドキュメントをクライアントに送信します。

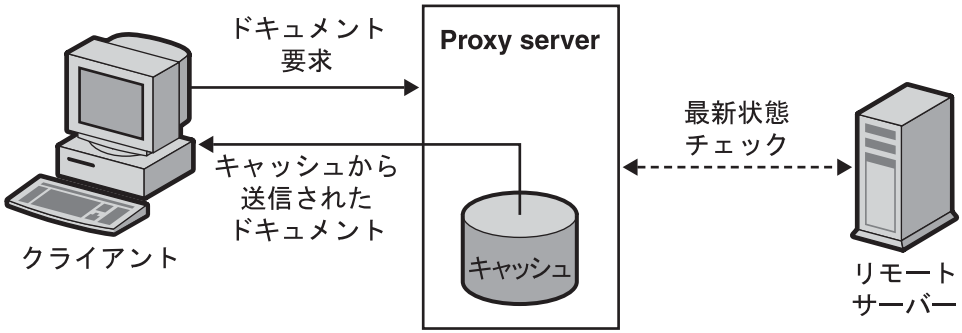


図 12–1　プロキシのドキュメントの取得

キャッシュ内のファイルは、Sun Java System Web Proxy Server のガベージコレクションユーティリティー (CacheGC) により自動的に管理されます。CacheGC は定期的にキャッシュを自動消去し、キャッシュが古いドキュメントでいっぱいにならないようにします。

キャッシュ構造について

キャッシュは 1 つ以上のパーティションから構成されます。概念上、パーティションとはキャッシュのために確保しておくディスク上のストレージ領域です。キャッシュを複数のディスクに分散させる場合、各ディスクに 1 つ以上のキャッシュパーティションを設定する必要があります。各パーティションは、個別に管理できます。つまり、ほかのすべてのパーティションとは別に各パーティションを有効化、無効化、および設定できます。

キャッシュされた大量のファイルを一箇所に保存すると、パフォーマンスが低下する場合があるため、各パーティションに複数のディレクトリ、またはセクションを作成してください。セクションは、キャッシュ構造内のパーティションの 1 つ下のレベルです。キャッシュ内には、パーティション全体で最大 256 のセクションを作成できます。キャッシュセクションの数は、必ず 2 の累乗 (1、2、4、8、16、...、256) にしてください。

キャッシュ構造の階層の最下位レベルはサブセクションです。サブセクションはセクション内のディレクトリです。各セクションは 64 のサブセクションから構成されます。キャッシュされたファイルは、キャッシュ内の最下位レベルであるサブセクションに格納されます。

次の図に、パーティションとセクションからなるキャッシュ構造の例を示します。この図では、キャッシュ全体を 3 つのパーティションに分割したキャッシュディレクトリ構造になっています。最初のパーティションには 4 つのキャッシュセクションが含まれ、2 番目の 2 つのパーティションにはそれぞれ 2 つのセクションが含まれます。

キャッシュの各セクションには、セクションを示す「s」とセクション番号が付けられています。3.4 というセクションの場合、3 はキャッシュセクションの数を表す 2 の累乗を示し ($2^3 = 8$)、4 はセクションの番号を意味します (8 つのセクションに 0 ～ 7 のラベルが付けられる)。したがって、s3.4 はセクション 5/8 を意味します。

図 12–2 キャッシュ構造の例

キャッシュへのファイルの分散

プロキシサーバーは特定のアルゴリズムを使用して、ドキュメントを格納するディレクトリを決定します。このアルゴリズムにより、ディレクトリ内でドキュメントを均等に分散させることができます。ディレクトリに大量のドキュメントを格納すると、パフォーマンスの問題が生じる傾向があるため、均等に分散することが重要です。

プロキシサーバーは RSA MD5 (Message Digest 5) アルゴリズムを使用して、URL を 16 バイトのバイナリデータに縮小し、このデータの 8 バイトを使用して、キャッシュにドキュメントを格納するために使用する 16 文字の 16 進数ファイル名を算出します。

キャッシュの詳細設定

キャッシュの詳細を設定することにより、キャッシュを有効にし、プロキシサーバーがキャッシュするプロトコルの種類を制御できます。キャッシュの詳細には次の項目が含まれます。

- キャッシュの有効化または無効化
- キャッシュが一時ファイルを格納する作業ディレクトリ
- キャッシュされた URL が記録されるディレクトリ名
- キャッシュのサイズ
- キャッシュの容量
- キャッシュされるプロトコルの種類
- キャッシュされたドキュメントの更新頻度
- プロキシがドキュメントへのアクセス回数を追跡し、その値をリモートサーバーに報告するかどうか

注 – 大きなキャッシュの詳細の設定には時間がかかるため、管理インタフェースがタイムアウトする可能性があります。そのため、大きなキャッシュを作成する場合は、コマンド行ユーティリティーを使用して、キャッシュの詳細を設定してください。キャッシュのコマンド行ユーティリティーの詳細については、278 ページの「キャッシュのコマンド行インタフェースの使用」を参照してください。

▼ キャッシュの詳細を設定するには

1. サーバーマネージャーにアクセスし、「**Caching**」タブをクリックします。
2. 「**Set Cache Specifics**」リンクをクリックします。
   「Set Cache Specifics」ページが表示されます。
3. 適切なオプションを選択して、キャッシュを有効または無効にできます。
   キャッシュはデフォルトで有効になっています。
4. 作業ディレクトリを指定します。
   デフォルトでは、作業ディレクトリはプロキシインスタンスの下にあります。この場所は変更できます。詳細については、260 ページの「キャッシュの作業ディレクトリの作成」を参照してください。

5 パーティション設定リンクをクリックします。

「Add/Edit Cache Partitions」ページが表示されます。新しいキャッシュのパーティションを追加したり、既存のキャッシュのパーティションを編集したりできます。キャッシュサイズは、キャッシュを増やすことができる最大のサイズです。最大キャッシュサイズは32Gバイトです。詳細については、260 ページの「キャッシュサイズの設定」を参照してください。

6 キャッシュの容量設定リンクをクリックします。

「Set Cache Capacity」ページが表示されます。「Set Cache Capacity」ページで、キャッシュの容量を設定できます。

7 「**Cache HTTP**」を選択して、**HTTP** ドキュメントのキャッシュを有効にします。

プロキシサーバーでHTTPドキュメントのキャッシュを行うことにした場合、キャッシュ内のドキュメントの最新状態チェックを常に行うか、あるいは定期的にチェックを行うかを決定する必要があります。また、プロキシサーバーがリモートサーバーにキャッシュのヒットを報告する機能を有効または無効にできます。詳細については、261 ページの「HTTP ドキュメントのキャッシュ」を参照してください。利用できるオプションは次のとおりです。

- 「Always Check That The Document Is Up To Date」オプションを選択し、HTTP ドキュメントを常に最新の状態に保ちます。
  - 「Check Only If Last Check More Than」ドロップダウンリストから時間数を選択し、プロキシサーバーの更新間隔を指定します。最新状態チェックは、次のオプションのいずれかを使用して実行します。
    - 「**Use Last-modified Factor**」: 配信元のサーバーからドキュメントとともに送信される、最後に変更されたヘッダー。
    - 「**Use Only Explicit Expiration Information**」: プロキシサーバーは Expires ヘッダーを使用して、キャッシュエントリが新しいか古いかを判断します。

    「Never Report Accesses To Remote Server」オプションを選択し、プロキシサーバーからリモートサーバーにアクセス回数を報告しないようにします。
  - 「Report Cache Hits To Remote Server」オプションを選択し、ドキュメントへのアクセス回数を追跡し、リモートサーバーに報告します。

8 「**Yes; Reload If Older Than**」チェックボックスを選択して、キャッシュされた **FTP** ドキュメントの更新間隔を設定し、ドロップダウンリストから値を選択して間隔を設定します。詳細については、**263** ページの「**FTP** ドキュメントと **Gopher** ドキュメントのキャッシュ」を参照してください。

9 キャッシュされた **Gopher** ドキュメントに更新間隔を設定できます。「**Yes; Reload If Older Than**」チェックボックスを選択し、ドロップダウンリストから値を選択して間

隔を設定します。詳細については、263 ページの「FTP ドキュメントと Gopher ドキュメントのキャッシュ」を参照してください。

10 「了解」をクリックします。

11 「**Restart Required**」をクリックします。「**Apply Changes**」ページが表示されます。

12 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

キャッシュの作業ディレクトリの作成

キャッシュファイルは、キャッシュパーティションに置かれています。「Set Cache Specifics」ページで指定する作業ディレクトリは、多くの場合、キャッシュの親ディレクトリになります。キャッシュされたすべてのファイルは、キャッシュディレクトリに、体系付けられたディレクトリ構造として表現されます。キャッシュディレクトリ名を変更する場合、またはキャッシュディレクトリを別の場所に移動する場合、プロキシに新しい場所を指定する必要があります。

キャッシュのディレクトリ構造は複数のファイルシステムに拡張できるため、大容量のキャッシュ構造を 1 つの大容量ディスクで管理するのではなく、複数の容量の小さなディスクに分割することができます。各プロキシサーバーには専用のキャッシュディレクトリ構造が必要です。つまり、キャッシュディレクトリは、複数のプロキシサーバーで同時に共有することはできません。

キャッシュサイズの設定

キャッシュサイズは、パーティションサイズを表します。キャッシュサイズは、常にキャッシュ容量より小さくする必要があります。キャッシュ容量はキャッシュを増加させることができる最大サイズであるためです。すべてのパーティションサイズの合計は、必ずキャッシュサイズ以下にする必要があります。

プロキシのキャッシュに利用できるディスク容量は、キャッシュのパフォーマンスに大きく影響します。キャッシュが小さすぎる場合、Cache GC がキャッシュされたドキュメントを削除してディスクに空き容量を作る頻度が増加し、ドキュメントがコンテンツサーバーから取得される頻度が高くなります。これらの動作によってパフォーマンスが低下します。

より多くのドキュメントをキャッシュでき、ネットワークトラフィックの負荷を減少させ、プロキシの応答時間を速くできるため、キャッシュサイズを大容量にすると効率が向上します。また、キャッシュされたドキュメントは、ユーザー側で不要になれば GC によって削除されます。ファイルシステムの制限をなくすことにより、キャッシュサイズが大きくなり過ぎることはなくなります。余分な容量は未使用領域として残るだけです。

またキャッシュは複数のディスクパーティションに分割できます。

HTTP ドキュメントのキャッシュ

HTTP ドキュメントは、ほかのプロトコルのドキュメントにはないキャッシュ機能を備えています。ただし、キャッシュを正しくセットアップし、設定することで、プロキシサーバーでの HTTP、FTP、および Gopher の各ドキュメントを効果的にキャッシュできるようになります。

注 – Proxy Server 4 は HTTPS ドキュメントのキャッシュをサポートしていません。

HTTP ドキュメントにはすべて詳細ヘッダーセクションがあり、プロキシサーバーはこのセクションを使用してプロキシキャッシュ内のドキュメントとリモートサーバー上のドキュメントを比較し、評価します。プロキシが HTTP ドキュメントに対して最新状態チェックを実行する場合、プロキシはサーバーに 1 つの要求を送信し、キャッシュ内のドキュメントが古ければサーバーにドキュメントを返すように指示します。ドキュメントは最後の要求から、変更されていないことが多く、したがって転送されません。このような HTTP ドキュメントの最新状態チェックにより、帯域幅が節約され、遅延が減少します。

リモートサーバーとのトランザクションを減らすために、プロキシサーバーで HTTP ドキュメント用にキャッシュ有効期限設定を設定できます。キャッシュ有効期限設定では、サーバーに要求を送信する前に HTTP ドキュメントの最新状態チェックが必要かどうかを評価するようにプロキシに情報を提供します。プロキシはこの評価を、ヘッダー内の HTTP ドキュメントが最後に変更された (Last-Modified) 日付に基づいて行います。

HTTP ドキュメントでは、キャッシュ更新設定も使用できます。このオプションは、プロキシが常に最新状態チェックを行うかどうか (有効期限の設定がオーバーライドされる)、プロキシが一定期間待機してからチェックを行うかどうかを指定します。次の表に、有効期限設定と更新設定の両方を指定した場合のプロキシの動作を示します。更新設定を使用すると、遅延が減り、帯域幅を大幅に節約できます。

表 12–1 HTTP でのキャッシュ有効期限設定とキャッシュ更新設定の使用

更新の設定	有効期限の設定	結果
常に最新状態チェックを行う	(適用なし)	常に最新状態チェックを行う
ユーザー指定の間隔	ドキュメントの「expires」ヘッダーを使用	更新期間が過ぎれば最新状態チェックを行う

表 12–1 HTTP でのキャッシュ有効期限設定とキャッシュ更新設定の使用 (続き)

更新の設定	有効期限の設定	結果
	ドキュメントの Last-Modified ヘッダーで評価	評価および expires ヘッダーに小さな値を設定*

注 – * 小さな値を使用すると、ドキュメントが頻繁に変更されるため、キャッシュのデータが古くなりません。

HTTP キャッシュの更新間隔の設定

プロキシサーバーで HTTP ドキュメントのキャッシュを行うことにした場合、プロキシサーバーでキャッシュ内のドキュメントの最新状態チェックを常に行うか、あるいはプロキシサーバーでキャッシュ更新設定 (最新状態チェックの間隔) に基づいてチェックを行うかを指定する必要があります。HTTP ドキュメントの場合、更新間隔は 4 ～ 8 時間が適当です。更新間隔を長くすると、プロキシがリモートサーバーに接続する回数が少なくなります。プロキシが次の更新までの間に最新状態チェックを行わない場合でも、クライアントで「Reload」ボタンをクリックすると更新を実行できます。この操作により、プロキシはリモートサーバーで強制的に最新状態チェックを実行します。

HTTP ドキュメントの更新間隔は、「Set Cache Specifics」ページまたは「Set Caching Configuration」ページで設定できます。「Set Cache Specifics」ページでは、グローバルなキャッシュプロシージャーを設定できます。また「Set Caching Configuration」ページでは、特定の URL およびリソースのキャッシュプロシージャーを制御できます。

HTTP キャッシュの有効期限ポリシーの設定

また、Last-Modified 要素または明示的な有効期限情報のみを使用して、キャッシュされたドキュメントが最新かどうかを確認するようにサーバーを設定できます。

明示的な有効期限情報とは、一部の HTTP ドキュメント内にある、ファイルが期限切れになる日時を指定するヘッダーです。明示的な Expires ヘッダーを使用する HTTP ドキュメントは多くないため、Last-modified ヘッダーに基づいて評価してください。

HTTP ドキュメントを Last-modified ヘッダーに基づいてキャッシュすることを決定した場合、有効期限の評価で使用する割合を選択する必要があります。LM 要素として知られるこの割合に、最後の変更からドキュメントで最後に最新状態チェックが実行された時間までの間隔を乗算します。この結果の数字を、最後の最新状態チェックからの経過時間と比較します。この数字が間隔時間よりも小さい場合、ドキュメントの期限は切れていません。割合が小さくなると、プロキシのドキュメントチェックの頻度が高くなります。

たとえば、最後の変更が 10 日前に行われたドキュメントを考えてみます。Last-Modified 要素を 0.1 に設定した場合、プロキシはこの要素の意味を、ドキュメントが 1 日 (10 * 0.1 = 1) の間変更されない、と解釈します。その場合、ドキュメントのチェックが 1 日以内に実行されている場合、プロキシはキャッシュからドキュメントを返します。

同じ例で、HTTP ドキュメントのキャッシュ更新設定が 1 日未満に設定された場合、プロキシは 1 日に何度か最新状態チェックを実行します。プロキシは常に、ファイルがより頻繁に更新される値 (キャッシュの更新またはキャッシュの有効期限) を使用します。

HTTP ドキュメントの有効期限設定は、「Set Cache Specifics」ページまたは「Set Caching Configuration」ページで設定できます。「Set Cache Specifics」ページでは、グローバルなキャッシュプロシージャを設定できます。また「Set Caching Configuration」ページでは、特定の URL およびリソースのキャッシュプロシージャを制御できます。

HTTP アクセスのリモートサーバーへの報告

Sun Java System Web Proxy Server でドキュメントがキャッシュされると、そのドキュメントには再度更新されるまでに、何度もアクセスできます。リモートサーバーの場合、プロキシにコピーを 1 つ送信し、プロキシがそれをキャッシュする動作が 1 回のアクセス、すなわち「ヒット」になります。プロキシサーバーは、指定されたドキュメントが現在の最新状態チェックから次のチェックまでの間にプロキシキャッシュからアクセスされた回数をカウントし、次回ドキュメントが更新されるときに追加 HTTP 要求ヘッダー (Cache-Info) でそのヒットカウントをリモートサーバーに送信します。このように、リモートサーバーがこの種類のヘッダーを認識するように設定されていれば、リモートサーバーはドキュメントのアクセス回数をより正確に受信できます。

FTP ドキュメントと Gopher ドキュメントのキャッシュ

FTP と Gopher では、ドキュメントの最新状態をチェックするための方法が使用できません。したがって、FTP ドキュメントと Gopher ドキュメントのキャッシュを最適化する唯一の方法は、キャッシュ更新間隔の設定になります。キャッシュ更新間隔は、リモートサーバーから最新バージョンのドキュメントを取得するまでにプロキシサーバーが待機する時間です。キャッシュ更新間隔を設定しない場合、プロキシはキャッシュ内のバージョンが最新である場合でも、リモートサーバーからこれらのドキュメントを取得します。

FTP および Gopher にキャッシュ更新間隔を設定する場合、プロキシがドキュメントを安全に取得できると考えられる間隔を選択します。たとえば、変更がまれな情報を格納している場合、数日間の高い数値を使用します。データが常に変更されてい

る場合、最低でも数時間間隔でファイルが取得されるようにします。更新中は、古いファイルをクライアントに送信してしまう危険性があります。この間隔が十分に短い場合 (数時間)、このような危険性をほぼ完全に除外でき、応答時間も著しく向上します。

FTP ドキュメントおよび Gopher ドキュメントのキャッシュの更新間隔は、「Set Cache Specifics」ページまたは「Set Caching Configuration」ページで設定できます。「Set Cache Specifics」ページでは、グローバルなキャッシュプロシージャを設定できます。また「Set Caching Configuration」ページでは、特定の URL およびリソースのキャッシュプロシージャを制御できます。「Set Cache Specifics」ページの使用の詳細については、258 ページの「キャッシュの詳細設定」を参照してください。「Set Caching Configuration」ページの使用の詳細については、268 ページの「キャッシュの設定」を参照してください。

注 – FTP ドキュメントおよび Gopher ドキュメントでドキュメント間に大きな相違がある場合 (ドキュメントにより変更頻度が異なる場合)、「Set Caching Configuration」ページでドキュメントの種類ごとに個別にテンプレートを作成し (たとえば、リソース `ftp://.*.gif` でテンプレートを作成するなど)、次にそのリソースに適した更新間隔を設定します。

キャッシュの作成と変更

キャッシュパーティションは、キャッシュ用に確保される、予約されたディスクまたはメモリーの一部です。キャッシュ容量を変更する場合、パーティションを変更または追加することができます。

▼ キャッシュパーティションを追加するには

1. サーバーマネージャーにアクセスし、「**Caching**」タブをクリックします。
2. 「**Add/Edit Cache Partitions**」リンクをクリックします。
   「Add/Edit Cache Partitions」ページが表示されます。
3. 「**Add Cache Partition**」ボタンをクリックします。
   「Cache Partition Configuration」ページが表示されます。
4. 新しいパーティションの適切な値を指定します。
5. 「了解」をクリックします。

6 「**Restart Required**」をクリックします。
「Apply changes」ページが表示されます。

7 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

▼ キャッシュパーティションを変更するには

1 サーバーマネージャーにアクセスし、「**Caching**」タブをクリックします。

2 「**Add/Edit Cache Partitions**」リンクをクリックします。
「Add/Edit Cache Partitions」ページが表示されます。

3 変更するパーティションの名前をクリックします。

4 情報を編集します。

5 「了解」をクリックします。

6 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

7 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

キャッシュ容量の設定

キャッシュ容量の値は、キャッシュのディレクトリ構造を導き出すために使用します。このキャッシュ容量から、キャッシュディレクトリに含められるセクションの数が導き出されます。キャッシュ容量とキャッシュディレクトリのキャッシュ階層には直接的な関係があります。容量が大きくなると、階層も大規模になります。
キャッシュ容量は、キャッシュのサイズ以上である必要があります。外部ディスクを追加するなど、あとでキャッシュサイズを増やす予定があるとわかっている場合は、キャッシュ容量をキャッシュサイズよりも大きく設定しておくと便利です。
キャッシュ容量の最大値は 32G バイトで、この中に 256 のセクションを作成できます。

▼ キャッシュ容量を設定するには

1. サーバーマネージャーにアクセスし、「**Caching**」タブをクリックします。
2. 「**Set Cache Capacity**」リンクをクリックします。
   「Set Cache Capacity」ページが表示されます。
3. 「**New Capacity Range**」ドロップダウンリストから、容量を選択します。
4. 「了解」をクリックします。
5. 「**Restart Required**」をクリックします。
   「Apply Changes」ページが表示されます。
6. 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

キャッシュセクションの管理

プロキシキャッシュは 1 つ以上のキャッシュセクションに分割されます。最大 256 のセクションを作成できます。キャッシュセクションの数は、必ず 2 の累乗 (1、2、4、8、16、...、256) にしてください。最大容量は 32G バイト (最適) で、256 のキャッシュセクションに分割できます。

キャッシュ容量に 500M バイトを選択した場合、インストーラは 4 つのキャッシュセクション (500 d6 125 = 4) を作成します。キャッシュ容量に 2G バイトを選択した場合、インストーラは 16 のセクション (2000 d6 125 = 16) を作成します。各セクションでセクション数を取得するための最適な値は 125 M バイトです。セクション数が多くなると、格納および分散される URL の数が多くなります。

▼ キャッシュセクションを管理するには

1. サーバーマネージャーにアクセスし、「**Caching**」タブをクリックします。
2. 「**Manage Sections**」リンクをクリックします。
   「Manage Sections」ページが表示されます。
3. テーブル内の情報を変更します。
   セクションは既存のパーティション間で移動することができます。
4. 「了解」をクリックします。

5 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

6 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

ガベージコレクションの詳細設定

キャッシュガベージコレクタを使用して、キャッシュからファイルを削除できます。ガベージコレクションは、自動モードまたは明示モードのいずれかで実行できます。明示モードは、管理者が外部からスケジュールします。いずれかのモードを選択し、「了解」をクリックします。「Restart Required」をクリックします。「Apply Changes」ページが表示されます。「Restart Proxy Server」ボタンをクリックして、変更を適用します。

ガベージコレクションのスケジュール

「Schedule Garbage Collection」ページでは、ガベージコレクションが実行される日時を指定できます。

▼ ガベージコレクションを設定するには

1 サーバーマネージャーにアクセスし、「**Caching**」タブをクリックします。

2 「**Schedule Garbage Collection**」リンクをクリックします。
「Schedule Garbage Collection」ページが表示されます。

3 「**Schedule Garbage Collection At**」リストから、ガベージコレクションを実行する時間を選択します。

4 ガベージコレクションを実行する曜日を指定します。

5 「了解」をクリックします。

6 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

7 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

キャッシュの設定

URL には指定する正規表現パターンに一致する複数の設定パラメータ値を指定できます。この機能により、キャッシュされたドキュメントの種類に基づいて、プロキシキャッシュを詳細に制御できます。キャッシュの設定では、次の項目を指定します。

- キャッシュのデフォルト
- 認証が必要なページをキャッシュする方法
- クエリーをキャッシュする方法
- キャッシュファイルの最小および最大サイズ
- キャッシュされたドキュメントの更新頻度
- キャッシュの有効期限ポリシー
- クライアント割り込みの場合のキャッシュ動作
- 配信元サーバーへの接続が失敗した場合のキャッシュ動作

注 – 特定のリソースのキャッシュデフォルトを「Derived configuration」か「Don't cache」のいずれかに設定した場合、キャッシュ設定オプションは「Set Caching Configuration」ページに表示されません。ただし、リソースにキャッシュデフォルトの「Cache」を選択した場合は、複数のほかの設定項目を指定することができます。

▼ キャッシュを設定するには

1. サーバーマネージャーにアクセスし、「**Caching**」タブをクリックします。
2. 「**Set Caching Configuration**」ページをクリックします。
   「Set Caching Configuration」ページが表示されます。
3. ドロップダウンリストからリソースを選択するか、「**Regular Expression**」ボタンをクリックして正規表現を入力し、「了解」をクリックします。
4. 設定情報を変更します。
5. 「了解」をクリックします。
6. 「**Restart Required**」をクリックします。
   「Apply Changes」ページが表示されます。
7. 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

設定要素のキャッシュ

次の節では、ニーズに最も適した設定を決定する際に役に立つ情報が含まれています。

キャッシュデフォルトの設定

プロキシサーバーでは、特定のリソースに対するキャッシュデフォルトを指定できます。リソースは指定した特定の基準と一致する、ファイルの一種です。たとえば、サーバーがドメイン `company.com` からすべてのドキュメントを自動的にキャッシュするように設定するには、次の正規表現を作成します。

```
[a-z] *://[^/:]\\.company\\.com.*.
```

デフォルトでは、「Cache」オプションが選択されています。サーバーはキャッシュ可能なすべてのドキュメントを上記のドメインから自動的にキャッシュします。

注 – 特定のリソースのキャッシュデフォルトを、「Derived configuration」か「Don't cache」のいずれかに設定した場合、そのリソースのキャッシュを設定する必要はありません。ただし、リソースにキャッシュデフォルトの「Cache」を選択した場合は、複数のほかの設定項目を指定することができます。これらの項目のリストについては、268 ページの「キャッシュの設定」を参照してください。

HTTP、FTP、および Gopher のキャッシュデフォルトも設定できます。

認証が必要なページのキャッシュ

サーバーのキャッシュにユーザー認証が必要なファイルを保存できます。プロキシサーバーはキャッシュ内のファイルにタグを付け、ユーザーがファイルを要求した場合に、リモートサーバーからの認証が必要なファイルとして認識できるようにします。

プロキシサーバーはリモートサーバーの認証方式を判断できず、ユーザーの ID やパスワードのリストを持たないため、認証を必要とするドキュメントが要求されるたびにリモートサーバーで最新状態チェックを強制的に実行します。したがって、ユーザーはファイルにアクセスするために ID とパスワードを入力する必要があります。ユーザーがブラウザセッションで以前にそのサーバーにアクセスしている場合、ブラウザにより認証情報が自動的に送信され、ユーザーに入力が要求されません。

認証を必要とするページのキャッシュを有効に設定していない場合、プロキシはそれらをキャッシュしません。これはデフォルトの動作です。

クエリーのキャッシュ

クエリーのキャッシュはHTTP ドキュメントでのみ有効です。キャッシュされるクエリーの長さを制限できます。またはクエリーのキャッシュを完全に禁止することができます。クエリーが長くなると、繰り返される可能性が低くなるため、キャッシュの有用性も低くなります。

クエリーには次のキャッシュの制限が適用されます。

- アクセス方式を GET にする、ドキュメントは保護されない (認証されたページのキャッシュを有効にしている場合を除く)、応答には少なくとも Last-modified ヘッダーを含める。この制限では、クエリーエンジンに対して、クエリー結果のドキュメントがキャッシュ可能であることを示す必要があります。
- Last-modified ヘッダーが含まれている場合は、クエリーエンジンはキャッシュを有効にするために条件付き GET メソッド (If-modified-since ヘッダーを使用) をサポートする必要があります。含まれていない場合は、Expires ヘッダーを返す必要があります。

キャッシュファイルの最小および最大サイズの設定

プロキシサーバーにキャッシュされるファイルに最小および最大サイズを設定できます。高速ネットワーク接続を確立している場合、最小サイズを設定してください。高速接続を使用している場合、小さい容量のファイルはすぐに取得できるため、サーバーはファイルをキャッシュする必要がありません。この例では、大容量のファイルのみキャッシュが必要になります。最大ファイルサイズを設定することで、大容量のファイルによりプロキシのディスク容量が占有されるのを防ぐことができます。

最新状態チェックポリシーの設定

最新状態のチェックポリシーにより、HTTP ドキュメントを常に最新状態に保つことができます。また、プロキシサーバーに更新間隔を指定することもできます。

有効期限ポリシーの設定

Last-modified 要素または明示的な有効期限情報を使用して、有効期限ポリシーを設定できます。

クライアント割り込みの場合のキャッシュ動作の設定

ドキュメントの一部のみが取得され、クライアントがデータ転送に割り込む場合、プロキシにはドキュメントをキャッシュするために、ドキュメントの取得を終了させることができます。プロキシのデフォルトでは、ドキュメントの 25 パーセント以上が取得された場合、キャッシュのためドキュメントの取得を終了します。それ以外の場合、プロキシはリモートサーバー接続を終了し、不完全なファイルを削除します。クライアント割り込みの割合は、高くしたり低くしたりすることができます。

サーバー接続に失敗した場合の動作

配信元サーバーに到達できないため、古いドキュメントの最新状態チェックに失敗する場合、プロキシがキャッシュの古いドキュメントを送信するかどうかを指定できます。

ローカルホストのキャッシュ

ローカルホストから要求されたURLにドメイン名が含まれていない場合、プロキシサーバーはそのURLをキャッシュしません。この動作によりキャッシュの重複を避けることができます。たとえば、ユーザーがローカルサーバーから `http://machine/filename.html` と `http://machine.example.com/filename.html` を要求する場合、どちらのURLもキャッシュに格納されている可能性があります。これらのファイルはローカルサーバーのファイルであるため、すばやく取得でき、キャッシュする必要がありません。

ただし、会社で複数のリモートの場所にサーバーを配置している場合、すべてのホストのドキュメントをキャッシュして、ネットワークトラフィックを減らし、ファイルへのアクセス時間を短縮することができます。

▼ ローカルホストのキャッシュを有効にするには

1. サーバーマネージャーにアクセスし、「**Caching**」タブをクリックします。
2. 「**Cache Local Hosts**」リンクをクリックします。
   「Cache Local Hosts」ページが表示されます。
3. ドロップダウンリストからリソースを選択するか、「**Regular Expression**」ボタンをクリックして正規表現を入力し、「了解」をクリックします。
   正規表現の詳細については、第16章を参照してください。
4. 「**Enabled**」ボタンをクリックします。
5. 「了解」をクリックします。
6. 「**Restart Required**」をクリックします。
   「Apply Changes」ページが表示されます。
7. 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

ファイルキャッシュの設定

ファイルキャッシュはデフォルトで有効になっています。ファイルキャッシュの設定は、`server.xml` ファイルに保存されています。ファイルキャッシュの設定は、サーバーマネージャーを使用して変更することができます。

注 – 「Configure File Cache」ページはユーザーインタフェースに表示されますが、今回の Proxy Server 4 のリリースでは実装されていません。

▼ ファイルキャッシュを設定するには

1 サーバーマネージャーから「**Caching**」タブをクリックします。

2 「**File Cache Configuration**」リンクをクリックします。
「File Cache Configuration」ページが表示されます。

3 まだ選択されていない場合には、「**Enable File Cache**」を選択します。

4 ファイルを転送するかどうかを選択します。
転送ファイルを有効にすると、サーバーはファイルキャッシュにファイルのコンテンツではなく、オープンしているファイルの記述子をキャッシュします。クライアントへのファイルコンテンツの送信には `PR_TransmitFile` が使用されます。「Transmit File」を有効にした場合、オープンしているファイルの記述子のみがキャッシュされるため、通常はファイルキャッシュによって行われる、小、中、大の異なるサイズのファイルの区別が行われなくなります。デフォルトでは、Windows では「Transmit File」は有効に、UNIX では無効に設定されています。UNIX では、ネイティブ OS で PR_TransmitFile がサポートされているプラットフォームのみで「Transmit File」を有効にするようにしてください。現在 HP-UX と AIX が該当します。UNIX/Linux プラットフォームでは使用しないことをお勧めします。

5 ハッシュテーブルのサイズを入力します。
デフォルトサイズはファイルの最大数の 2 倍に 1 を加えた数です。たとえば、ファイルの最大数が 1024 に設定されている場合、デフォルトのハッシュテーブルのサイズは 2049 になります。

6 有効なキャッシュエントリの最大時間を秒数で入力します。
デフォルトの設定は 30 です。この設定により、ファイルがキャッシュされてから、キャッシュされた情報を継続して使用できる時間が制御されます。「MaxAge」よりも古いエントリは、同じファイルがキャッシュから参照された場合、同じファイルの新しいエントリに置き換えられます。コンテンツを定期的に更新するかどうかに基づいて、最大時間を設定します。たとえば、コンテンツが定期的に 1 日に 4 回更新

される場合、最大時間を 21600 秒 (6 時間) に設定できます。それ以外の場合、最大時間には、ファイルが変更されたあと、以前のバージョンのコンテンツファイルを提供する最大時間を設定することを検討してください。

7 キャッシュするファイルの最大数を入力します。
デフォルトの設定は 1024 です。

8 ファイルサイズ (中) およびファイルサイズ (小) の上限をバイトで入力します。
デフォルトでは、「Medium File Size Limit」は 537600 に設定され、「Small File Size Limit」は 2048 に設定されています。

キャッシュは小ファイル、中ファイル、大ファイルを異なる方法で処理します。UNIX/Linux プラットフォームのみ、中ファイルのコンテンツは、ファイルが仮想メモリーにマッピングされることにより、キャッシュされます。小ファイルのコンテンツは、ヒープスペースが割り当てられ、ファイルをそのスペース内に読み込むことにより、キャッシュされます。大ファイルのコンテンツはキャッシュされませんが、大ファイルの情報はキャッシュされます。小ファイルと中ファイルを区別する利点は、小ファイルが多数ある場合に、仮想メモリーで多くのページが消費されるのを防ぐことにあります。このため、「Small File Size Limit」の値は、通常は VM ページサイズよりもやや低い値になります。

9 中ファイルおよび小ファイルの容量を設定します。
中ファイルの容量は、中サイズの全ファイルのマップに使用される仮想メモリーのサイズをバイトで表したものです。サイズはデフォルトで 10485760 に設定されています。小ファイルの容量は、小ファイルのキャッシュに使用されるヒープスペースを含めた、キャッシュに使用されるヒープスペースをバイトで表したものです。UNIX/Linux の場合、サイズはデフォルトで 1048576 に設定されています。

10 「了解」をクリックします。

11 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

12 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

URL データベースの表示

アクセスプロトコルとサイト名によってグループ分けされ、記録されたすべてのキャッシュされた URL の名前と属性を表示できます。この情報にアクセスすることで、ドキュメントの期限切れやキャッシュからの削除などのさまざまなキャッシュ管理機能を実行できます。

▼ データベースの URL を表示するには

1 サーバーマネージャーにアクセスし、「**Caching**」タブをクリックします。

2 「**View URL Database**」リンクをクリックします。
「View URL Database」ページが表示されます。

3 「**Regenerate**」ボタンをクリックし、キャッシュされた **URL** の最新のリストを生成します。

4 **(**オプション**)** 特定の **URL** の情報を表示する場合は、「**Search**」フィールドに **URL** または正規表現を入力し、「**Search**」ボタンをクリックします。

5 ドメイン名とホストで分類されたキャッシュデータベース情報を表示する場合は、

   a. リストからドメイン名を選択します。
   そのドメイン内にあるホストのリストが表示されます。ホスト名をクリックすると、URL のリストが表示されます。

   b. **URL** の名前をクリックします。
   その URL に関する詳細な情報が表示されます。

   c. **URL** の名前をクリックして、その **URL** に関する詳細情報が表示されます。

▼ キャッシュされた URL を期限切れにするか削除するには

1 サーバーマネージャーにアクセスし、「**Caching**」タブをクリックします。

2 「**View URL Database**」リンクをクリックします。
「View URL Database」ページが表示されます。

3 「**Regenerate**」ボタンをクリックして、キャッシュデータベースのスナップショットを生成します。
このスナップショットが、以降の手順の基準になります。

4 期限切れにするまたは削除する特定の **URL** についての情報を得たい場合は、「**Search**」フィールドにその **URL** またはその **URL** に対応するまたは正規表現を入力し、「**Search**」ボタンをクリックします。

5 ドメイン名とホストで分類された **URL** を操作する場合、

   a. リストからドメイン名を選択します。
   そのドメイン内にあるホストのリストが表示されます。

   b. ホスト名をクリックすると、**URL** のリストが表示されます。

6 個々のファイルを期限切れにするには、

   a. それらのファイルの **URL** の横の「**Ex**」オプションを選択します。

   b. 「**Exp/Rem Marked**」ボタンをクリックします。

7 リスト内のすべてのファイルを期限切れにするには、フォームの下部の「**Exp All**」ボタンをクリックします。

8 キャッシュから個々のファイルを削除するには、

   a. 削除するファイルの **URL** の横の「**Rm**」オプションを選択します。

   b. 「**Exp/Rem Marked**」ボタンをクリックします。

9 リスト内のすべてのファイルを削除するには、「**Rem All**」ボタンをクリックします。

10 「**Regenerate**」ボタンをクリックし、スナップショットを再生成します。

注 – 「Ex」オプションまたは「Remove」オプションを使用する場合、関連ファイルは処理されますが、スナップショットに変更が反映されません。変更を反映させるには、スナップショットを再生成する必要があります。

キャッシュのバッチ更新の使用

指定された Web サイトにファイルをプリロードしたり、プロキシサーバーがビジー状態の場合を除いて、キャッシュ内のドキュメントの最新状態チェックを実行したりすることができます。URL のバッチを作成、編集、削除し、バッチの更新を有効、無効にできます。

バッチ更新の作成

バッチ内の更新するファイルを指定して、ファイルをアクティブにキャッシュできます。現在キャッシュにある複数のファイルに対して、最新状態チェックを実行したり、特定の Web サイト内の複数のファイルをプリロードしたりできます。

▼ バッチ更新を作成するには

1. サーバーマネージャーにアクセスし、「**Caching**」タブをクリックします。
2. 「**Set Cache Batch Updates**」リンクをクリックします。
   「Set Cache Batch Updates」ページが表示されます。
3. 「**Create/Select a Batch Update Configuration**」の横のドロップダウンリストから、「**New and Create**」を選択します。
4. 「了解」をクリックします。「**Set Cache Batch Updates**」ページが表示されます。
5. 「**Name**」セクションに、新しいバッチ更新のエントリの名前を入力します。
6. このページの「**Source**」セクションで、作成するバッチ更新の種類を選択します。
   キャッシュ内のすべてのドキュメントに対して最新状態チェックを実行する場合は、最初のラジオボタンをクリックします。指定されたソース URL から再帰的に URL をキャッシュする場合は、2 つ目のラジオボタンをクリックします。
7. 「**Source**」セクションフィールドで、バッチ更新で使用するドキュメントを指定します。
8. 「**Exceptions**」セクションで、バッチ更新から除外するファイルをすべて指定します。
9. 「**Resources**」セクションに、同時接続の最大数と走査するドキュメントの最大数を入力します。
10. 「了解」をクリックします。
    「Create/Select a Batch Update Configuration」の横のドロップダウンリストから、新しく追加したバッチ名と「Schedule」 を選択します。
11. 「了解」をクリックします。

注 – バッチ更新設定は、この機能を有効にせずに作成、編集、および削除できます。ただし、「Set Cache Batch Updates」ページで設定した時間に従ってバッチ更新を実行する場合は、更新を有効にする必要があります。

12 「**Schedule Batch Updates**」ページが表示されます。

13 「**Update On**」または「**Update Off**」オプションを選択します。

14 ドロップダウンリストから時間を選択し、更新を実行する日付を選択します。

15 「了解」をクリックします。

16 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

17 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

バッチ更新設定の編集または削除

特定のファイルを除外する必要がある場合、またはバッチの更新回数を増やす場合、バッチ更新を編集することができます。また、バッチ更新設定を完全に削除することもできます。

▼ バッチ更新設定を編集または削除するには

1 サーバーマネージャーにアクセスし、「**Caching**」タブをクリックします。

2 「**Set Cache Batch Updates**」リンクをクリックします。
「Set Cache Batch Updates」ページが表示されます。

3 バッチを編集するには、そのバッチの名前を選択し、「**Create/Select a Batch Update Configuration**」の横のドロップダウンリストから「**Edit**」を選択します。

4 「了解」をクリックします。
「Set Cache Batch Updates」ページが表示されます。

5 必要に応じて、情報を変更します。

6 「了解」をクリックします。

7 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

8 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

▼ バッチ更新設定を削除するには

1 サーバーマネージャーにアクセスし、「**Caching**」タブをクリックします。

2 「**Set Cache Batch Updates**」リンクをクリックします。

3 バッチを削除するには、そのバッチの名前を選択し、「**Create/Select a Batch Update Configuration**」の横のドロップダウンリストから「**Delete**」を選択します。

4 「了解」をクリックします。

5 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

6 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

キャッシュのコマンド行インタフェースの使用

プロキシサーバーには、いくつかのコマンド行ユーティリティーが付属しており、これを使用してキャッシュのディレクトリ構造を設定、変更、生成、および修正することができます。これらのユーティリティーの大半は、サーバーマネージャーの各ページの機能と重複しています。たとえば、cron ジョブとして保守をスケジュールする必要がある場合に、ユーティリティーを使用することもできます。すべてのユーティリティーは `extras` ディレクトリにあります。

▼ コマンド行ユーティリティーを実行するには

1 コマンド行プロンプトから *server_root* `/proxy-server`*id* ディレクトリに移動します。

2 **`./start -shell`** と入力します。
次の節では各種のユーティリティーについて説明します。

キャッシュディレクトリ構造の構築

cbuild と呼ばれるプロキシユーティリティーは、オフラインでキャッシュデータベースを管理します。このユーティリティーを使用すると、コマンド行インタフェースを使用して新しいキャッシュ構造を作成したり、既存のキャッシュ構造を変更することができます。サーバーマネージャーのページを使用して、プロキシで新しく作成されたキャッシュを使用できるようにすることができます。

注 – このユーティリティーは server.xml ファイルを更新しません。 cbuild は複数のパーティションを持つキャッシュのサイズを変更できません。cbuild でキャッシュを作成または変更する場合、server.xml ファイルの cachecapacity パラメータを手動で更新してください。

```
<PARTITION partitionname="part1" partitiondir="/home/build/install9
/proxy-server1/cache" maxsize="1600" minspace="5" enabled="true"/>
<CACHE enabled="true" cachecapacity="2000" cachedir="/tmp/cache">
```

---

cbuild ユーティリティーは 2 つのモードで起動できます。1 つ目のモードは次のとおりです。

```
cbuild -d conf-dir -c cache-dir -s cache size
cbuild -d conf-dir -c cache-dir -s cache size -r
```

次に例を示します。

```
cbuild -d server_root/proxy-serverid/config
    -c server_root/proxy-serverid/cache -s 512
cbuild -d server_root/proxy-serverid/config
    -c server_root/proxy-serverid/cache -s 512 -r
```

ここで

- *conf-dir* は *server_root* /proxy-server*id*/config ディレクトリにあるプロキシインスタンスの設定ディレクトリ。
- *cache-dir* はキャッシュ構造のディレクトリ。
- *cache size* はキャッシュの最大サイズ。このオプションは *cache-dim* パラメータと一緒に使用できません。最大サイズは 65135 M バイトです。
- -r は、単一パーティション構成の場合、既存のキャッシュ構造のサイズを変更します。新しいキャッシュを作成する場合は必要ありません。

2 つ目のモードは次のとおりです。

```
cbuild -d conf-dir -c cache-dir -n cache-dim
cbuild -d conf-dir -c cache-dir -n cache-dim -r
```

次に例を示します。

```
cbuild -d server_root/proxy-serverid/config
    -c server_root/proxy-serverid/cache -n 3
cbuild -d server_root/proxy-serverid/config
    -c server_root/proxy-serverid/cache -n 3 -r
```

ここで

- *conf-dir* は *server_root* /proxy-server*id*/config ディレクトリにあるプロキシインスタンスの設定ディレクトリ。
- *cache-dir* はキャッシュ構造のディレクトリ。
- *cache-dim* はセクション数を指定します。たとえば、図 12–2の s3.4 として示されるセクションでは、3 は次元を示します。*cache-dim* のデフォルト値は 0 で、最大値は 8 です。
- -r は、単一パーティション構成の場合、既存のキャッシュ構造のサイズを変更します。このオプションは、新しいキャッシュを作成する場合は必要ありません。

キャッシュ URL リストの管理

プロキシユーティリティー urldb はキャッシュ内の URL リストを管理します。このユーティリティーを使用して、キャッシュされる URL を一覧表示することができます。また、キャッシュされたオブジェクトを選択して有効期限切れにしたり、キャッシュデータベースから削除することができます。

urldb コマンドは -o オプションに基づいて 3 つのグループに分類できます。

- ドメイン
- サイト
- URL
- ドメインを一覧表示するには、コマンド行で次のコマンドを入力します。

  ```
  urldb -o matching_domains -e reg-exp -d conf-dir
  ```

  次に例を示します。

  ```
  urldb -o matching_domains -e ".*phoenix.*" -d server-root/proxy-serverid/config
  ```

  ここで

  - *matching_domains* は正規表現に一致するドメインを一覧表示します。
  - *reg-exp* は使用する正規表現。
  - *conf-dir* は *server-root* /proxy-server*id*/config ディレクトリにあるプロキシインスタンスの設定ディレクトリ。

- ドメイン内で一致するサイトをすべて一覧表示するには、コマンド行で次のように入力します。

  ```
  urldb -o matching_sites_in_domain -e reg-exp -m domain_name -d conf-dir
  ```

  次に例を示します。

  ```
  urldb -o matching_sites_in_domain -e ".*atlas" -m phoenix.com
      -d server-root/proxy-serverid/config
  ```

  ここで

  - `matching_sites_in_domain` は正規表現に一致するドメイン内のすべてのサイトを一覧表示します。
  - *reg-exp* は使用する正規表現。
  - *domain_name* はドメインの名前。
  - *conf-dir* は *server-root* `/proxy-server`*id*`/config` ディレクトリにあるプロキシインスタンスの設定ディレクトリ。
- 一致するサイトをすべて一覧表示するには、コマンド行で次のように入力します。

  ```
  urldb -o all_matching_sites -e reg-exp -d conf-dir
  ```

  次に例を示します。

  ```
  urldb -o all_matching_sites -e ".*atlas.*" -d server-root/proxy-serverid/config
  ```

  ここで

  - `all_matching_sites` は、正規表現に一致するすべてのサイトを一覧表示します。
  - *reg-exp* は使用する正規表現。
  - *conf-dir* は *server-root* `/proxy-server`*id*`/config` ディレクトリにあるプロキシインスタンスの設定ディレクトリ。
- サイト内で一致する URL を一覧表示するには、コマンド行で次のように入力します。

  ```
  urldb -o matching_urls_from_site -e reg-exp -s site_name -d conf-dir
  ```

  次に例を示します。

  ```
  urldb -o matching_urls_from_site -e "http://.*atlas.*" -s atlas.phoenix.com
      -d server-root/proxy-serverid/config
  ```

  ここで

- `matching_urls_from_site` は、正規表現に一致するサイトのすべての URL を一覧表示します。
- *reg-exp* は使用する正規表現。
- *site_name* はサイトの名前。
- *conf-dir* は *server-root* `/proxy-server`*id*`/config` ディレクトリにあるプロキシインスタンスの設定ディレクトリ。

  ディレクトリにないことがあります。

- サイト内で一致する URL を有効期限切れまたは削除するには、コマンド行で次のように入力します。

```
urldb -o matching_urls_from_site -e reg-exp -s site_name -x e -d conf-dir
urldb -o matching_urls_from_site -e reg-exp -s site_name -x r -d conf-dir
```

次に例を示します。

```
urldb -o matching_urls_from_site -e “http://.*atlas.*” -s atlas.phoenix.com
    -x e -d server-root/proxy-serverid/config
```

ここで

- `matching_urls_from_site` は、正規表現に一致するサイトのすべての URL を一覧表示します。
- *reg-exp* は使用する正規表現。
- *site_name* はサイトの名前。
- `-x e` はキャッシュデータベースで一致する URL を有効期限切れにするオプション。このオプションは、ドメインモードおよびサイトモードでは使用できません。
- `-x r` は、キャッシュデータベースで一致する URL を削除するオプション。
- *conf-dir* は、プロキシインスタンスの設定ディレクトリ。これは *server-root* `/proxy-server`*id*`/config` ディレクトリにあります。

- 一致する URL をすべて一覧表示するには、コマンド行で次のように入力します。

```
urldb -o all_matching_urls -e reg-exp -d conf-dir
```

次に例を示します。

```
urldb -o all_matching_urls -e “.*cgi-bin.*” -d
    server-root/proxy-serverid/config
```

ここで

- `all_matching_urls` は、正規表現に一致するすべての URL を一覧表示します。
- *reg-exp* は使用する正規表現。

- *conf-dir* は *server-root* /proxy-server*id*/config ディレクトリにあるプロキシインスタンスの設定ディレクトリ

- 一致する URL をすべて有効期限切れまたは削除するには、コマンド行で次のように入力します。

```
urldb -o all_matching_urls -e reg-exp -x e -d conf-dir
urldb -o all_matching_urls -e reg-exp -x r -d conf-dir
```

次に例を示します。

```
urldb -o all_matching_urls -e ".*cgi-bin.*" -x e -d server-root/proxy-serverid/config
```

ここで

- all_matching_urls は、正規表現に一致するすべての URL を一覧表示します。
- *reg-exp* は使用する正規表現。
- -x e は、キャッシュデータベースで一致する URL を有効期限切れにするオプション。
- -x r は、キャッシュデータベースで一致する URL を削除するオプション。
- *conf-dir* は *server-root* /proxy-server*id*/config ディレクトリにあるプロキシインスタンスの設定ディレクトリ

- URL のリストを期限切れまたは削除するには、コマンド行に次のコマンドを入力します。

```
urldb -l url-list -x e -e reg-exp -d conf-dir
urldb -l url-list -x r -e reg-exp -d conf-dir
```

次に例を示します。

```
urldb -l url.lst -x e -e ".*cgi-bin.*" -d server-root/proxy-serverid/config
```

ここで

- url-list は有効期限切れにする必要がある URL のリストです。 このオプションを使用して、URL リストを表示できます。
- -x e は、キャッシュデータベースで一致する URL を有効期限切れにするオプション。
- -x r は、キャッシュデータベースで一致する URL を削除するオプション。
- *reg-exp* は使用する正規表現。
- *conf-dir* は *server-root* /proxy-server*id*/config ディレクトリにあるプロキシインスタンスの設定ディレクトリ。

キャッシュガベージコレクションの管理

cachegc ユーティリティーを使用すると、キャッシュデータベースにある有効期限切れのオブジェクト、または古くなりすぎてキャッシュサイズの制約によりディレクトリにキャッシュできなくなったオブジェクトを消去できます。

注 – cachegc ユーティリティーを使用する場合は、プロキシインスタンスで CacheGC が実行されていないことを確認してください。

---

cachegc ユーティリティーは、次のように使用します。

```
cachegc -f leave-fs-full-percent -u gc-high-margin-percent -l gc-low-margin-percent -e
    extra-margin-percent -d conf-dir
```

次に例を示します。

```
cachegc -f 50 -u 80 -l 60 -e 5 -d server-root/proxy-serverid/config
```

ここで

- *leave-fs-full-percent* は、ガベージコレクションが実行されない範囲のキャッシュパーティションサイズの割合を決定します。
- *gc-high-margin-percent* は、最大キャッシュサイズの割合を制御します。その割合に達するとガベージコレクションが開始されます。
- *gc-low-margin-percent* はガベージコレクタがターゲットとする最大キャッシュサイズの割合を制御します。
- *extra-margin-percent* は、削除するキャッシュの割合を決定するために、ガベージコレクタによって使用されます。
- *conf-dir* は *server-root* /proxy-server*id*/config ディレクトリにあるプロキシインスタンスの設定ディレクトリ。

バッチ更新の管理

bu ユーティリティーはキャッシュを更新し、2 つのモードで動作します。1 つ目のモードでは、キャッシュデータベース内を反復し、それぞれに HTTP 要求を送信することで、キャッシュ内に存在するすべての URL を更新します。2 つ目のモードでは、指定された URL から開始し、その URL から指定する深さまでのすべてのリンクに対して幅優先の反復を行い、キャッシュにページをフェッチします。bu は RFC 準拠のロボットです。

```
bu -n hostname -p port -t time-lmt -f contact-address -s sleep-time -o object -r n -d conf-dir
```

次に例を示します。

```
bu -n phoenix -p 80 -t 3600 -f admin@phoenix.com -s 60 -o nova -r n
    -d server-root/proxy-serverid/config
```

ここで

- *hostname* は、プロキシが稼働しているマシンのホスト名。デフォルト値は localhost。
- *port* はプロキシサーバーが稼働しているポート。デフォルトのポートは `8080` です。
- *time-lmt* はユーティリティーが実行される制限時間。
- *contact-address* は、`bu` から送信される HTTP 要求で送信される連絡先アドレスを指定します。デフォルト値は `worm@proxy-name` です。
- *sleep-time* は 2 つの連続する要求の間のスリープ時間。デフォルト値は 5 秒です。
- *object* は現在実行している `bu.conf` で指定されるオブジェクト。
- `-r n` オプションは、`-robot.txt` ポリシーに従うかどうかを指定します。デフォルト値は `y` です。
- *conf-dir* は *server-root* `/proxy-serverid/config` ディレクトリにあるプロキシインスタンスの設定ディレクトリ。

Internet Cache Protocol (ICP) の使用

Internet Cache Protocol (ICP) はオブジェクトロケーションプロトコルで、キャッシュ間の対話を可能にします。キャッシュでは ICP を使用して、キャッシュされた URL の存在と、それらの URL を取得するための最適な場所について、クエリーの送信と応答を行います。通常の ICP 交換では、1 つのキャッシュから特定の URL に関する 1 つの ICP クエリーが隣接するすべてのキャッシュに送信されます。次に、隣接するキャッシュに、その URL が含まれているかどうかを示す ICP 応答を返します。キャッシュに URL が含まれていない場合、miss を返します。URL が含まれている場合、hit を返します。

隣接 ICP を経由したルーティング

ICP は異なる管理ドメインに存在するプロキシ間の通信に使用できます。ICP により、ある管理ドメインのプロキシキャッシュは、別の管理ドメイン内のプロキシキャッシュと通信できます。ICP は複数のプロキシサーバー間で通信する必要があるが、プロキシ配列内にあるため、1 つのマスタープロキシからすべてのプロキシサーバーを設定できないような場合に適しています。図 12–3 に、異なる管理ドメインにあるプロキシ間の ICP の交換を示します。

ICP を介して通信するプロキシはネイバーと呼ばれます。隣接 ICP 内に 64 を超えるネイバーを配置できません。隣接 ICP の 2 つの種類のネイバーは *parent* と *sibling* です。要求される URL がほかのネイバーに存在しない場合、parent のみがリモート

サーバーにアクセスできます。隣接 ICP には parent が存在しなくても、また複数の parent が存在していてもかまいません。隣接 ICP の parent 以外のすべてのネイバーは sibling と見なされます。Sibling は ICP のデフォルト経路としてマークされている場合を除き、リモートサーバーからドキュメントを取得できず、ICP はデフォルトを使用します。

ネイバーがクエリーを受け取る順序を決定する場合は、ポーリングラウンドを使用できます。ポーリングラウンドは ICP クエリーのサイクルです。各ネイバーに対して、ポーリングラウンドを割り当てる必要があります。すべてのネイバーをポーリングラウンド 1 に設定する場合、すべてのネイバーは同時に 1 サイクルで照会されます。一部のネイバーをポーリングラウンド 2 に設定した場合、ポーリングラウンド 1 のすべてのネイバーが最初に照会され、どのネイバーも HIT を返さない場合、ラウンド 2 のすべてのプロキシが照会されます。ポーリングラウンドの最大回数は 2 です。

ICP の parent はネットワークのボトルネックになりやすいため、ポーリングラウンドを使用して負荷を軽減できます。一般的な設定では、すべての sibling をポーリングラウンド 1 に設定し、すべての parent をポーリングラウンド 2 に設定します。このようにすることで、ローカルプロキシが URL を要求した場合、要求はまず隣接内のすべての sibling に照会されます。どの sibling にも要求された URL が存在しない場合、要求は parent に照会されます。parent にも URL が存在しない場合、ローカルプロキシはリモートサーバーから URL を取得します。

隣接 ICP の各ネイバーは、少なくとも 1 台の ICP サーバーを稼働させている必要があります。ネイバーが ICP サーバーを稼働させていない場合、ネイバーからの ICP 要求に答えることができません。プロキシサーバーで ICP を有効にすると、まだ稼働していない場合は、ICP サーバーが起動されます。

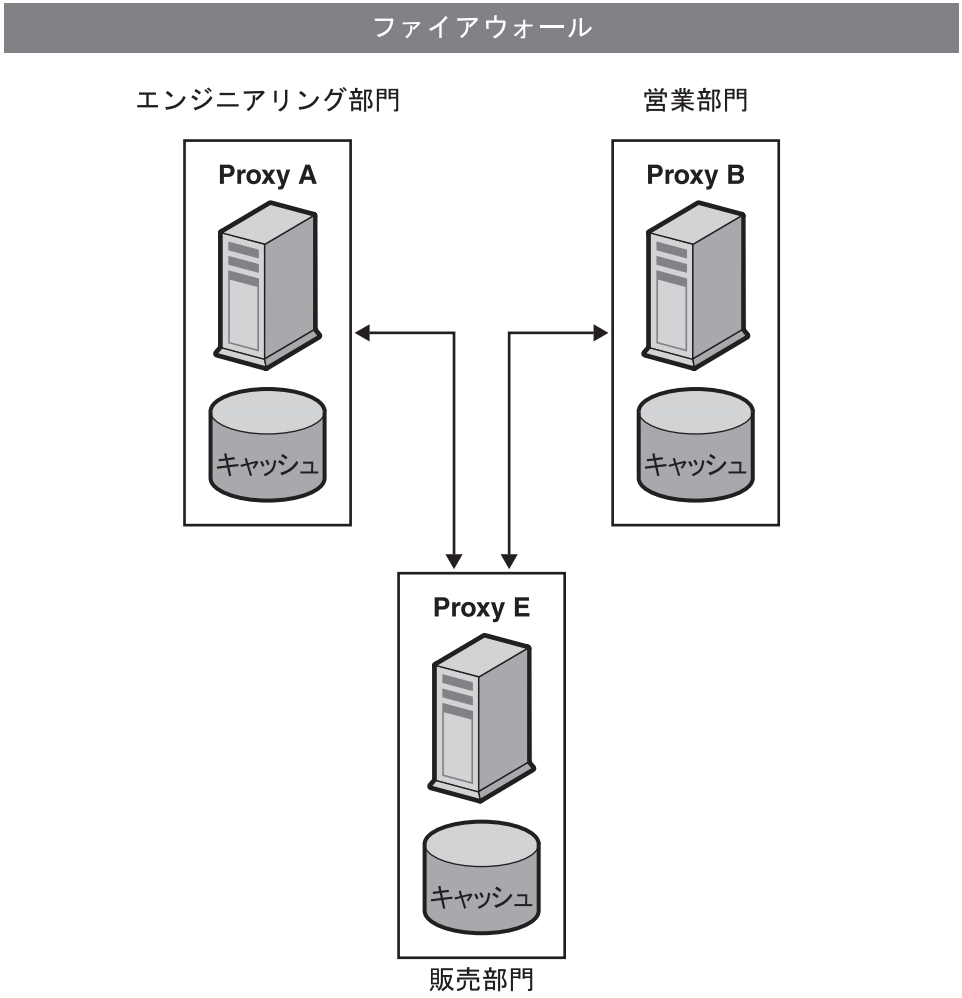


図 12–3　ICP 交換

ICP の設定

ここでは、ICP の設定に関する詳細を説明します。ICP の設定に必要な一般的な手順は次のとおりです。

1. (オプション) 隣接 ICP に parent を追加します。

   詳細については、288 ページの「隣接 ICP に parent または sibling プロキシを追加するには」を参照してください。

2. 隣接 ICP に sibling を追加します。

詳細については、288 ページの「隣接 ICP に parent または sibling プロキシを追加するには」を参照してください。

3. 隣接 ICP の各ネイバーを設定します。

   詳細については、289 ページの「隣接 ICP の設定を編集するには」を参照してください。

4. ICP を有効にします。

   詳細については、292 ページの「ICP を有効にするには」を参照してください。

5. プロキシの隣接 ICP に sibling または parent がある場合、隣接 ICP を経由したルーティングを有効にします。

   詳細については、292 ページの「隣接 ICP を経由したルーティングを有効にするには」を参照してください。

▼ 隣接 ICP に parent または sibling プロキシを追加するには

1 サーバーマネージャーにアクセスし、「**Caching**」タブをクリックします。

2 「**Configure ICP**」リンクをクリックします。

「Configure ICP」ページが表示されます。

3 このページの「**Parent List**」セクションで、「**Add**」ボタンをクリックします。

「ICP Parent」ページが表示されます。

- parent プロキシを追加するには、ページの「Add in the Parent List」セクションをクリックします。

  「ICP Parent」ページが表示されます。

- sibling プロキシを追加するには、ページの「Add in the Sibling List」セクションをクリックしてください。

  「ICP Sibling」ページが表示されます。

4 「**Machine Address**」フィールドに、隣接 **ICP** に追加する **parent** プロキシの **IP** アドレスまたはホスト名を入力します。

5 「**ICP Port**」フィールドに、**parent** プロキシが **ICP** メッセージを待機するポート番号を入力します。

6 (オプション)「**Multicast Address**」フィールドに、**parent** が待機するマルチキャストアドレスを入力します。マルチキャストアドレスは、複数のサーバーが待機できる **IP** アドレスです。

マルチキャストアドレスを使用することによって、そのマルチキャストアドレスを待機しているすべてのネイバーで表示できるネットワークに対して、プロキシから 1 つのクエリーを送信することができます。この手法によって、各ネイバーに個別にクエリーを送信する必要がなくなります。マルチキャストの使用はオプションです。

注 – 別のポーリングラウンドのネイバーは、同じマルチキャストアドレスを待機できません。

---

7 「**TTL**」フィールドに、マルチキャストメッセージが転送されるサブネットの数を入力します。

TTL を 1 に設定すると、マルチキャストメッセージはローカルサブネットにのみ転送されます。TTL を 2 に設定すると、メッセージは、1 レベル先のすべてのサブネットに送信され、3 の場合は 2 レベル先のように拡大されます。

注 – マルチキャストを使用した場合、2 つの関連性のないネイバーで相互に ICP メッセージを送信することができます。したがって、隣接 ICP のプロキシからの ICP メッセージを関連性のないネイバーに受け取られるのを防ぐには、「TTL」フィールドの TTL の値を低く設定します。

---

8 「**Proxy Port**」フィールドに、**parent** のプロキシサーバー用のポートを入力します。

9 「**Polling Round**」ドロップダウンリストから、**parent** に割り当てるポーリングラウンドを選択します。デフォルトのポーリングラウンドは **1** です。

10 「了解」をクリックします。

11 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

12 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

▼ 隣接 **ICP** の設定を編集するには

1 サーバーマネージャーにアクセスし、「**Caching**」タブをクリックします。

2 「**Configure ICP**」リンクを選択します。「**Configure ICP**」ページが表示されます。

3 編集するプロキシの横のラジオボタンを選択します。

4 「編集」ボタンをクリックします。

5 該当する情報を変更します。

6 「了解」をクリックします。

7 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

8 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

▼ 隣接 **ICP** からプロキシを削除するには

1 サーバーマネージャーにアクセスし、「**Caching**」タブをクリックします。

2 「**Configure ICP**」リンクを選択します。「**Configure ICP**」ページが表示されます。

3 削除するプロキシの横のラジオボタンを選択します。

4 ［削除］をクリックします。

5 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

6 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

▼ 隣接 **ICP** のローカルプロキシサーバーを設定するには

隣接 ICP では各ネイバー、すなわちローカルプロキシを設定する必要があります。

1 サーバーマネージャーにアクセスし、「**Caching**」タブをクリックします。

2 「**Configure ICP**」リンクを選択します。
「Configure ICP」ページが表示されます。

3 「**Binding Address**」フィールドに、隣接サーバーがバインドする **IP** アドレスを入力します。

4 「ポート」フィールドに、隣接サーバーが **ICP** を待機するポート番号を入力します。

5 「**Multicast Address**」フィールドに、ネイバーが待機するマルチキャストアドレスを入力します。

マルチキャストアドレスは、複数のサーバーが待機できる IP アドレスです。マルチキャストアドレスを使用することによって、そのマルチキャストアドレスを待機しているすべてのネイバーで表示できるネットワークに対して、プロキシから 1 つのクエリーを送信することができます。この手法によって、各ネイバーに個別にクエリーを送信する必要がなくなります。

ネイバーにマルチキャストアドレスとバインドアドレスの両方が指定されている場合、ネイバーは、応答の送信にはバインドアドレスを使用し、待機にはマルチキャストを使用します。バインドアドレスもマルチキャストアドレスも指定されない場合、オペレーティングシステムによりデータの送信に使用するアドレスが決定されます。

6 「**Default Route**」フィールドに、隣接するプロキシが **HIT** を応答しない場合にネイバーが応答をルーティングするプロキシの名前または **IP** アドレスを入力します。

このフィールドに「origin」と入力した場合、または空白のままにした場合、デフォルト経路で配信元サーバーにルーティングされます。

「No Hit Behavior」ドロップダウンリストから「最初に応答する parent」を選択した場合、「Default Route」フィールドに入力した経路は無効になります。プロキシがこの経路を使用するのは、デフォルトの「ヒット動作なし」を選択した場合のみです。

7 **2** つ目の「ポート」フィールドに、「**Default Route**」フィールドに入力したデフォルト経路のマシンのポート番号を入力します。

8 「**On No Hits, Route Through**」ドロップダウンリストから、隣接 **ICP** のどの **sibling** もキャッシュ内に要求された **URL** がない場合のネイバーの動作を選択します。

利用できるオプションは次のとおりです。

- 最初に応答する **parent**: ネイバーは最初に「miss」で応答する parent から、要求された URL を取得します。
- デフォルトの経路: ネイバーは、「Default Route」フィールドに指定されたマシンから、要求された URL を取得します。

9 「**Server Count**」フィールドに、**ICP** 要求を処理するプロセス数を入力します。

10 「**Timeout**」フィールドに、各ラウンドでネイバーが **ICP** 応答を待機する最大時間を入力します。

11 「了解」をクリックします。

12 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

13 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

▼ ICP を有効にするには

1 サーバーマネージャーにアクセスし、「**Preferences**」タブをクリックします。

2 「**Configure System Preferences**」リンクをクリックします。
「Configure System Preferences」ページが表示されます。

3 **ICP** の「**Yes**」ラジオボタンを選択して、「了解」をクリックします。

4 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

5 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

▼ 隣接 ICP を経由したルーティングを有効にするには

隣接 ICP を経由したルーティングを有効にする必要があるのは、プロキシが隣接 ICP にほかの sibling または parent を持っている場合のみです。プロキシが別のプロキシの parent であり、独自の sibling または parent を持っていない場合、そのプロキシに対してのみ ICP を有効にする必要があります。隣接 ICP を経由したルーティングを有効にする必要はありません。

1 サーバーマネージャーにアクセスし、「**Routing**」タブをクリックします。

2 「**Set Routing Preferences**」リンクをクリックします。
「Set Routing Preferences」ページが表示されます。

3 ドロップダウンリストからリソースを選択するか、「**Regular Expression**」ボタンをクリックして正規表現を入力し、「了解」をクリックします。

4 「**Route Through**」オプションの横のラジオボタンを選択します。

5 **ICP** の横のチェックボックスを選択します。

6 (オプション) クライアントが別のネイバーを経由してドキュメントを取得するのではなく、そのドキュメントを持つ隣接 **ICP** から直接ドキュメントを取得するように設定するには、「**Text Redirect**」の横のチェックボックスを選択します。

7 「了解」をクリックします。

注意 – 現在リダイレクトはどのクライアントでもサポートされていないため、今回はこの機能を使用しないでください。

8 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

9 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

プロキシ配列の使用

分散キャッシュのためにプロキシ配列を使用すると、複数のプロキシが 1 つのキャッシュとして機能します。配列内の各プロキシに、ブラウザまたはダウンストリームのプロキシサーバーにより取得可能な、さまざまなキャッシュ URL が格納されます。プロキシ配列を使用すると、複数のプロキシサーバーを使用するとしばしば発生する、キャッシュの重複を避けることができます。ハッシュベースのルーティングを通じて、プロキシ配列はプロキシ配列内の正しいキャッシュに要求をルーティングします。

またプロキシ配列では、増分スケーラビリティが可能です。プロキシ配列へのプロキシの追加を決定した場合、各メンバーのキャッシュは無効になりません。各メンバーのキャッシュ内の 1/*n* の URL のみが、ほかのメンバーに再割り当てされます (*n* は配列のプロキシの数)。

プロキシ配列を経由したルーティング

プロキシ配列からの各要求について、ハッシュ機能は配列内の各プロキシに、要求された URL、プロキシ名、およびプロキシの負荷要因に基づくスコアを割り当てます。要求はスコアが最も高いプロキシにルーティングされます。

URL の要求はクライアントとプロキシの両方から送られる可能性があるため、プロキシ配列を経由するルーティングには次の 2 種類あります。クライアントからプロキシへのルーティングとプロキシからプロキシへのルーティング。

クライアントからプロキシへのルーティングでは、クライアントはプロキシ自動設定 (Proxy Auto Configuration、PAC) メカニズムを使用して、経由するプロキシを決定します。ただし、クライアントは、標準の PAC ファイルを使用する代わりに、

ハッシュアルゴリズムを計算する特別な PAC ファイルを使用し、要求された URL に適切な経路を決定します。図 12–4 にクライアントからプロキシへのルーティングを示します。この図では、プロキシ配列の各メンバーがロードされ、マスタープロキシに対してPAT ファイルに加えられた更新をポーリングします。PAC ファイルを取得したクライアントに必要となるのは、設定が変更されるたびにこのファイルをダウンロードすることだけです。通常、クライアントは再起動時に PAC ファイルをダウンロードします。

プロキシサーバーは、管理インタフェースから作成された Proxy Array Membership Table (PAT) 仕様に基づいて特別な PAC ファイルを自動的に生成できます。

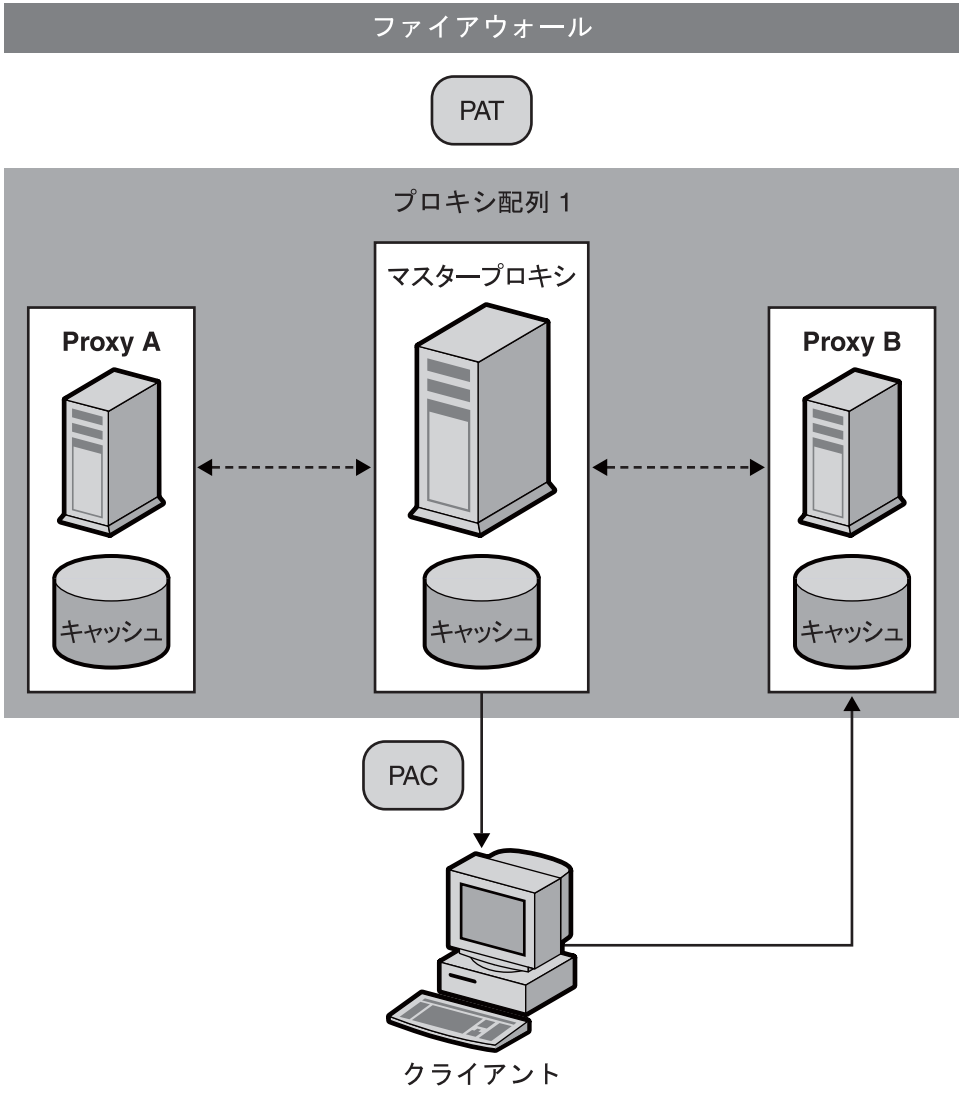


図 12–4　クライアントからプロキシへのルーティング

プロキシからプロキシへのルーティングでは、プロキシは、クライアントが使用する PAC ファイルの代わりに PAT (Proxy Array Table) ファイルを使用して、ハッシュアルゴリズムを計算します。PAT ファイルは、プロキスのマシン名、IP アドレス、ポート、負荷係数、キャッシュサイズなどを含むプロキシ配列に関する情報を格納した ASCII ファイルです。サーバー側でハッシュアルゴリズムを計算する場合、PAC ファイル (実行時に解釈が必要な JavaScript ファイル) よりも PAT ファイルを使用する

方がはるかに効率的です。ただし、ほとんどのクライアントは PAT ファイル形式を認識しないため、PAC ファイルを使用する必要があります。図 12–5 にプロキシからプロキシへのルーティングを示します。

PAT ファイルはプロキシ配列のマスタープロキシで作成されます。プロキシの管理者は、マスタープロキシにするプロキシを決定する必要があります。管理者は、このマスタープロキシサーバーから PAT ファイルを変更できます。プロキシ配列のほかのすべてのメンバーは、これらの変更について、マスタープロキシを手動または自動でポーリングできます。これらの変更に基づき、PAC ファイルを自動的に生成するように各メンバーを設定できます。

また、プロキシ配列を連鎖して、階層ルーティングを作成することもできます。プロキシサーバーが受信した要求を上流のプロキシ配列にルーティングする場合、上流のプロキシ配列は親配列となります。つまり、クライアントがプロキシ X のドキュメントを要求し、プロキシ X にそのドキュメントがない場合、クライアントは直接リモートサーバーに要求を送信する代わりに、プロキシ配列 Y に要求を送信します。このため、プロキシ配列 Y が親配列になります。

図 12–5で、プロキシ配列 1 はプロキシ配列 2 の親配列です。プロキシ配列 2 のメンバーは親配列の PAT ファイルへの更新をロードし、ポーリングします。通常は、親配列のマスタープロキシに対してポーリングします。要求された URL のハッシュアルゴリズムは、ダウンロードされた PAT ファイルを使用して計算されます。プロキシ配列 2 のメンバーは、プロキシ配列 1 の中で最もスコアの高いプロキシを通して要求された URL を取得します。図では、クライアントから要求された URL に関して最もスコアが高いのは、プロキシ B です。

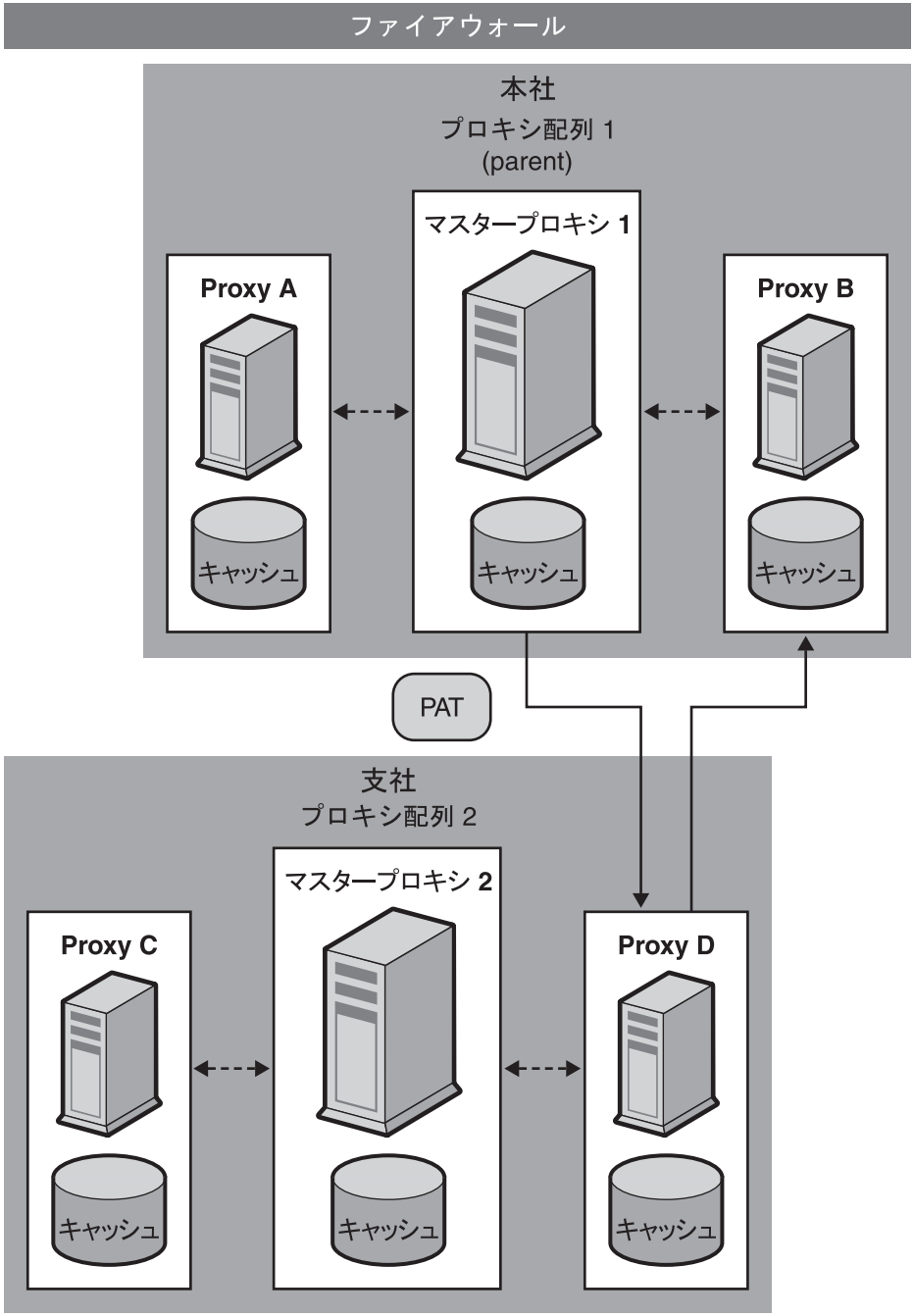


図 12–5　プロキシからプロキシへのルーティング

プロキシ配列を設定する一般的な手順は次のようになります。

マスタープロキシから、次の手順を実行します。

1. プロキシ配列を作成します。

   メンバーリストの作成の詳細については、299 ページの「プロキシ配列メンバーリストの作成」を参照してください。
2. PAT ファイルから PAC ファイルを生成します。

   PAC ファイルを生成する必要があるのは、クライアントからプロキシへのルーティングを使用する場合のみです。詳細については、304 ページの「PAT ファイルからの PAC ファイルの生成」を参照してください。
3. 配列のマスターメンバーを設定します。詳細については、301 ページの「プロキシ配列メンバーの設定」を参照してください。
4. プロキシ配列を経由したルーティングを有効にします。詳細については、303 ページの「プロキシ配列を経由したルーティングの有効化」を参照してください。
5. URL `/pat` を PAT ファイルにマップする PAT マッピングを作成します。
6. プロキシ配列を有効にします。

   詳細については、303 ページの「プロキシ配列の有効化または無効化」を参照してください。

マスター以外の各プロキシから、次の手順を実行します。

1. 配列のマスター以外のメンバーを設定します。

   詳細については、301 ページの「プロキシ配列メンバーの設定」を参照してください。
2. プロキシ配列を経由したルーティングを有効にします。

   詳細については、303 ページの「プロキシ配列を経由したルーティングの有効化」を参照してください。
3. プロキシ配列を有効にします。

   詳細については、303 ページの「プロキシ配列の有効化または無効化」を参照してください。

注 – プロキシ配列が親配列を経由してルーティングされる場合、親配列を有効にし、各メンバーが親配列を経由して要求される URL にルーティングされるように設定する必要があります。詳細については、306 ページの「親配列を経由したルーティング」を参照してください。

プロキシ配列メンバーリストの作成

プロキシ配列のメンバーリストの作成および更新は、配列のマスタープロキシからのみ行うことができます。プロキシ配列のメンバーリストは1度だけ作成する必要がありますが、いつでも変更できます。プロキシ配列メンバーリストを作成することにより、配列内のすべてのプロキシとダウンストリームプロキシに配布されるPATファイルが生成されます。

注– プロキシ配列のメンバーリストへの変更または追加は、配列内のマスタープロキシからのみ実行します。配列内のほかのメンバーはすべて、メンバーリストの読み込みのみ可能です。

▼ プロキシ配列メンバーリストを作成するには

1 サーバーマネージャーにアクセスし、「**Caching**」タブをクリックします。

2 「**Configure Proxy Array**」リンクをクリックします。
「Configure Proxy Array」ページが表示されます。

3 「**Array name**」フィールドに、配列の名前を入力します。

4 「**Reload Configuration Every**」フィールドに、**PAT**ファイルのポーリング間隔を分で入力します。

5 「**Array Enabled**」チェックボックスをクリックします。

6 「**Create**」ボタンをクリックします。
「Create」ボタンは、プロキシ配列が作成されたあと、「了解」ボタンに変わります。

注– メンバーをメンバーリストに追加する前に、必ず「了解」をクリックしてください。

---

7 「了解」をクリックします。

8 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

9 プロキシ配列の各メンバーについて、次の内容を入力し、「了解」をクリックします。
最初にマスターメンバーを追加してから、ほかのメンバーを追加してください。

- 「名前」: メンバーリストに追加するプロキシサーバーの名前
- 「**IP** アドレス」: メンバーリストに追加するプロキシサーバーの IP アドレス
- 「ポート」: メンバーが PAT ファイルをポーリングするポート
- 「負荷係数」: メンバーを経由してルーティングする相対的負荷を示す整数
- 「状態情報」: メンバーの状態。値はオンまたはオフ。プロキシ配列メンバーを無効にした場合、メンバーの要求は別のメンバーに再ルーティングされます。

注 – 追加するプロキシ配列メンバーごとに情報を入力したあと、必ず「了解」をクリックしてください。

---

10 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

11 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

プロキシ配列メンバーリスト情報の編集

プロキシ配列メンバーリスト内のメンバー情報は、いつでも変更できます。プロキシ配列メンバーリストは、マスタープロキシからのみ編集できます。

注 – プロキシ配列のメンバーリストへの変更または追加は、配列内のマスタープロキシからのみ実行します。配列のほかのメンバーからこのリストを変更した場合、すべての変更内容は失われます。

▼ メンバーリスト情報を編集するには

1 サーバーマネージャーにアクセスし、「**Caching**」タブをクリックします。

2 「**Configure Proxy Array**」リンクをクリックします。
「Configure Proxy Array」ページが表示されます。

3 「**Member List**」で、編集するメンバーの横のラジオボタンを選択します。

4 「編集」ボタンをクリックします。
「Configure Proxy Array Member」ページが表示されます。

5 該当する情報を編集します。

6 「了解」をクリックします。

7 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

8 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

注 – 変更内容を有効にし、プロキシ配列のメンバーに配布する場合、「Configure Proxy Array」ページの「Configuration ID」を更新し、「了解」をクリックします。設定 ID を更新する場合、ID に 1 を加えます。

プロキシ配列メンバーの削除

プロキシ配列のメンバーを削除すると、プロキシ配列から削除されます。プロキシ配列のメンバーは、マスタープロキシからのみ削除できます。

▼ プロキシ配列からメンバーを削除するには

1 サーバーマネージャーにアクセスし、「**Caching**」タブをクリックします。

2 「**Configure Proxy Array**」リンクをクリックします。
「Configure Proxy Array」ページが表示されます。

3 「**Member List**」で、削除するメンバーの横のラジオボタンを選択します。

4 「**Delete**」をクリックします。

注 – 変更内容を有効にし、プロキシ配列のメンバーに配布する場合、「Configure Proxy Array」ページの「Configuration ID」を更新し、「了解」をクリックします。設定 ID を更新する場合、ID に 1 を加えます。

---

5 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

6 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

プロキシ配列メンバーの設定

プロキシ配列内のメンバーは 1 度だけ設定する必要があります。これは各メンバーから行います。配列のメンバーの設定は、別のメンバーからは行うことができません。また、マスタープロキシを設定する必要があります。

▼ プロキシ配列の各メンバーを設定するには

1 サーバーマネージャーにアクセスし、「**Caching**」タブをクリックします。

2 「**Configure Proxy Array Member**」リンクをクリックします。
「Configure Proxy Array Member」ページが表示されます。

3 「**Proxy Array**」セクションで、適切なラジオボタンを選択し、メンバーが **PAT** ファイルをポーリングする必要があるかどうかを指定します。

- 「**Non-Master Member**」: 設定するメンバーがマスタープロキシ以外の場合にこのオプションを選択します。マスタープロキシ以外のプロキシ配列メンバーは、マスタープロキシから PAT ファイルを取得するために、PAT ファイルをポーリングする必要があります。
- 「**Master Member**」: マスタープロキシを設定する場合はこのオプションを選択します。マスタープロキシを設定する場合、PAT ファイルはローカルのため、ポーリングする必要はありません。

4 「**Poll Host**」フィールドに、**PAT** ファイルをポーリングするマスタープロキシの名前を入力します。

5 「**Port**」フィールドに、マスタープロキシが **HTTP** 要求を受け取るポートを入力します。

6 「**URL**」フィールドに、マスタープロキシの **PAT** ファイルの **URL** を入力します。マスタープロキシで、**PAT** ファイルを **URL** `/pat` にマッピングする **PAT** マッピングを作成している場合、「**URL**」フィールドに **`/pat`** と入力します。

7 **(**オプション**)**「**Headers File**」フィールドに、**PAT** ファイルに対する **HTTP** 要求 **(**認証情報など**)** とともに送信する必要がある特殊ヘッダー付きのファイルのフルパス名を入力します。

8 「了解」をクリックします。

9 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

10 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

プロキシ配列を経由したルーティングの有効化

▼ プロキシ配列を経由したルーティングを有効にするには

1 サーバーマネージャーにアクセスし、「**Routing**」タブをクリックします。

2 「**Set Routing Preferences**」リンクをクリックします。
「Set Routing Preferences」ページが表示されます。

3 ドロップダウンリストからリソースを選択するか、「**Regular Expression**」ボタンをクリックして正規表現を入力し、「了解」をクリックします。

4 「**Route Through**」オプションを選択します。

5 「**proxy array**」または「**parent array**」のチェックボックスを選択します。
設定するプロキシサーバーがプロキシ配列のメンバーである場合のみ、プロキシ配列のルーティングを有効にできます。親配列が存在する場合、親ルーティングのみを有効にできます。いずれのルーティングオプションも、独立して使用されます。

6 プロキシ配列経由でルーティングし、要求を別の **URL** にリダイレクトする場合、「**redirect**」チェックボックスを選択します。
リダイレクトは、プロキシ配列のメンバーが処理できない要求を受け取った場合、クライアントに、その要求を照会するプロキシを伝えることを意味します。

7 「了解」をクリックします。

8 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

9 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

プロキシ配列の有効化または無効化

プロキシ配列を経由したルーティングを行わない場合、プロキシ配列オプションを無効にする前に、すべてのクライアントが特別な PAC ファイルを使用して正しくルーティングしていることを確認する必要があります。親配列オプションを無効にする場合、「Set Routing Preferences」ページで、明示的プロキシまたは直接接続など、有効な代替ルーティングオプションを設定する必要があります。

▼ プロキシ配列を有効または無効にするには

1. サーバーマネージャーにアクセスし、「**Preferences**」タブをクリックします。
2. 「**Configure System Preferences**」リンクをクリックします。
   「Configure System Preferences」ページが表示されます。
3. プロキシ配列を有効または無効にします。
   - プロキシ配列を有効にするには、有効にする配列の種類(通常のプロキシ配列または親配列) の「Yes」オプションをクリックします。
   - プロキシ配列を無効にするには「No」をクリックします。
4. 「了解」をクリックします。
5. 「**Restart Required**」をクリックします。
   「Apply Changes」ページが表示されます。
6. 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

プロキシ配列の要求のリダイレクト

プロキシ配列経由のルーティングを選択する場合、要求を別の URL にリダイレクトするかどうかを指定する必要があります。リダイレクトは、プロキシ配列のメンバーが処理できない要求を受け取った場合、クライアントに、その要求を照会するプロキシを伝えることを意味します。

PAT ファイルからの PAC ファイルの生成

ほとんどのクライアントは PAT ファイル形式を認識しないため、クライアントからプロキシへのルーティングを使用するクライアントはプロキシ自動設定 (PAC) メカニズムを使用して、どのプロキシを経由するかについての情報を受け取ります。ただし、標準の PAC ファイルを使用する代わりに、クライアントは PAT ファイルから生成された特別な PAC ファイルを使用します。この特別な PAC ファイルは、ハッシュアルゴリズムを計算して、要求された URL に最適な経路を決定します。

PAC ファイルは PAT ファイルから手動で、または自動的に生成できます。プロキシ配列の特定のメンバーから手動で PAC ファイルを生成する場合、そのメンバーは PAT ファイルの現在の情報に基づいてただちに PAC ファイルを再生成します。プロキシ配列メンバーにPAC ファイルの自動生成を設定する場合、PAT ファイルが変更されたことを検出するたびに、メンバーは自動的にファイルを再生成します。

注 – プロキシサーバーのプロキシ配列機能を使用していない場合、「Create/Edit Autoconfiguration File」ページを使用して PAC ファイルを生成します。詳細については、第 17 章を参照してください。

▼ PAT ファイルから PAC ファイルを手動で生成するには

PAC ファイルは、マスタープロキシからのみ生成できます。

1. マスタープロキシのサーバーマネージャーにアクセスし、「**Caching**」タブをクリックします。
2. 「**Configure Proxy Array**」リンクをクリックします。
   「Configure Proxy Array」ページが表示されます。
3. 「**Generate PAC**」ボタンをクリックします。
   「PAC Generation」ページが表示されます。
4. **PAC** ファイルでカスタムロジックを使用する場合、「**Custom logic file**」フィールドに、**PAC** ファイルの生成時に組み込むカスタムロジックを含むファイル名を入力します。
   このロジックは、`FindProxyForURL` 関数のプロキシ配列の選択ロジックの前に挿入されます。この関数は、通常はプロキシ配列を経由する必要のないローカル要求で使用されます。

   プロキシ配列メンバーの設定時に、すでにカスタムロジックファイルを指定している場合、このフィールドにはその情報が表示されます。ここでカスタムロジックファイルの名前を編集できます。
5. 「**Default Route**」フィールドに、配列内のプロキシが利用できないときにクライアントが利用する経路を入力します。
   プロキシ配列メンバーの設定時に、すでにデフォルト経路を指定している場合、このフィールドにはその情報が表示されます。ここでデフォルト経路を編集できます。
6. 「了解」をクリックします。
7. 「**Restart Required**」をクリックします。
   「Apply Changes」ページが表示されます。
8. 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

▼ PAC ファイルを自動生成するには

1 サーバーマネージャーにアクセスし、「**Caching**」タブをクリックします。

2 「**Configure Proxy Array Member**」リンクをクリックします。
「Configure Proxy Array Member」ページが表示されます。

3 「**Auto-generate PAC File**」チェックボックスを選択します。

4 PAC ファイルでカスタムロジックを使用する場合、「**Custom Logic File**」フィールドに、PAC ファイルの生成時に組み込むカスタムロジックを含むファイル名を入力します。
このロジックは、`FindProxyForURL` 関数のプロキシ配列の選択ロジックの前に挿入されます。

プロキシ配列の設定時に、すでにカスタムロジックファイルを指定し、保存している場合、このフィールドにはその情報が表示されます。ここでカスタムロジックファイルの名前を編集できます。

5 「**Default Route**」フィールドに、配列内のプロキシが利用できないときにクライアントが利用する経路を入力します。
プロキシ配列の設定時に、すでにデフォルト経路を指定し、保存している場合、このフィールドにはその情報が表示されます。デフォルト経路を編集できます。

6 「了解」をクリックします。

7 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

8 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

親配列を経由したルーティング

直接リモートサーバーに接続する代わりに、上流の親配列を経由してルーティングするよう、プロキシまたはプロキシ配列のメンバーを設定できます。

▼ 親配列を経由してルーティングするには

1 親配列を有効にします。
詳細については、303 ページの「プロキシ配列の有効化または無効化」を参照してください。

2. 親配列を経由するルーティングを有効にします。
   詳細については、303 ページの「プロキシ配列を経由したルーティングの有効化」を参照してください。
3. サーバーマネージャーにアクセスし、「**Caching**」タブをクリックします。
4. 「**Configure Proxy Array Member**」リンクをクリックします。
   「Configure Proxy Array Member」ページが表示されます。
5. このページの「**Parent Array**」セクションにある「**Poll Host**」フィールドに、**PAT** ファイルのポーリング対象にする親配列にあるプロキシのホスト名を入力します。
   このプロキシは、通常、親配列のマスタープロキシになります。
6. このページの「**Parent Array**」セクションにある「ポート」フィールドに、**PAT** ファイルをポーリングする親配列にあるプロキシのポート番号を入力します。
7. 「**URL**」フィールドに、マスタープロキシの **PAT** ファイルの **URL** を入力します。
   マスタープロキシで PAT マッピングを作成している場合、この「URL」フィールドにマッピングを入力する必要があります。
8. **(**オプション**)** このフォームの「**Parent Array**」セクションにある「**Headers File**」フィールドに、**PAT** ファイルに対する **HTTP** 要求 **(**認証情報など**)** とともに送信する必要がある特殊ヘッダー付きのファイルのフルパス名を入力します。
   このフィールドは省略可能です。
9. 「了解」をクリックします。
10. 「**Restart Required**」をクリックします。
    「Apply Changes」ページが表示されます。
11. 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

親配列情報の表示

プロキシ配列が親配列経由でルーティングする場合、親配列のメンバーの情報が必要です。この情報は PAT ファイルの形式で親配列から送信されます。

▼ 親配列情報を表示するには

1. サーバーマネージャーにアクセスし、「**Caching**」タブをクリックします。
2. 「**View Parent Array Configuration**」リンクをクリックします。
   「View Parent Array Configuration」ページが表示されます。
3. 情報を表示します。

◆ ◆ ◆ 第 13 章

13 プロキシを使用したコンテンツのフィルタリング

この章では、URL にフィルタをかけて、Proxy Server がその URL へのアクセスできないようにしたり、URL からクライアントに返される HTML および JavaScript コンテンツを変更できないようにする方法について説明します。また、この章では、クライアントが使用している Web ブラウザ (ユーザーエージェント) に基づいて、プロキシを使用してアクセスを制限できるようにする方法についても説明します。

URL フィルタファイルを使用して、サーバーがサポートする URL を決定することができます。たとえば、サポートする URL のワイルドカードパターンを手動で入力する代わりに、制限する URL が含まれた 1 つのテキストファイルを作成または購入することができます。この機能によって、URL の格納されたファイルを 1 つ作成すれば、それを多くの異なるプロキシサーバーに対して使用することができます。

また、MIME タイプに基づいて URL にフィルタをかけることもできます。たとえば、プロキシに HTML ファイルと GIF ファイルのキャッシュおよび送信は許可しても、コンピュータウイルスによるリスクを回避するために、バイナリファイルや実行可能ファイルの受信は許可しないようにすることもできます。

この章の内容は次のとおりです。

URLのフィルタリング

URLのファイルを使用して、プロキシサーバーが取得するコンテンツを設定することができます。プロキシが常にサポートするURLのリストと、プロキシがサポートしないURLのリストを設定できます。

たとえば、子どもに適したコンテンツを持つ、プロキシサーバーを実行するインターネットサービスプロバイダであれば、子どもが閲覧することが認可されたURLのリストを設定します。そうすると、認可されたURLのみを取得するようにプロキシサーバーを設定することができます。クライアントがサポートされていないURLにアクセスしようとすると、プロキシからデフォルトの「Forbidden」メッセージを返すようにするか、クライアントがそのURLにアクセスできなかった理由を示すカスタムメッセージを作成することができます。

URLに基づいてアクセスを制限するには、許可または制限するURLのファイルを作成する必要があります。このファイルは、サーバーマネージャーで作成できます。ファイルを作成したら、制限を設定できます。次の節では、これらの方法について説明します。

URLのフィルタファイルの作成

フィルタファイルとは、URLのリストが含まれているファイルのことです。プロキシサーバーが使用するフィルタファイルは、次のようなパターンでURLの行が含まれているプレーンテキストファイルです。

```
protocol://host:port/path/filename
```

3つのセクション`protocol`、`host:port`、および`path/filename`では、正規表現を使用することができます。たとえば、`netscape.com`ドメインにアクセスするすべてのプロトコルのURLパターンを作成する場合は、ファイルに次の行を追加します。

```
.*://.*\\.example\\.com/.*
```

この行は、ポート番号を指定しない場合のみ機能します。正規表現の詳細については、第16章の「正規表現について」を参照してください。

サーバーマネージャーを使用せずに独自のファイルを作成する場合は、サーバーマネージャーのページを使用して空のファイルを作成し、そのファイルにテキストを追加するか、そのファイルを正規表現を格納するファイルと置き換えます。

▼ フィルタファイルを作成するには

1 サーバーマネージャーにアクセスし、「**Filters**」タブをクリックします。

2 「**Restrict URL Filter Access**」リンクをクリックします。
「Restrict URL Filter Access」ページが表示されます。

3 「**Create/Edit**」ボタンの横のドロップダウンリストから、「**New Filter**」を選択します。

4 ドロップダウンリストの右のテキストボックスにフィルタファイルの名前を入力して、「**Create/Edit**」ボタンをクリックします。
「Filter Editor」ページが表示されます。

5 スクロール可能な「**Filter Content**」テキストボックスを使用して、**URL** と **URL** の正規表現を入力します。
「Reset」ボタンを押すと、このフィールドのすべてのテキストがクリアされます。
正規表現の詳細については、第 16 章の「正規表現について」を参照してください。

6 「了解」をクリックします。
Proxy Server によってファイルが作成され、「Restrict URL Filter Access」ページに戻ります。フィルタファイルは `proxy-server`*id*`/conf_bk` ディレクトリに作成されます。

フィルタファイルに対するデフォルトアクセスの設定

使用する URL を含むフィルタファイルを作成したら、これらの URL に対するデフォルトアクセスを設定できます。

▼ フィルタファイルに対するデフォルトアクセスを設定するには

1 サーバーマネージャーにアクセスし、「**Filters**」タブをクリックします。

2 「**Restrict URL Filter Access**」リンクをクリックします。
「Restrict URL Filter Access」ページが表示されます。

3 フィルタで使用するテンプレートを選択します。
通常は、プロキシサーバー全体に対するフィルタファイルを作成しますが、HTTP 用に 1 セットのフィルタファイル、FTP 用にもう 1 セットのフィルタファイルを作成することもできます。

4 「**URL Filter To Allow**」リストを使用して、**Proxy Server** がサポートする **URL** を含むフィルタファイルを選択します。

5 「**URL Filter To Deny**」リストを使用して、**Proxy Server** がアクセスを拒否する **URL** を含むフィルタファイルを選択します。

6 拒否された **URL** を要求したクライアントに **Proxy Server** が返すテキストを選択します。

- プロキシが生成するデフォルトの「Forbidden」応答を返信します。
- カスタムテキストを含むテキストファイルか HTML ファイルを送信します。このファイルへの絶対パスをテキストボックスに入力します。

7 「了解」をクリックします。

8 「**Restart Required**」をクリックします。「**Apply Changes**」ページが表示されます。

9 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

コンテンツ **URL** のリライト

Proxy Server では、クライアントに返されるコンテンツを調べて、URL などのパターンを別の文字列に置き換えることができます。元の文字列と置換後の文字列の 2 つのパラメータを設定できます。Proxy Server は元の文字列と一致するテキストを検索し、置換後の文字列に置き換えます。この機能は、逆プロキシモードでのみ動作します。

▼ **URL** のリライトパターンを作成するには

1 サーバーマネージャーにアクセスし、「**Filters**」タブをクリックします。

2 「**Set Content URL Rewriting**」リンクをクリックします。
「Set Content URL Rewriting」ページが表示されます。

3 ドロップダウンリストからリソースを選択するか、正規表現を指定します。
正規表現の詳細については、第 16 章の「正規表現について」を参照してください。

4 「**Source Pattern**」テキストボックスで、元の文字列を指定します。

5 「**Destination Pattern**」テキストボックスで、置換後の文字列を指定します。

6 「**MIME Pattern**」テキストボックスで、コンテンツタイプを指定します。

7 「了解」をクリックします。

8 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

9 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

▼ URL のリライトパターンを編集するには

1 サーバーマネージャーにアクセスし、「**Filters**」タブをクリックします。

2 「**Set Content URL Rewriting**」リンクをクリックします。
「Set Content URL Rewriting」ページが表示されます。

3 編集する **URL** リライトパターンの横の「**Edit**」リンクをクリックします。

4 「了解」をクリックして変更内容を保存し、

5 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

6 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

▼ URL のリライトパターンを削除するには

1 サーバーマネージャーにアクセスし、「**Filters**」タブをクリックします。

2 「**Set Content URL Rewriting**」リンクをクリックします。
「Set Content URL Rewriting」ページが表示されます。

3 削除する **URL** リライトパターンの横の「**Remove**」リンクをクリックします。
「了解」をクリックして、削除を確認します。

4 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

5 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

特定の Web ブラウザへのアクセス制限

クライアントの Web ブラウザの種類やバージョンに基づいて、プロキシサーバーへのアクセスを制限することができます。すべての Web ブラウザが要求の際にサーバーに送信する user-agent ヘッダーに基づいて制限が課されます。

▼ クライアントの Web ブラウザに基づいてプロキシへのアクセスを制限するには

1. サーバーマネージャーにアクセスし、「**Filters**」タブをクリックします。
2. 「**Set User-Agent Restriction**」リンクをクリックします。
   「Set User-Agent Restriction」ページが表示されます。
3. ドロップダウンリストからリソースを選択するか、**Proxy Server** がサポートするブラウザの **user-agent** 文字列と一致する正規表現を入力します。
   複数のクライアントを指定する場合は、正規表現を括弧で囲み、「|」文字を使用して複数のエントリを区切ります。正規表現の詳細については、第 16 章の「正規表現について」を参照してください。
4. 「**Allow Only User-Agents Matching**」オプションにチェックマークを付けます。
5. 「了解」をクリックします。
6. 「**Restart Required**」をクリックします。
   「Apply Changes」ページが表示されます。
7. 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

要求のブロック

アップロードするコンテンツタイプに基づいて、ファイルのアップロードおよびその他の要求をブロックすることもできます。

▼ MIMEタイプに基づいて要求をブロックするには

1. サーバーマネージャーにアクセスし、「**Filters**」タブをクリックします。
2. 「**Set Request Blocking**」リンクをクリックします。
   「Set Request Blocking」ページが表示されます。
3. ドロップダウンリストからリソースを選択するか、「**Regular Expression**」ボタンをクリックして正規表現を入力し、「了解」をクリックします。
4. 必要な要求ブロックのタイプを選択します。
   - 「Disabled」— 要求のブロックを無効にします。
   - 「Multipart MIME (File Upload)」— すべてのファイルのアップロードをブロックします。
   - 「MIME Types Matching Regular Expression」— 入力した正規表現と一致するMIMEタイプの要求をブロックします。正規表現の詳細については、第16章の「正規表現について」を参照してください。
5. すべてのクライアントの要求をブロックするのか、入力した正規表現と一致する**user-agents**の要求をブロックするのかを選択します。
6. 要求をブロックするメソッドを選択します。
   次のオプションがあります。
   - 「Any Method With Request Body」—メソッドに関係なく、要求のボディーを含むすべての要求をブロックします。
   - メソッドは次のものに限定されます。
     - 「POST」— POSTメソッドを使用するファイルアップロード要求をブロックします。
     - 「PUT」— PUTメソッドを使用するファイルアップロード要求をブロックします。
   - 「Methods Matching Regular Expression」— 入力したメソッドを使用するすべてのファイルアップロード要求をブロックします。
7. 「了解」をクリックします。
8. 「**Restart Required**」をクリックします。
   「Apply Changes」ページが表示されます。
9. 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

送信されるヘッダーの抑止

通常セキュリティー上の理由で、送信されるヘッダーを要求から削除するように Proxy Server を設定することができます。たとえば、From ヘッダーを送信しないようにして、ユーザーの電子メールアドレスが漏れることを防いだり、user-agent ヘッダーをフィルタで除外して、組織で使用する Web ブラウザを外部のサーバーから特定できないようにすることもできます。また、要求がインターネットに転送される前に、イントラネット内でしか使用しないログやクライアント関連のヘッダーを削除することもできます。

この機能は、プロキシ自体により特別に処理、または生成されるヘッダーや、If-Modified-Since や Forwarded などプロトコルが適切に動作するために必要なヘッダーには影響しません。

生成元のプロキシからの Forwarded ヘッダーはセキュリティー上の問題にはなりません。リモートサーバーは、接続しているプロキシホストを接続から検出できます。プロキシチェーン内では、内部のプロキシから生成される Forwarded ヘッダーを、外部のプロキシによって抑止することができます。内部プロキシやクライアントホスト名をリモートサーバーに漏らしたくない場合は、このようにサーバーを設定することをお勧めします。

▼ 送信されるヘッダーを抑止するには

1. サーバーマネージャーにアクセスし、「**Filters**」タブをクリックします。
2. 「**Suppress Outgoing Headers**」リンクをクリックします。
   「Suppress Outgoing Headers」ページが表示されます。
3. 「**Suppress Headers**」テキストボックスに、抑止する要求ヘッダーの一覧をコンマで区切って入力します。
4. 「**Restart Required**」をクリックします。
   「Apply Changes」ページが表示されます。
5. 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

MIMEタイプによるフィルタリング

MIMEタイプが一致する特定のファイルをブロックするようにProxy Serverを設定することができます。たとえば、Proxy Serverを使用するクライアントがコンピュータウィルスをダウンロードしないように、実行可能ファイルやほかのバイナリファイルをブロックするようにProxy Serverを設定できます。

Proxy Serverが新しいMIMEタイプをサポートするように設定するには、サーバーマネージャーで、「Preferences」>「Create/Edit MIME Types」の順に選択して、タイプを追加します。MIMEタイプの作成の詳細については、144ページの「MIMEタイプの作成」を参照してください。

MIMEタイプのフィルタリングとテンプレートを組み合わせて、特定のURLに対しては特定のMIMEタイプのみがブロックされるように設定できます。たとえば、`.edu`ドメインの任意のコンピュータから送られてくる実行可能ファイルをブロックすることもできます。

▼ MIMEタイプによってフィルタをかけるには

1. サーバーマネージャーにアクセスし、「**Filters**」タブをクリックします。
2. 「**SetMIME Filters**」リンクをクリックします。
   「Set MIME Filters」ページが表示されます。
3. **MIME**タイプのフィルタリングに使用するテンプレートを選択するか、サーバー全体を編集することを確認します。
4. 「**Current filter**」テキストボックスには、ブロックする**MIME**タイプと一致する正規表現を入力できます。
   たとえば、すべてのアプリケーションをフィルタで除外する場合は、正規表現に**`application/.*`**と入力します。この方法のほうが、すべてのアプリケーションタイプに対して各MIMEタイプのチェックマークを付けるよりも速く指定できます。正規表現では大文字と小文字は区別されません。正規表現の詳細については、第16章の「正規表現について」を参照してください。
5. フィルタをかける**MIME**タイプにチェックマークを付けます。
   クライアントがブロックされたファイルにアクセスを試みると、Proxy Serverから「403 Forbidden」メッセージが返されます。
6. 「了解」をクリックします。
7. 「**Restart Required**」をクリックします。
   「Apply Changes」ページが表示されます。

8 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

HTML タグによるフィルタリング

ファイルをクライアントに渡す前にフィルタで除外する HTML タグを指定することができます。この方法によって、HTML ファイルに埋め込まれた Java アプレットや JavaScript などのオブジェクトをフィルタで除外することができます。HTML タグにフィルタをかけるには、HTML の開始タグと終了タグを指定します。これで、ファイルがクライアントに送信される前に、これらのタグで囲まれたすべてのテキストとオブジェクトが、プロキシによってブランクに置き換えられます。

元の (編集前の) ファイルをキャッシュするようにプロキシを設定すれば、プロキシによってそのリソースがキャッシュ内に保存されます。

▼ HTML タグをフィルタで除外するには

1 サーバーマネージャーにアクセスし、「**Filters**」タブをクリックします。

2 「**Set HTML Tag Filters**」リンクをクリックします。
「Set HTML Tag Filters」ページが表示されます。

3 変更するテンプレートを選択します。
HTTP を選択するか、特定の URL (`.edu` ドメイン内のホストからの URL など) のみを指定したテンプレートを選択します。

4 フィルタをかけるデフォルトの **HTML** タグを選択します。

- APPLET は通常 Java アプレットを囲んでいます。
- SCRIPT は、JavaScript コードの開始を示しています。
- IMG インラインイメージファイルを指定しています。

5 フィルタをかける任意の **HTML** タグを入力できます。
HTML の開始タグと終了タグを入力します。

たとえば、フォームをフィルタで除外する場合は、「Start Tag」ボックスに **`FORM`**、「End Tag」ボックスに **`/FORM`** と入力します。HTML タグでは大文字と小文字は区別されません。`OBJECT` や `IMG` のように、フィルタをかけるタグに終了タグが存在しない場合、「End Tag」ボックスは空のままでかまいません。

6 「了解」をクリックします。

7 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

8 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

コンテンツを圧縮するためのサーバー設定

Proxy Server は HTTP コンテンツの圧縮をサポートしています。コンテンツを圧縮することで、ハードウェアに負担をかけることなくクライアントへの配信速度を向上させ、コンテンツのボリュームを増やすことができます。コンテンツ圧縮により、コンテンツのダウンロード時間が減少します。これは、ダイアルアップ接続やトラフィックの多い接続を使用するユーザーにとって非常に重要な利点です。

コンテンツを圧縮した場合、Proxy Server は圧縮されたデータを送信し、そのデータを直ちに展開 (解凍) するようにブラウザに指示を出します。これにより、送信するデータの容量が減り、ページの表示速度が速くなります。

コンテンツをオンデマンドで圧縮するためのサーバー設定

転送データを直ちに圧縮するように Proxy Server を設定できます。動的に生成される HTML ページは、ユーザーがそれを要求するまで生成されません。

▼ コンテンツをオンデマンドで圧縮するようにサーバーを設定するには

1 サーバーマネージャーにアクセスし、「**Filters**」タブをクリックします。

2 「**Compress Content on Demand**」リンクをクリックします。
「Compress Content on Demand」ページが表示されます。

3 ドロップダウンリストからリソースを選択するか、正規表現を入力します。
正規表現の詳細については、第 16 章の「正規表現について」を参照してください。

4 次の情報を指定します。

- 「**Activate Compress Content on Demand?**」: サーバーが、選択したリソースの圧縮コンテンツを配信するかどうかを選択します。

- 「**Vary Header**」: Vary: Accept-encoding ヘッダーを挿入するかどうかを指定します。「yes」または「no」を選択します。「yes」に設定すると、ファイルの圧縮バージョンが選択された場合は常に Vary: Accept-encoding ヘッダーが挿入されます。

  「no」に設定すると、Vary: Accept-encoding ヘッダーは挿入されません。

  デフォルトでは、「yes」に設定されています。
- 「**Fragment Size**」: 圧縮ライブラリ (`zlib`) が一度に圧縮する量を制御するために使用するメモリフラグメントのサイズをバイト単位で指定します。デフォルト値は 8096 です。
- 「**Compression Level**」: 圧縮のレベルを指定します。1 ～ 9 の値を選択します。値 1 では速度が最高になり、値 9 では圧縮率が最高になります。デフォルト値は、速度と圧縮率の両方を考慮した 6 です。

5 「了解」をクリックします。

6 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

7 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

◆ ◆ ◆ 第 14 章

14 逆プロキシの使用

この章では、Proxy Server を逆プロキシとして使用する方法について説明します。逆プロキシは、ファイアウォールの外側で使用され、外部のクライアントに対してはセキュリティー保護されたコンテンツサーバーに見せかけます。これは会社のサーバーのデータが外部から直接、監視されずにアクセスされることを防ぎます。また、レプリケーション用に使用することもできます。つまり、アクセス回数の多いサーバーの前に複数のプロキシを接続して、負荷を分散することができます。この章では、Proxy Server をファイアウォールの内側または外側で交互に使用できるように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。

逆プロキシのしくみ

逆プロキシには 2 つの方法を使用できます。一方の方法では Proxy Server のセキュリティー機能を利用してトランザクションを処理します。他方の方法ではキャッシング機能を利用してアクセス回数の多いサーバーの負荷を分散します。どちらの方法も、厳密にファイアウォール上で動作しないため、従来のプロキシの使用法とは異なっています。

サーバーの代理としてのプロキシ

クレジットカード番号のデータベースなど、セキュリティー保護する必要のある機密情報を保持するコンテンツサーバーがある場合、このコンテンツサーバーの代理として、プロキシをファイアウォールの外側に設定することができます。外部のクライアントがコンテンツサーバーにアクセスしようとすると、代わりに Proxy Server に送られます。実際のコンテンツは、ファイアウォールの内側でセキュリティー保

護されたコンテンツサーバー内にあります。Proxy Server はファイアウォールの外側にあり、クライアントからはコンテンツサーバーに見えます。

クライアントがサイトに対して要求を送ると、その要求は Proxy Server に送られます。次に、Proxy Server は、クライアントの要求を、ファイアウォール内の特定の経路を経由させてコンテンツサーバーに送信します。コンテンツサーバーも、この経路を経由させて結果をプロキシに返します。図 14–1 に示すように、プロキシは、実際のコンテンツサーバーのように見せかけて、取得した情報をクライアントに送信します。コンテンツサーバーがエラーメッセージを返してきた場合、Proxy Server は、このメッセージをクライアントに送信する前に、メッセージを遮断して、ヘッダーに表示された URL を変更することができます。この動作によって、外部クライアントにより内部のコンテンツサーバーに対するリダイレクト URL が入手されることを防ぐことができます。

このように、プロキシはセキュリティー保護されたデータベースを悪意のある攻撃から守るためのバリアとして機能します。万が一悪意のある攻撃が成功してしまった場合でも、侵入者は 1 つのトランザクションに関連する情報にしかアクセスできないように制限される可能性が高くなり、データベース全体にアクセスすることはありません。ファイアウォールの経路には Proxy Server しかアクセスできないため、承認されていないユーザーは実際のコンテンツサーバーにアクセスすることはできません。

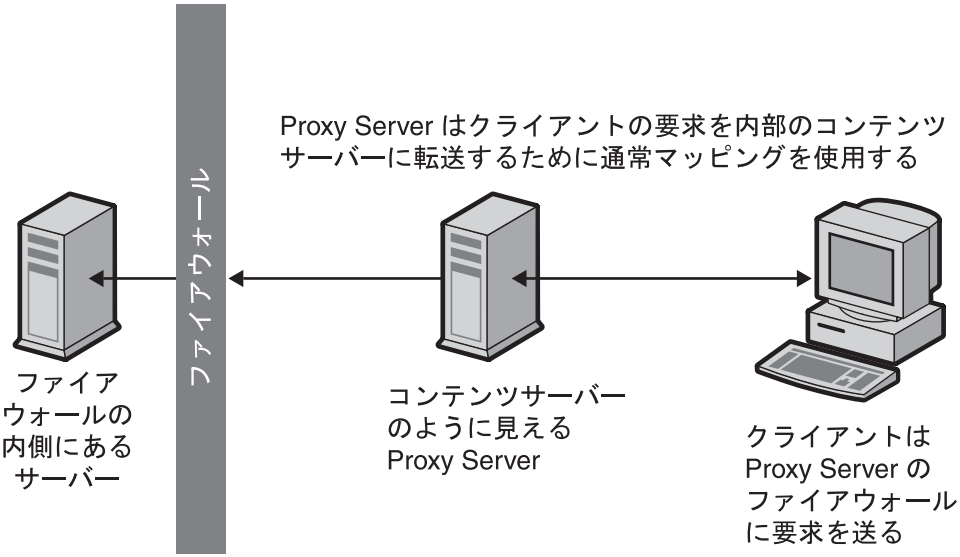


図 14–1 逆プロキシのしくみ

ファイアウォールルーターを設定することによって、特定のポート上の特定のサーバー (この場合は割り当てられたポート上のプロキシ) が、ほかのコンピュータのアクセスを許可することなく、ファイアウォールを通過してアクセスできるようになります。

セキュリティー保護された逆プロキシ

セキュリティー保護された逆プロキシが実行されるのは、Proxy Server とほかのマシンの間の 1 つ以上の接続で Secure Sockets Layer (SSL) プロトコルが使用され、データが暗号化されている場合です。

セキュリティー保護された逆プロキシには多くの使用法があります。

- ファイアウォールの外側のプロキシサーバーからファイアウォールの内側のセキュリティー保護されたコンテンツサーバーへの接続を暗号化します。
- クライアントがプロキシサーバーに安全に接続できるようにし、情報 (クレジットカード番号など) の転送のセキュリティーを保護します。

セキュリティー保護された逆プロキシを使用すると、データの暗号化に伴うオーバーヘッドによって、セキュリティー保護された各接続の速度が低下します。しかし、SSL の提供するキャッシングメカニズムによって、接続している両者は以前に更新済みのセキュリティーパラメータを再利用することができ、以降の接続のオーバーヘッドを大幅に減らすことができます。

セキュリティー保護された逆プロキシは、次の 3 つの方法で設定できます。

- セキュリティー保護されたクライアントからプロキシへ: このシナリオは、次の図に示すように、プロキシとコンテンツサーバー間でやりとりされる情報が、承認されていないユーザーからアクセスされる可能性がほとんど、またはまったくない場合に効果的です。

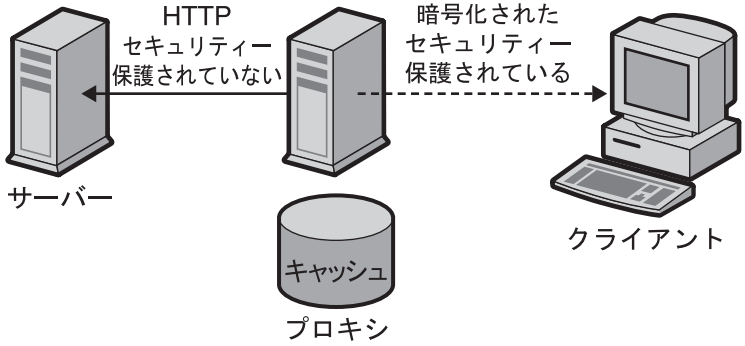


図 14–2 セキュリティー保護されたクライアントからプロキシへの接続

- セキュリティー保護されたプロキシからコンテンツサーバーへ: このシナリオは、ファイアウォールの内側にクライアントがあり、コンテンツサーバーがファイアウォールの外側にある場合に効果的です。このシナリオでは、次の図に示すように、Proxy Server はサイト間のセキュリティー保護されたチャネルとして機能します。

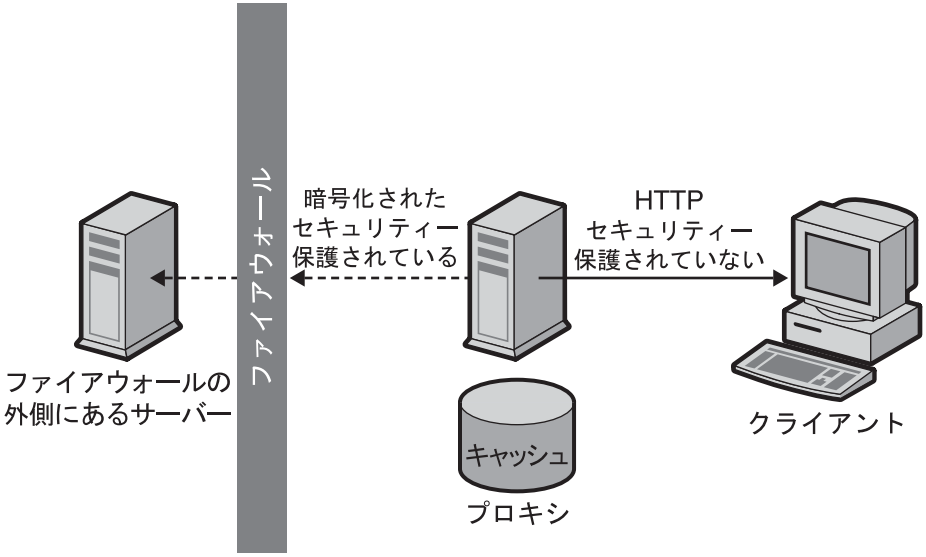


図 **14–3** セキュリティー保護されたプロキシからコンテンツサーバーへの接続

- セキュリティー保護されたクライアントからプロキシへ、およびセキュリティー保護されたプロキシからコンテンツサーバーへ: このシナリオは、サーバー、プロキシ、およびクライアント間でやりとりされる情報をセキュリティー保護する必要がある場合に効果的です。このシナリオでは、次の図に示すように、Proxy Server はサイト間のセキュリティー保護されたチャネルのように機能し、クライアント認証についてもセキュリティー保護します。

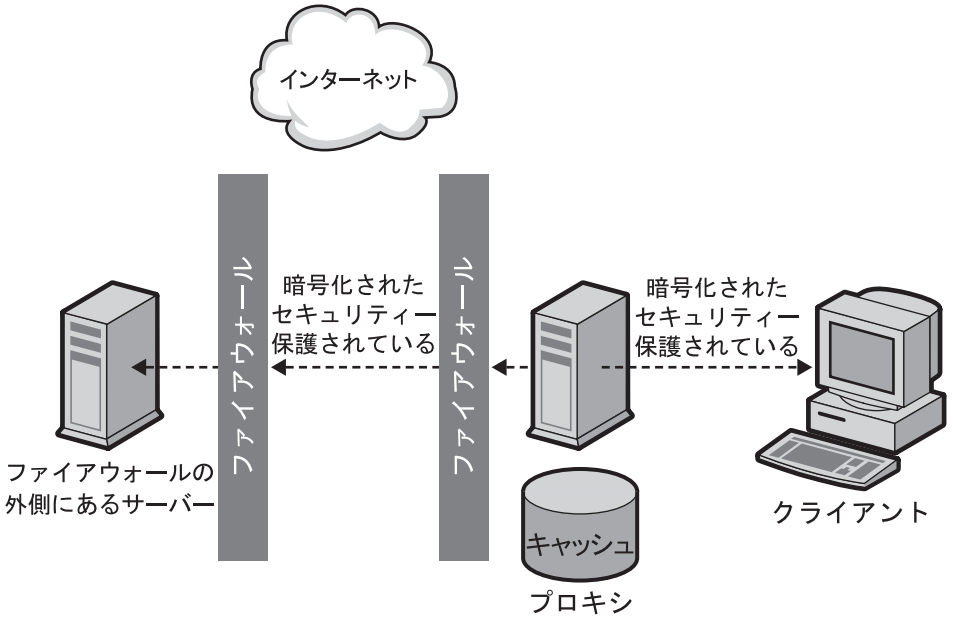


図 14–4　セキュリティー保護されたクライアントからプロキシへの接続と、セキュリティー保護されたプロキシからコンテンツサーバーへの接続

これらの設定の各設定方法については、327 ページの「逆プロキシの設定」を参照してください。

プロキシは SSL のほかにもクライアント認証を使用することができます。このためには、プロキシに対して要求を作成するコンピュータが、識別情報を検証するための証明書または他の ID の形式を提示する必要があります。

負荷分散のためのプロキシ

組織内で複数のプロキシサーバーを使用して、Web サーバー間のネットワーク負荷を分散することができます。このモデルでは、Proxy Server のキャッシュ機能を利用して、負荷分散のためのサーバープールを作成します。この場合、Proxy Server はファイアウォールのどちら側に配置してもかまいません。1 日に大量の要求を受信する Web サーバーがある場合、プロキシサーバーを使用することで、この Web サーバーの負荷を減らし、ネットワークアクセスの効率を上げることができます。

Proxy Server は、実際のサーバーに対するクライアント要求の橋渡し役として機能します。Proxy Server は、要求されたドキュメントをキャッシュします。複数のプロキシサーバーが存在する場合、DNS は IP アドレスを「ラウンドロビン」方式で選択し

て、要求をランダムにルーティングすることができます。クライアントは毎回同じ URL を使用しますが、要求は毎回異なるプロキシを経由して送信されることがあります。

アクセス回数の多い 1 台のコンテンツサーバーへの要求を、複数のプロキシを使用して処理する方法には、サーバーが 1 台の場合と比べて、より大きな負荷を、より効率的に処理できるようになるという利点があります。最初の起動時に、プロキシがコンテンツサーバーからドキュメントをはじめて取得した後は、コンテンツサーバーへの要求数が大幅に減少する可能性があります。

コンテンツサーバーに送る必要があるのは、CGI 要求とときどき発生する新規の要求だけです。それ以外の要求は、プロキシで処理できます。たとえば、サーバーへの要求の 90% は CGI 以外の要求 、つまり、キャッシュ可能な要求で、1 日あたりのコンテンツサーバーへのヒット数は 200 万回であるとします。この状況で、3 つのプロキシに接続し、各逆プロキシが 1 日あたり 200 万回のアクセスを処理すれば、1 日あたり約 600 万回のヒット数を処理することもできるようになります。コンテンツサーバーに到達する要求は 10% なので、各プロキシからの 1 日あたりのヒット数は合計 20 万回、つまり全体で 60 万回程度となり、こちらの方がはるかに効率的です。ヒット数が約 200 万回から 600万回に増加しても、コンテンツサーバー上の負荷はそれに伴って 200 万回から 60 万回に減少します。ただし、実際の結果はユーザーの状況によって異なります。

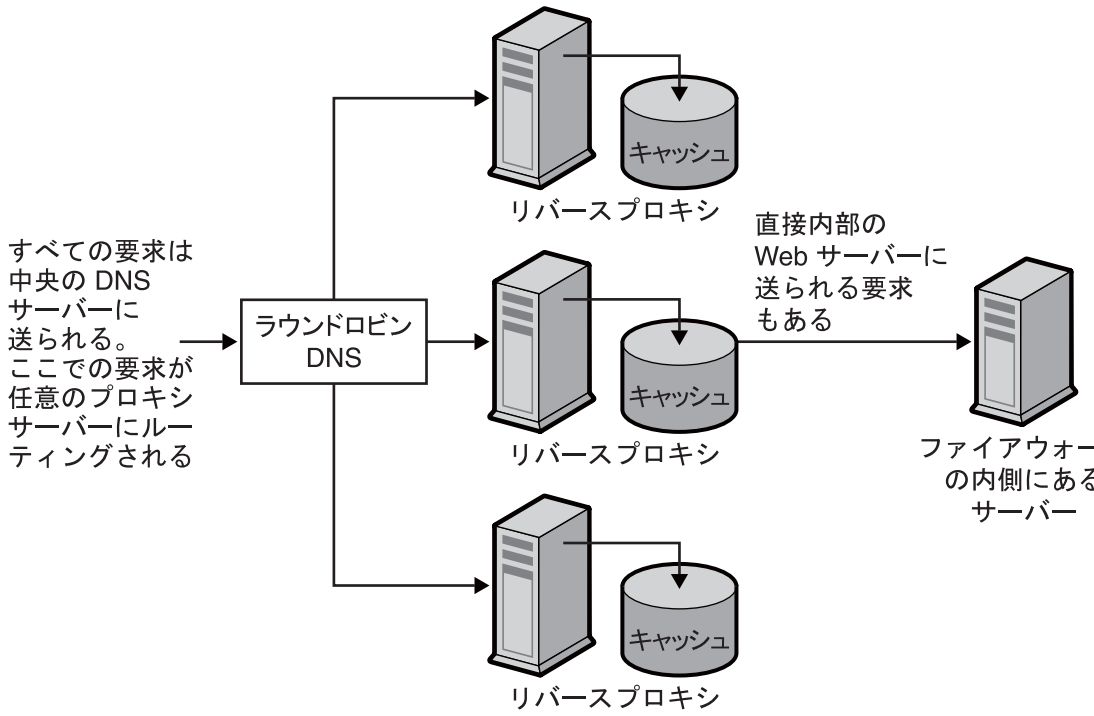


図 14–5　負荷分散のためのプロキシ

逆プロキシの設定

逆プロキシを設定するには、2 つのマッピング、通常マッピングと逆マッピングが必要です。

- 通常マッピングは、要求をコンテンツサーバーにリダイレクトします。クライアントが Proxy Server にドキュメントを要求した場合、Proxy Server は通常マッピングによって、実際のドキュメントの取得先を知る必要があります。

注意 – 自動設定ファイルを使用するプロキシがある場合は、逆プロキシを使用しないでください。プロキシが不正な結果を返すことがあるためです。

- 逆マッピングでは、Proxy Server がコンテンツサーバーからのリダイレクトにトラップをかけます。プロキシはリダイレクトを遮断し、リダイレクトされた URL を変更して Proxy Server にマップします。たとえば、クライアントが要求するドキュメントがほかの場所に移動されていたり、見つからなかったりした場合、コンテンツサーバーは、要求された URL でドキュメントが見つからないことを説明するメッセージをクライアントに返します。コンテンツサーバーは、返されたメッセージに、移動されたファイルの取得に使用する URL を示した HTTP

ヘッダーを追加します。内部のコンテンツサーバーのプライバシーを守るために、プロキシは、逆マッピングを使用して URL をリダイレクトすることができます。

ここでは、`http://http.site.com/` という Web サーバーがあり、このサーバーの逆プロキシサーバーを設定する場合を考えてみます。この場合の逆プロキシは `http://proxy.site.com/` とします。

▼ 通常マッピングと逆マッピングを作成するには

1. サーバーマネージャーにアクセスし、「**URL**」タブをクリックします。
2. 「**Create Mapping**」リンクをクリックします。
   「Create Mapping」ページが表示されます。
3. 表示されるページで、通常マッピングの「**source prefix**」と「**source destination**」を指定します。
   たとえば、次のようになります。

   「Source prefix」: `http://proxy.site.com`

   「Source destination」: `http://http.site.com/`
4. 「了解」をクリックします。
   ページに戻り、たとえば、逆マッピングを作成します。

   「逆マッピング」:

   「Source prefix」: `http://http.site.com/`

   「Source destination」: `http://proxy.site.com/`
5. 変更を行うには、「了解」をクリックします。
   「了解」ボタンをクリックすると、プロキシサーバーは 1 つ以上のマッピングを追加します。マッピングを表示するには、「View/Edit Mappings」というリンクをクリックします。追加されたマッピングは、次のような形式で表示されます。

   「from」: `/`

   「to」: `http://http.site.com/`

   これらの追加の自動マッピングは、逆プロキシを通常のサーバーとして接続してくるユーザーを対象としています。最初のマッピングは、逆プロキシを通常のプロキシとして接続してくるユーザーを対象としています。"/" マッピングは、ユーザーが管理 GUI によって自動的に指定された「Map Source Prefix」テキストボックスの内容

を変更していない場合にのみ追加されます。設定方法によりますが、通常、必要とされるのは2つ目のマッピングのみですが、余分なマッピングを設定してもプロキシに問題が発生することはありません。

注 – Web サーバーに複数の DNS エイリアスが存在する場合、各エイリアスに対応する通常マッピングが必要です。Web サーバーが複数の DNS エイリアスによって、自分自身に対するリダイレクトを複数生成する場合、これらのエイリアスには、それぞれに対応する逆マッピングが必要です。

配信元サーバー上で CGI アプリケーションがまだ実行されています。Proxy Server が単独で CGI アプリケーションを実行することはありません。ただし、CGI スクリプトによって結果をキャッシュするように指示されている (Last-modified または Expires ヘッダーによってゼロ以外の time-to-live を示唆することにより) 場合、プロキシは結果をキャッシュします。

Web サーバーのコンテンツをオーサリングする場合は、コンテンツが逆プロキシからも提供されることに注意してください。したがって、Web サーバー上のファイルに対するすべてのリンクを、相対リンクにする必要があります。HTML ファイルでは、ホスト名を参照しないでください。すべてのリンクを次のページのものにする必要があります。

```
/abc/def
```

この場合の完全修飾ホスト名は次のとおりです。

```
http://http.site.com/abc/def
```

注 – 逆プロキシモードで発生するエラーのカスタムエラーページを指定できます。これらのエラーページは、プロキシによって生成されるエラーより優先されます。これにより、クライアントから、Proxy Server が設定されていることがわからないようにすることができます。

セキュリティー保護された逆プロキシの設定

セキュリティー保護された逆プロキシを設定する前に、デジタル署名、認証局 (Certificate Authority、ACA)、および認証についてよく理解しておく必要があります。

セキュリティー保護された逆プロキシの設定は、セキュリティー保護されていない逆プロキシの設定とほぼ同じです。唯一異なる点は、暗号化するファイルのプロトコルとして HTTPS を指定する必要があるということだけです。

セキュリティー保護されたクライアントからプロキシへ

次の手順に、ユーザーが選択する設定シナリオに応じた、セキュリティー保護された逆プロキシの設定方法を説明します。マッピングの設定方法を説明するために、ここでは、`http.site.com` という Web サーバーがあり、`proxy.site.com` というセキュリティー保護された逆プロキシを設定する場合を考えてみます。手順を実際に実行する場合は、下記の例で使用されている名前を、実際の Web サーバー名とプロキシ名に置き換えてください。

▼ セキュリティー保護されたクライアントからプロキシへのマッピングを設定するには

1. サーバーマネージャーにアクセスし、「**URL**」タブをクリックします。
2. 「**Create Mapping**」リンクをクリックします。
   「Create Mapping」ページが表示されます。
3. 表示されたページで、次の方法に従って通常マッピングと逆マッピングを設定します。
   「**Regular mapping**」:
   「Source prefix」: `https://proxy.mysite.com`
   「Source destination」: `http://http.mysite.com/`
   「**Reverse mapping**」:
   「Source prefix」: `http://http.mysite.com/`
   「Source destination」: `https://proxy.mysite.com/`
4. 変更を保存して適用します。
   作成したマッピングを表示するには、「View/Edit Mappings」リンクをクリックします。

注 – この設定は、プロキシサーバーがセキュリティーモードで実行されている場合にのみ機能します。つまり、暗号化を有効にして、プロキシをコマンド行から再起動する必要があります。プロキシをコマンド行から再起動するには、プロキシディレクトリに移動して **`./start`** と入力します。

▼ セキュリティー保護されたプロキシからコンテンツサーバーへのマッピングを設定するには

1 サーバーマネージャーにアクセスし、「**URL**」タブをクリックします。

2 「**Create Mapping**」リンクをクリックします。
「Create Mapping」ページが表示されます。

3 表示されたページで、次の方法に従って通常マッピングと逆マッピングを設定します。

「**Regular mapping**」:

「Source prefix」:`http://proxy.mysite.com`

「Source destination」:`https://http.mysite.com/`

「**Reverse mapping**」:

「Source prefix」:`https://http.mysite.com/`

「Source destination」:`http://proxy.mysite.com/`

4 変更を保存して適用します。
作成したマッピングを表示するには、「View/Edit Mappings」というリンクをクリックします。

注 – この設定は、コンテンツサーバーがセキュリティーモードで実行されている場合にのみ機能します。

▼ セキュリティー保護されたクライアントからプロキシへ、およびセキュリティー保護されたプロキシからコンテンツサーバーへを設定するには

1 サーバーマネージャーにアクセスし、「**URL**」タブをクリックします。

2 「**Create Mapping**」リンクをクリックします。
「Create Mapping」ページが表示されます。

3 表示されたページで、次の方法に従って通常マッピングと逆マッピングを設定します。

「**Regular mapping**」:

「Source prefix」:`https://proxy.mysite.com`

「Source destination」: `https://http.mysite.com/`

「**Reverse mapping**」:

「Source prefix」: `https://http.mysite.com/`

「Source destination」: `https://proxy.mysite.com/`

4 変更を保存して適用します。

作成したマッピングを表示するには、「View/Edit Mappings」というリンクをクリックします。

注 – この設定は、プロキシサーバーとコンテンツサーバーがセキュリティーモードで実行されている場合にのみ機能します。つまり、プロキシに対して暗号化を有効にし、プロキシをコマンド行から再起動する必要があります。プロキシをコマンド行から再起動するには、プロキシディレクトリに移動して **`./restart`** と入力します。

逆プロキシでの仮想マルチホスティング

仮想マルチホスティングは、配信元サーバー (逆プロキシサーバーなど) が、複数の DNS エイリアスに対して、それぞれのアドレスに別々のサーバーがインストールされているかのように応答できるようにする機能です。例として、次のような DNS ホスト名があるとします。

- `www`
- `specs`
- `phones`

これらのホスト名は、同じ IP アドレス (逆プロキシの IP アドレス) にマッピングされます。次に、アクセスに使用された DNS 名に基づいて、逆プロキシに異なる動作をさせることができます。

仮想マルチホスティングを使用すると、複数の異なるドメインも 1 台の逆プロキシサーバーでホストできるようになります。次に例を示します。

- `www.domain-1.com`
- `www.domain-2.com`
- `www.domain-3.com`

複数のローカルホスト名と複数のドメインの組み合わせを、すべて 1 台のプロキシサーバーに保持できます。

- `www`
- `specs`
- `phones`
- `www.domain-1.com`

- www.domain-2.com
- www.domain-3.com

仮想マルチホスティング機能の詳細

仮想マルチホスティング機能は、DNS ホスト名およびドメイン名またはエイリアスを指定して、ターゲット URL のプレフィックスに、そのホスト名に送信された要求が送られる場所を指定することによって機能します。例として、次の 2 つのマッピングがあるとします。

- engr.domain.com -> http://int-engr.domain.com
- mktg.domain.com -> http://int-mktg.domain.com

マッピングはルート対ルートで行う必要はありません。ターゲット URL 内に URL パスのプレフィックスを追加することもできます。

- engr.domain.com -> http://internal.domain.com/engr
- mktg.domain.com -> http://internal.domain.com/mktg

仮想ドメインのマッピングにも同じことが当てはまります。たとえば、次のようなマッピングを使用できます。

- www.domain-1.com -> http://int-engr.domain.com
- www.domain-2.com -> http://int-mktg.domain.com

システムは、HTTP の「Host:ヘッダーを見にいきます。そのヘッダーに基づいて、一致する仮想マルチホスティングのマッピングが選択されます。一致するマルチホスティングのマッピングがない場合、サーバーは、続けて設定ファイルに表示される順にほかのマッピングを見にいきます。一致するものが見つからなければ、マッピングは実行されません。一致するものがない場合、通常、「プロキシは要求の実行を拒否しました」という応答がプロキシから返されます。

▼ 仮想マルチホスティングを設定するには

1. サーバーマネージャーにアクセスし、「**URL**」タブをクリックします。
2. 「**Configure Virtual Multihosting**」リンクをクリックします。
   「Configure Virtual Multihosting」ページが表示されます。
3. 「**Source Hostname (alias)**」フィールドで、このマッピングを適用するローカルホスト名 (または **DNS** エイリアス) を入力します。
4. 「**Source Domain Name**」フィールドで、このマッピングを適用するローカルドメイン名を入力します。
   通常、複数の異なる DNS ドメインをマルチホストする場合を除き、この名前はユーザー自身のネットワークのドメイン名になります。

5 「**Destination URL Prefix**」フィールドに、ホスト名とドメイン名が上の指定と一致する場合に要求が送信されるターゲット **URL** のプレフィックスを入力します。

6 **I**テンプレートを使用している場合は、「**Use This Template**」ドロップダウンリストからテンプレート名を選択します。テンプレートを適用しない場合は「**NONE**」のままにします。

7 「了解」をクリックします。

8 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

9 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。
作成するそれぞれの仮想マルチホスティングのマッピングについて、上記の手順を繰り返します。

仮想マルチホスティングのマッピングはすべて、「Configure Virtual Multihosting」ページの下部に表示されます。「Source Hostname (alias)」と「Source Domain Name」フィールドは、プロキシのポート番号とともに 1 つの正規表現にマージされ、「Host:」ヘッダーとのマッチングに使用されることに注意してください。

たとえば、ホスト名が `www`、ドメインが `example.com`、ポート番号が 8080 の場合、次のような正規表現が表示されます。

```
www(|.example.com)(|:8080)
```

これは、ユーザーが入力するか、クライアントが送信する、次のようなすべての組み合わせと必ず一致します。ポート番号は、80 以外の場合でも、サーバーがそのポートで待機していたため、クライアントソフトウェアによって省略されることがあります。

- `www`
- `www:8080`
- `www.example.com`
- `www.example.com:8080`

仮想マルチホスティングに関する注意事項

- 逆プロキシのマッピングを設定する前には、クライアント自動設定機能を無効にする必要があります。クライアント自動設定機能は転送プロキシを対象としており、逆プロキシは対象としていません。
- 仮想マルチホスティング機能により、自動逆マッピングが設定されます。「Virtual Multihosting」ページを使用して指定したマッピングに対して、逆マッピングを作成しないでください。
- 仮想マッピングは、`obj.conf` 内で `virt-map` 関数を使用して指定します。

- 仮想マッピングは、 obj.conf 設定ファイル内で指定された順にマッチングされます。仮想マッピングを行う前に、通常マッピング、逆マッピング、正規表現によるマッピング、またはクライアント自動設定によるマッピングが存在する場合、これらが最初に適用されます。同様に、仮想マッピング内に一致するものが存在しない場合、obj.conf 内の仮想マッピングセクションの次にあるマッピングに進んで変換が行われます。

  注 – 仕様の順序では、逆マッピングを他のマッピングの前に指定する必要があります。

- Proxy Server のポート番号が変更された場合、新しいポート番号を反映するため、仮想マルチホスティングのマッピングを再作成する必要があります。

◆ ◆ ◆ 第 15 章

SOCKS の使用

この章では、Sun Java System Web Proxy Server に組み込まれている SOCKS サーバーの設定方法と使用方法について説明します。Proxy Server は SOCKS バージョン 4 および 5 をサポートしています。

この章の内容は次のとおりです。

SOCKS について

SOCKS は、反対側の SOCKS サーバー上のホストからの接続要求をリダイレクトするネットワークプロキシのプロトコルで、これによって直接 IP に接続しなくても、一方のホストが他方のホストに完全アクセスができるようになります。SOCKS は一般的にネットワークファイアウォールとして使用され、SOCKS サーバーの背後にあるホストがインターネットに完全アクセスできるようにする一方で、インターネットから内部ホストへの不正アクセスを防ぎます。

SOCKS サーバーは、ポイントツーポイント単位で、ファイアウォールを経由してアクセスを制御する汎用ファイアウォールデーモンです。SOCKS サーバーは、要求を認証および承認し、プロキシ接続を確立し、データを中継します。SOCKS サーバーは、アプリケーションレベルではなくネットワークレベルで動作するため、要求の

転送に使用されるプロトコルやメソッドを認識しません。SOCKS サーバーはプロトコルを認識しないため、Telnet などの Proxy Server がサポートしていないプロトコルの経路として使用することができます。

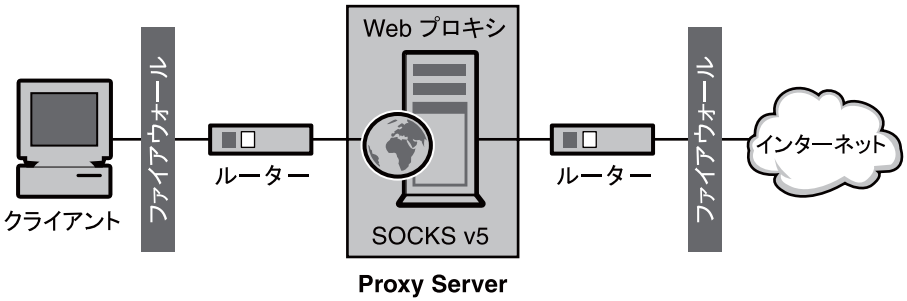


図 15–1　ネットワークにおける SOCKS サーバーの位置

バンドルされた SOCKS v5 サーバーの使用

Sun Java System Web Proxy Server には、ほかの SOCKS デーモンが使用する標準の `socks5.conf` ファイル形式を認識する独自の SOCKS デーモンが組み込まれています。このデーモンは、Proxy Server で使用して要求をルーティングしたり、Proxy Server から実行してネットワークに追加機能を提供したりすることができます。SOCKS サーバーを経由して要求をルーティングするように Proxy Server を設定する方法については、349 ページの「ルーティングエントリの設定」を参照してください。

Proxy Server に含まれる SOCKS デーモンは、デフォルトで無効になっています。デーモンは、サーバーマネージャーインタフェースの「SOCKS」タブまたはコマンド行から有効にできます。詳細は、340 ページの「SOCKS v5 サーバーの起動と停止」を参照してください。

注 – Proxy Server 4 で、SOCKS デーモンの名前が `ns-sockd` から `sockd` に変更されました。

Proxy Server に組み込まれている SOCKS サーバーを使用するための手順は次のとおりです。

▼ SOCKS を使用するには

1. SOCKS サーバーを設定します。**341 ページの「SOCKS v5 サーバーの設定」**を参照してください。

2. **SOCKS** サーバーを複数のインタフェースを持つコンピュータで実行する場合は、**SOCKS** ルーティングエントリを作成します。**349** ページの「ルーティングエントリの設定」を参照してください。
3. 認証エントリを作成します。**343** ページの「**SOCKS v5** の認証エントリの設定」を参照してください。
4. 接続エントリを作成します。**345** ページの「**SOCKS v5** の接続エントリの設定」を参照してください。
5. **SOCKS** サーバーを有効にします。**340** ページの「**SOCKS v5** サーバーの起動と停止」を参照してください。

socks5.conf について

Sun Java System Web Proxy Server は、`socks5.conf` ファイルを使用して SOCKS サーバーとそのサービスへのアクセスを制御します。各エントリでは、そのエントリと一致した要求を受信したときの Proxy Server の動作が定義されています。サーバーマネージャーでの選択内容は `socks5.conf` に書き込まれます。ファイルは手動で編集することもできます。`socks5.conf` ファイルは、次に示すインストールルートディレクトリ *server-root* に置かれます。

server-root`/proxy-`*serverid*`/config` ディレクトリ

この節では、`socks5.conf` に関する一般情報について説明します。このファイルやこのファイルで使用する指令と構文については、『Proxy Server *Configuration File Reference*』 を参照してください。

認証

SOCKS デーモンは、そのサービスを使用するための認証を要求するように設定できます。認証は、ホスト名と接続クライアントのポートに基づいて行われます。ユーザー名とパスワードを入力するように選択した場合は、 `socks5.conf` ファイルによって参照されるユーザー名とパスワードのファイルに対して入力情報が認証されます。入力されたユーザー名とパスワードが、パスワードファイルのリストと一致しない場合、アクセスが拒否されます。パスワードファイル内のユーザー名とパスワードの書式は *username password* で、ユーザー名とパスワードはスペースによって区切られます。

ユーザーを拒否することもできます。ユーザー名とパスワードによる認証を要求するようにするには、`SOCKS5_PWDFILE` 指令を `socks5.conf` に追加する必要があります。この指令と構文の詳細については、『Proxy Server *Configuration File Reference*』にある `socks5.conf` の節を参照してください。

ユーザー名とパスワードによる認証は、設定済みの LDAP サーバーに対して実行されることもあり、ファイルに対してのみ実行されるものではありません。

アクセス制御

アクセス制御は、`socks5.conf` ファイル内で順番に並べられた一連の行を使用して実行されます。各行に、リソースへのアクセスを許可または拒否する 1 つの指令が含まれています。これらの指令は、この設定ファイルに表示される順番で処理されます。許可の指令のいずれにも一致しない要求は、アクセスを拒否されます。

ロギング

SOCKS デーモンは、SOCKS ログファイルにエラーメッセージとアクセスメッセージの両方を記録します。ログファイルの場所とロギングのタイプは、 `socks5.conf` 内で指定されます。

SOCKS デーモンは、1 時間ごとに stat エントリを生成し、デーモンの統計情報となります。

チューニング

`socks5.conf`ファイルを使用して、SOCKS サーバーによって使用されるワークスレッドと受け入れスレッドの数を指定することができます。これらの数は、SOCKS サーバーのパフォーマンスに影響します。

ワークスレッドおよび受け入れスレッドの設定と、これらがパフォーマンスに及ぼす影響については、341 ページの「SOCKS v5 サーバーの設定」の関連する節を参照してください。

SOCKS v5 サーバーの起動と停止

SOCKS サーバーの起動と停止は、サーバーマネージャーから、またはコマンド行から実行できます。

▼ サーバーマネージャーから SOCKS サーバーの起動または停止を実行するには

1. サーバーインスタンスのサーバーマネージャーにアクセスし、「**SOCKS**」タブをクリックします。
2. 「**Start/Stop SOCKS Server**」リンクをクリックします。
3. **SOCKS** サーバーを起動または停止します。

SOCKS サーバーの起動と停止をコマンド行から実行するには

server-root/proxy- *serverid* ディレクトリにある次のスクリプトを実行します。ここで *server-root* はインストールルートです。

- `start-sockd` は、SOCKS デーモンを起動します。
- `stop-sockd` は、SOCKS デーモンを停止させます。
- `restart-sockd` は、SOCKS デーモンを再起動します。

SOCKS v5 サーバーの設定

▼ SOCKS サーバーを設定するには

1. サーバーインスタンスのサーバーマネージャーにアクセスし、「**SOCKS**」タブをクリックします。
2. 「**Configure SOCKS v5**」リンクをクリックします。
3. 「**SOCKS Port**」フィールドに、**SOCKS** サーバーが待機するポート番号を入力します。デフォルトでは **1080** です。
4. 使用する **SOCKS** のオプションを選択します。
   有効なオプションは次のとおりです。
   - 「*Disable Reverse DNS Lookup*」:SOCKS サーバーの DNS 逆検索を無効にします。逆引き DNS は、IP アドレスをホスト名に変換します。DNS 逆検索を無効にすると、ネットワークリソースを節約できます。DNS 検索はデフォルトで無効になっています。DNS 逆検索が無効になっており、ホスト名で URL が要求された場合、サーバーはホスト名を IP アドレスにはマップしません。IDNS 逆検索が有効になっている場合、サーバーはマッピングを実行し、SOCKS ログファイルにエントリが追加され、DNS 変換が一覧されます。
     - 「**Use Client-specific Bind Port**」**:**BIND 要求でクライアントがポートを指定できるようにします。このオプションを無効にすると、SOCKS はクライアントの要求したポートを無視して、ランダムにポートを割り当てます。このオプションは、デフォルトで無効になっています。
     - 「**Allow** *Wildcard As Bind IP Address*」*:*クライアントが BIND 要求で IP アドレスをすべてゼロ (0.0.0.0) に指定できるようにします。これで任意の IP アドレスに接続できるようになります。このオプションを無効にすると、クライアントはバインドポートに接続する IP アドレスを指定しなくてはならなくなり、

SOCKS サーバーは、0.0.0.0. に対するバインド要求を拒否します。このオプションは、デフォルトで無効になっています。

- 「*Quench Updates*」1 時間ごとの stat ファイルへの自動書き込みを無効にします。これを無効にすると、要求のたびに書き込みが行われます。詳細は、340 ページの「ロギング」を参照してください。

「Quench Updates」要素はユーザーインタフェースには表示されますが、今回の Proxy Server 4 のリリースでは実装されていません。

5 「ログファイル」フィールドに、**SOCKS** ログファイルのフルパス名を入力します。

デフォルトは *server-root*/proxy- *serverid*/logs/socks5.log です。

6 「**Log Level**」ドロップダウンリストから、ログファイルに記録するのは警告とエラーのみか、すべての要求か、またはデバッグメッセージかを選択します。

7 **RFC 1413 ident** 応答を選択します。

ident によって、SOCKS サーバーがクライアントのユーザー名を判別できるようになります。一般に、この機能は、クライアントが UNIX の一部のバージョンを実行している場合にのみ有効になります。次のオプションが利用できます。

- 「**Don't Ask**」**:**ident を使用してクライアントのユーザー名を判別しません。これは推奨される、デフォルトの設定です。
  - 「**Ask But Don't Require**」**:**すべてのクライアントのユーザー名を尋ねますが、要求はしません。このオプションでは、ident をログ目的のみに使用します。
  - 「**Require**」**:**すべてのクライアントのユーザー名を尋ね、有効な応答のあったクライアントに対するアクセスのみを許可します。

8 「**SOCKS Tuning**」セクションで、**SOCKS** サーバーが使用するワークスレッドと受け入れスレッドの数を指定します。これらの数は、**SOCKS** サーバーのパフォーマンスに影響します。「了解」をクリックします。

- 「**Number Of Worker Threads**」**:**デフォルトは 40 です。SOCKS サーバーの速度が遅すぎる場合は、ワークスレッドの数を増やします。サーバーが不安定な場合は、この数を減らします。この数を変更する場合は、デフォルト設定から始めて、必要に応じて数を増減させてください。一般的なワークスレッド数は 10 ～ 150 です。絶対最大値は 512 ですが、150 を超えると無駄が多く、動作も不安定になります。
  - 「**Number Of Posted Accepts**」**:**デフォルトは 1 です。SOCKS サーバーが接続をドロップしている場合は、受け入れスレッドの数を増やします。サーバーが不安定な場合は、サーバーの数を減らします。この数を変更する場合は、デフォルト設定から始めて、必要に応じて数を増減させてください。一般的な受け入れスレッド数は 1 ～ 10 です。絶対最大値は 512 ですが、60 を超えると無

駄が多く、動作も不安定になります。SOCKS サーバーに負荷がかかっていて接続がドロップされているために、要求が失敗している場合は、この設定を調整します。

SOCKS v5 の認証エントリの設定

SOCKS の認証エントリは、SOCKS デーモンが接続を受け入れるホストと、SOCKS デーモンがこれらのホストの認証に使用する認証タイプを識別します。

▼ SOCKS の認証エントリを作成するには

1. サーバーインスタンスのサーバーマネージャーにアクセスし、「**SOCKS**」タブをクリックします。
2. 「**Set SOCKS v5 Authentication**」リンクをクリックします。
3. 「追加」ボタンをクリックします。
4. 「**Host Mask**」フィールドに、**SOCKS** サーバーが認証するホストの **IP** アドレスまたはホスト名を入力します。

   IP アドレスを入力する場合、アドレスの後にスラッシュを入力し、その後に受信する IP アドレスに適用されるマスクを入力します。SOCKS サーバーはこのマスクを IP アドレスに適用し、有効なホストかどうかを判別します。ホストマスクのエントリには空白文字を使用しないでください。ホストマスクを入力しない場合、認証エントリはすべてのホストに適用されます。

   たとえば、「host mask」フィールドには 155.25.0.0/255.255.0.0 と入力できます。ホストの IP アドレスが 155.25.3.5 の場合、SOCKS サーバーは IP アドレスにマスクを適用して、ホストの IP アドレスが認証レコードの適用される IP アドレス (155.25.0.0) と一致するかを判別します。
5. 「**Port Range**」フィールドに、**SOCKS** サーバーが認証するホストコンピュータ上のポートを入力します。

   ポート範囲のエントリには空白文字を使用しないでください。ポート範囲を入力しない場合、認証エントリはすべてのポートに適用されます。

   角括弧 [ ] を使用すると範囲の開始と終了のポートを含む設定となり、丸括弧 ( ) を使用すると範囲の開始と終了のポートは除外されます。たとえば、[1000-1010] は、1000 ～ 1010 までの間のすべてのポート番号を意味し、ここでは 1000 と 1010 が含まれます。一方、(1000-1010) は 1000 ～ 1010 までの間のすべてのポート番号を意味しますが、ここでは 1000 と 1010 は含まれません。角括弧と丸括弧は合わせて使用することもできます。たとえば、(1000-1010] は、1000 ～ 1010 までの間のすべてのポート番号を意味しますが、1000 は含まれず、1010 は含まれます。

6 「**Authentication Type**」ドロップダウンリストから、認証タイプを選択します。
有効なオプションは次のとおりです。

- 「**Require user-password**」**:**SOCKS サーバーにアクセスするためには、ユーザー名とパスワードが要求されます。
- 「**User-password, if available**」**:**ユーザー名とパスワードが利用できる場合は、SOCKS サーバーにアクセスするためにこれらを使用します。ただし、アクセスの必要条件ではありません。
- 「**Ban**」**:**SOCKS サーバーから拒否されます。
- 「**None**」**:**SOCKS サーバーにアクセスするために、認証は要求されません。

7 「**Insert**」ドロップダウンリストから、このエントリを挿入する `socks5.conf` ファイル内の位置を選択し、「了解」をクリックします。
複数の認証方法が存在する可能性もあるので、これらが評価される順序を指定する必要があります。したがって、リスト内の最初の認証方法をクライアントがサポートしていない場合は、2 番目の方法が代わりに使用されます。リスト内のどの認証方法もクライアントがサポートしていない場合、SOCKS サーバーは要求を受け入れずに接続を切ります。

▼ 認証エントリを編集するには

1 サーバーインスタンスのサーバーマネージャーにアクセスし、「**SOCKS**」タブをクリックします。

2 「**Set SOCKS v5 Authentication**」リンクをクリックします。

3 編集する認証エントリを選択し、「**Edit**」ボタンをクリックします。

4 必要な変更を行います。

5 「了解」をクリックします。

▼ 認証エントリを削除するには

1 サーバーインスタンスのサーバーマネージャーにアクセスし、「**SOCKS**」タブをクリックします。

2 「**Set SOCKS v5 Authentication**」リンクをクリックします。

3 削除する認証エントリを選択します。

4 「削除」をクリックします。

▼ 認証エントリを移動するには

エントリは、socks5.conf ファイルに表示される順番で評価されます。この順番は、エントリを移動させることで変更できます。

1 サーバーインスタンスのサーバーマネージャーにアクセスし、「**SOCKS**」タブをクリックします。

2 「**Set SOCKS v5 Authentication**」リンクをクリックします。

3 移動する認証エントリを選択し、「**Move**」ボタンをクリックします。

4 「**Move**」ドロップダウンリストから、このエントリを挿入する socks5.conf ファイル内の位置を選択します。

5 「了解」をクリックします。

SOCKS v5 の接続エントリの設定

SOCKS の接続エントリは、SOCKS デーモンが要求を許可するのか、拒否するのかを指定します。

▼ 接続エントリを作成するには

1 サーバーインスタンスのサーバーマネージャーにアクセスし、「**SOCKS**」タブをクリックします。

2 「**Set SOCKS v5 Connections**」リンクをクリックします。

3 「追加」ボタンをクリックします。

4 「**Authentication Type**」ドロップダウンリストから、このアクセス制御行が適用される認証方法を選択します。

5 「**Connection Type**」ドロップダウンリストから、この行と一致するコマンドタイプを選択します。次のコマンドタイプを選択できます。

- **Connect**
- **Bind**

- **UDP**
- **All**

6 「**Source Host Mask**」フィールドに、接続制御エントリが適用されるホストの **IP** アドレスまたはホスト名を入力します。

IP アドレスを入力する場合、アドレスの後にスラッシュを入力し、その後に接続元の IP アドレスに適用されるマスクを入力します。SOCKS サーバーはこのマスクを接続元の IP アドレスに適用し、有効なホストかどうかを判別します。ホストマスクのエントリには空白文字を使用しないでください。ホストマスクを入力しない場合、接続エントリはすべてのホストに適用されます。

たとえば、「host mask」フィールドには 155.25.0.0/255.255.0.0 と入力できます。ホストの IP アドレスが 155.25.3.5 の場合、SOCKS サーバーは IP アドレスにマスクを適用して、ホストの IP アドレスが接続制御エントリの適用される IP アドレス (155.25.0.0) と一致するかを判別します。

7 「**Port Range**」フィールドに、接続制御エントリが適用される接続元コンピュータ上のポートを入力します。

ポート範囲のエントリには空白文字を使用しないでください。ポート範囲を指定しない場合、接続エントリはすべてのポートに適用されます。

角括弧 [ ] を使用すると範囲の開始と終了のポートを含む設定となり、丸括弧 ( ) を使用すると範囲の開始と終了のポートは除外されます。たとえば、[1000-1010] は、1000 ～ 1010 までの間のすべてのポート番号を意味し、ここでは 1000 と 1010 が含まれます。一方、(1000-1010) は 1000 ～ 1010 までの間のすべてのポート番号を意味しますが、ここでは 1000 と 1010 は含まれません。角括弧と丸括弧は合わせて使用することもできます。たとえば、(1000-1010] は、1000 ～ 1010 までの間のすべてのポート番号を意味しますが、1000 は含まれず、1010 は含まれます。

8 「**Destination Host Mask**」フィールドに、接続エントリが適用される **IP** アドレスまたはホスト名を入力します。

IP アドレスを入力する場合、アドレスの後にスラッシュを入力し、その後に受信する IP アドレスに適用されるマスクを入力します。SOCKS サーバーはこのマスクを接続先コンピュータの IP アドレスに適用し、有効な接続先ホストかどうかを判別します。ホストマスクのエントリには空白文字を使用しないでください。接続先のホストマスクを入力しない場合、接続エントリはすべてのホストに適用されます。

たとえば、「destination host mask」フィールドには 155.25.0.0/255.255.0.0 と入力できます。接続先ホストの IP アドレスが 155.25.3.5 の場合、SOCKS サーバーは IP アドレスにマスクを適用して、接続先ホストの IP アドレスがプロキシエントリの適用される IP アドレス (155.25.0.0) と一致するかを判別します。

9 「**Port Range**」フィールドに、接続制御エントリが適用される接続先ホストコンピュータ上のポートを入力します。

ポート範囲のエントリには空白文字を使用しないでください。ポート範囲を入力しない場合、接続エントリはすべてのポートに適用されます。

注 – 大部分の SOCKS アプリケーションは、バインド要求のためにポート 0 を要求します。これは、ポートの詳細が設定されていないことを意味します。したがって、バインドの接続先のポート範囲には、常に 0 を含めることをお勧めします。

角括弧 [ ] を使用すると範囲の開始と終了のポートを含む設定となり、丸括弧 ( ) を使用すると範囲の開始と終了のポートは除外されます。たとえば、[1000-1010] は、1000 ～ 1010 までの間のすべてのポート番号を意味し、ここでは 1000 と 1010 が含まれます。一方、(1000-1010) は 1000 ～ 1010 までの間のすべてのポート番号を意味しますが、ここでは 1000 と 1010 は含まれません。角括弧と丸括弧は合わせて使用することもできます。たとえば、(1000-1010] は、1000 ～ 1010 までの間のすべてのポート番号を意味しますが、1000 は含まれず、1010 は含まれます。

10 「**User Group**」フィールドに、アクセスを許可または拒否するグループを入力します。

グループを指定しない場合、接続エントリはすべてのユーザーに適用されます。

11 「**Action**」ドロップダウンリストから、作成する接続のアクセスを許可するか、拒否するかを選択します。

12 「**Insert**」ドロップダウンリストから、このエントリを挿入する `socks5.conf` ファイル内の位置を選択し、「了解」をクリックします。

複数の接続指令が存在する可能性もあるので、これらが評価される順序を指定する必要があります。

▼ 接続エントリを編集するには

1 サーバーインスタンスのサーバーマネージャーにアクセスし、「**SOCKS**」タブをクリックします。

2 「**Set SOCKS v5 Connections**」リンクをクリックします。

3 編集する接続エントリを選択し、「**Edit**」ボタンをクリックします。

4 必要な変更を行います。

5 「了解」をクリックします。

▼ 接続エントリを削除するには

1. サーバーインスタンスのサーバーマネージャーにアクセスし、「**SOCKS**」タブをクリックします。
2. 「**Set SOCKS v5 Connections**」リンクをクリックします。
3. 削除する接続エントリを選択します。
4. 「削除」をクリックします。

▼ 接続エントリを移動するには

エントリは、`socks5.conf` ファイルに表示される順番で評価されます。この順番は、エントリを移動させることで変更できます。

1. サーバーインスタンスのサーバーマネージャーにアクセスし、「**SOCKS**」タブをクリックします。
2. 「**Set SOCKS v5 Connections**」リンクをクリックします。
3. 移動する接続エントリを選択します。
4. 「移動」ボタンをクリックします。
5. 「**Insert**」ドロップダウンリストから、このエントリを挿入する `socks5.conf` ファイル内の位置を選択し、「了解」をクリックします。

SOCKS v5 サーバーの連鎖の設定

複数の SOCKS サーバーは、Proxy Server と同じ方法で連鎖させることができます。つまり、別の SOCKS サーバーを経由して SOCKS サーバーをルーティングできます。

▼ SOCKS サーバーの連鎖を設定するには

1. サーバーインスタンスのサーバーマネージャーにアクセスし、「**SOCKS**」タブをクリックします。
2. 「**Set SOCKS v5 Routing**」リンクをクリックします。

3 「**Server Chaining**」セクションで、プロキシの連鎖内の下流にあるプロキシが要 求を処理するために認証を必要とする場合、連鎖させる **Proxy Server** に対する認 証のためのユーザー名とパスワードを入力します。「了解」をクリックします。

ルーティングエントリの設定

ルーティングエントリを使用して、SOCKS サーバーを経由して要求をルーティングするように Proxy Server を設定できます。ルーティングエントリには、SOCKS v5 ルートと SOCKS v5 プロキシルートの 2 つのタイプがあります。

- SOCKS v5 ルートは、特定の IP アドレスに対して SOCKS デーモンが使用するインタフェースを識別します。
- SOCKS v5 プロキシルートは、別の SOCKS サーバーを経由してアクセスできる IP アドレスを識別し、その SOCKS サーバーがホストに直接接続しているかどうかを見分けます。SOCKS サーバーを経由してルーティングする場合、プロキシルートは重要です。

▼ ルーティングエントリを作成するには

1 サーバーインスタンスのサーバーマネージャーにアクセスし、「**SOCKS**」タブをクリックします。

2 「**Set SOCKS v5 Routing**」リンクをクリックします。

3 「**Routing**」セクションで、「**Add**」ボタンをクリックします。

4 「**Host Mask**」フィールドに、受信および送信の接続が指定のインタフェースを経由する必要のある **IP** アドレスまたはホスト名を入力します。

IP アドレスを入力する場合、アドレスの後にスラッシュを入力し、その後に受信する IP アドレスに適用されるマスクを入力します。SOCKS サーバーはこのマスクを IP アドレスに適用し、有効なホストかどうかを判別します。ホストマスクのエントリには空白文字を使用しないでください。ホストマスクを入力しない場合、SOCKS v5 エントリはすべてのホストに適用されます。

たとえば、「host mask」フィールドには 155.25.0.0/255.255.0.0 と入力できます。ホストの IP アドレスが 155.25.3.5 の場合、SOCKS サーバーは IP アドレスにマスクを適用して、ホストの IP アドレスがルーティングエントリの適用される IP アドレス (155.25.0.0) と一致するかを判別します。

5 「**Port Range**」フィールドに、受信および送信の接続が指定のインタフェースを経由する必要のあるポートを入力します。ポート範囲には空白文字を使用しないでください。

ポート範囲を指定しない場合、SOCKS v5 エントリはすべてのポートに適用されます。

角括弧 [ ] を使用すると範囲の開始と終了のポートを含む設定となり、丸括弧 ( ) を使用すると範囲の開始と終了のポートは除外されます。たとえば、[1000-1010] は、1000 ～ 1010 までの間のすべてのポート番号を意味し、ここでは 1000 と 1010 が含まれます。一方、(1000-1010) は 1000 ～ 1010 までの間のすべてのポート番号を意味しますが、ここでは 1000 と 1010 は含まれません。角括弧と丸括弧は合わせて使用することもできます。たとえば、(1000-1010] は、1000 ～ 1010 までの間のすべてのポート番号を意味しますが、1000 は含まれず、1010 は含まれます。

6 「**Interface/Address**」フィールドに、受信および送信の接続が通過する必要のある **IP** アドレスまたはインタフェース名を入力します。

7 「**Insert**」ドロップダウンリストから、このエントリを挿入する `socks5.conf` ファイル内の位置を選択し、「了解」をクリックします。

複数のルーティング方法が存在する可能性もあるので、これらが評価される順序を指定する必要があります。

注 – 指定したインタフェースは、受信および送信の接続の両方で使用する必要があります。そうしないと、受信ルートが設定したインタフェースと異なるために、エラーメッセージが表示されます。

▼ プロキシルーティングエントリを作成するには

1 サーバーインスタンスのサーバーマネージャーにアクセスし、「**SOCKS**」タブをクリックします。

2 「**Set SOCKS v5 Routing**」リンクをクリックします。

3 「**Proxy Routing**」セクションで、「**Add**」ボタンをクリックします。

4 「**Proxy Type**」ドロップダウンリストから、経由する **Proxy Server** の種類を選択します。次のオプションが利用できます。

- **SOCKS v5**
  - **SOCKS v4**
  - **Direct connection**

5 「**Destination Host Mask**」フィールドに、接続エントリが適用される **IP** アドレスまたはホスト名を入力します。

IP アドレスを入力する場合、アドレスの後にスラッシュを入力し、その後に受信する IP アドレスに適用されるマスクを入力します。SOCKS サーバーはこのマスクを接続先コンピュータの IP アドレスに適用し、有効な接続先ホストかどうかを判別します。ホストマスクのエントリには空白文字を使用しないでください。接続先のホストマスクを入力しない場合、接続エントリはすべてのホストに適用されます。

たとえば、「destination host mask」フィールドには 155.25.0.0/255.255.0.0 と入力できます。接続先ホストの IP アドレスが 155.25.3.5 の場合、SOCKS サーバーは IP アドレスにマスクを適用して、接続先ホストの IP アドレスがプロキシエントリの適用される IP アドレス (155.25.0.0) と一致するかを判別します。

6 「**Destination Port Range**」フィールドに、プロキシエントリが適用される接続先ホスト上のポートを入力します。

ポート範囲のエントリには空白文字を使用しないでください。ポート範囲を指定しない場合、プロキシエントリはすべてのポートに適用されます。

角括弧 [ ] を使用すると範囲の開始と終了のポートを含む設定となり、丸括弧 ( ) を使用すると範囲の開始と終了のポートは除外されます。たとえば、[1000-1010] は、1000 ～ 1010 までの間のすべてのポート番号を意味し、ここでは 1000 と 1010 が含まれます。一方、(1000-1010) は 1000 ～ 1010 までの間のすべてのポート番号を意味しますが、ここでは 1000 と 1010 は含まれません。角括弧と丸括弧は合わせて使用することもできます。たとえば、(1000-1010] は、1000 ～ 1010 までの間のすべてのポート番号を意味しますが、1000 は含まれず、1010 は含まれます。

7 「**Destination Proxy Address**」フィールドに、使用する **Proxy Server** のホスト名または **IP** アドレスを入力します。

8 「**Destination Proxy Port**」フィールドに、**Proxy Server** が **SOCKS** 要求を待機するポート番号を入力します。

9 「**Insert**」ドロップダウンリストから、このエントリを挿入する `socks5.conf` ファイル内の位置を選択し、「了解」をクリックします。

複数のルーティング方法が存在する可能性もあるので、これらが評価される順序を指定する必要があります。

▼ ルーティングエントリを編集するには

1 サーバーインスタンスのサーバーマネージャーにアクセスし、「**SOCKS**」タブをクリックします。

2 「**Set SOCKS v5 Routing**」リンクをクリックします。

3 編集するエントリを選択します。

4 「編集」ボタンをクリックします。

5 必要な変更を行います。

6 「了解」をクリックします。

▼ ルーティングエントリを削除するには

1 サーバーインスタンスのサーバーマネージャーにアクセスし、「**SOCKS**」タブをクリックします。

2 「**Set SOCKS v5 Routing**」リンクをクリックします。

3 削除するエントリを選択します。

4 「削除」をクリックします。

▼ ルーティングエントリを移動するには

エントリは、`socks5.conf` ファイルに表示される順番で評価されます。この順番は、エントリを移動させることで変更できます。

1 サーバーインスタンスのサーバーマネージャーにアクセスし、「**SOCKS**」タブをクリックします。

2 「**Set SOCKS v5 Routing**」リンクをクリックします。

3 移動するエントリを選択します。

4 「移動」ボタンをクリックします。

5 「**Move**」ドロップダウンリストから、このエントリを挿入する `socks5.conf` ファイル内の位置を選択し、「了解」をクリックします。

◆ ◆ ◆ 第 16 章

16 テンプレートとリソースの管理

テンプレートを使用すると、複数の URL をグループ化して、プロキシによる処理方法を設定できます。クライアントが取得しようとする URL によって異なるプロキシの動作を設定できます。たとえば、特定のドメインから URL にアクセスするときに、ユーザー名とパスワードを入力して、クライアントの認証を求めることもできます。または、イメージファイルをポイントする URL へのアクセスを拒否するようにすることもできます。ファイルタイプに基づいて、別々のキャッシュ更新方法を設定できます。

この章の内容は次のとおりです。

テンプレートについて

テンプレートとは、リソースと呼ばれる URL の集合です。リソースは、単一の URL、なんらかの共通点を持つ複数の URL のグループ、またはプロトコル全体である場合があります。テンプレートを作成して名前を付けたら、そのテンプレートに、正規表現を使用して URL を割り当てます。このようにして、さまざまな URL の要求に対して、異なる処理ができるように、Proxy Server を設定できます。正規表現を使用して作成できる URL パターンはすべて、テンプレートに含めることができます。次の表は、デフォルトのリソースを表示し、ほかのテンプレートの内容をいくつか示しています。

表 16–1 リソースの正規表現のワイルドカードパターン

正規表現のパターン	設定内容
`ftp://.*`	すべての FTP 要求

表 16–1 リソースの正規表現のワイルドカードパターン (続き)

正規表現のパターン	設定内容
`http://.*`	すべての HTTP 要求
`https://.*`	すべてのセキュリティー保護された HTTP 要求
`gopher://.*`	すべての Gopher 要求
`connect://.*:443`	HTTPS ポートに対するすべての SSL (セキュリティー保護された) トランザクション
`http://home\.example\.com.*`	`home.example.com` Web サイト上のすべてのドキュメント
`.*\.gif.*`	文字列 `.gif` を含むすべての URL
`.*\.edu.*`	文字列 `.edu` を含むすべての URL
`http://.*\.edu.*`	`.edu` ドメイン内のコンピュータに移動するすべての URL

正規表現について

Proxy Server では、正規表現を使用してリソースを特定することができます。正規表現では、文字列のパターンを指定します。Proxy Server では、URL 内の一致パターンを見つけるために正規表現が使用されます。

正規表現の例を次に示します。

```
[a-z]*://[^:/]*\.abc\.com.*
```

この正規表現は、`.abc.com` ドメインにある任意のドキュメントに一致します。どのプロトコルのドキュメントでも、どのファイル拡張子が付いていてもかまいません。

次の表は、正規表現と、それぞれに対応する意味を示します。

表 16–2 正規表現とその意味

表現	意味
。	改行文字を除く、任意の単一文字に一致します。
x?	正規表現の *x* が 0 回または 1 回出現するものに一致します。
*x**	正規表現の *x* が 0 回以上出現するものに一致します。
x+	正規表現の *x* が 1 回以上出現するものに一致します。
x{n,m}	文字 *x* に一致します。ここで *x* の出現は *n* 回以上、*m* 回以下です。

表 16–2 正規表現とその意味 (続き)

表現	意味
x{n,}	文字 *x* に一致します。ここで *x* の出現は *n* 回以上です。
x{n}	文字 *x* に一致します。ここで *x* の出現はちょうど *n* 回です。
[abc]	角括弧に囲まれた任意の文字に一致します。
[^abc]	角括弧に囲まれていない任意の文字に一致します。
[a-z]	角括弧内に示された範囲内の任意の文字に一致します。
x	文字 *x* に一致します。ここで *x* は特殊文字ではありません。
\x	特殊文字 *x* の意味を削除します。
"x"	特殊文字 *x* の意味を削除します。
xy	正規表現 *x* が出現し、それに続いて正規表現 *y* が出現するものに一致します。
x|y	正規表現 *x* か、正規表現 *y* のどちらかが出現するものに一致します。
^	文字列の冒頭に一致します。
$	文字列の末尾に一致します。
(x)	正規表現をグループ化します。

次の例は、354 ページの「正規表現について」に示した正規表現のうち、いくつかの使用方法について示したものです。

```
[a-z]*://([^.:/]*[:/]|.*\.local\.com).*
```

- `[a-z]*` は、任意のプロトコルのドキュメントに一致します。
- `://` は、(:) の後に (//) が続くものに一致します。
- `[^.:/]*[:/]` は、(.)、(:)または (/) を含まない任意の文字列の後に、(:) または (/) が続くものに一致します。つまり、この表現は完全修飾されていないホスト名と、ポート番号の付いたホストに一致します。
- `|.*\.local\.com` は、完全修飾ドメイン名のホスト名 (`local.com` など) には一致しませんが、`.local.com` ドメイン内のドキュメントには一致します。
- `.*` は、任意のファイル拡張子を持つドキュメントに一致します。

354 ページの「正規表現について」に示したように、円記号は、特殊文字の意味をエスケープまたは削除するために使用することができます。ピリオドや疑問符などの文字には特別な意味があるので、それ自体を表すものとしてこれらの文字を使用する場合は、エスケープする必要があります。特に、ピリオドは多くの URL 内で使用

されています。したがって、正規表現内でピリオドの特殊な意味を削除するために、ピリオドの前に円記号を付ける必要があります。

ワイルドカードパターンについて

ワイルドカードパターンのリストを作成すると、サイトからアクセス可能な URL を指定できるようになります。ワイルドカードは、使用方法によって、正規表現かシェル表現のどちらかの形式になります。一般的な規則は次のとおりです。

- 目的の URL に一致する任意のパターンの正規表現を使用します。ここには、<Object ppath=...>、URL フィルタ、および NameTrans、PathCheck、ObjectType などの関数が含まれます。
- 受信するクライアントまたはユーザー ID に一致する任意のパターンのシェル表現を使用します。ユーザー ID には、ユーザー名、アクセス制御のグループ、および受信するユーザーの IP アドレスか DNS 名 (たとえば <Client dns=...>) が含まれます。

正規表現のワイルドカードパターンを使用して、複数の URL を指定することができます。ワイルドカードでは、任意のドメイン名、または指定した語が URL に含まれる任意の URL によってフィルタをかけることができます。たとえば、「careers」という文字列を含む URL へのアクセスをブロックすることもできます。このためには、テンプレートの正規表現として `http://.*careers.*` を指定します。

テンプレートでの作業

▼ テンプレートを作成するには

正規表現のワイルドカードパターンを使用して、テンプレートを作成できます。その後、そのテンプレートで指定した URL のみに影響する要素を設定することができます。たとえば、あるタイプのキャッシュ設定を `.GIF` イメージに対して使用し、プレーンな `.html` ファイルに対しては別のタイプのキャッシュ設定を使用することもできます。

1 サーバーマネージャーにアクセスし、「**Templates**」タブをクリックします。

「Create Template」リンクをクリックします。「Create Template」ページが表示されます。

2 「**Template Name**」フィールドに、作成するテンプレートの名前を入力して、「了解」をクリックします。

名前は覚えやすいものにしてください。サーバーマネージャーから、変更を保存して適用することを求めるプロンプトが表示されます。テンプレート用の正規表現を作成したら、下記の手順に従って変更を保存できます。

▼ テンプレートを適用するには

1 サーバーマネージャーにアクセスし、「**Templates**」タブをクリックします。

2 「**Apply Template**」リンクをクリックします。
「Apply Template」ページが表示されます。

3 「**URL Prefix Wildcard**」フィールドに、テンプレートに含めるすべての **URL** を含む正規表現のワイルドカードパターンを入力します。

4 「**Template**」リストから、追加したばかりの新しいテンプレート名を選択します。

5 「了解」をクリックします。

6 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

7 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

▼ テンプレートを削除するには

既存のテンプレートを削除することもできます。テンプレートを削除すると、テンプレートに関連するすべての設定が削除されます。たとえば、テンプレート TEST にすべての URL に対してアクセス制御を設定した場合、TEST テンプレートを削除すると、テンプレートに含まれている URL へのアクセス制御も削除されます。

1 サーバーマネージャーにアクセスし、「**Templates**」タブをクリックします。

2 「**Remove Template**」リンクをクリックします。
「Remove Template」ページが表示されます。

3 「**Remove**」リストからテンプレートを選択します。

4 「了解」をクリックします。

5 「**Restart Required**」をクリックします。
「Apply Changes」ページが表示されます。

6 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

▼ テンプレートを編集するには

サーバーマネージャーで作成したテンプレートを表示および編集することができます。

1. サーバーマネージャーにアクセスし、「**Templates**」タブをクリックします。
2. 「**View Template**」リンクをクリックします。
   「View Template」ページが表示されます。テンプレートの表には、テンプレート用の正規表現とテンプレート名が表示されます。
3. 既存のテンプレートを編集するには、「**Edit Template Assignment**」リンクをクリックします。「**Apply Template**」ページが表示されます。

リソースの削除

「Remove Resource」ページでは、正規表現オブジェクト全体と、それに対応する設定を削除できます。たとえば、Gopher のリソースを削除することで、そのリソースに関連するすべての設定を、Proxy Server の設定ファイルから削除することができます。

▼ リソースを削除するには

1. サーバーマネージャーにアクセスし、「**Templates**」タブをクリックします。
2. 「**Remove Resource**」リンクをクリックします。
   「Remove Resource」ページが表示されます。
3. 「**Remove**」ドロップダウンリストから、削除するリソースを選択します。
4. 「了解」をクリックします。
5. 「**Restart Required**」をクリックします。
   「Apply Changes」ページが表示されます。
6. 「**Restart Proxy Server**」ボタンをクリックして、変更を適用します。

◆ ◆ ◆ 第 17 章

17 クライアント自動設定ファイルの使用

多数のクライアントをサポートする複数のプロキシサーバーがある場合、クライアント自動設定ファイルを使用して、すべてのブラウザクライアントを設定することができます。自動設定ファイルには、さまざまな URL にアクセスする場合、ブラウザが使用するプロキシがあれば、どのプロキシを使用するかを決定する JavaScript 関数が含まれています。

ブラウザは、起動時に自動設定ファイルを読み込みます。ユーザーがリンクをクリックするか URL を入力するたびに、ブラウザは設定ファイルを使用して、プロキシを使用する必要があるかどうか、使用する場合はどのプロキシを使用するのかを決定します。この機能によって、組織内のブラウザのすべてのインスタンスを簡単に設定することができます。自動設定ファイルは、次のような複数の方法でクライアントに配布できます。

- 自動設定ファイルを返す Web サーバーとしてプロキシサーバーを使用することができます。ブラウザに、プロキシの URL を指定します。プロキシを Web サーバーとして動作させることで、自動設定ファイルを一箇所に保持できるので、更新の必要が生じたときに 1 つのファイルを変更するだけで済みます。
- 自動設定ファイルは、Web サーバー、FTP サーバー、またはブラウザがアクセスできるネットワークディレクトリに保存することができます。ファイルへの URL を指定することでファイルを見つけられるようにブラウザを設定します。これは一般の URL の機能と同じです。複雑な計算が必要な場合 (たとえば、組織内に大規模なプロキシチェーンがある場合など) は、ファイルにアクセスするユーザーに応じて異なるファイルを出力する Web サーバーの CGI プログラムを作成することもできます。
- 各ブラウザをコピーすることによって自動設定ファイルをローカルに保存することができます。ただし、ファイルを更新する必要がある場合は、ファイルのコピーを各クライアントに配布する必要があります。

自動設定ファイルは次の 2 つの方法で作成することができます。サーバーマネージャーのページを使用するか、またはファイルを手動で作成できます。ファイルを作成するための指令については、この章の後半で説明します。

この章の内容は次のとおりです。

自動設定ファイルについて

Proxy Server の管理者としてクライアント自動設定ファイルを作成して配布する場合があります。

自動設定ファイルの機能

自動設定ファイルは JavaScript で記述されています。JavaScript はコンパクトな、オブジェクトベースのスクリプト言語で、クライアントおよびサーバーのインターネットアプリケーションの開発に使用されます。ブラウザが JavaScript ファイルを解釈します。

ブラウザは最初の読み込み時に自動設定ファイルをダウンロードします。このファイルは、ブラウザが URL を使用してアクセスできる任意の場所に保存できます。たとえば、自動設定ファイルを Web サーバー上に保存することができます。ブラウザがファイル `://` `URL` を使用して自動設定ファイルを取得できる場合には、自動設定ファイルをネットワークファイルシステム上に保存することができます。

プロキシ設定ファイルは JavaScript で記述されています。この JavaScript ファイルでは 1 つの関数 (*FindProxyForURL*) が定義されています。この関数は、各 URL に対してブラウザが使用するプロキシサーバーがあれば、どれを使用するのかを決定するものです。ブラウザは、JavaScript 関数に 2 つのパラメータを送信します。ブラウザが実行されているシステムのホスト名と、取得しようとしている URL です。JavaScript 関数は、処理方法を示す値をブラウザに返します。

自動設定ファイルを使用すると、さまざまなタイプの URL、さまざまなサーバー、または 1 日のうちのさまざまな時間に対して、異なるプロキシを指定できるようになります。またはプロキシをまったく指定しないこともできます。つまり、複数の専用のプロキシを用意できるので、たとえば、あるプロキシを `.com` ドメイン、別のプロキシを `.edu` ドメイン、さらに別のプロキシをその他すべてのドメイン用にすることができます。これによって、負荷を分散して、プロキシのディスクをさらに効率的に使用できるようになります。同じドキュメントを複数のプロキシすべてで保存するのではなく、どのファイルについてもキャッシュ内にコピーが 1 つ存在するだけだからです。

自動設定ファイルはプロキシのフェイルオーバーもサポートしているので、プロキシサーバーが使用できない場合、ブラウザはユーザーに意識させずに別のプロキシサーバーに切り替えを行います。

Web サーバーとしてのプロキシへのアクセス

プロキシサーバー上に 1 つ以上の自動設定ファイルを保存し、そのプロキシサーバーを、自動設定ファイルを唯一のドキュメントとする Web サーバーとして機能させることができます。これによって、プロキシの管理者は、組織内のクライアントが必要とするプロキシ自動設定ファイルを保持することができます。また、ファイルを中央で保持できるので、ファイルを更新する必要がある場合は、中央のファイルを一度更新するだけで、すべてのブラウザクライアントにより自動的に更新が取得されます。

プロキシ自動設定ファイルは、*server- root*/proxy-server*id*/pac/ ディレクトリで保持されます。ブラウザでは、プロキシ自動設定ファイルへの URL を「Proxies」タブに入力します。プロキシの URL は次の書式になります。

`http://`*proxy.domain:port/URI*

たとえば、URL は `http://proxy.example.com` となります。この後に、host:port の組み合わせが続きます。URI を使用すれば、テンプレートを使用してさまざまな自動設定ファイルへのアクセスを制御できます。たとえば、`/proxy.pac` という自動設定ファイルを格納する `/test` という URI を作成すると、リソースパターン `http://proxy.mysite.com:8080/test/.*` を使用したテンプレートを作成できます。次に、このテンプレートを使用して、そのディレクトリ固有のアクセス制御を設定することができます。

複数の自動設定ファイルを作成し、それぞれが異なる URL 経由でアクセスするように設定できます。次の表は、クライアントがアクセスするために使用する URI と URL の例をいくつか示しています。

表 17–1　URI の例とそれに対応する URL

URI (パス)	プロキシへの URL
/	http://proxy.mysite.com
/employees	http://proxy.mysite.com/employees
/group1	http://proxy.mysite.com/group1
/managers	http://proxy.mysite.com/managers

逆プロキシによる pac ファイルの使用

逆プロキシの機能から考えて、プロキシサーバーを使用して、`.pac` ファイルを処理することは困難です。プロキシサーバーがファイルに対する要求を受け取ったときに、その要求がローカルの `.pac` ファイルに対するものなのか、それともリモートドキュメントに対するものなのかを判断する必要があるからです。

.pac ファイルの管理および配布に加えて、プロキシサーバーを逆プロキシとして動作させるには、obj.conf ファイルを手動で編集して、NameTrans 関数の順序が正しいことを確認する必要があります。

プロキシサーバーが逆プロキシとして動作するように、通常マッピングを作成します。このマッピングは、通常、すべての要求をリモートコンテンツサーバーに向けてルーティングするようプロキシに指示します。プロキシ自動設定ファイルを追加し、これを特定のディレクトリ (/pac など) にマップすることができます。ここでは、.pac ファイルを受け取るクライアントはすべて、次のような URL を使用します。

```
http://proxy.mysite.com/pac
```

注意 – このマッピングでは、リモートコンテンツサーバーに同じようなディレクトリが存在していないことを確認してください。

obj.conf ファイルを編集し、プロキシ自動設定ファイルに関する指令および関数が、ほかのマッピングより先に表示されていることを確認します。プロキシサーバーは通常、要求を処理する前にすべての NameTrans 関数を実行するので、この指令と関数が先に置かれている必要があります。ただし、自動設定ファイルの場合は、プロキシが即座にパスを認識して、.pac ファイルを返します。

次に、逆プロキシを使用し、自動設定ファイルを管理する obj.conf ファイルの記述例を示します。

```
<Object name="default">
NameTrans from="file:" fn="map" to="ftp:"
NameTrans from="/pac" fn="pac-map" name="file"
    to="/ns-home/proxy/pac/proxy.pac"
NameTrans fn="redirect" from="http://foo.*" url="http://www.acme.com"
NameTrans from="/ns-icons" fn="pfx2dir" dir="/ns-home/ns-icons" name="file"
NameTrans fn="reverse-map" from="http://web.acme.com"
    to="http://proxy.acme.com:8080"
NameTrans fn="map" from="http://proxy.acme.com:8080"
    to="http://web.acme.com"
NameTrans fn="map" from="/" to="http://web.acme.com"
PathCheck fn="url-check"
Service fn="deny-service"
AddLog fn="flex-log" name="access"
AddLog fn="urldb-record"
</Object>
```

サーバーマネージャーのページを使用した自動設定ファイルの作成

▼ サーバーマネージャーのページを使用して自動設定ファイルを作成するには

1 サーバーマネージャーにアクセスし、「**Routing**」タブを選択します。

2 「**Create/Edit Autoconfiguration**」リンクをクリックします。

ユーザーのプロキシシステム上にある自動設定ファイルのリストを示すページが表示されます。自動設定ファイルをクリックすると、それを編集できます。以下の手順では、新しいファイルの作成方法について説明します。

3 クライアントが自動設定ファイルをプロキシから入手するときに使用するオプションの **URI (URL** のパス部分**)** を入力します。

たとえば、/ を入力して、プロキシのメインドキュメント (Web サーバーの `index.html` ファイルのようなもの) となるファイルにクライアントがアクセスできるようにします。これでクライアントは、自動設定ファイルのプロキシにアクセスするときに、ドメイン名のみを使用するようになります。複数の URI を使用して、各 URI に個別の自動設定ファイルを作成することができます。

4 `.pac` 拡張子を使用して、自動設定ファイルの名前を入力します。

ファイルが 1 つだけの場合は、単に `proxy.pac` (pac は proxy autoconfiguration の略) という名前にすることもできます。すべての自動設定ファイルは、JavaScript 関数が 1 つ含まれた ASCII テキストファイルです。

5 ［了解］をクリックします。別のページが表示されます。

このページを使用して、自動設定ファイルを作成します。このページ上の項目は、クライアントごとの順序で表示されています。次のような項目があります。

- 「Never Go Direct To Remote Server」は、ブラウザに対して、常にプロキシを使用するよう指示します。プロキシサーバーが実行されていない場合は、2 番目のプロキシサーバーを使用するように指定できます。
- 「Go Direct To Remote Server When」では、プロキシサーバーをバイパスする状況を指定します。ブラウザは、このページに示されたオプションの順序に従って、これらの状況を識別します。
- 「Connecting To Non-fully Qualified Host Names」では、ユーザーがコンピュータ名のみを指定した場合に、直接サーバーにアクセスするようブラウザに指示します。たとえば、内部 Web サーバーが `winternal.mysite.com` である場合、ユーザー

が、完全修飾ドメイン名を入力せずに `http://winternal` とだけ入力することがあります。このような場合、ブラウザはプロキシではなく、直接 Web サーバーにアクセスします。

- 「Connecting To A Host In Domain」では、ブラウザが直接アクセスできるドメイン名を 3 つまで指定できます。ドメインを指定する場合は、ドットから始めます。たとえば、`.example.com` と入力できます。
- 「Connecting To A Resolvable Host」では、クライアントがホストを解決できる場合、ブラウザが直接そのサーバーにアクセスするようにします。このオプションは、通常、DNS がローカル (内部) ホストのみを解決するように設定されている場合に使用します。ローカルネットワークの外側にあるサーバーに接続する場合、クライアントはプロキシサーバーを使用します。

注意 – 上のオプションを選択すると、クライアントはすべての要求について DNS を参照します。このため、クライアント側から見た場合のパフォーマンスに悪影響を及ぼします。

- 「Connecting To A Host In Subnet」では、クライアントが特定のサブネット内のサーバーにアクセスする場合、ブラウザが直接そのサーバーにアクセスするようにします。このオプションは、ある地域内に組織が多数のサブネットを抱えている場合に便利です。たとえば、1 つのドメイン名を世界中の複数のサブネットに適用していても、各サブネットは特定の地域に固有である企業のような場合です。

注意 – 上のオプションを選択すると、クライアントはすべての要求について DNS を参照します。このため、クライアント側から見た場合のパフォーマンスに悪影響を及ぼします。

- 「Except When Connecting To Hosts」では、サーバーに直接アクセスするときのルールに対して例外を指定することができます。たとえば、直接アクセスするドメインとして `.example.com` と入力した場合、`home.example.com` にアクセスする例外を作成することができます。ブラウザは、`home.example.com` にアクセスするときにはユーザーのプロキシを使用するものの、`example.com` ドメイン内のその他のサーバーには直接アクセスするようになります。
- 「Secondary Failover Proxy」では、プロキシサーバーが実行されていない場合に使用する 2 番目のプロキシを指定します。
- 「Failover Direct」では、プロキシサーバーが実行されていない場合、サーバーに直接アクセスするようブラウザに指示します。二次フェイルオーバープロキシを指定している場合、Navigator はサーバーに直接アクセスする前に、2 番目のプロキシサーバーへのアクセスを試行します。

6 「了解」をクリックして、自動設定ファイルを作成します。

このファイルは、ディレクトリ *server-root*/proxy-server *id*/pac に格納されます。

ファイルが正常に作成されたことを示す確認メッセージが表示されます。ここまでの手順を繰り返して、必要な数だけ自動設定ファイルを作成します。

自動設定ファイルを作成したら必ず、プロキシサーバーを使用するすべてのユーザーに正しい自動設定ファイルを指定するように指示するか、ブラウザのコピーを自分で設定してください。

自動設定ファイルの手動による作成

この節では、自動設定ファイルを手動で作成する方法について説明します。

プロキシ設定ファイルは、クライアントサイド JavaScript を使用して記述されています。各ファイルには、FindProxyForURL() という 1 つの JavaScript 関数が含まれています。この関数は、各 URL に対してブラウザが使用するプロキシサーバーがあれば、どれを使用するのかを決定するものです。ブラウザは、JavaScript 関数に 2 つのパラメータを送信します。接続先の配信元サーバーのホスト名と、取得しようとしている URL です。JavaScript 関数は、処理方法を示す値を Navigator に返します。次の節では、関数の構文と考えられる戻り値について説明します。

FindProxyForURL() 関数

FindProxyFor() URL 関数の構文は次のとおりです。

```
function FindProxyForURL(url, host){   ...}
```

ブラウザがアクセスするすべての URL に対して、*url* および *host* パラメータが送られ、次のようにして関数が呼び出されます。

```
ret = FindProxyForURL(url, host);
```

url は、ブラウザからアクセスされる完全な URL です。

host は、アクセスされた URL から抽出されるホスト名です。これは便宜上使用しているだけで、**://** と最初の **:** またはそのあとの **/** の間にある文字列と同じものです。このパラメータにはポート番号は含まれません。ポート番号は必要に応じて URL から抽出できます。

ret (戻り値) は、設定を記述した文字列です。

関数の戻り値

自動設定ファイルには、関数 FindProxyForURL () が含まれています。この関数は、クライアントホスト名と、アクセスしている URL をパラメータとして使用します。この関数は、ブラウザに処理方法を指示する 1 つの文字列を返します。この文字列が null の場合は、プロキシは使用されません。この文字列には、次の表に示す構成ブロックを、セミコロンで区切っていくつでも含めることができます。

表 17–2 FindProxyForURL() の戻り値

戻り値	ブラウザの結果のアクション
DIRECT	プロキシを経由せずに、サーバーに直接接続します。
PROXY *host:port*	指定されたプロキシとポート番号を使用します。複数の値がセミコロンで区切られている場合、最初のプロキシが使用されます。そのプロキシで失敗した場合は、それ以降のプロキシが順に使用されます。
SOCKS *host:port*	指定された SOCKS サーバーを使用します。複数の値がセミコロンで区切られている場合、最初のプロキシが使用されます。そのプロキシで失敗した場合は、それ以降のプロキシが順に使用されます。

ブラウザは、使用できないプロキシサーバーに遭遇すると、30 分後に以前に応答のなかったプロキシに自動的に再度アクセスします。それでも応答がなければ 1 時間後というように、30 分間隔で再試行します。つまり、プロキシサーバーを一時的にシャットダウンした場合でも、再起動後 30 分以内には、クライアントがプロキシの使用を再開することになります。

すべてのプロキシが停止しており、DIRECT 戻り値が指定されていない場合、ブラウザは、一時的にプロキシを無視して直接接続を試みるかどうかをユーザーに問い合わせます。ブラウザは、プロキシを 20 分後に再試行し、さらにまた 20 分後というように、20 分間隔で再試行するかどうかを問い合わせてきます。

次の例では、戻り値は、ポート 8080 上で w3proxy.example.com というプロキシを使用するようにブラウザに指示していますが、このプロキシが使用できない場合、ブラウザはポート 8080 上で proxy1.example.com というプロキシを使用します。

```
PROXY w3proxy.example.com:8080; PROXY proxy1.example.com:8080
```

次の例では、一次プロキシは w3proxy.example.com:8080 で、このプロキシが使用できない場合、ブラウザは proxy1.example.com:8080 を使用します。これらのプロキシがどちらも使用できない場合、ブラウザはサーバーに直接アクセスします。さらに 20 分後、ブラウザは最初のプロキシを再試行するかどうかを問い合わせてきます。

```
PROXY w3proxy.example.com:8080; PROXY proxy1.example.com:8080; DIRECT
```

JavaScript の関数および環境

JavaScript 言語には、複数の事前定義された関数と環境条件があり、これらはプロキシの処理に役立ちます。これらの関数はそれぞれ、一定の条件が満たされているかどうかをチェックして、true または false の値を返します。ただし、関連するユーティリティー関数は例外で、DNS ホスト名または IP アドレスを返します。メインの `FindProxyForURL()` 関数内でこれらの関数を使用して、ブラウザに送る戻り値を決定することができます。この章の後半で挙げる例では、これらの関数の使用に関する考え方を説明します。

この節では、それぞれの関数や環境条件について説明します。プロキシとのブラウザ統合に適用される関数と環境変数は、次のとおりです。

- ホスト名ベースの関数
  - `dnsDomainIs()()`
  - `isInNet()()`
  - `isPlainhost name()()`
  - `isResolvable()()`
  - `localHostOrDomainIs()()`
- ユーティリティー関数
  - dnsDomainLevels()
  - dnsResolve()
  - myIpAddress()
- URL/ホスト名ベースの条件
  - shExpMatch()
- 時間ベースの条件
  - dateRange()
  - timeRange()
  - weekdayRange()

ホスト名ベースの関数

ホスト名ベースの関数では、ホスト名または IP アドレスを使用して、使用するプロキシがあれば決定します。

`dnsDomainIs()`(host, domain)

`dnsDomainIs()()` 関数は、URL ホスト名が、指定された DNS ドメインに属しているかどうかを検出します。この関数は、ローカルドメインに対してプロキシを使用しないようにブラウザを設定する場合に役立ちます。376 ページの「例 1: ローカルホスト以外の全サーバーのプロキシ」と 376 ページの「例 2: ファイアウォールの外側にあるローカルサーバーのプロキシ」を参照してください。

この関数は、要求を受け取るプロキシが、URL の属している DNS ドメインに基づいてプロキシのグループから選択される状況において、複数のプロキシを使用して負荷分散をはかる場合にも役立ちます。たとえば、.edu を含む複数の URL をあるプロキシに割り当て、.com を含む複数の URL を別のプロキシに割り当てることで負荷分散をはかる場合、dnsDomainIs()() を使用して URL ホスト名を確認することができます。

パラメータ

host は、URL からのホスト名です。

domain は、ホスト名のテストで照合するドメイン名です。

戻り値

true または false

例

次の文は true になります。

```
dnsDomainIs("www.example.com", ".example.com")
```

次の文は false になります。

```
dnsDomainIs("www", ".example.com") dnsDomainIs("www.mcom.com",
    ".example.com")
```

isInNet()(host, pattern, mask)

isInNet()() 関数を使用すると、URL ホスト名を IP アドレスに解決し、それがマスクによって指定されたサブネットに属しているかどうかをテストすることができます。これは、SOCKS が使用するものと一致する IP アドレスパターンです。377 ページの「例 4: サブネットへの直接接続」を参照してください。

パラメータ:

host は、DNS ホスト名または IP アドレスです。ホスト名が渡されると、この関数はこれを IP アドレスに解決します。

pattern は、ドット区切り形式の IP アドレスパターンです。

mask は、IP アドレスパターンマスクで、IP アドレスのどの部分をマッチングするのかを指定します。値 0 は無視、値 255 は一致を意味しています。この関数は、ホストの IP アドレスが指定された IP アドレスパターンと一致すると true になります。

戻り値

true または false

例

次の文では、ホストの IP アドレスが 198.95.249.79 と完全に一致した場合のみ true になります。

```
isInNet(host, "198.95.249.79", "255.255.255.255")
```

次の文では、ホストの IP アドレスが 198.95.*.* と一致した場合のみ true になります。

```
isInNet(host, "198.95.0.0", "255.255.0.0")
```

isPlainhost name()(host)

isPlainhost name()() 関数は、要求された URL のホスト名がプレーンなホスト名なのか完全修飾ドメイン名なのかを検出します。この関数は、ブラウザをローカルサーバーに直接接続させる場合に役立ちます。376 ページの「例 1: ローカルホスト以外の全サーバーのプロキシ」と 376 ページの「例 2: ファイアウォールの外側にあるローカルサーバーのプロキシ」を参照してください。

パラメータ

host は、ホスト名にドメイン名がない (ドットの付いたセグメントがない) 場合のみの、URL からのホスト名です (ポート番号を除く)。

戻り値

host がローカルの場合は true、*host* がリモートの場合は false

例

```
isPlainhost name("host")
```

host が www などの場合は true が返されます。host が www. example.com などの場合は false が返されます。

isResolvable()(host)

ファイアウォールの内側の DNS が内部ホストのみを認識する場合、isResolvable()() 関数を使用して、ホスト名がネットワークの内部か外部かをテストすることができます。この関数を使用すると、内部サーバーには直接接続を使用し、外部サーバーに対してのみプロキシを使用するように、ブラウザを設定できます。これは、ファイアウォールの内側の内部ホストが、ほかの内部ホストの DNS ドメイン名は解決できても、外部ホストは一切解決できないサイトで役立ちます。isResolvable()() 関数は DNS を参照し、ホスト名を IP アドレスに解決しようとします。377 ページの「例 3: 未解決のホストのみのプロキシ」を参照してください。

パラメータ

`host()` は、URL からのホスト名です。

戻り値

ホスト名が解決できたら true、できなければ false

例

```
isResolvable("host")
```

`host()` が `www` などで、DNS を介して解決できる場合、この関数は true を返します。

`localHostOrDomainIs()`(host, hostdom)

`localHostOrDomainIs()()` 関数は、完全修飾ドメイン名かプレーンなホスト名のどちらかによってアクセスされる可能性のあるローカルホストを特定します。376 ページの「例 2: ファイアウォールの外側にあるローカルサーバーのプロキシ」を参照してください。

`localHostOrDomainIs()()` 関数は、ホスト名が指定されたホスト名と完全に一致する場合か、ホスト名の中に非修飾ホスト名と一致するドメイン名部分がない場合に、`true` を返します。

パラメータ

host は、URL からのホスト名です。

hostdom は、一致する完全修飾ホスト名です。

戻り値

true または false

例

次の文は true (完全一致) です。

```
localHostOrDomainIs("www.example.com", "www.example.com")
```

次の文は true (ホスト名は一致、ドメイン名は指定なし) です。

```
localHostOrDomainIs("www", "www.example.com")
```

次の文は false (ドメイン名が不一致) です。

```
localHostOrDomainIs("www.mcom.com", "www.example.com")
```

次の文は false (ホスト名が不一致) です。

```
localHostOrDomainIs("home.example.com", "www.example.com")
```

ユーティリティー関数

ユーティリティー関数を使用すると、ドメインレベル、ブラウザが実行されているホスト、あるいはホストの IP アドレスを検出することができます。

dnsDomainLevels()(host)

`dnsDomainLevels()()` 関数は、URL ホスト名の中の DNS レベルの数 (ドットの数) を検出します。

パラメータ

host は、URL からのホスト名です。

戻り値

DNS ドメインレベルの数 (整数)。

例

`dnsDomainLevels("www")` は 0 を返します。

`dnsDomainLevels("www.example.com")` は 2 を返します。

dnsResolve()(host)

`dnsResolve()()` 関数は、指定されたホスト (通常は URL からのもの) の IP アドレスを解決します。この関数は、JavaScript 関数が、既存の関数でできるものより高度なパターンマッチングを実行する必要がある場合に役立ちます。

パラメータ

host は、解決するホスト名です。指定された DNS ホスト名を IP アドレスに解決し、ドット区切り形式の文字列として返します。

戻り値

ドットの付いた Quad 表記の IP アドレス (文字列値)

例

次の例では、文字列 198.95.249.79 が返されます。

```
dnsResolve("home.example.com")
```

myIpAddress()()

myIpAddress()() 関数は、ブラウザが実行されているホストによって、JavaScript 関数が異なる動作をする必要のある場合に役立ちます。この関数は、ブラウザを実行しているコンピュータの IP アドレスを返します。

戻り値

ドットの付いた Quad 表記の IP アドレス (文字列値)

例:

次の例では、コンピュータ home.example.com 上で Navigator を実行している場合に、文字列 198.95.249.79 を返します。

```
myIpAddress()
```

URL/ホスト名ベースの条件

負荷分散やルーティングを行うために、ホスト名または URL をマッチングすることができます。

shExpMatch()(str, shexp)

shExpMatch()() 関数は、URL ホスト名または URL そのものをマッチングします。この関数は主に、負荷を分散して、さまざまなプロキシサーバーに URL をインテリジェントにルーティングする場合に使用します。

パラメータ

str は、比較対象の文字列です (たとえば、URL またはホスト名)。

shexp は、比較するシェル表現です。

この表現は、文字列が指定されたシェル表現と一致したときに true になります。379 ページの「例 6: shExpMatch()() によるプロキシの負荷分散」を参照してください。

戻り値

true または false

例

最初の例は true を返し、2 つ目の例は false を返します。

```
shExpMatch("http://home.example.com/people/index.html",
           ".*/people/.*")
shExpMatch("http://home.example.com/people/yourpage/index.html",
           ".*/mypage/.*")
```

時間ベースの条件

日付、時間、または曜日によって、**FindProxyForURL** 関数が異なる動作をするように設定できます。

dateRange() (day, month, year...)

dateRange()() 関数は、1996 年 4 月 19 日から 1996 年 5 月 3 日のような、特定の日付や日付範囲を検出します。これは、日付によって FindProxyForURL() 関数に異なる動作をさせる場合に役立ちます。たとえば、プロキシの 1 つに対してメンテナンスのための停止時間を定期的にスケジュールするような場合です。

日付の範囲は、次に示すいくつかの方法で指定できます。

```
dateRange(day)dateRange(day1, day2)dateRange(mon)dateRange(month1,
month2)dateRange(year)dateRange(year1, year2)dateRange(day1, month1, day2,
month2)dateRange(month1, year1, month2, year2)dateRange(day1, month1, year1,
day2, month2, year2)dateRange(day1, month1, year1, day2, month2, year2, gmt)
```

パラメータ

day は、月の日付を表す 1 ～ 31 までの整数です。

month には、月を表す次のいずれかの文字列が入ります。JAN FEB MAR APR MAY JUN JUL AUG SEP OCT NOV DEC

year には、西暦を表す 4 桁の整数が入ります (たとえば 1996)。

gmt には、グリニッジ標準時で時間の比較を行う場合は文字列「GMT」を指定し、時間を現地時間帯とする場合は空白のままにします。GMT パラメータは、常に最後のパラメータとして、どの呼び出しプロファイル内でも指定できます。1 つの値だけが指定 (日、月、年の各カテゴリから) された場合、この関数は、その指定と一致する日に対してのみ true の値を返します。2 つの値が指定された場合、最初の指定時間から 2 つ目の指定時間までが true になります。

例

次の文は、ローカル時間帯の各月の最初の日について true になります。

```
dateRange(1)
```

次の文は、グリニッジ標準時の各月の最初の日について true になります。
`dateRange(1, "GMT")`

次の文は、各月の前半について true になります。`dateRange(1, 15)`

次の文は、各年の 12 月 24 日について true になります。`dateRange(24, "DEC")`

次の文は、1995 年 12 月 24 日について true になります。`dateRange(24, "DEC", 1995)`

次の文は、年の第 1 四半期について true になります。`dateRange("JAN", "MAR")`

次の文は、各年の 6 月 1 日から 8 月 15 日までについて true になります。
`dateRange(1, "JUN", 15, "AUG")`

次の文は、1995 年 6 月 1 日から 1995 年 8 月 15 日までについて true になります。
`dateRange(1, "JUN", 15, 1995, "AUG", 1995)`

次の文は、1995 年 10 月から 1996 年 3 月までについて true になります。
`dateRange("OCT", 1995, "MAR", 1996)`

次の文は、1995 年全体を通じて true になります。`dateRange(1995)`

次の文は、1995 年の始めから 1997 年の終わりまでについて true になります。
`dateRange(1995, 1997)`

timeRange (hour, minute, second...)

`timeRange()` 関数は、午後 9 時から午前 12 時のような、特定の時間や時間範囲を検出します。これは、時間によって `FindProxyForURL()` 関数に異なる動作をさせる場合に役立ちます。

```
timeRange(hour)timeRange(hour1, hour2)timeRange(hour1, min1, hour2,
min2)timeRange(hour1, min1, sec1, hour2, min2, sec2)
```

パラメータ:

hour は、0 時から 23 時までの時間です (0 は午前 0 時、23 は午後 11 時)。

min は、0 ～ 59 までの分数です。

sec は、0 ～ 59 までの秒数です。

gmt は、グリニッジ標準時に対しては文字列 GMT を指定し、現地時間帯の場合は何も指定しません。このパラメータは、各パラメータプロファイルで使用でき、常に最後のパラメータとなります。

戻り値

true または false

例:

次の文は、正午から午後 1 時までが true になります。m: `timerange(12, 13)`

次の文は、グリニッジ標準時の正午から午後 12 時 59 分までが true になります。
`timerange(12, "GMT")`

次の文は、午前 9 時から午後 5 時までが true になります。m: `timerange(9, 17)`

次の文は、午前 0 時から午前 0 時の 30 秒後までの間について true になります。
`timerange(0, 0, 0, 0, 0, 30)`

weekdayRange()(wd1, wd2, gmt)

weekdayRange() 関数は、月曜日から金曜日などの、特定の曜日または曜日の範囲を検出します。これは、曜日によって *FindProxyForURL* 関数に異なる動作をさせる場合に役立ちます。

パラメータ

wd1 と *wd2* には、曜日を表す次のいずれかの文字列が入ります。`SUN MON TUE WED THU FRI SAT`

gmt には、グリニッジ標準時を表す GMT を指定するか、現地時間帯の場合は空白のままにします。

最初のパラメータ *wd1* のみが必須です。*wd2* または *gmt*、あるいはその両方とも空白のままにすることができます。

パラメータが 1 つしか指定されていない場合、関数は、パラメータの表す曜日についてのみ true の値を返します。2 つ目のパラメータとして文字列 GMT が指定された場合、時間はグリニッジ標準時と見なされ、それ以外の場合は、現地時間帯と見なされます。

wd1 と *wd2* の両方が定義された場合、現在の曜日がこれら 2 つの曜日の間にあれば条件は true になります。両端の曜日も含まれます。この場合、パラメータの順序が重要になります。"MON", "WED" は、月曜日から水曜日までを表しますが、"WED", "MON" は水曜日から次週の月曜日までを表します。

例

次の文は、月曜日から金曜日までが true です (現地時間帯)。`weekdayRange("MON", "FRI")`

次の文は、グリニッジ標準時の月曜日から金曜日までが true です。
`weekdayRange("MON", "FRI", "GMT")`

次の文は、現地時間の土曜日が true です。`weekdayRange("SAT")`

次の文は、グリニッジ標準時の土曜日が true です。`weekdayRange("SAT", "GMT")`

次の文は、金曜日から月曜日までが true です (曜日の順序が重要)。
`weekdayRange("FRI", "MON")`

関数の例

この節では、JavaScript 関数の詳細な例を説明します。

例 1: ローカルホスト以外の全サーバーのプロキシ

次の例では、ブラウザは、完全修飾されていないすべてのホストと、ローカルドメイン内のホストに直接接続しています。それ以外はすべて `w3proxy.example.com:8080` というプロキシを経由します。

注 – プロキシが停止すると、自動的に直接接続に切り替わります。

```
function FindProxyForURL(url, host)
{
    if (isPlainhost name(host) ||
        dnsDomainIs(host, ".example.com") ||
        dnsDomainIs(host, ".mcom.com"))
        return "DIRECT";
    else
        return "PROXY w3proxy.example.com:8080; DIRECT";
}
```

例 2: ファイアウォールの外側にあるローカルサーバーのプロキシ

この例は376 ページの「例 1: ローカルホスト以外の全サーバーのプロキシ」とよく似ていますが、ここではファイアウォールの外側にあるローカルサーバーに対してプロキシを使用しています。ローカルドメインに属しているが、ファイアウォールの外側にあり、プロキシサーバーを経由しないとアクセスできないホスト (主要な Web サーバーなど) が存在する場合、これらの例外は `localHostOrDomainIs()()` 関数を使用して処理されます。

```
function FindProxyForURL(url, host)
{
    if ((isPlainhost name(host) ||
    dnsDomainIs(host, ".example.com")) &&
    !localHostOrDomainIs(host, "www.example.com") &&
    !localHostOrDoaminIs(host, "merchant.example.com"))
        return "DIRECT";
    else
        return "PROXY w3proxy.example.com:8080; DIRECT";
}
```

この例では、example.com ドメイン内のローカルホストを除くすべてに対してプロキシを使用しています。ホスト www.example.com と merchant.example.com も、プロキシを経由します。

例外の順序によって効率を上げることができます。localHostOrDomainIs() () 関数は、すべての URL に対してではなく、ローカルドメイン内の URL に対してのみ実行されます。特に、*and* 式の前の *or* 式を囲む括弧に注意してください。

例 3: 未解決のホストのみのプロキシ

この例は、内部ホスト名だけを解決できるように内部 DNS が設定されている環境で機能します。解決できないホストに対してのみプロキシを使用することを目的とします。

```
function FindProxyForURL(url, host)
{
    if (isResolvable(host))
            return "DIRECT";
        else
            return "PROXY proxy.mydomain.com:8080";
}
```

この例では、DNS を毎回参照する必要があるので、ほかのルールとグループ化して、ほかのルールで結果が得られない場合のみ DNS を参照するように設定します。

```
function FindProxyForURL(url, host)
{
    if (isPlainhost name(host) ||
        dnsDomainIs(host, ".mydomain.com") ||
        isResolvable(host))
        return "DIRECT";
    else
        return "PROXY proxy.mydomain.com:8080";
}
```

例 4: サブネットへの直接接続

この例では、指定されたサブネット内のホストはすべて直接接続し、それ以外はプロキシを経由します。

```
function FindProxyForURL(url, host)
{
    if (isInNet(host, "198.95.0.0", "255.255.0.0"))
        return "DIRECT";
    else
        return "PROXY proxy.mydomain.com:8080";
}
```

この例では、先頭に冗長なルールを追加して、DNS の使用を最小限に抑えることができます。

```
function FindProxyForURL(url, host)
{
    if (isPlainhost name(host) ||
        dnsDomainIs(host, ".mydomain.com") ||
        isInNet(host, "198.95.0.0", "255.255.0.0"))
        return "DIRECT";
    else
        return "PROXY proxy.mydomain.com:8080";
}
```

例 5: dnsDomainIs()() によるプロキシの負荷分散

次に挙げる例は、より詳細なものです。ここにはプロキシサーバーが 4 台あり、そのうちの 1 台がほかのサーバーに対するホットスタンバイとして機能しています。したがって、ほかの 3 台のサーバーのいずれかが停止したら、4 台目のサーバーが引き継ぎます。ほかの 3 台のプロキシサーバーは URL パターンに基づいて負荷を分散しており、これによってキャッシュが効率的に行われています。3 台のサーバー上のドキュメントはすべてコピーが 1 つしかありません。それぞれに 1 つずつコピーがあるわけではありません。負荷分散は、次の表に示すように行われています。

表 17–3 プロキシの負荷分散

プロキシ (Proxy)	目的
#1	.com ドメイン
#2	.edu ドメイン
#3	ほかのすべてのドメイン
#4	ホットスタンバイ

ローカルアクセスはすべて直接行われる必要があります。プロキシサーバーはすべて、ポート 8080 上で実行されます。+ 演算子を使用して、文字列を連結することができます。

```
function FindProxyForURL(url, host)
{
    if (isPlainhost name(host) || dnsDomainIs(host, ".mydomain.com"))
        return "DIRECT";

    else if (dnsDomainIs(host, ".com"))
        return "PROXY proxy1.mydomain.com:8080; " +
               "PROXY proxy4.mydomain.com:8080";
```

```
    else if (dnsDomainIs(host, ".edu"))
        return "PROXY proxy2.mydomain.com:8080; " +
               "PROXY proxy4.mydomain.com:8080";

    else
        return "PROXY proxy3.mydomain.com:8080; " +
               "PROXY proxy4.mydomain.com:8080";
}
```

例 6: shExpMatch()() によるプロキシの負荷分散

この例は、基本的には378 ページの「例 5: dnsDomainIs()() によるプロキシの負荷分散」と同じですが、dnsDomainIs()() を使用する代わりに、この例では shExpMatch()() が使用されています。

```
    function FindProxyForURL(url, host)
    {
    if (isPlainhost name(host) || dnsDomainIs(host, ".mydomain.com"))
        return "DIRECT";
    else if (shExpMatch(host, "*.com"))
        return "PROXY proxy1.mydomain.com:8080; " +
               "PROXY proxy4.mydomain.com:8080";
    else if (shExpMatch(host, "*.edu"))
        return "PROXY proxy2.mydomain.com:8080; " +
               "PROXY proxy4.mydomain.com:8080";
    else
        return "PROXY proxy3.mydomain.com:8080; " +
               "PROXY proxy4.mydomain.com:8080";
    }
```

例 7: 固有のプロトコルのプロキシ

固有のプロトコルに対するプロキシを設定できます。標準の JavaScript 機能のほとんどは、FindProxyForURL()() 関数で使用できるようになっています。たとえば、プロトコルに基づいて異なるプロキシを設定する場合は、substring()() 関数を使用できます。

```
    function FindProxyForURL(url, host)
    {
        if (url.substring(0, 5) == "http:") {
            return "PROXY http-proxy.mydomain.com:8080";
        }
        else if (url.substring(0, 4) == "ftp:") {
            return "PROXY ftp-proxy.mydomain.com:8080";
        }
        else if (url.substring(0, 7) == "gopher:") {
            return "PROXY gopher-proxy.mydomain.com:8080";
```

```
    }
    else if         (url.substring(0, 6) == "https:" ||
            url.substring(0, 6) == "snews:") {
        return "PROXY security-proxy.mydomain.com:8080";
    }
    else {
        return "DIRECT";
    }
}
```

この処理は、次の例のように、shExpMatch() () 関数を使用して実行することもできます。

```
...
if (shExpMatch(url, "http:*")) {
    return "PROXY http-proxy.mydomain.com:8080;
}
...
```

◆ ◆ ◆ 第 18 章

18 ACL ファイルの構文

アクセス制御リスト (Access Control List、ACL) ファイルは、Proxy Server のリソースにアクセスできるユーザーを定義したリストを含むテキストファイルです。デフォルトでは、Proxy Server はサーバーへのアクセスに関するすべてのリストを含む ACL ファイルを 1 つ使用します。また、`obj.conf` ファイルでは、複数の ACL ファイルを作成または参照できます。

Proxy Server 4 では、Proxy Server 3.x で使用されていたものとは異なる ACL ファイルの構文が使用されています。この付録では、ACL ファイルとその構文について説明します。Proxy Server とそのリソースへのアクセスの制御については、第 8 章を参照してください。第 16 章で説明しているように、Proxy Server 4 リリースではリソーステンプレートがサポートされています。

この付録は、次の節で構成されています。

ACL ファイルと ACL ファイルの構文について

すべての ACL ファイルは、特定の書式と構文に従って記述する必要があります。ACL ファイルは 1 つまたは複数の ACL を含むテキストファイルです。すべての ACL ファイルの先頭に、構文のバージョン番号を記述する必要があります。次に例を示します。

```
version 3.0;
```

バージョン行は、何行のコメント行の後にも指定できます。Proxy Server は、構文バージョン 3.0 を使用します。コメント行の先頭に # を付けて、ファイルにコメントを挿入することができます。

ファイル内の各 ACL は、タイプを定義する文で始まります。パス、リソース、または名前付きのいずれかです。

- パス ACL では、影響が及ぶリソースへの絶対パスを指定します。
- リソース ACL では、影響が及ぶテンプレートを指定します。たとえば `http://`、`https://`、`ftp://` などです。テンプレートについては、第 16 章を参照してください。
- 名前付き ACL では、 `obj.conf` ファイル内のリソースで参照される名前を指定します。サーバーには、すべてのユーザーに読み取りアクセスを許可し、LDAP ディレクトリのユーザーに書き込みアクセスを許可する、デフォルト名の付いたリソースが付属しています。Proxy Server のユーザーインタフェースから名前付き ACL を作成することはできますが、`obj.conf` ファイル内のリソースで、名前付き ACL を手動で参照する必要があります。

パス ACL とリソース ACL ではワイルドカードを使用することができます。ワイルドカードについては、第 16 章を参照してください。

タイプの行は `acl` という文字で始まり、タイプ情報は二重引用符で囲まれ、次にセミコロン (;) が続きます。次に例を示します。

```
acl "default";acl "http://*.*";
```

異なる ACL ファイルの間でも、すべての ACL の各タイプ情報に固有の名前を付ける必要があります。ACL のタイプを定義したあと、ACL で使用されるメソッド (認証文) と、アクセスを許可、または拒否されるユーザーやコンピュータ (承認文) を定義する 1 つまたは複数の文を記述することができます。次の節では、このような文の構文について説明します。

認証文

ACL では、必要に応じて、サーバーが ACL を処理するときに使用する必要のある認証方法を指定します。主に次の 3 つのメソッドがあります。

- 基本 (デフォルト)
- ダイジェスト
- SSL

基本メソッドとダイジェストメソッドでは、リソースにアクセスしようとしているユーザーに対してユーザー名とパスワードの入力を要求します。

SSL メソッドでは、ユーザーに対してクライアント証明書を持っていることを要求します。認証されるには、Proxy Server の暗号化を有効にする必要があり、信頼されている CA のリストにユーザーの証明書発行元が記載されている必要があります。

デフォルトでは、サーバーはメソッドを指定しない ACL に対して基本メソッドを使用します。サーバーの認証データベースは、ユーザーから送信されたダイジェスト認証をサポートする必要があります。

各認証行では、サーバーが認証を行う属性を指定する必要があります。ユーザー、グループ、またはその両方です。次に示す ACL タイプ行の後に表示される認証文では、データベースまたはディレクトリ内の各ユーザーと一致するユーザーの基本認証を指定します。

```
authenticate(user) { method = "basic";};
```

次の例では、ユーザーとグループの認証方法として SSL を使用します。

```
authenticate(user, group) { method = "ssl";};
```

次の例では、ユーザー名が sales という語で始まるユーザーを認証します。

```
allow (all) user = "sales*";
```

最後の行が group = sales に変更された場合、グループの属性が認証されないため、ACL がエラーになります。

承認文

各 ACL エントリには、1 つ以上の承認文を指定できます。承認文では、サーバーリソースへのアクセスを許可、または拒否するユーザーを指定します。

承認文の作成

承認文を作成する場合、次の構文を使用します。

```
allow|deny [absolute] (right[,right...]) attribute expression;
```

各行の先頭を allow または deny にします。規則は階層化されているため、通常の場合、最初の規則ですべてのユーザーに対してアクセスを拒否し、2 番目以降の規則で個別のユーザー、グループ、またはコンピュータに対してアクセスを許可してください。たとえば、すべてのユーザーに対して /my_files という名前のディレクトリへのアクセスを許可し、一部のユーザーのみに対してサブディレクトリ /my_files/personal へのアクセスを許可する場合、/my_files ディレクトリへのアクセスを許可されたユーザーは /my_files/personal ディレクトリへのアクセスも許可されるため、サブディレクトリに対するアクセス制御は機能しません。これを避けるには、すべてのユーザーのアクセスを拒否してからアクセスする必要のあるユーザーだけにアクセスを許可する規則をサブディレクトリに対して作成します。

ただし、デフォルトの ACL を設定してすべてのユーザーに対してアクセスを拒否した場合、ほかの ACL 規則では「deny all」規則は必要ありません。

次の行では、すべてのユーザーに対してアクセスを拒否します。

```
deny (all) user = "anyone";
```

承認文の階層

ACL には、リソースに応じて異なる階層があります。サーバーが特定のリソースに対する要求を受信した場合、そのリソースに適用する ACL のリストを構築します。サーバーはまず、check-acl 文の obj.conf ファイルにリスト表示する名前付き ACL を追加します。次に、一致するパス ACL とリソース ACL を追加します。このリストは同じ順序で処理されます。「無条件な」ACL 文がない場合は、すべての文が順序どおりに評価されます。「無条件に許可」の文または「無条件に拒否」の文が「true」かどうかを評価する場合、サーバーは処理を停止し、この結果の処理を受け取ります。

一致する ACL が複数ある場合、サーバーは一致する最後の文を使用します。ただし、無条件文を使用する場合は、ほかの一致する文の検索を停止し、無条件文のある ACL を使用します。同一のリソースに対する無条件文が 2 つある場合は、ファイル内の最初の文を使用し、一致するほかのリソースの検索を停止します。

```
version 3.0;acl "default";authenticate (user,group)
    { prompt="Sun Java System Web Proxy Server";};
    allow (read,execute,list,info) user = "anyone";
    allow (write,delete) user = "all";acl "http://*.*";
    deny (all) user = "anyone";allow (all) user = "joe";
```

属性式

属性式は、ユーザー名、グループ名、ホスト名、または IP アドレスに基づいて、アクセスを許可、または拒否するユーザーを定義します。次の行は、複数のユーザーまたはコンピュータに対してアクセスを許可する方法の例を示しています。

- user = "anyone"
- user = "smith*"
- group = "sales"
- dns = "*.mycorp.com"
- dns = "*.mycorp.com,*.company.com"
- ip = "198.*"
- ciphers = "rc4"
- ssl = "on"

また、timeofday 属性を使用すると、サーバーのローカル時間を基準にした時刻で、サーバーへのアクセスを制御できます。たとえば、timeofday 属性を使用すると、特定のユーザーによる特定の時間のアクセスを制御することができます。

時間を指定するには、24 時間形式を使用します。たとえば、午前 4 時を指定するには 0400、午後 10 時 30 分を指定するには 2230 と指定します。次の例では、guests というユーザーのグループの午前 8 時から午後 4 時 59 分までの間のアクセスを制御します。

```
allow (read) (group="guests") and (timeofday<0800 or timeofday=1700);
```

また、曜日によってアクセスを制御することもできます。3 文字の省略形を使用して曜日を指定します。つまり、Sun、Mon、Tue、Wed、Thu、Fri、Sat です。

次の文で、premium グループのユーザーは、曜日や時間に関係なくアクセスが許可されます。discount グループのユーザーは、週末 (土曜日と日曜日) は時間に制限なく、平日 (月曜日から金曜日まで) は午前 8 時から午後 4 時 59 分までを除く任意の時間にアクセスできます。

```
allow (read) (group="discount" and dayofweek="Sat,Sun") or (group="discount" and
(dayofweek="mon,tue,wed,thu,fri" and(timeofday<0800 or timeofday=1700)))or
(group="premium");
```

式の演算子

属性式では、各種の演算子を使用できます。括弧で演算子の優先度を示します。user、group、 dns、および ip では、次の演算子を使用できます。

- and
- or
- not
- = (等号)
- != (等しくない)

timeofday および dayofweek では、次の演算子を使用できます。

- より大きい
- < (より小さい)
- = (以上)
- <= (以下)

デフォルト ACL ファイル

インストールのあと、サーバーは *server_root*/httpacl/generated.proxy- *serverid*.acl ファイルに含まれているデフォルト設定を使用できるようになります。ユーザーインタフェースで設定が作成されるまで、サーバーは作業ファイル genwork.proxy-*serverid*.acl を使用します。ACL ファイルを編集する場合、genwork ファイルに対して変更を行い、Proxy Server を使用して変更を保存し、適用します。

汎用構文の項目

入力文字列には、次の文字を使用できます。

- a ～ z までの文字
- 0 ～ 9 までの文字

- ピリオド (.) と下線 (_)

その他の文字については、二重引用符で文字を囲む必要があります。

1 つの文は 1 行で表示し、末尾にセミコロンを付けます。複数の文は括弧で囲みます。項目のリストはコンマで区切り、二重引用符で囲む必要があります。

obj.conf ファイルでの ACL ファイルの参照

名前付き ACL ファイルまたは個別の ACL ファイルは、obj.conf ファイル内で参照することができます。このためには、PathCheck 指令で check-acl 関数を使用します。この行には、次の構文があります。

```
PathCheck fn="check-acl" acl="aclname "
```

ここで *aclname* は、ACL ファイルに表示される、ACL の固有の名前です。

たとえば、testacl という名前の付いた ACL を使用してディレクトリへのアクセスを制限する場合、obj.conf ファイルに次のような行を追加します。

```
<Object ppath="https://"PathCheck fn="check-acl" acl="testacl"</Object>
```

上の例では、1 行目が、アクセスを制御するサーバーリソースを示すオブジェクトです。2 行目はPathCheck 指令で、check-acl 関数を使用して名前付き ACL (testacl) を、指令が表示されるオブジェクトにバインドします。testacl ACL は、server.xml で参照される ACL ファイルに表示できます。

◆ ◆ ◆ 第 19 章

19 サーバーパフォーマンスの調整

Proxy Server 環境におけるパフォーマンスには、プロキシクライアント、Proxy Server、配信元サーバー、ネットワークなど、多数の要素が影響を与えます。この付録では、Proxy Server のパフォーマンスを向上させるために実行できる調整について説明します。

この付録は、上級管理者のみを対象としています。サーバーの調整を行う場合は細心の注意を払い、変更前に必ず設定ファイルをバックアップするようにしてください。

この付録は、次の節で構成されています。

パフォーマンスに関する一般的な注意事項

この節では、Proxy Server のパフォーマンスを分析するときに考慮する一般的な領域について説明します。

ここでは、次の内容について説明します。

アクセスログ

アクセスログを無効にすると、Proxy Server のパフォーマンスが向上する場合があります。ただし、Proxy Server にアクセスしているユーザーや、ユーザーの要求しているページに関する情報を確認できなくなるため、不便も生じます。

Proxy Server のアクセスログは、obj.conf ファイルで次の指令をコメントアウトすることで無効にできます。

```
Init fn=“flex-init” access=“$accesslog” format.access=“%Ses->client.ip% -
%Req->vars.auth-user% [%SYSDATE%] \\”%Req->reqpb.clf-request%\\“
%Req->srvhdrs.clf-status% %Req->srvhdrs.content-length%” ...AddLog
fn=“flex-log” name=“access”
```

ACL キャッシュの調整

デフォルトでは、Proxy Server によるユーザーとグループの認証の結果が、ACL ユーザーキャッシュに保存されます。magnus.conf ファイルの ACLCacheLifetime 指令を使用して、ACL ユーザーキャッシュを有効にする期間を制御することができます。キャッシュのエントリが参照されるたびにその経過時間が計算され、ACLCacheLifetime と照合されます。経過時間が ACLCacheLifetime と同じか、それよりも長い場合、このエントリは使用されません。

ACLCacheLifetime のデフォルト値は 120 秒です。これは、Proxy Server と LDAP サーバーの同期が 2 分間にわたってとられない可能性があることを意味しています。この値を 0 (ゼロ) に設定すると、キャッシュがオフになり、Proxy Server はユーザー認証のたびに LDAP サーバーに照会することになります。アクセス制御を実装する場合、この設定は Proxy Server のパフォーマンスに悪影響を及ぼします。ACLCacheLifetime に大きな値を設定した場合、LDAP エントリに変更を行うたびに Proxy Server を再起動する必要が生じる可能性もあります。この設定によって Proxy Server が LDAP サーバーに問い合わせるようにするためです。LDAP ディレクトリが頻繁に変更される可能性が低い場合にだけ、大きな値を設定します。

ACLUserCacheSize は、キャッシュ内に保持できるエントリの最大数を設定する `magnus.conf` のパラメータです。このデフォルト値は 200 です。新しいエントリがリストの先頭に追加され、キャッシュが最大サイズに達すると、新しいエントリが追加できるように、このリストの末尾のエントリが再利用されます。

また、*ACLGroupCacheSize* パラメータを使用して、ユーザーエントリごとにキャッシュできるグループメンバーシップの最大数を設定することもできます。デフォルト値は 4 です。ただし、グループのメンバーではないユーザーはキャッシュされないため、要求ごとに何回か LDAP ディレクトリにアクセスすることになります。

バッファーサイズ

サーバーのソケットの送信バッファー (SndBufSize) と受信バッファー (RcvBufSize) のサイズを指定できます。これらのパラメータは、`magnus.conf` ファイルで設定できます。推奨される値は UNIX や Linux オペレーティングシステムの種類によって異なります。これらのパラメータの適切な設定値については、オペレーティングシステムのマニュアルを参照してください。

接続タイムアウト

`magnus.conf` ファイル内の *AcceptTimeout* パラメータを使用すると、接続をクローズする前に、サーバーがクライアントからのデータの到着を待機する秒数を指定できます。タイムアウトの制限時間内にデータが届かない場合、接続がクローズされます。デフォルトでは、30 秒に設定されています。ほとんどの環境では、この設定を変更する必要はありません。この値をデフォルトよりも少なく設定してスレッドを解放することもできますが、接続に時間のかかるユーザーが切断されてしまう可能性もあります。

エラーログレベル

`server.xml()` ファイルの LOG タグにある loglevel 属性の値を大きくすると、サーバーがエラーログに生成して格納する情報量が多くなります。ただし、このファイルにエントリを書き込むときのパフォーマンスに影響を与えます。ログレベルを大きくするのは問題をデバッグするときだけにして、障害追跡モード以外の場合はログレベルを最小に設定します。

セキュリティー要件

SSL を有効にすると Proxy Server のプライバシーとセキュリティーが向上しますが、パケットの暗号化と復号化によってオーバーヘッドが発生するため、パフォーマンスに影響を与えます。ハードウェアアクセラレータカードに対する暗号化および復号化処理の負荷を低減させることを考慮してください。

Solaris ファイルシステムキャッシュ

Proxy Server キャッシュはランダムアクセスメモリには保存されません。キャッシュからドキュメントが取り出されるたびに、ファイルシステムに対してファイルアクセスが行われます。Solaris のファイルシステムキャッシュを使用して、Proxy Server キャッシュをメモリーにプリロードしてください。これによって、キャッシュされたファイルへの参照が、ファイルシステムではなくメモリから取り出されるようになります。

タイムアウト値

タイムアウトは、サーバーのパフォーマンスに大きく影響します。Proxy Server に対して最適なタイムアウトを設定することは、ネットワークリソースの節約に役立ちます。

2 つのインスタンス固有の SAF (サーバーアプリケーション関数) と、1 つのグローバルパラメータを使用して、Proxy Server 内のタイムアウト値を設定することができます。

- 390 ページの「`init-proxy()` SAF (`obj.conf` ファイル)」
- 391 ページの「`http-client-config()` SAF (`obj.conf` ファイル)」
- 392 ページの「`KeepAliveTimeout()` SAF (`magnus.conf` ファイル)」

`init-proxy()` SAF (`obj.conf` ファイル)

`init-proxy()` 関数は、Proxy Server の内部設定を初期化します。この関数は Proxy Server の初期化時に呼び出されますが、値が正しく初期化されるように obj.conf ファイル内で指定する必要もあります。

この関数の構文は次のとおりです。

```
Init fn=init-proxy timeout=seconds timeout-2=seconds
```

上の例では、次のパラメータを、init-proxy SAF の Proxy Server タイムアウト設定に直接適用することができます。

- timeout (プロキシタイムアウト)– プロキシタイムアウトパラメータは、アイドル接続を中断するまでのサーバーの待機時間を指示します。プロキシのタイムアウトを高く設定すると、時間のかかるプロキシスレッドによってクライアントが長時間停止させられる可能性があります。タイムアウト値を低く設定すると、結果が出るまでに時間のかかる CGI スクリプト (データベースクエリーゲートウェイなど) が中断されてしまいます。

  サーバーに対して最適なプロキシタイムアウトを決定するには、次の問題について検討します。

- Proxy Server は、多数のデータベースクエリーや CGI スクリプトを処理するか。
- Proxy Server の処理する要求数が少なく、常に 1 つのプロセスに時間を割くことができるか。

  上のどちらの質問に対する答えも「はい」である場合は、プロキシタイムアウトに高い値を設定することになります。プロキシタイムアウトの推奨される最高値は 1 時間です。デフォルト値は 300 秒 (5 分) です。

  サーバーマネージャーの「Preferences」タブにある「Configure System Preferences」ページにアクセスすると、プロキシタイムアウトの値を表示または変更できます。このパラメータは、「Proxy Timeout」として参照されます。

timeout-2 (中断後のタイムアウト)– 中断後のタイムアウト値は、クライアントがトランザクションを中断した後、キャッシュファイルに書き込みを続行する必要のある時間を Proxy Server に指示します。つまり、Proxy Server がドキュメントのキャッシュをほぼ終了していて、クライアントが接続を中断した場合、サーバーは、中断後のタイムアウト値に達するまでドキュメントのキャッシュを続行することができます。

推奨される中断後のタイムアウトの最高値は 5 分です。デフォルト値は 15 秒です。

`http-client-config()` SAF (`obj.conf` ファイル)

http-client-config 関数は、Proxy Server の HTTP クライアントを設定します。

この関数の構文は次のとおりです。

```
Init fn=http-client-config
  keep-alive=(true|false)
  keep-alive-timeout=seconds
  always-use-keep-alive=(true|false)
  protocol=HTTP Protocol
  proxy-agent="Proxy-agent HTTP request header"
```

設定値は次のように定義されます。

- keep-alive– (オプション) HTTP クライアントが持続的接続を使用するかどうかを示すブール値。デフォルトは true です。
- keep-alive-timeout– (オプション) 持続的接続を開いたままにしておく最大秒数。デフォルトは 29 です。
- always-use-keep-alive– (オプション) すべてのタイプの要求に対して HTTP クライアントが既存の持続的接続を再使用できるかどうかを示すブール値。デフォルトは false です。つまり、GET 以外の要求またはボディーを含む要求に対しては、持続的接続は再使用されません。

- protocol– (オプション) HTTP プロトコルバージョンの文字列。デフォルトでは、HTTP クライアントは、HTTP 要求のコンテンツによって HTTP/1.0 か HTTP/1.1 のどちらかを使用します。プロトコルの相互運用について特定の問題が発生しないかぎり、protocol パラメータは使用しないでください。
- proxy-agent– (オプション) Proxy-agent HTTP 要求ヘッダーの値。デフォルトは、Proxy Server の製品名とバージョンを含む文字列です。

`KeepAliveTimeout()` SAF (`magnus.conf` ファイル)

`KeepAliveTimeout()` パラメータは、サーバーが、クライアントと Proxy Server 間の HTTP keep-alive 接続または持続的接続を開いたままにしておく最大時間 (秒) を決定します。デフォルトは 30 秒です。アイドル時間が 30 秒を超えると、接続が切断されます。最大値は 300 秒 (5 分) です。

注意 – `magnus.conf` ファイル内のタイムアウト設定は、クライアントと Proxy Server 間の接続に適用されます。`obj.conf` ファイル内の `http-client-config` SAF にあるタイムアウト設定は、Proxy Server と配信元サーバー間の接続に適用されます。

最新状態チェック

Proxy Servers は、ドキュメントを配信元サーバーから取得するのではなく、ローカルキャッシュから提供することによって、パフォーマンスを向上させることができます。この方法の欠点の 1 つは、期限切れのドキュメントが提供される可能性があることです。

Proxy Server は、ドキュメントが最新のものかどうかをチェックして、ドキュメントが古い場合は、キャッシュを更新することができます。この最新状態チェックは、必要な場合のみ実行することをお勧めします。ドキュメントのチェックを頻繁に行うと、Proxy Server 全体のパフォーマンスが低下してしまう可能性があるからです。

最新状態チェックは、「Caching」タブの「Set Cache Specifics」で設定します。デフォルトは、2 時間おきに新しいドキュメントをチェックする設定になっています。この情報は、ObjectType 指令で max-uncheck パラメータを使用して設定します。

ドキュメントを最新に保ちつつ、サーバーのパフォーマンスを向上させるには、適切なドキュメントの期限とともに、last-modified 要素を特定することによって最新状態チェックをカスタマイズします。

last-modified 要素

last-modified 要素は、通知されてきたこれまでの変更に基づいて、今後のドキュメントの変更の可能性を判断するのに役立ちます。

last-modified 要素は、.02 ～ 1.0 の間の割合です。ドキュメントの実際の最後の変更から最後に最新状態チェックが実行された時間までの間隔に、この割合を掛け合わせます。この結果の数字を、最後の最新状態チェックからの経過時間と比較します。この数字が間隔時間より小さい場合、ドキュメントの期限は切れていません。この数字が間隔時間より大きい場合は、ドキュメントの期限が切れており、配信元サーバーから新しいバーションを取得します。

last-modified 要素によって、最近チェックされたドキュメントを、古いドキュメントよりも頻繁にチェックするように設定できます。

last-modified 要素は、0.1 ～ 0.2 の間に設定することをお勧めします。

DNS 設定

DNS は、標準の IP アドレスをホスト名に関連付けるために使用されるシステムです。このシステムは、重要な Proxy Server リソースが適切に設定されていない場合に、占有することができます。パフォーマンスを最適化するには、次の点に注意します。

- DNS キャッシュを有効にする。

  DNS キャッシュを有効にするには、サーバーマネージャーの「Preferences」タブにある「Configure DNS Cache」リンクを選択します。DNS キャッシュの「Enabled」ラジオボタンを選択します。
- クライアントの DNS 名をログに記録せず、IP アドレスだけを記録する。

  クライアントの DNS 名のログを無効にするには、サーバーマネージャーの「Server Status」タブにある「Set Access Log Preferences」リンクを選択します。クライアントのホスト名ではなく IP アドレスをログに記録するには、「IP Addresses」ラジオボタンを選択します。
- 逆引き DNS を無効にする。

  逆引き DNS は、IP アドレスをホスト名に変換します。逆引き DNSを無効にするには、サーバーマネージャーの「Preferences」タブにある「Configure System Preferences」リンクを選択します。逆引き DNS を無効にするには、「No」ラジオボタンを選択します。
- クライアントのホスト名に基づいたアクセス制御をしない

  アクセス制御文では、ホスト名ではなくクライアントの IP アドレスを使用します (可能な場合)。

スレッド数

magnus.conf ファイル内の RqThrottle パラメータによって、Proxy Server が同時に処理できるトランザクションの最大数を指定します。デフォルト値は 128 です。この値を変更することで、サーバー処理を低速にし、実行されるトランザクションの待ち時間を最小限に抑えることができます。

同時処理する要求数を計算するために、サーバーはアクティブな要求数をカウントし、そこに新しい要求が届いたら 1 を足し、要求が終了したら 1 を引きます。新しい要求が届いたら、サーバーは、要求の最大数がすでに処理済みであるかどうかを確認します。制限数に達してしまった場合は、アクティブな要求数が最大数を下回るまで、新しい要求の処理が延期されます。

同時処理されている要求数を監視するには、`perfdump` によって生成されるデータの SessionCreationInfo 部分か、`proxystats.xml` データを確認します。この情報から、同時処理する要求の最大数 (ピーク) を、スレッドの合計数 (制限) と比較して決定することができます。`perfdump` 出力で表示される情報は次のとおりです。

```
SessionCreationInfo:
-----------------------
Active Sessions          1
Keep-Alive Sessions      0
Total Sessions Created   48/128
```

Active Sessions は、現在要求を処理しているセッションの数 (要求処理のスレッド) を示しています。Keep-Alive Sessions は Active Sessions と似ていますが、クライア ントが keep-alive 接続を要求している場合に特定されています。Total Sessions Created は、作成されたセッション数と許可されるセッションの最大数の両方を示しています。これらは、RqThrottle 値の最小値と最大値です。

注意 – RqThrottleMin は、サーバーが起動時に開始するスレッドの最小数です。デフォルト値は 48 です。このパラメータは `magnus.conf` ファイルでも設定できますが、デフォルトでは表示されません。

設定されたスレッドの最大数に達してしまってもかまいません。RqThrottle 値を反射的に増やす必要もありません。この最大限度に達したということは、サーバーがピークロード時にこれだけの数のスレッドを必要としたことを意味しています。しかし、要求がタイムリーに処理されているかぎり、サーバーは適切に調整されているといえます。ただし、この時点で接続は接続キューに入れられるため、オーバーフローする可能性もあります。`perfdump` 出力を定期的にチェックし、作成されたセッションの合計数がしばしば RqThrottle の最大数に接近する場合は、スレッドの制限数を大きくすることを検討してください。

適切な RqThrottle の値の範囲は、負荷によって 100 ～ 500 になります。

インバウンド接続プール

インバウンド接続プールを調整するには、KeepAlive* と、magnus.conf 内にある次のような関連の設定を使用します。

- MaxKeepAliveConnections
- KeepAliveThreads
- KeepAliveTimeout
- KeepAliveQueryMaxSleepTime
- KeepAliveQueryMeanTime
- ConnQueueSize
- RqThrottle
- acceptorthreads

これらのパラメータについては、『Sun ONE Web Server 6.1 Performance Tuning, Sizing, and Scaling Guide』の第 2 章を参照してください。このマニュアルは次の Web サイトから入手できます。

http://docs.sun.com/app/docs/doc/819-6516/

今回の Proxy Server のリリースでは、アウトバウンド接続プールの設定はできません。

FTP リストの幅

FTP リストの幅を広げると、長いファイル名が表示できるようになるため、ファイル名の切り詰めが少なくなります。デフォルトの幅は、80 文字です。

FTP リストの幅を変更するには、サーバーマネージャーの「Preferences」タブにある「Tune Proxy」リンクを選択します。

キャッシュアーキテクチャー

キャッシュをうまく設定することで、サーバーのパフォーマンスを向上させることができます。キャッシュを設計する場合の留意事項は次のとおりです。

- 負荷を分散させる
- 複数のプロキシキャッシュパーティションを使用する
- 複数のディスクドライブを使用する
- 複数のディスクコントローラを使用する

Proxy Server のパフォーマンスにとって、適切なキャッシュの設定は非常に重要です。プロキシキャッシュを設計する場合に忘れてはならない最も重要なルールは、負荷を分散させることです。キャッシュは、パーティションあたり約 1G バイトに設

定し、複数のディスクおよび複数のディスクコントローラ間に分散させる必要があります。こうした調整によって、単一の大きなサイズのキャッシュを使用するよりも、ファイルの作成や取得が高速になります。

キャッシュのバッチ更新

キャッシュバッチ更新機能によって、指定された Web サイトからファイルをプリロードしたり、すでにキャッシュ内にあるドキュメントに対して最新状態チェックを実行したりできます。これは通常、Proxy Server 上の負荷が最も低い状態にあるときに開始されます。「Cache Batch Updates」ページから、URL のバッチを作成、編集、削除することができ、バッチの更新を有効または無効にすることができます。

バッチ更新を実行するファイルを指定することによって、オンデマンドキャッシュとは異なり、コンテンツをアクティブにキャッシュできます。Proxy Server では、現在キャッシュ内にある複数のキャッシュに対して最新状態チェックを実行したり、特定の Web サイト内の複数のファイルをプリロードしたりできます。

サーバーとプロキシによるネットワークを含む大規模なサイトでは、バッチ更新機能を使用して、指定された Web 領域をプリロードすることもできます。バッチ処理では、ドキュメント内のリンクを再帰的に下降し、コンテンツをローカルにキャッシュします。この機能はリモートサーバー上では負荷が重くなる可能性があるので、注意が必要です。再帰が無制限に実行されないような方法が取られ、`bu.conf` 設定ファイル内のパラメータによって、このプロセスの一部を制御できます。

Proxy Server のアクセスログを使用して、一般に最もアクティブなサイトを判別し、これらのサイトに対してバッチ更新を実行してパフォーマンスを向上させます。

ガベージコレクション

ガベージコレクションは、Proxy Server のキャッシュを調査して、古い無効なファイルを削除するプロセスです。ガベージコレクションはリソース集中型のプロセスなので、パフォーマンスを向上させるために、一部のガベージコレクション設定を調整する必要がある場合もあります。

次のパラメータを使用すると、ガベージコレクションのプロセスを微調整することができます。これらのパラメータは、「Tune Garbage Collection」フォーム上で表示または変更することができます。このフォームは、サーバーマネージャーの「Caching」タブにある「Tune GC」を選択すると表示されます。パラメータは次のとおりです。

- *gc hi margin percent*
- *gc lo margin percent*

- *gc extra margin percent*
- *gc leave fs full percent*

gc hi margin percent 変数

gc hi margin percent 変数は、最大キャッシュサイズの割合を制御します。その割合に達するとガベージコレクションが開始されます。

この値は gc lo margin percent の値よりも高く設定する必要があります。

gc hi margin percent の有効な値の範囲は、10 〜 100 % です。デフォルト値は 80 % です。キャッシュが 80 % になるとガベージコレクションが開始されます。

gc lo margin percent 変数

gc lo margin percent 変数は、最大キャッシュサイズの割合を制御します。その割合に達するとガベージコレクションが開始されます。

この値は gc hi margin percent の値よりも低く設定する必要があります。

gc lo margin percent の有効な値の範囲は、5 〜 100 % です。デフォルト値は 70 % です。ガベージコレクション後にキャッシュが 70 % になった状態をターゲットとします。

gc extra margin percent 変数

パーティションのサイズが最大許容値 (gc hi margin percent) に接近したという理由以外でガベージコレクションが開始された場合、ガベージコレクタは gc extra margin percent 変数によって設定された割合を使用して、削除するキャッシュの割合を決定します。

gc extra margin percent の有効な値の範囲は、0 〜 100 % です。デフォルト値は 30 % です。既存のキャッシュファイルの 30 % を削除します。

gc leave fs full percent 変数

gc leave fs full percent の値は、ガベージコレクションが実行されない範囲のキャッシュパーティションサイズの割合を決定します。この値によって、ほかのアプリケーションがディスク領域を占有していても、ガベージコレクタがキャッシュからすべてのファイルを削除しないように設定できます。

gc leave fs full percent の有効な値の範囲は、0 (完全に削除できる) 〜 100 % (何も削除しない) です。デフォルト値は 60 % です。現在の 60 % までキャッシュサイズを小さくできます。

Solaris のパフォーマンス調整

Solaris カーネルでは、さまざまなパラメータを使用して Proxy Server のパフォーマンスを微調整できます。次の表は、これらのパラメータの一部を示しています。

表 19–1 Solaris のパフォーマンス調整パラメータ

パラメータ	スコープ	デフォルト値	調整値	コメント
rlim_fd_max	/etc/system	1024	8192	オープンファイル記述子の制限を処理します。予想される負荷 (存在する場合は、関連付けられたソケット、ファイル、パイプなど) を考慮してください。
rlim_fd_cur	/etc/system	64	8192	
sq_max_size	/etc/system	2	0	ストリームドライバのキューサイズを制限します。0 に設定すると、パフォーマンスはバッファー領域の不足による影響を受けなくなります。このパラメータはクライアント上でも設定します。
tcp_close_wait_interval	ndd/dev/tcp	240000	60000	このパラメータはクライアント上でも設定します。
tcp_conn_req_max_q	ndd/dev/tcp	128	1024	
tcp_conn_req_max_q0	ndd/dev/tcp	1024	4096	
tcp_ip_abort_interval	ndd/dev/tcp	480000	60000	
tcp_keepalive_interval	ndd/dev/tcp	7200000	900000	アクセスの多い Web サイトでは、この値を低く設定します。
tcp_rexmit_interval_initial	ndd/dev/tcp	3000	3000	再転送率が 30 〜 40% を超える場合、この値を大きくします。
tcp_rexmit_interval_max	ndd/dev/tcp	240000	10000	
tcp_rexmit_interval_min	ndd/dev/tcp	200	3000	

表 19–1 Solarisのパフォーマンス調整パラメータ (続き)

パラメータ	スコープ	デフォルト値	調整値	コメント
tcp_smallest_anon_port	ndd/dev/tcp	32768	1024	このパラメータはクライアント上でも設定します。
tcp_slow_start_initial	ndd/dev/tcp	1	2	データが少量であればやや高速に転送します。
tcp_xmit_hiwat	ndd/dev/tcp	8129	32768	送信バッファーを増やすために使用します。
tcp_recv_hiwat	ndd/dev/tcp	8129	32768	受信バッファーを増やすために使用します。

これらのパラメータについては、『Sun ONE Web Server 6.1 Performance Tuning, Sizing, and Scaling Guide』の第5章を参照してください。このマニュアルは次のWebサイトから入手できます。

http://docs.sun.com/app/docs/doc/819-6516/

索引

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

あ

き

く

け

こ

さ

そ

た

ち

つ

て