管理者ガイド

iPlanet™ Messaging Server

Release 5.2

816-5017-01
2002 年 2 月

目 次

表目次

図目次

このマニュアルについて

本書では、iPlanet Messaging Server の管理および構成方法について説明します。iPlanet Messaging Server は、オープンインターネット規格を使用するさまざまな規模の企業およびメッセージングホストの電子メールに関するニーズに応え、強力で柔軟なクロスプラットフォーム対応のソリューションを提供します。

この章には、以下の節があります。

- 対象読者
- 予備知識
- 本書の構成
- マニュアルの表記規則
- 関連情報

対象読者

このマニュアルは、自分のサイトで iPlanet Messaging Server を管理および構成する担当者を対象としています。

予備知識

このマニュアルは、読者に以下の予備知識があることを前提としています。

- インターネットおよび WWW (ワールドワイドウェブ)
- iPlanet Administration Server
- Netscape Directory Server および LDAP
- 電子メールとその概念
- Netscape Console

本書の構成

本書には、次の章および付録が含まれています。

- このマニュアルについて (本章)
- 第 1 章「はじめに」

  この章では、iPlanet Messaging Servr の高度な概要を提供します。
- 第 2 章「一般的なメッセージング機能を設定する」

  この章では、サービスの開始と停止、およびディレクトリアクセスの構成など、Messaging Server の全般的なタスクについて説明します。
- 第 3 章「POP、IMAP、および HTTP サービスの設定」

  この章では、iPlanet Console またはコマンドラインユーティリティを使って 1 つ以上のサービスをサポートするように構成する方法について説明します。
- 第 4 章「マルチプレクササービスを設定および管理する」

  この章では、複数の Messaging Server の単一接続ポイントとして機能する特別な Messaging Server である、iPlanet Messaging Multiplexor および iPlanet Messenger Express Multiplexor の概念について説明します。
- 第 5 章「MTA の概念」

  この章では、MTA の概念について説明します。
- 第 6 章「MTA サービスと設定について」

  この章では、サーバでの MTA サービスの構成についての全般的な情報を提供します。
- 第 7 章「書き換え規則を設定する」

  この章では、MTA 設定ファイル `imta.cnf` での書き換え規則 (アドレスの書き換え) を構成する方法について説明します。
- 第 8 章「チャネル定義を設定する」

  この章では、MTA 設定ファイル `imta.cnf` でのチャネル定義の構成方法について説明します。
- 第 9 章「定義済みチャネルを使用する」

  この章では、保留チャネルや変換チャネルなど、定義済みの MTA チャネル定義の使用方法を説明します。
- 第 10 章「メールのフィルタリングとアクセス制御」

  この章では、メールサービスへのアクセスの制御方法、およびマッピングテーブルやサーバ側規則 (SSR) を使ったメールのフィルタリング方法について説明します。

- 第 11 章「メッセージストアを管理する」

  この章では、メッセージストアディレクトリのレイアウト、メッセージストアパーティションの構成方法、制限容量の設定、存続期間決定ポリシーの設定などについて説明します。

- 第 12 章「セキュリティとアクセス制御を設定する」

  この章では、iPlanet Messaging Server で利用できるセキュリティおよびアクセス制御の機能について説明します。

- 第 13 章「ログ記録とログ解析」

  この付録では、MTA のサービスログおよびメッセージストアとメッセージアクセスサービスのサービスログを表示および構成する方法を説明します。

- 第 14 章「MTA のトラブルシューティング」

  この章では、MTA (Message Transfer Agent) のトラブルシューティングのための一般的なツール、方法、手順について説明します。

- 第 15 章「iPlanet Messaging Server をモニタする」

  この章では、iPlanet Message Server のモニタ機能について説明します。

- 付録 A「SNMP サポート」

  この章では、Messaging Server の SNMP サポートを有効にする方法について説明します。また、SNMP から得られる情報の種類についても簡単に説明します。

- 付録 B「MTA ダイレクト LDAP 操作」

  この付録では、MTA のダイレクト LDAP の動作について説明します。

- 付録 C「iPlanet Messaging Server の Event Notification Service を管理する」

  この付録では、iPlanet Messaging Server で iPlanet Event Notification Service を有効にして管理するために必要な事項について説明します。

- 付録 D「メールユーザとメーリングリストを管理する」

  この付録では、Console インタフェースを使ってユーザのメールアカウントとメーリングリストを作成および管理する方法について説明します。

- 用語集

  用語集では、用語および命名規則について定義しています。

マニュアルの表記規則

表 1　　マニュアルの表記規則

フォント	意味	例
AaBbCc123	コマンド名、ファイル名、コード、ディレクトリ名、ホスト名、識別名、およびコンピュータ画面に表示されるテキストを示します。	msg.conf ファイルを編集します。すべてのファイルを一覧表示するには、ls -a を使用します。 Error: illegal port #
AaBbCc123	ユーザが入力するテキストを示します。	% **cd madonna**
the_variable	コマンドラインのプレースホルダまたは変数を示します。実際の名前または値で置き換えます。	# *Instance_Root*/**start-msg**

コマンドラインプロンプト

このマニュアルの例では、概して、コマンドラインプロンプト (C シェルの %、Korn シェルの $ など) は示していません。お使いのオペレーティングシステムによって、コマンドラインプロンプトが異なるためです。コマンドは、プロンプトとは無関係に、本書で示されているとおりに入力してください。

関連情報

iPlanet Messaging Server には、以下の補足情報も用意されています。

`http://docs.iplanet.com/docs/manuals/messaging.html`

利用できる関連マニュアルは次のとおりです。

- 『iPlanet Messaging Server 管理者ガイド』
- 『iPlanet Messaging Server インストールガイド』
- 『iPlanet Messaging Server リファレンスマニュアル』
- 『iPlanet Messaging Server Schema Reference』
- 『iPlanet Messaging Server プロビジョニングガイド』
- 『iPlanet Delegated Administrator for Messaging and Collaboration インストールおよび管理ガイド』

第 1 章

はじめに

iPlanet Messaging Server は、企業とサービスプロバイダの両方で要求される大容量で信頼性の高いメッセージング処理のために設計された、強力な標準ベースのインターネットメッセージングサーバです。サーバはモジュール化された、個別に構成可能な複数のコンポーネントから成ります。これらのコンポーネントは、さまざまな標準ベースの電子メールプロトコルをサポートしています。

Messaging Server は、ユーザ、グループ、およびドメインについての情報を格納するために一元化された LDAP データベースを使用します。サーバ構成の情報には、LDAP データベースに格納されるものと、設定ファイルのセットに格納されるものがあります。

Messaging Server 製品群には、ユーザのプロビジョニングやサーバの構成をサポートするツールが含まれています。

この章には、以下の節があります。

- 標準プロトコルのサポート
- ホストドメインのサポート
- ユーザのプロビジョニングのサポート
- 統一メッセージングのサポート
- Web メールのサポート
- 強力なセキュリティとアクセス制御
- 使いやすいユーザインタフェース
- インストール後のディレクトリとファイルの編成

標準プロトコルのサポート

iPlanet Messaging Server は、電子メッセージングに関連するほとんどの国内規格、国際規格、および業界規格をサポートしています。完全なリストは、『iPlanet Messaging Server リファレンスマニュアル』の付録 A を参照してください。

ホストドメインのサポート

Messaging Server は、ISP にアウトソースされた電子メールドメインのようなホストドメインを完全にサポートしています。つまり、ISP は組織の電子メールサービスをリモートで操作および管理することにより組織をホスティングする電子メールドメインを提供します。ホストドメインは、ほかのホストドメインと同じ Messaging Server ホストを共有することができます。初期の LDAP ベースの電子メールシステムでは、1 つのドメインが 1 つまたは複数の電子メールサーバホストによってサポートされていました。Messaging Server では、複数のドメインを単一のサーバ上でホストできます。各ホストドメインには、そのドメインのユーザとグループのコンテナを指し、さまざまなドメイン固有のデフォルト設定を提供する LDAP エントリがあります。

ユーザのプロビジョニングのサポート

Messaging Server は、ユーザ、グループ、およびドメインについての情報を格納するために一元化された LDAP データベースを使用します。iPlanet Delegated Administrator for Messaging 製品には、組織内のユーザ、グループ、およびドメインを管理するために、Console グラフィカルユーザインタフェースとコマンドラインユーティリティが用意されています。

ユーザ、グループ、およびドメインのメッセージングの詳細は、以下のマニュアルを参照してください。

- 『iPlanet Messaging Server プロビジョニングガイド』- LDAP を使ってドメイン、ユーザ、グループ、または管理者のエントリを作成する方法を説明しています。
- 『iPlanet Messaging Server Schema Reference Manual』- iPlanet Messaging Server のスキーマについて説明しています。
- 『iPlanet Messaging Server リファレンスマニュアル』- ユーザ、グループ、およびドメインを管理するための Delegated Administrator コマンドラインユーティリティについて説明しています。

- iPlanet Messaging Server Delegated Administrator Console オンラインヘルプ

注 Console インタフェースを使用してユーザやグループを作成することもできますが、Delegated Administrator でこれらのエントリの表示や修正ができなくなるため、この方法はできるかぎり避けてください。

統一メッセージングのサポート

iPlanet Messaging Server では、電子メール、ボイスメール、FAX、およびその他の通信形態に関して単一のメッセージストアを使用するという、完全な統一メッセージングソリューションの基盤を提供します。

Web メールのサポート

iPlanet Messaging Server には、Messnger Express という Web で使用する電子メールプログラムが含まれており、エンドユーザは HTTP でインターネットに接続されているコンピュータシステム上で動作しているブラウザを使って自分のメールボックスにアクセスすることができます。Messenger Express クライアントは、iPlanet Messaging Server の一部である特殊な Web サーバにメールを送信します。HTTP サービスは、ルーティングまたは配信のために、そのメッセージをローカルの MTA またはリモートの MTA に送信します。

強力なセキュリティとアクセス制御

iPlanet Messaging Server には、次のセキュリティとアクセス制御の機能があります。

- POP、IMAP、HTTP、または SMTP へのパスワードによるログインおよび証明書に基づくログインのサポート
- 標準セキュリティプロトコル、TLS (Transport Layer Security)、SSL (Secure Socket Layer)、および SASL (Simple Authentication and Security Layer) のサポート
- ACI (Access Control Instruction) による委任管理
- POP、IMAP、SMTP および HTTP へのクライアントアクセスのフィルタリング
- システム全体およびユーザごとのサーバ側規則による不特定多数宛てのメールのフィルタリング

使いやすいユーザインタフェース

Messaging Server はモジュール化された、個別に構成可能な複数のコンポーネントから成ります。これらのコンポーネントは、電子メールの転送とアクセスプロトコルをサポートしています。

MTA (Message Transfer Agent) を構成するために、Messaging Server には設定ファイルの完全なセットとコマンドラインユーティリティのセットが用意されています。設定ファイルのセットはサーバにローカルに格納されています。また、メッセージストアおよびメッセージアクセスサービスを構成するために、Console グラフィカルユーザインタフェースとコマンドラインユーティリティの完全なセットが用意されています。

MTA および MTA へのアクセスの構成方法については、このマニュアルの次の章を参照してください。

- 第 5 章「MTA の概念」
- 第 6 章「MTA サービスと設定について」
- 第 7 章「書き換え規則を設定する」
- 第 8 章「チャネル定義を設定する」
- 第 9 章「定義済みチャネルを使用する」
- 第 10 章「メールのフィルタリングとアクセス制御」
- 第 12 章「セキュリティとアクセス制御を設定する」
- 第 14 章「MTA のトラブルシューティング」
- 第 15 章「iPlanet Messaging Server をモニタする」

『iPlanet Messaging Server リファレンスマニュアル』も参照してください。

メッセージストアとストアへのアクセスの構成方法については、このマニュアルの次の章を参照してください。

- 第 3 章「POP、IMAP、および HTTP サービスの設定」
- 第 11 章「メッセージストアを管理する」
- 第 12 章「セキュリティとアクセス制御を設定する」

『iPlanet Messaging Server リファレンスマニュアル』も参照してください。

さらに、このマニュアルの次の章も確認してください。

- 第 2 章「一般的なメッセージング機能を設定する」。サービスの開始と停止、およびディレクトリアクセスの構成など、Messaging Server の全般的なタスクについて説明しています。

- 第 4 章「マルチプレクササービスを設定および管理する」。複数の Messaging Server の単一接続ポイントとして機能する特別な Messaging Server である iPlanet Messaging Multiplexor (MMP) について説明しています。

インストール後のディレクトリとファイルの編成

iPlanet Messaging Server をインストールすると、そのディレクトリとファイルは表 1-1 に示すように編成されます。この表はすべてを網羅したものではありません。典型的なサーバ管理タスクに関連の深いディレクトリとファイルのみを示しています。

表 1-1 インストール後のディレクトリとファイル

ディレクトリ	デフォルトの場所と説明
サーバルートディレクトリ : (*server_root*)	`/usr/iplanet/server5/` (デフォルトの場所)
	指定したサーバグループのすべてのサーバ (指定した Administration Server で管理するすべてのサーバ) がインストールされているディレクトリ。この中には、Messaging Server 以外にほかの iPlanet サーバが含まれることもある
	このディレクトリには、Administration Server の開始と停止 (`start-admin`、`stop-admin`)、Console の起動 (`startconsole`) などを行うためのバイナリ実行ファイルも含まれる
インスタンスディレクトリ (*instance_root* または *instance_directory*)	*server_root*/msg-*instance_name*/ (必要な場所)
	instance_name は、インストール時に指定した Messaging Server のこのインスタンスの名前 (デフォルト = サーバマシンのホスト名)
	このディレクトリには、Messaging Server の特定のインスタンスを定義する設定ファイルが含まれる。特定のホストマシンに、同じバイナリファイルを使用する Messaging Server インスタンスが複数存在する場合がある
	このディレクトリには、`configutil`、`start-msg`、`stop-msg` など、インストールした Messaging Server の実行ファイルの一部が含まれることもある
インストールディレクトリ (*installDirectory*)	*server_root*`/bin/msg/` (必要な場所)
	このディレクトリには、インストールした Messaging Server のバイナリ実行ファイルの一部が含まれる

表 1-1　インストール後のディレクトリとファイル (続き)

ディレクトリ	デフォルトの場所と説明
設定ディレクトリ config	*instance_root*/config/ (必要な場所)
	local.conf、msg.conf、sslpassword.conf などの一般的な設定ファイルが含まれる
	msg.conf ファイルの値はインストール時に設定される。Messaging Server はこのファイルを使って、LDAP ホスト名やポート番号など、起動時に必要な情報を取得する
MTA ディレクトリ imta	*instance_root*/imta/ (必要な場所)
	MTA の構成に関連する次のディレクトリが含まれる。bin、config、db、dl、programs、queue、tmp
MTA 構成ディレクトリ config	*instance_root*/imta/config/ (必要な場所)
	imta.cnf、dispatcher.cnf、job_controller.cnf、aliases、imta_tailor などの MTA 設定ファイルが含まれる
MTA 構成ディレクトリ queue	*instance_root*/imta/queue/ (必要な場所)
	メッセージキューサブディレクトリが含まれる。各チャネルキューのこのディレクトリの下には、次のようなサブディレクトリが 1 つある。ims-ms、tcp_intranet、tcp_local、autoreply
MTA プログラムディレクトリ programs	*instance_root*/imta/programs/ (必要な場所)
	ユーザのメールを処理するサイト提供実行可能プログラムがある場合は、このディレクトリに含まれる
MTA データベースディレクトリ db	*instance_root*/imta/db/ (必要な場所)
	MTA が使用する次のデータベースが含まれる。aliasesdb.db、domaindb.db、profiledb.db、reversedb.db、ssrdb.db など
メッセージストアディレクトリ store	*instance_root*/store (必要な場所)
	メッセージストアに関連する次のディレクトリが含まれる。mboxlist、partition、user
	詳細は、345 ページの「メッセージストアのディレクトリレイアウト」を参照してください。

表 1-1　インストール後のディレクトリとファイル (続き)

ディレクトリ	デフォルトの場所と説明
マニュアルのディレクトリ `manual`	*server_root*/`manual` (必要な場所) サーバと一緒にインストールされたマニュアルがあるディレクトリ `manual/en/admin/` には Administration Server のマニュアルがある `manual/en/msg/` には Messaging Server のマニュアルがある `manual/en/slapd/` には Directory Server のマニュアルがある

第 2 章

一般的なメッセージング機能を設定する

この章では、サービスの起動と停止、ディレクトリアクセスの設定など、Netscape Console (以下、省略して Console という) またはコマンドラインユーティリティを使って実行できる Messaging Server の一般的なタスクについて説明します。個々の Messaging Server サービス (POP、IMAP、HTTP、および SMTP など) に固有なタスクについては、あとの章で説明します。この章には、以下の節があります。

注 エンドユーザアカウント情報およびドメイン固有の情報は、主に Delegated Administrator for Messaging のインタフェースを使用して管理します。詳細については、Delegated Administrator の『Delegated Administrator for Messaging インストールおよび管理ガイド』およびオンラインヘルプを参照してください。

メールユーザとメーリングリストを管理する

すべてのユーザおよびメーリングリストの情報は、LDAP ディレクトリ内のエントリとして保存されています。LDAP ディレクトリには、従業員、顧客、または組織に何らかのかかわりを持つその他の人々に関する詳細な情報を保存しておくことができます。これらの人々は、組織のユーザとして扱われます。

LDAP ディレクトリ内のユーザ情報は、各ユーザエントリのさまざまな属性に基づいて効率的に検索できるようになっています。ユーザエントリに関連付けられている属性には、氏名やその他の ID、部署、職名、勤務地、マネージャ名、直属の上司名、組織内の各部へのアクセス権限、およびその他の詳細設定があります。

組織内に電子メッセージングサービスがある場合は、大部分またはすべてのユーザがメールアカウントを持っているはずです。iPlanet Messaging Server の場合、メールアカウント情報はサーバにローカルには保存されません。これは、LDAP ユーザディレクトリの一部です。各メールアカウントの情報は、ディレクトリ内のユーザのエントリに付加されたメール属性として保存されます。

メールユーザとメーリングリストの作成と管理は、ディレクトリ内のユーザおよびメーリングリストのエントリを作成および変更することによって行います。これは、iPlanet Delegated Administrator for Messaging、Delegated Administrator コマンドラインユーティリティを使用するか、LDAP ディレクトリを直接変更することによって行います。また、ユーザおよびメーリングリストのエントリは Console を使って作成することもできますが、この方法はできるかぎり避けてください (付録 D を参照)。

Delegated Administrator for Messaging は、ユーザ、グループ、ファミリーグループ、およびホストドメインの管理を完全にサポートしています。Delegated Administrator では、ユーザやグループの管理を委任したり、ホストドメインごとに管理者を設定することができます。また、Delegated Administrator は GUI インタフェースを備えているため、管理者がユーザやグループを管理したり、エンドユーザが自分のメールアカウントを管理する場合に便利です。さらに、管理者はユーザやグループの管理に Delegated Administrator コマンドラインユーティリティを使用することもできます (『iPlanet Messaging Server リファレンスマニュアル』を参照)。Delegated Administrator の詳細については、『Delegated Administrator インストールおよび管理ガイド』および Delegated Administrator のオンラインヘルプを参照してください。また、LDAP ツールを使用してユーザ、グループ、およびドメインを管理する方法については、『iPlanet Messaging Server プロビジョニングガイド』を参照してください。

サーバの基本情報を表示するには

インストールした Messaging Server に関する基本情報を確認するには、Console を使って情報フォームを表示します。

注	iPlanet Directory Server 5.1 をインストールしている場合は、同時にインストールした iPlanet Console 5.0 から管理します。iPlanet Messaging Server 5.2 は、同時にインストールした Netscape Console 4.2 から管理します。

情報フォームを表示するには、次の手順に従います。

1. Console で、情報を表示する Messaging Server を開きます。
2. 左側のペインにあるサーバのアイコンを選択します。
3. 左側のペインの「構成」タブをクリックします。
4. 右側のペインの「情報」タブをクリックします。

   情報フォームが表示されます。このフォームには、サーバ名、サーバのルートディレクトリ、インストールディレクトリ、およびインスタンスディレクトリが表示されます。

サービスを起動および停止する

サービスを起動または停止する方法は、そのサービスが HA 環境にインストールされているかどうかによって異なります。

HA 環境でサービスを起動および停止するには

Messaging Server を HA 制御下で実行している場合は、個々の Messaging Server サービスを制御するための通常の Messaging Server コマンド (起動、再起動、停止) を使用することはできません。これらのコマンドを使うと、HA 制御により 1 つ以上のサービスが予期しない状況で停止したとみなされ、すべての Messaging Server の再起動が試みられるか、またはほかのクラスタノードへのフェイルオーバーが行われます。

以下の表に、適切な起動、停止、再起動のコマンドを示します。単一の Messaging Server サービス (たとえば、SMTP) を起動、再起動、停止するための Sun Cluster コマンドはないことに注意してください。Sun Cluster の最小単位は、個々のリソースです。Messaging Server は Sun Cluster でリソースとして認識されるため、`scswitch` コマンドがすべての Messaging Server サービスに影響を及ぼします。

表 2-1 Sun Cluster 3.0 環境での起動、停止、再起動

動作	個々のリソース	リソースグループ全体
起動	`scswitch -e -j` *resource*	`sscswitch -Z -g` *resource_group*
再起動	`scswitch -n -j` *resource* `scswitch -e -j` *resource*	`scswitch -R -g` *resource_group*
停止	`scswitch -n -j` *resource*	`scswitch -F -g` *resource_group*

表 2-2 Sun Cluster 2.2 環境での起動、停止、再起動

動作	個々のデータサービス	すべての登録済みデータサービス
起動	`hareg -y` *data_service*	`hareg -Y`
再起動	`hareg -n` *data_service* `hareg -y` *data_service*	`hareg -N` `hareg -Y`
停止	`hareg -n` *data_service*	`hareg -N`

表 2-3 Veritas 1.1 環境での起動、停止、再起動

動作	個々のリソース	リソースグループ全体
起動	`hares -online` *resource* `-sys` *system*	`hagrp -online` *group* `-sys` *system*
再起動	`hares -offline` *resource* `-sys` *system* `hares -online` *resource* `-sys` *system*	`hagrp -offline` *group* `-sys` *system* `hagrp -online` *group* `-sys` *system*
停止	`hares -offline` *resource* `-sys` *system*	`hagrp -offline` *group* `-sys` *system*

HA 環境以外でサービスを起動および停止するには

サービスは、Console またはコマンドラインを使って起動および停止できます。

必要な操作は、サーバが実際に使用しているサービスを実行するだけです。たとえば、MTA (Message Transfer Agent) として、一時的に特定の Messaging Server インスタンスを 1 つだけ使用している場合は、MTA だけを起動できます。また、メンテナンス、修復、セキュリティ上の必要からサーバをシャットダウンしなければならない場合は、影響が及ぶサービスだけを停止できます。実行する予定のないサービスは、停止するのではなく無効にしてください。

注	POP、IMAP、および HTTP の各サービスを起動または停止するには、まずそれらを使用可能な状態にする必要があります。詳細は、54 ページの「サービスの有効化と無効化」を参照してください。

重要 : サーバプロセスがクラッシュすると、ほかのプロセスがハングする可能性があります。これは、それらのプロセスがクラッシュしたサーバプロセスによって保持されていたロックを待機しているためです。したがって、サーバプロセスがクラッシュした場合は、すべてのプロセスを停止し、再起動するようにしてください。これには、POP、IMAP、HTTP、MTA の各プロセス、`stored` (メッセージストア) プロセス、およびメッセージストアを変更するすべてのユーティリティが含まれます。このユーティリティには、`mboxutil`、`deliver`、`reconstruct`、`readership`、`upgrade` などがあります。

コンソール : Console には、個々のサービスを起動または停止したり、各サービスに関するステータス情報を表示するためのフォームがあります。

フォームには、IMAP、POP、SMTP、および HTTP の各サービスに対し、現在の状態 (オンまたはオフ) が表示されます。また、サービスが実行中である場合には、そのサービスが最後に起動した時刻やほかのステータス情報も表示されます。

メッセージングサービスを起動またはシャットダウンしたり、そのステータスを表示するには、次の手順に従います。

1. Console で、サービスを起動または停止する Messaging Server を開きます。
2. 次のいずれかの方法で、「サービスの一般構成」フォームを表示します。
   a. 「タスク」タブをクリックし、「サービスの起動 / 停止」をクリックします。
   b. 「構成」タブをクリックし、左側のペインの「サービス」フォルダ選択します。次に、右側のペインで「一般」タブをクリックします。
3. 「サービスの一般構成」フォームが表示されます。

「プロセスコントロール」フィールドの左側のカラムには、サーバによってサポートされているサービスの一覧が表示されます。右側のカラムには、各サービスの基本ステータスが表示されます (オンまたはオフ。オンの場合は、前回起動したときの時刻)。

4. 現在実行中のサービスに関するステータス情報を表示するには、「プロセスコントロール」フィールドでそのサービスを選択します。

   「サービスステータス」フィールドに、そのサービスに関するステータス情報が表示されます。

   POP、IMAP、および HTTP の場合、フィールドには、最終接続時間、合計接続数、現在の接続数、最後にサービスを起動してから接続に失敗した回数、最後にサービスを起動してからログインに失敗した回数が表示されます。

   このフィールドの情報を確認すれば、サーバにかかる負荷やそのサービスの信頼性などを把握できます。また、サーバのセキュリティに対する攻撃を調べるのにも役立ちます。

5. サービスを起動するには、「プロセスコントロール」フィールドでそのサービスを選択し、「起動」をクリックします。

6. サービスを停止するには、「プロセスコントロール」フィールドでそのサービスを選択し、「停止」をクリックします。

7. 有効なサービスをすべて起動または停止するには、「すべて起動」ボタンまたは「すべて停止」ボタンをクリックします。

コマンドライン : `start-msg` および `stop-msg` コマンドを使って、任意のメッセージングサービス (`pop`、`imap`、`http`、`smtp`、`store`) を起動または停止できます。以下に、その例を示します。

```
server_root/msg-instance/start-msg imap
server_root/msg-instance/stop-msg pop
server_root/msg-instance/stop-msg smtp
```

注 `start-msg smtp` および `stop-msg smtp` コマンドを実行すると、SMTP サーバだけでなく、すべての MTA サービスが起動または停止します。特定の MTA サービスだけを起動または停止する場合は、`imsimta start` または `imsimta stop` コマンドを使用します。詳細については、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

グリーティングメッセージを設定するには

Messaging Server を使って、新規ユーザに送るグリーティングメッセージを作成できます。

コンソール : Console を使って新規ユーザへのグリーティングメッセージを作成するには、次の手順に従います。

1. Console で、新規ユーザへのグリーティングを設定する Messaging Server を開きます。
2. 「構成」タブをクリックします。左側のペインでサーバのアイコンが強調表示されていない場合は、アイコンを選択します。
3. 右側のペインの「その他」タブを選択します。
4. 必要に応じて、新規ユーザへのグリーティングを作成または変更します。

   電子メールメッセージと同じように、グリーティングメッセージの書式を設定する必要があります。まずヘッダー (少なくとも件名行を含める) を入力し、1 行空けて、メッセージ本文を入力します。

   メッセージを作成する際は、メッセージフィールドの上にあるドロップダウンリストを使って言語を指定します。必要に応じて、複数の言語で複数のメッセージを作成することも可能です。サーバは、「自動返信メッセージ用の言語の設定」の節で説明している情報に基づいて、新規ユーザに適切な言語のメッセージを送信します。
5. 「保存」をクリックします。

コマンドライン : コマンドラインを使って新規ユーザへのグリーティングメッセージを作成するには、次のように入力します。

```
configutil -o gen.newuserforms -v value
```

自動返信メッセージ用の言語を設定する

この節では、サーバから送信される通知やメッセージの言語がどのようにして選択されるかについて説明します。また、ユーザが言語を指定する方法やデフォルトのサーバサイト言語を指定する方法についても説明します。

ユーザは、指定した特定の条件が満たされたときにサーバから自動的に送られるメッセージを作成できます。たとえば、すべての受信メールに対して「現在、休暇中です。」というようなメッセージを自動返信することが可能です。このようなメッセージを作成するときは、そのメッセージが特定の言語で表示されるように指定できます。つまり、サーバが送信するメッセージをいくつかの異なる言語で作成しておくことが可能です。

ユーザが、自分が受け取る自動返信メッセージの言語を指定することもできます。ただし、その言語で作成されたメッセージが準備されている必要があります。

サーバは、以下の規則に従って特定言語の送信メッセージを選択します。

1. メッセージを受け取るユーザが言語を選択しており (42 ページの「ユーザの優先言語を設定するには」を参照)、その言語で作成されたメッセージが準備されている場合は、その言語のメッセージが送信されます。たとえば、ユーザが日本語を選択しており、日本語で作成されたメッセージが準備されている場合は、日本語のメッセージが送信されます。ドメインの優先言語が使用可能な場合と、一致する自動返信メッセージがある場合は、これが使われます。
2. ユーザが言語を選択していない場合、または言語を選択しているがその言語のメッセージが準備されていなかったり、ドメインの優先言語が指定されていない場合は、デフォルトのサーバサイト言語 (43 ページの「サーバサイト言語を設定するには」を参照) のメッセージが送信されます。たとえば、デフォルトのサーバサイト言語がスペイン語で、ユーザがフランス語を選択しているのにフランス語版のメッセージが準備されていない場合は、スペイン語版のメッセージが送信されます。
3. メッセージが 1 つの言語でしか作成されなかった場合は、ユーザが選択した言語やサイト言語に関係なく、準備されている言語のメッセージが送信されます。
4. ユーザの優先言語またはデフォルトのサイト言語と一致するメッセージが準備されていない場合、ドメインの優先言語が使用可能な場合、および複数の言語バージョンがある場合は、ユーザの LDAP エントリ内の最初のメッセージテキストが送信されます。

ユーザの優先言語を設定するには

ユーザは、Delegated Administrator for Messaging のインタフェースを使って優先言語を選択できます。また、メールクライアントの中には、優先言語を指定できるものもあります。Delegated Administrator を使って優先言語を設定した場合、その情報は Directory Server に保存されます。

サーバの管理ドメイン外のユーザにメッセージを送信する場合、サーバはそのユーザの優先言語は判断できません。ただし、そのメッセージが、ヘッダーに優先言語が指定された受信メッセージへの応答である場合を除きます。これらのヘッダーフィールド (`accept-language`、`Preferred-Language`、または `X-Accept-Language`) は、ユーザのメールクライアントで指定された属性に応じて設定されています。

優先言語に対して複数の設定がある場合、たとえば、Directory Server に保存されている優先言語属性とメールクライアントで指定された優先言語があるような場合は、以下の順序で優先言語が選択されます。

1. 元のメッセージの `accept-language` ヘッダー

2. 元のメッセージの Preferred-Language ヘッダー
3. 元のメッセージの X-Accept-Language ヘッダー
4. 差出人の優先言語属性 (LDAP ディレクトリで見つかった場合)

ドメインの優先言語を設定するには

ドメインの優先言語は、特定のドメイン用に指定されているデフォルトの言語です。たとえば、mexico.siroe.com というドメイン用にスペイン語を指定するとします。管理者は、Delegated Administrator for Messaging インタフェースでドメインを作成する際に優先言語オプションを選択することによって、あるいはドメインの LDAP エントリに LDAP 属性 preferredLanguage を追加することによって、ホストドメインの優先言語を選択することができます。

サーバサイト言語を設定するには

以下の手順に従って、サーバのデフォルトサイト言語を指定できます。ユーザの優先言語が設定されていない場合は、サイト言語を使用して特定言語のメッセージを送信します。

コンソール : Console からサイト言語を指定するには、次の手順に従います。

1. 設定を行う Messaging Server を開きます。
2. 「構成」タブをクリックします。
3. 右側のペインの「その他」タブをクリックします。
4. 「サイト言語」ドロップダウンリストで、使用する言語を選択します。
5. 「保存」をクリックします。

コマンドライン : 次に示すように、コマンドラインでサイト言語を指定することもできます。

```
configutil -o gen.sitelanguage -v value
```

value には、ローカルでサポートされている以下のいずれかの言語を指定できます。

```
af      アフリカーンス語
ca      カタロニア語
da      デンマーク語
de      ドイツ語
en      英語
es      スペイン語
```

fi	フィンランド語
fr	フランス語
ga	アイルランド語
gl	ガリシア語
is	アイスランド語
it	イタリア語
ja	日本語
nl	オランダ語
no	ノルウェー語
pt	ポルトガル語
sv	スウェーデン語

シングルサインオン (SSO) を有効にする

シングルサインオン機能を使うと、1 つのアプリケーションにログインしたユーザがほかのアプリケーションも使用できるようになります。たとえば、Messenger Express にログインしたユーザは、認証プロセスを繰り返さなくても Delegated Administrator for Messaging を使用できるようになります。

2 つのアプリケーション間でシングルサインオンを有効にするには、各アプリケーションを設定する必要があります。この節では、Messenger Express と Delegated Administrator の間でシングルサインオンを有効にする方法について説明します。46 ページの「Messenger Express と Delegated Administrator for Messaging 間のシングルサインオンを有効にするには」を参照してください。

Messenger Express の SSO 設定パラメータ

`configutil` コマンドを使うと、Messenger Express のシングルサインオン設定パラメータを変更できます。表 2-4 に、パラメータを示します。`configutil` の詳細については、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

表 2-4　Messenger Express のシングルサインオンパラメータ

パラメータ	説明
local.webmail.sso.enable	ログインページが取り込まれたときにクライアントが提示する SSO cookie を受け入れ確認する機能、ログイン成功時に SSO cookie を返す機能、ほかの SSO パートナーからの要求に応答して独自の cookie を確認する機能など、すべてのシングルサインオン機能を有効または無効にする ゼロ以外の値に設定した場合、サーバはすべての SSO 機能を実行する ゼロに設定した場合、サーバはどの SSO 機能も実行しない デフォルト値はゼロ
local.webmail.sso.prefix	このパラメータの文字列値は、HTTP サーバによって設定された SSO cookie をフォーマットするときの接頭辞として使用される。この接頭辞の付いた SSO cookie だけがサーバによって認識され、ほかの SSO cookie はすべて無視される このパラメータの値が null (空白) の場合は、事実上、サーバ上のすべての SSO 機能が無効になる デフォルト値は null (空白)
local.webmail.sso.id	このパラメータの文字列値は、Messenger Express HTTP サーバによって設定された SSO cookie をフォーマットするときのアプリケーション ID 値として使用される デフォルト値は null (空白)
local.webmail.sso. cookiedomain	このパラメータの文字列値は、Messenger Express HTTP サーバによって設定されたすべての SSO cookie の cookie ドメイン値を設定するために使用される デフォルト値は null (空白)
local.webmail.sso. singlesignoff	このパラメータの整数値がゼロ以外に設定されている場合は、クライアントがログアウトするときに、local.webmail.sso.prefix の値に一致する接頭辞値を持つクライアント上の SSO cookie がすべて消去される ゼロに設定されている場合は、クライアントがログアウトするときに、Messenger Express がその独自の SSO cookie を消去する デフォルト値はゼロ

表 2-4　Messenger Express のシングルサインオンパラメータ (続き)

パラメータ	説明
`local.sso.`*appid*`.verifyurl`	ピア SSO アプリケーションの確認 URL 値を設定する。*appid* は、処理される SSO cookie を生成するピア SSO アプリケーションのアプリケーション ID である。たとえば、Delegated Administrator の *appid* は `nda45` である 信頼されている各 SSO アプリケーションに対し、1 つのパラメータが定義されている必要がある。確認 URL の標準形は次のようになる `http://`*nda-host:port*`/VerifySSO?`

したがって、Messenger Express でシングルサインオンを有効にするには、以下のように各パラメータを設定します (デフォルトのドメインは eng.siroe.com です)。

```
configutil -o local.sso.appid.verifyurl -v "http://nda-host:port/verifySSO?"
configutil -o local.webmail.sso.enable -v 1
configutil -o local.webmail.sso.prefix -v ssogrp1
configutil -o local.webmail.sso.id -v msg50
configutil -o local.webmail.sso.cookiedomain -v ".siroe.com"
configutil -o local.webmail.sso.singlesignoff -v 1
```

Messenger Express と Delegated Administrator for Messaging 間のシングルサインオンを有効にするには

Messenger Express と Delegated Administrator との間でシングルサインオンを有効にするには、次の手順に従います。

1. Directory Server を設定します。
   a. Directory Server でプロキシユーザアカウントのエントリを作成します。
   b. プロキシ認証の ACI (Access Control Instructions) を作成します。
2. Delegated Administrator を設定します。
   a. プロキシユーザ証明書を追加します。
   b. シングルサインオン cookie 情報を追加します。
   c. 対象となるサーバの確認 URL を追加します。

3. Enterprise Server を再起動します。

Directory Server を設定するには、ldapmodify ユーティリティを使用します。このユーティリティの詳細については、Directory Server のマニュアルを参照してください。

Delegated Administrator を設定するには、以下の設定ファイルを変更します。

iDA_server_root/nda/classes/netscape/nda/servlet/resource.properties

Web_server_root/https-*instancename*/config/servlets.properties

Web_server_root/https-*instancename*/config/contexts.properties

手順 1a : プロキシユーザアカウントを作成する

プロキシユーザアカウントを使用すると、ユーザはプロキシ認証のために Directory Server にバインドできるようになります。以下に、プロキシユーザアカウントエントリの例を示します。

```
dn: uid=proxy, ou=people, o=siroe.com, o=isp
objectclass: top
objectclass: person
objectclass: organizationalperson
objectclass: inetorgperson
uid: proxy
givenname: Proxy
sn: Auth
cn: Proxy Auth
userpassword: proxypassword
```

手順 1b : プロキシ認証の ACI を作成する

次に、ldapmodify ユーティリティを使って、インストール時に作成した接尾辞の ACI を作成します。

- osiroot - ユーザデータを保存するために入力した接尾辞
- dcroot - ドメイン情報を保存するために入力した接尾辞
- osiroot - 設定情報を保存するために入力した接尾辞 (デフォルト : osiroot)

以下に、ACI エントリの例を示します。

```
dn:  o=isp
changetype:  modify
add:  aci
aci:  (target="ldap:///o=isp")(targetattr="*")(version 3.0; acl
     "proxy";allow (proxy) userdn="ldap:///uid=proxy, ou=people,
     o=siroe.com, o=isp";)
```

手順 2a : resource.properties ファイルにプロキシユーザ証明書を追加する

Delegated Administrator をプロキシ認証用に設定するには、Delegated Administrator の *iDA_server_root*/nda/classes/netscape/nda/servlet/resource.properties ファイルの以下のエントリのコメントを解除し、修正します。

LDAPDatabaseInterface-ldapauthdn=*Proxy_Auth_DN*

LDAPDatabaseInterface-ldapauthpw=*Proxy_Auth_Password*

たとえば、以下のようになります。

```
LDAPDatabaseInterface-ldapauthdn=
   uid=proxy, ou=people,o=siroe.com, o=mailqa
   LDAPDatabaseInterface-ldapauthpw=proxypassword
```

手順 2b : シングルサインオン cookie 情報を追加する

シングルサインオン情報を追加するには、Delegated Administrator のコンテキスト識別子を定義し、そのコンテキストの cookie 名を指定します。

- コンテキスト識別子を定義するには、Enterprise Server の *Web_Server_Root*/https-*instancename*/config/servlets.properties ファイルを編集し、servlet.xxxxx.context=ims50 というテキストが含まれている行のコメントをすべて解除する
- Delegated Administrator の設定でコンテキストの cookie 名を指定するには、以下のエントリを Delegated Administrator の *iDA_server_root*/nda/classes/netscape/nda/servlet/resource.properties ファイルに追加する

  ```
  NDAAuth-singleSignOnId=ssogrp1-
  NDAAuth-applicationId=nda45
  ```

- Enterprise Server の設定でコンテキストの cookie 名を指定するには、以下のエントリを Enterprise Server の *Web_Server_Root*/https-*instancename*/config/contexts.properties ファイルに追加する

  ```
  context.ims50.sessionCookie=ssogrp1-nda45
  ```

手順 2c : 対象となるサーバの確認 URL を追加する

受け取ったシングルサインオン cookie を確認するには、Delegated Administrator にその連絡先を指定しておく必要があります。対象となるすべてのサーバに、確認 URL を指定します。

以下の例では、Messenger Express がインストールされており、そのアプリケーション ID が msg50 であると仮定しています。Delegated Administrator の *iDA_server_root*/nda/classes/netscape/nda/servlet/resource.properties ファイルを編集し、以下のようなエントリを追加します。

```
verificationurl-ssogrp1-msg50=http://webmail_hostname:port/VerifySSO?
verificationurl-ssogrp1-nda45=http://nda_hostname:port/VerifySSO?
```

手順 3 : Enterprise Server を再起動する

手順 1a 〜 2c の説明に従って設定を変更したら、その変更内容が反映されるように Enterprise Server を再起動します。

ディレクトリ検索をカスタマイズするには

iPlanet Messaging Server は、iPlanet Directory Server などの LDAP ベースのディレクトリシステムがないと機能しません。Messaging Server および Console には、以下の 3 つの目的を果たすためにディレクトリアクセスが必要です。

- Messaging Server をはじめてインストールする際、管理者はサーバの構成設定を入力します。これらの設定は、中央設定ディレクトリに保存されます。また、インストール時には、そのディレクトリへの接続も設定します。
- 管理者がメールユーザまたはメールグループ用のアカウント情報を作成または更新すると、その情報はユーザディレクトリと呼ばれるディレクトリに保存されます。サーバグループの Administration Server はインストール時に設定されており、管理者が「ユーザ」や「グループ」にアクセスすると、デフォルトでは、Console は 管理トポロジを定義するユーザディレクトリに接続します。管理トポロジとは、同じ設定ディレクトリおよびユーザディレクトリを共有する iPlanet サーバの集まりです。

- メッセージのルーティング時やメールボックスへのメールの配信時に、Messaging Server はユーザディレクトリ内で差出人または受取人に関する情報を検索します。デフォルトでは、Messaging Server は Administration Server が使用するのと同じユーザディレクトリ内を検索します。

これらのディレクトリの構成設定は、以下の方法で変更できます。

- Console の「Administration Server」インタフェースを使用すると、設定ディレクトリの接続設定を変更できます。詳細については、『Netscape Console によるサーバの管理』の「Administration Server」の章を参照してください。
- ユーザやグループの情報を変更する場合は、Console の「ユーザおよびグループ」インタフェースを使用すると、デフォルトとは別のユーザディレクトリに一時的に接続することができます。詳細については、『Netscape Console によるサーバの管理』の「ユーザおよびグループ」の章を参照してください。
- Console の「Messaging Server」インタフェースを使用すると、Administration Server で定義されているデフォルトとは別のユーザディレクトリに接続するように Messaging Server を設定できます。これが、この節で説明している設定作業です。

別のユーザディレクトリに接続してユーザやグループを検索するように Messaging Server を再設定するかどうかは、管理者の判断次第です。通常は、サーバの管理ドメインを定義しているユーザディレクトリがドメイン内のすべてのサーバによって使用されます。

注 Messaging Server の検索用にカスタムユーザディレクトリを指定した場合は、Console の「ユーザおよびグループ」インタフェースにアクセスして、そのディレクトリのユーザ情報またはグループ情報を変更するときにも同じディレクトリを指定する必要があります。詳細は、付録 D「メールユーザとメーリングリストを管理する」を参照してください。

コンソール : Console を使って Messaging Server の LDAP ユーザ検索設定を変更するには、次の手順に従います。

1. Console から、LDAP 接続をカスタマイズする Messaging Server を開きます。
2. 「構成」タブをクリックします。
3. 左側のペインで「サービス」フォルダを選択します。
4. 右側のペインで「LDAP」タブを選択します。LDAP フォームが表示されます。

   LDAP フォームには、設定ディレクトリとユーザディレクトリの構成設定が表示されます。ただし、このフォーム内の設定ディレクトリの設定は読み取り専用です。これらの設定の変更方法については、『Netscape Console によるサーバの管理』の「Administration Server」の章を参照してください。
5. ユーザディレクトリの接続設定を変更するには、「メッセージングサーバ固有のディレクトリ設定を使用」ボックスをクリックします。

6. 以下に示す情報を入力または変更して、LDAP 構成を更新します (識別名などの用語の定義やディレクトリの概念については、『iPlanet Directory Server 管理者ガイド』を参照)。

   ホスト名：インストールのユーザ情報を含むディレクトリがあるホストマシンの名前。通常、これは Messaging Server ホストとは別のものです。ただし、非常に小規模のインストールでは、同じ場合もあります。

   ポート番号：Messaging Server がユーザ検索用のディレクトリにアクセスするときに使用するディレクトリホストのポート番号。この番号は、ディレクトリ管理者が定義するもので、必ずしもデフォルトのポート番号 (389) である必要はありません。

   ベース DN： 検索ベース (ユーザ検索の開始点を示すディレクトリエントリの識別名)。ディレクトリツリー内で検索ベースが目的の情報に近いほど、検索処理は速くなります。ディレクトリツリーに「people」や「users」などの分岐がある場合は、それを開始点にするのが妥当です。

   バインド DN： Messaging Server が検索を行うために Directory Server に接続する際、その Messaging Server を識別するために使われる名前。バインド DN は、ディレクトリのユーザ部分に対する検索特権がある、ユーザディレクトリのエントリの識別名でなければなりません。ディレクトリに対して匿名検索アクセスを許可する場合は、このエントリを指定しないことも可能です。

7. ユーザ検索のために LDAP ディレクトリに対してこの Messaging Server の認証を行う際に、バインド DN とともに使用するパスワードを変更するには、「バインドパスワードの変更」ボタンをクリックします。「パスワード入力」ウィンドウが表示されたら、そこに新しいパスワードを入力します。

   この場合に使用するパスワードは、個別のセキュリティポリシーによって決まります。最初、パスワードは「パスワードなし」に設定されています。「バインド DN」フィールドに何も入力しないで匿名アクセスを指定した場合、パスワードは使用しません。

   この手順により、サーバ構成に保存されているパスワードは更新されますが、LDAP サーバ内のパスワードは変更されません。また、このアカウントは、デフォルトで PAB 検索にも使用されます。パスワードを変更したら、以下の 2 つの操作を行う必要があります。

8. 設定属性 `local.ugldapbinddn` で指定されているユーザのパスワードを変更します。このユーザアカウントは、設定属性 `local.ugldaphost` に指定されているディレクトリサーバ内にあります。

9. `local.service.pab.ldapbinddn` および `local.service.pab.ldaphost` 属性で指定されているものと同じアカウントが PAB で使用されている場合は、`local.service.pab.ldappasswd` に保存されているパスワードも更新する必要があります。

デフォルトのユーザディレクトリに戻るには、「メッセージングサーバ固有のディレクトリ設定を使用」ボックスのチェックマークを外します。

コマンドライン : 以下に示すように、コマンドラインでユーザディレクトリの接続設定値を指定することもできます。上記の手順 8 および 9 で説明しているように、LDAP および PAB パスワードも必ず設定してください。

メッセージングサーバ固有のディレクトリ設定を使用するかどうかを指定するには、次のように入力します。

```
configutil -o local.ugldapuselocal -v [ yes | no ]
```

ユーザ検索用の LDAP ホスト名を指定するには、次のように入力します。

```
configutil -o local.ugldaphost -v name
```

ユーザ検索用の LDAP ポート番号を指定するには、次のように入力します。

```
configutil -o local.ugldapport -v number
```

ユーザ検索用の LDAP ベース DN を指定するには、次のように入力します。

```
configutil -o local.ugldapbasedn -v basedn
```

ユーザ検索用の LDAP バインド DN を指定するには、次のように入力します。

```
configutil -o local.ugldapbinddn -v binddn
```

暗号化の設定

Console を使用すると、Messaging Server の SSL (Secure Sockets Layer) 暗号化および認証を有効にしたり、サーバがすべてのサービスにわたってサポートする特定の符合化方式を選択できます。

この作業は一般的な設定タスクですが、第 12 章「セキュリティとアクセス制御を設定する」の「SSL を有効化し符号化方式を選択するには」で説明します。この章には、すべてのセキュリティに関する背景情報や Messaging Server のアクセス制御に関するトピックが記載されています。

第 3 章

POP、IMAP、および HTTP サービスの設定

iPlanet Messaging Server は、クライアントのメールボックスへのアクセス用に Post Office Protocol 3 (POP3)、Internet Mail Access Protocol 4 (IMAP4)、および Hyper Text Transfer Protocol (HTTP) をサポートしています。IMAP と POP はいずれもインターネットの標準メールボックスプロトコルです。Web で使用する電子メールプログラムの Messnger Express で、エンドユーザは HTTP でインターネットに接続されたコンピュータシステム上で動作しているブラウザを使って自分のメールボックスにアクセスすることができます。

この章では、iPlanet Console またはコマンドラインユーティリティを使って 1 つ以上のサービスをサポートするように構成する方法について説明します。

注	iPlanet Directory Server 5.1 をインストールしている場合は、同時にインストールした iPlanet Console 5.0 から管理します。iPlanet Messaging Server 5.2 は、同時にインストールした Netscape Console 4.2 から管理します。

Simple Mail Transfer Protocol (SMTP) サービスの設定については、第 6 章「MTA サービスと設定について」を参照してください。

この章には、以下の節があります。

全般的な設定

Messaging Server の POP、IMAP、および HTTP サービスの全般的な機能の設定には、サービスの有効無効の指定、ポート番号の割り当て、および接続するクライアントへ送信されるサービスバナーの修正 (省略可) が含まれます。この節では、そのための基礎的な情報を提供します。これらの設定を行う際に従う手順については、62 ページの「POP サービスを設定するには」、63 ページの「IMAP サービスを設定するには」、および 65 ページの「HTTP サービスを設定するには」を参照してください。

サービスの有効化と無効化

Messaging Server の特定のインスタンスがその POP、IMAP、または HTTP サービスを使用できるようにするかどうかを制御することができます。これは、サービスの開始や停止と同じではありません (37 ページの「サービスを起動および停止する」を参照してください)。POP、IMAP、または HTTP が機能するには、有効化されていることと開始されていることの両方が必要です。

サービスの有効化は、サービスの開始や停止よりも「グローバルな」処理です。たとえば、有効にする設定はシステムを再起動しても持続されますが、前に「停止」したサービスを再起動後に再び開始する必要があります。

使用する予定がないサービスは有効にする必要はありません。たとえば、Messaging Server インスタンスをメッセージ転送エージェント (MTA) としてのみ使用する場合、POP、IMAP、および HTTP は無効にする必要があります。POP サービス用にのみ使用する場合、IMAP と HTTP を無効にする必要があります。Web ベースの電子メール用にのみ使用する場合、POP と IMAP を無効にする必要があります。

サービスの有効化と無効化は、サーバレベルで行うことができます。この処理はこの章で説明されています。特定の LDAP 属性を設定することにより、ユーザレベルでサービスの有効化と無効化を行うことができます。詳細は、『iPlanet Messaging Server プロビジョニングガイド』を参照してください。

ポート番号を指定する

各サービスに対して、サーバがサービスの接続に使用するポート番号を指定することができます。

- POP サービスを有効にする場合、サーバが POP 接続に使用するポート番号を指定することができます。デフォルトは 110 です。
- IMAP サービスを有効にする場合、サーバが IMAP 接続に使用するポート番号を指定することができます。デフォルトは 143 です。

- HTTP サービスを有効にする場合、サーバが HTTP 接続に使用するポート番号を指定することができます。デフォルトは 80 です。

たとえば 1 つのホストマシンに複数の IMAP サーバインスタンスがある場合や、同じホストマシンを IMAP サーバおよび Messaging Multiplexor サーバとして使用している場合は、デフォルト以外のポート番号を指定する必要があります。Multiplexor については、第 4 章「マルチプレクササービスを設定および管理する」を参照してください。

ポート番号を指定する際には、次の点に注意してください。

- ポート番号は 1 から 65535 までの任意の値を指定できます。
- 選択したポートが別のサービス用にすでに使用されていたり、割り当てられていないことを確認してください。

暗号化通信用のポート

Messaging Server は、SSL (Secure Socket Layer) プロトコルを使用することにより、IMAP や HTTP クライアントの暗号化通信をサポートします。Messaging Server の SSL サポートについての詳細は、404 ページの「暗号化と証明書に基づく認証を構成する」を参照してください。

SSL を使用した IMAP

「SSL を使用した IMAP」のデフォルトポート番号 (993) を使用するか、または「SSL を使用した IMAP」に別のポートを指定することができます。

現在の IMAP クライアントの多くが個別の IMAP ポートおよび SSL を使用した IMAP ポートを必要としているため、Messaging Server ではオプションとしてそれぞれに個別のポートを使用できます。最近では、同じポートによる IMAP および「SSL を使用した IMAP」の通信が新たな標準となってきています。お使いの Messaging Server に SSL の証明書 (406 ページの「証明書の入手」を参照) がインストールされていれば、同じポートを使って IMAP および「SSL を使用した IMAP」の通信を行うことができます。

SSL を使用した HTTP

「SSL を使用した HTTP」のデフォルトポート番号 (443) を使用するか、または「SSL を使用した HTTP」に別のポートを指定することができます。

サービスの見出し

クライアントがはじめて Messaging Server の POP または IMAP のポートに接続すると、サーバがクライアントに確認用のテキスト文字列を送信します。このサービスの見出し (通常、クライアントのユーザには表示されません) は、サーバが iPlanet Messaging Server であることを証明するもので、そこにはサーバのバージョン番号が表示されます。一般に、この見出しはクライアントのデバッグまたは問題をつきとめるために使用されます。

接続中のクライアントに他のメッセージを送信したい場合、POP または IMAP サービスのデフォルトの見出しを変更できます。

iPlanet Console または `configutil` ユーティリティ (`service.imap.banner`、`service.pop.banner`) を使ってサービス見出しを設定することができます。`configutil` のシンタックスの詳細については、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

ログインの必要条件

ユーザは POP、IMAP、または HTTP サービスにログインしてメールを取り込みます。このユーザによるログインの方法は制御できます。パスワードに基づくログイン (すべてのサービス)、および証明書に基づくログイン (IMAP または HTTP サービス) を許可することができます。この節では、予備知識としての情報を提供しています。これらの設定の手順について知りたい場合は、62 ページの「POP サービスを設定するには」、63 ページの「IMAP サービスを設定するには」、または 65 ページの「HTTP サービスを設定するには」を参照してください。さらに、POP ログインの有効なログイン区切りを指定することもできます。

POP クライアントのログイン区切りを設定するには

POP メールクライアントによっては、Messaging Server で、ログイン区切りとして `@` を使用できない場合があります。アドレスに含まれる `@` が `uid@dmain` と似ているからです。これらのクライアントの例には、Windows 2000 上で動作する Netscape Messenger 4.76、Netscape Messenger 6.0、および Microsoft Outlook Express があります。これを回避するには次のようにします。

1. 次のコマンドを使って + を有効な区切りにします。

```
configutil -o service.loginseparator -v "@+"
```

2. POP クライアントユーザに @ ではなく + をログイン区切りとして使ってログインするよう知らせます。

パスワードに基づくログイン

一般的なメッセージングインストールでは、ユーザはメールクライアントにパスワードを入力して POP、IMAP、または HTTP メールボックスにアクセスします。クライアントがパスワードをサーバに送信すると、サーバはそのパスワードを使ってユーザを認証します。ユーザが認証されると、アクセス制御規則に基づき、そのサーバに保存されている特定のメールボックスへのアクセスを許可するかどうかが決定されます。

パスワードログインを認めると、ユーザはパスワードを入力することにより POP、IMAP、または HTTP にアクセスできるようになります。POP サービスにおける認証方法は、パスワードに基づくログインのみです。パスワードは LDAP ディレクトリに保存されます。パスワードの必要最小文字数などのポリシーは、ディレクトリポリシーによって決まります。

IMAP または HTTP サービスに対してパスワードログインを認めない場合は、パスワードに基づく認証は許可されません。その場合、次の節で説明する証明書に基づくログインを行わなければなりません。

IMAP および HTTP サービスにおけるパスワード送信のセキュリティを強化するために、サーバに送信する前にパスワードを暗号化するように要求できます。そのためには、ログインに必要な暗号化最小文字数を選択します。

- 暗号化の必要がない場合にはゼロを選択します。クライアントポリシーによって、パスワードは平文で、または暗号化されて送信されます。
- ゼロ以外の値を選択すると、クライアントは指定した値を満たすキー長の符号化方式を使って、サーバとの SSL セッションを確立しなければなりません。これにより、クライアントが送信する IMAP または HTTP のユーザパスワードがすべて暗号化されます。

クライアントにおける暗号化のキー長設定がサーバのサポートする最大長より大きい場合、またはサーバにおける暗号化のキー長設定がクライアントのサポートする最大長より大きい場合は、パスワードに基づくログインを行うことができません。さまざまな符号化方式とキー長をサポートするようにサーバを設定する方法については、410 ページの「SSL を有効化し符号化方式を選択するには」を参照してください。

証明書に基づくログイン

パスワードに基づく認証のほかに、iPlanet サーバはユーザのデジタル証明書を確認することにより認証を行うことができます。サーバとの SSL セッションを確立するときに、パスワードの代わりにユーザの証明書を提示します。証明書の妥当性が確認されると、ユーザが本人であるとみなされます。

IMAP または HTTP サービスに対し、証明書に基づくログインを認めるように Messaging Server を設定する方法については、413 ページの「証明書に基づくログインを設定するには」を参照してください。

証明書に基づくログインを有効にするために、IMAP または HTTP システムフォームの「パスワードログインの許可」チェックボックスをオフにする必要はありません。チェックボックスが選択されていても (デフォルト)、証明書に基づくログインの設定を行った場合は、パスワードに基づくログインと証明書に基づくログインの両方がサポートされます。その場合、クライアントが SSL セッションを確立して証明書を提示すると、証明書に基づくログインが使用されます。クライアントが SSL を使用しない場合や、クライアント証明書を提示しない場合には、代わりにパスワードが送信されます。

パフォーマンスパラメータ

Messaging Server の POP、IMAP、および HTTP サービスに対し、いくつかの基本的なパフォーマンスパラメータを設定できます。ハードウェアの容量に基づきユーザベースでもっとも効率的なサービスを実行できます。この節では、予備知識としての情報を提供しています。これらの設定の手順について知りたい場合は、62 ページの「POP サービスを設定するには」、63 ページの「IMAP サービスを設定するには」、または 65 ページの「HTTP サービスを設定するには」を参照してください。

プロセス数

Messaging Server は作業をいくつかの実行プロセスに分割することができます。こうすると、場合によっては効率が上がることがあります。この機能はマルチプロセッサのサーバマシンにおいて特に効果があります。多くのサーバプロセス数を調整することによりハードウェアプロセッサ間で複数のタスクをより効率よく分配できます。

ただし、タスクを複数のプロセスに割り当てたり、プロセッサ間で切り替えたりする際に、パフォーマンスオーバーヘッドが発生します。新たなプロセスが 1 つ追加されるごとに、複数のプロセスを持つ利点が薄れていきます。ほとんどの設定では、サーバマシンの各ハードウェアプロセッサ当たり 1 つのプロセス (最大でも 4 プロセス) を、割り当てるのが原則です。用途によっては最適とされる設定が異なることがあるため、この原則はあくまでも参考として把握しておいてください。

注 : プラットフォームによっては、パフォーマンスに影響を与える可能性のある、そのプラットフォーム固有のプロセスに対する制限 (最大ファイルディスクリプタ数など) を緩めるために、プロセス数を増やした方がよいこともあります。

POP、IMAP、および HTTP サービスのデフォルトのプロセス数は、1 です。

プロセス当たりの接続数

POP、IMAP、または HTTP サービスが同時に持てるクライアント接続の数が多いほど、クライアントにとって有利になります。空いている接続がないためにクライアントがサービスにアクセスできない場合、別のクライアントが接続を切断するまで待たなければなりません。

その一方で、各オープン接続がそれぞれメモリリソースを消費し、サーバマシンの入出力サブシステムに負担をかけるため、実際にサーバがサポートできる同時セッションの数には限界があります。サーバのメモリを増やすか入出力を拡大すれば、制限枠を上げることができます。

IMAP、HTTP、および POP には、それぞれ以下のような違いがあります。

- IMAP 接続は、POP や HTTP 接続に比べ、一般的に長く維持できます。メッセージをダウンロードするためにユーザが IMAP に接続すると、接続は通常ユーザが終了するか、タイムアウトになるまで維持されます。これに対し、POP 接続や HTTP 接続は、通常 POP または HTTP 要求が満たされるとすぐに閉じられます。
- 一般に、IMAP と HTTP 接続は、POP 接続に比べて非常に効率的です。POP 接続の場合は、再接続するたびにユーザの認証を必要とします。これに対し、IMAP 接続の場合は認証が必要なのは 1 回のみで、IMAP セッション (ログインからログアウトまで) が終わるまで接続が維持されます。HTTP 接続は短いですが、1 回の HTTP セッション (ログインからログアウトまで) で複数の接続が許可されているのでユーザは接続するたびに再び認証を行う必要はありません。そのため POP 接続は、IMAP または HTTP 接続よりも大幅なパフォーマンスオーバーヘッドを生じます。iPlanet Messaging Servr は、オープン IMAP 接続 (ただし、アイドル接続) と複数の HTTP 接続によって、オーバーヘッドを減らすように設計されています。

注	HTTP セッションのセキュリティの詳細については、399 ページの「HTTP のセキュリティについて」を参照してください。

したがって、所定の時間とユーザの要求により、Messaging Server はオープン IMAP 接続または HTTP 接続を POP 接続よりも多くサポートできる場合があります。

プロセス当たりの接続数は、IMAP のデフォルトが 4000、HTTP のデフォルトが 6000、POP のデフォルトが 600 です。これらの値は、一般的な設定のサーバマシンが処理できる要求とほぼ同等です。用途によっては最適とされる設定が異なることがあるため、これらのデフォルト値はあくまでも一般的なガイドラインとして参考にしてください。

プロセス当たりのスレッド数

複数のプロセスをサポートするだけでなく、Messaging Server ではタスクを複数のスレッドに分配することにより、さらにパフォーマンスを向上させています。サーバがスレッドを使用すると、処理中のコマンドがほかのコマンドの実行を妨げることがなくなるため、実行効率が向上します。スレッドは、設定した最大数の範囲内で、コマンドの実行中に、必要に応じて作成され破棄されます。

同時に実行されるスレッドが多いほど、より多くのクライアント要求を遅延なく処理することができます。そのためより多くのクライアントに迅速にサービスを提供できます。ただし、スレッド間のディスパッチがパフォーマンスオーバーヘッドになるため、実際にサーバが使用できるスレッド数には限界があります。

POP、IMAP、および HTTP のプロセス当たりの最大スレッド数は、デフォルトで 250 です。IMAP および HTTP のデフォルトの接続数が POP のデフォルト値より大きいにもかかわらず、同じ数値になります。同じ最大スレッド数で、より多くの IMAP および HTTP 接続が、より少なく、ただし頻度の高い POP 接続と同じくらい効率よく処理されると考えられます。用途によっては最適とされる設定が異なることがありますが、これらのデフォルト値は十分高いため、設定値を大きくする必要はおそらくありません。通常、これらのデフォルト値で十分なパフォーマンスが得られます。

アイドル接続を切断する

応答のないクライアントへの接続に使用されているシステムリソースを回復するために、IMAP4、POP3、および HTTP プロトコルは、一定の時間が過ぎたアイドル接続をサーバが一方的に切断することを許可します。

それぞれのプロトコル仕様により、サーバはアイドル接続を指定されている最小時間オープンにしておくことが要求されます。最低時間のデフォルト値は POP が 10 分、IMAP が 30 分、HTTP が 3 分です。アイドル時間を増やしてデフォルト値を増やすことはできますが、それ以下に減らすことはできません。

POP または IMAP 接続が切断されると、ユーザは新たに接続するときに再び認証される必要があります。これに対し、HTTP 接続が切断された場合は、HTTP セッションがオープンにされたままなので、再認証の必要はありません。HTTP セッションのセキュリティの詳細については、399 ページの「HTTP のセキュリティについて」を参照してください。

POP のアイドル接続は、通常クライアントが応答できない何らかの問題 (クラッシュやハングするなど) により起こります。一方、IMAP アイドル接続は正常な状態で発生します。IMAP ユーザが接続を一方的に切断されないようにするため、IMAP クライアントは通常 30 分以下の一定間隔で IMAP サーバにコマンドを送信します。

HTTP クライアントをログアウトする

HTTP セッションは複数の接続にわたって維持されます。HTTP クライアントは、接続が切断されてもログアウトされません。ただし、HTTP セッションが指定された時間以上アイドル状態であると、サーバは HTTP セッションを自動的に切断し、クライアントはログアウトされます (デフォルト値は 2 時間)。セッションが切断されると、クライアントのセッション ID が無効になり、クライアントは新たにセッションを確立するために、再び認証されなければなりません。HTTP のセキュリティおよびセッション ID の詳細については、399 ページの「HTTP のセキュリティについて」を参照してください。

クライアントアクセスの制御

iPlanet Messaging Server にはアクセス制御機能があり、POP、IMAP、または HTTP メッセージングサービス (および SMTP) にアクセスできるクライアントを決定することができます。さまざまな条件に基づき、クライアントのアクセスを許可または拒否する柔軟性のあるアクセスフィルタを作成できます。

クライアントアクセスの制御は、iPlanet Messaging Server に備わっている重要なセキュリティ機能です。クライアントアクセスの制御フィルタの作成と使用法の例については、417 ページの「POP、IMAP、および HTTP サービスへのクライアントアクセスを構成する」および 432 ページの「SMTP サービスへのクライアントアクセスを構成する」を参照してください。

POP サービスを設定するには

`configutil` コマンドまたは iPlanet Console を使用して、Messaging Server POP サービスの基本設定を行うことができます。この章では、一般的な POP サービスのオプションについて説明します。『iPlanet Messaging Server リファレンスマニュアル』に一覧表示されています。

詳細は、以下を参照してください。

- 54 ページの「サービスの有効化と無効化」
- 56 ページの「POP クライアントのログイン区切りを設定するには」
- 54 ページの「ポート番号を指定する」
- 59 ページの「プロセス当たりの接続数」
- 60 ページの「アイドル接続を切断する」
- 60 ページの「プロセス当たりのスレッド数」
- 58 ページの「プロセス数」

コンソール： 次のコンソールを使用して POP サービスを設定します

1. iPlanet Console で、設定する Messaging Server を開きます。
2. 「構成環境設定」タブをクリックし、左ペインで「サービス」フォルダを開きます。
3. 「POP」を選択します。
4. 右ペインで「システム」タブをクリックします。
5. サービスを有効にするには、「ポートで POP サービスを有効化」チェックボックスをオンにし、ポート番号を指定します。
6. 接続設定を次のように指定します。
   - プロセス当たりの最大ネットワーク接続数を設定します。詳細は、59 ページの「プロセス当たりの接続数」を参照してください。
   - 接続の最大アイドル時間を設定します。詳細は、60 ページの「アイドル接続を切断する」を参照してください。
7. プロセス設定を次のように指定します。
   - プロセス当たりの最大スレッド数を設定します。詳細は、60 ページの「プロセス当たりのスレッド数」を参照してください。
   - 最大プロセス数を設定します。詳細は、58 ページの「プロセス数」を参照してください。
8. 必要に応じて、POP サービスの見出しフィールドにサービスの見出しを指定します。

9. 「保存」をクリックします。

注	POP サービスの場合は、パスワードに基づくログインが自動的に有効になります。

コマンドライン： 次に示すように、コマンドラインから POP 属性の値を設定できます。

POP サービスを有効または無効にする

```
configutil -o service.pop.enable -v [ yes | no ]
```

ポート番号を指定する

```
configutil -o service.pop.port -v 番号
```

プロセス当たりの最大ネットワーク接続数を設定する

```
configutil -o service.pop.maxsessions -v 数値
```

接続の最大アイドル時間を設定する

```
configutil -o service.pop.idletimeout -v 数値
```

プロセス当たりの最大スレッド数を設定する

```
configutil -o service.pop.maxthreads -v 数値
```

最大プロセス数を設定する

```
configutil -o service.pop.numprocesses -v 数値
```

プロトコルにようこそ見出しを指定する

```
configutil -o service.pop.banner -v 見出し
```

IMAP サービスを設定するには

`configutil` コマンドまたは iPlanet Console を使用して、Messaging Server IMAP サービスの基本設定を行うことができます。この節では、一般的な IMAP サービスのオプションについて説明します。完全なリストは、『iPlanet Messaging Server リファレンスマニュアル』にあります。詳細は、以下を参照してください。

- 54 ページの「サービスの有効化と無効化」
- 54 ページの「ポート番号を指定する」
- 57 ページの「パスワードに基づくログイン」
- 59 ページの「プロセス当たりの接続数」

- 60 ページの「アイドル接続を切断する」
- 60 ページの「プロセス当たりのスレッド数」
- 58 ページの「プロセス数」

コンソール： 次のコンソールを使用して IMAP サービスを設定します

1. iPlanet Console で、設定する Messaging Server を開きます。
2. 「構成環境設定」タブをクリックし、左ペインで「サービス」フォルダを開きます。
3. 「IMAP」を選択します。
4. 右ペインで「システム」タブをクリックします。
5. サービスを有効にするには、「ポートで IMAP サービスを有効化」チェックボックスをオンにし、ポート番号を指定します。
6. 必要に応じて、パスワードに基づくログインを有効にします。
7. 接続設定を次のように指定します。
   - プロセス当たりの最大ネットワーク接続数を設定します。詳細は、59 ページの「プロセス当たりの接続数」を参照してください。
   - 接続の最大アイドル時間を設定します。詳細は、60 ページの「アイドル接続を切断する」を参照してください。
8. プロセス設定を次のように指定します。
   - プロセス当たりの最大スレッド数を設定します。詳細は、60 ページの「プロセス当たりのスレッド数」を参照してください。
   - 最大プロセス数を設定します。詳細は、58 ページの「プロセス数」を参照してください。
9. 必要に応じて、IMAP サービス見出しフィールドにサービスの見出しを指定します。
10. 「保存」をクリックします。

コマンドライン： 次に示すように、コマンドラインから IMAP 属性の値を設定できます。

IMAP サービスを有効または無効にする

```
configutil -o service.imap.enable -v [ yes | no ]
```

ポート番号を指定する

```
configutil -o service.imap.port -v 番号
```

「SSL を使用した IMAP」用に別のポートを有効にする

```
configutil -o service.imap.enablesslport -v [ yes | no ]
```

「SSL を使用した IMAP」のポート番号を指定する

```
configutil -o service.imap.sslport -v 番号
```

IMAP サービスでパスワードログインを有効または無効にする

```
configutil -o service.http.plaintextmincipher -v 値
```

value は次のいずれかになります。

```
 -1 - パスワードログインを無効にする
  0 - 暗号なしのパスワードログインを有効にする
 40 - パスワードログインを有効にし、暗号の強さを指定する
128 - パスワードログインを有効にし、暗号の強さを指定する
```

プロセス当たりの最大ネットワーク接続数を設定する

```
configutil -o service.imap.maxsessions -v 数値
```

接続の最大アイドル時間を設定する

```
configutil -o service.imap.idletimeout -v 数値
```

プロセス当たりの最大スレッド数を設定する

```
configutil -o service.imap.maxthreads -v 数値
```

最大プロセス数を設定する

```
configutil -o service.imap.numprocesses -v 数値
```

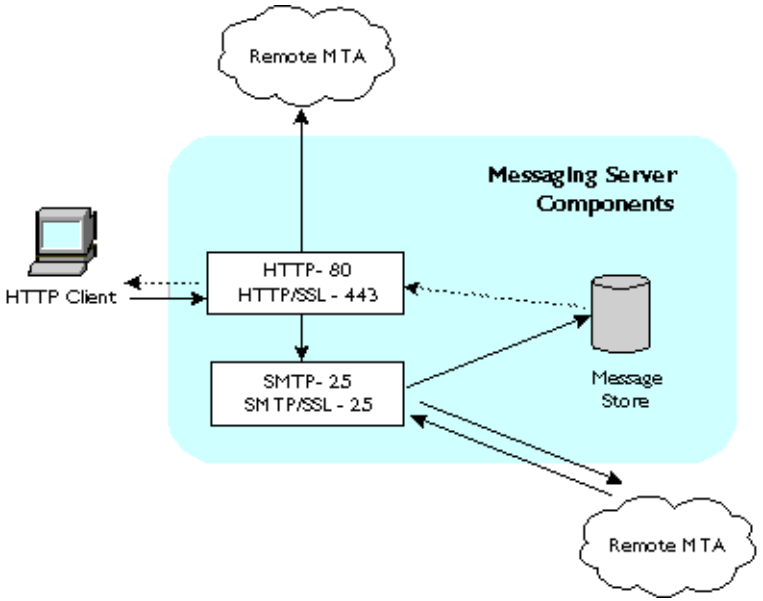
プロトコルにようこそ見出しを指定する

```
configutil -o service.imap.banner -v 見出し
```

HTTP サービスを設定するには

POP および IMAP クライアントは、ルーティングまたは配信するためにメールを直接 iPlanet Messaging Server の MTA に送信します。これに対し、HTTP クライアントは、メールを iPlanet Messaging Server の一部である特殊な Web サーバに送信します。その後、HTTP サービスは、図 3-1 に示すように、ルーティングまたは配信するためにメッセージをローカルの MTA またはリモート MTA に送信します。

図 3-1　HTTP サービスのコンポーネント



HTTP 設定パラメータの多くは、POP および IMAP サービスで提供されるパラメータに似ています。これらには、接続設定とプロセス設定のパラメータが含まれています。この節では、一般的な HTTP サービスのオプションについて説明します。『iPlanet Messaging Server リファレンスマニュアル』に一覧表示されています。詳細は、以下を参照してください。

- 54 ページの「サービスの有効化と無効化」
- 54 ページの「ポート番号を指定する」
- 57 ページの「パスワードに基づくログイン」
- 59 ページの「プロセス当たりの接続数」
- 60 ページの「アイドル接続を切断する」
- 61 ページの「HTTP クライアントをログアウトする」
- 60 ページの「プロセス当たりのスレッド数」
- 58 ページの「プロセス数」

パラメータの中には、メッセージ設定や MTA 設定など、HTTP サービスに独特なものもあります。

メッセージ設定： HTTP クライアントが添付ファイル付きのメッセージを構成すると、添付ファイルはサーバにアップロードされファイルに保存されます。ルーティングまたは配信するためにメッセージを MTA に送信する前に、HTTP サービスは添付ファイルを取得し、メッセージを構成します。この場合、デフォルトの添付スプールディレクトリを使用するか、または代わりのディレクトリを指定することができます。また、添付ファイルの最大サイズを指定することもできます。

MTA 設定： デフォルトでは、HTTP サービスは送信 Web メールをローカルの MTA に送信してルーティングまたは配信します。サイトがホストサービスで、ほとんどの受取人がローカルホストマシンと同じドメインではない場合には、メールをリモート MTA に送信するように HTTP サービスを設定できます。Web メールをリモート MTA に送信するには、リモートホスト名およびリモートホストの SMTP ポート番号を指定する必要があります。

コンソール： iPlanet Console を使用して HTTP サービスを設定する

1. iPlanet Console で、設定する Messaging Server を開きます。
2. 「構成環境設定」タブをクリックし、左ペインで「サービス」フォルダを開きます。
3. 「HTTP」を選択します。
4. 右ペインで「システム」タブをクリックします。
5. サービスを有効にするには、「ポートで HTTP サービスを有効化」チェックボックスをオンにし、ポート番号を指定します。
6. 必要に応じて、パスワードに基づくログインを有効にします。
7. 接続設定を次のように指定します。
   - プロセス当たりの最大ネットワーク接続数を設定します。詳細は、59 ページの「プロセス当たりの接続数」を参照してください。
   - 接続の最大アイドル時間を設定します。詳細は、60 ページの「アイドル接続を切断する」を参照してください。
   - クライアントセッションの最大アイドル時間を設定します。詳細は、61 ページの「HTTP クライアントをログアウトする」を参照してください。
8. プロセス設定を次のように指定します。
   - プロセス当たりの最大スレッド数を設定します。詳細は、60 ページの「プロセス当たりのスレッド数」を参照してください。
   - 最大プロセス数を設定します。詳細は、58 ページの「プロセス数」を参照してください。
9. メッセージ設定を次のように指定します。
   - 必要に応じて、添付スプールディレクトリを指定します。

- 必要に応じて、送信メールの最大サイズを指定します。このサイズは base64 でエンコードされたすべての添付ファイルが含まれること、および base64 でエンコードするには容量が 33% 多く必要になることに注意してください。このため、コンソールでの 5M バイトの容量制限を考慮すると 1 つのメッセージと添付ファイルの最大サイズは 3.75M バイトになります。

詳細は、67 ページの「メッセージ設定」を参照してください。

10. MTA 設定を次のように指定します。
    - 必要に応じて、代わりの MTA ホスト名を指定します。
    - 必要に応じて、代わりの MTA ポートを指定します。

    詳細は、67 ページの「MTA 設定」を参照してください。
11.「保存」をクリックします。

コマンドライン : 以下に示すように、コマンドラインを使用して HTTP 属性の値を設定できます。

HTTP サービスを有効または無効にする

```
configutil -o service.http.enable -v [ yes | no ]
```

ポート番号を指定する

```
configutil -o service.http.port -v 番号
```

「SSL を使用した HTTP」用に別のポートを有効にする

```
configutil -o service.http.enablesslport -v [ yes | no ]
```

「SSL を使用した HTTP」にポート番号を指定する

```
configutil -o service.http.sslport -v 番号
```

パスワードログインを有効または無効にする

```
configutil -o service.http.plaintextmincipher -v 値
```

value は次のいずれかになります。

```
 -1 - パスワードログインを無効にする
  0 - 暗号なしのパスワードログインを有効にする
 40 - パスワードログインを有効にし、暗号の強さを指定する
128 - パスワードログインを有効にし、暗号の強さを指定する
```

プロセス当たりの最大ネットワーク接続数を設定する

```
configutil -o service.http.maxsessions -v 数値
```

接続の最大アイドル時間を設定する

```
configutil -o service.http.idletimeout -v 数値
```

クライアントセッションの最大アイドル時間を設定する

```
configutil -o service.http.sessiontimeout -v 数値
```

プロセス当たりの最大スレッド数を設定する

```
configutil -o service.http.maxthreads -v 数値
```

最大プロセス数を設定する

```
configutil -o service.http.numprocesses -v 数値
```

クライアントの送信メールに対する添付スプールディレクトリを指定する

```
configutil -o service.http.spooldir -v ディスパッチ
```

メッセージの最大サイズを指定する

```
configutil -o service.http.maxmessagesize -v size
```

size はバイト単位です。このサイズは base64 でエンコードされたすべての添付ファイルが含まれること、および base64 でエンコードするには容量が 33% 多く必要になることに注意してください。このため、コンソールでの 5M バイトの容量制限を考慮すると 1 つのメッセージと添付ファイルの最大サイズは 3.75M バイトになります。

必要に応じて、代わりの MTA ホスト名を指定します。

```
configutil -o service.http.smtphost -v ホスト名
```

代わりの MTA ホスト名のポート番号を指定する

```
configutil -o service.http.smtpport -v ポート番号
```

第 4 章

マルチプレクササービスを設定および管理する

この章では、iPlanet Messaging Server に含まれる、標準メールプロトコル (POP、IMAP および SMTP) 対応の Messaging Multiplexor (MMP) および Messenger Express Web インタフェースで使用する Messenger Express Multiplexor の 2 つのマルチプレクサについて説明します。

この章には、以下の節があります。

- マルチプレクササービスについて
- iPlanet Messaging Multiplexor について (MMP)

注 MMP のインストール手順については、『iPlanet Messaging Server インストールガイド』を参照してください。MMP 設定パラメータの詳細については、『iPlanet MessagingServer リファレンスマニュアル』を参照してください。

- Messenger Express Multiplexor について

マルチプレクササービスについて

マルチプレクサは、複数のメールストアに間接的に接続する場合に使用する単一ドメイン名を提供します。このため、複数のマシンを追加することにより多くのユーザをサポートできる水平スケーラビリティ機能の実現には欠かすことができません。また、マルチプレクサにはセキュリティ上の利点もあります。

MMP は iPlanet Messaging Server から別途管理され、Messenger Express Multiplexor は iPlanet Message Store and Message Access のインストールに含まれる HTTP サービス (`mshttpd`) に組み込まれます。

マルチプレクサの利点

負荷の大きいメッセージングサーバでは、メッセージストアの容量が非常に大きくなることがあります。このような場合は、ユーザメールボックスとユーザ接続を複数のサーバに振り分けると、容量を拡張し、パフォーマンスを向上させることができます。また、大容量の大型マルチプロセッサマシンを 1 台使用するよりも、小さなサーバマシンを数台使う方が費用効率が高い場合があります。

メールサーバのインストールで複数のメッセージストアを使用する必要がある場合は、マルチプレクサを使用すると便利です。ユーザからメッセージストアへの接続が間接的であること、および複数のメッセージングサーバ間でのユーザアカウントの再設定が簡単であることから、以下のような利点が生まれます。

- **ユーザ管理の簡易化**

  すべてのユーザが 1 台 のサーバ (POP、IMAP、SMTP、Web アクセス用に別のマルチプレクサマシンがある場合は複数台) に接続するので、電子メールクライアントをあらかじめ設定しておき、すべてのユーザに同一のログイン情報を配布することができます。これにより管理タスクが簡易化され、間違ったログイン情報を配布する可能性が減ります。

  特に負荷が大きい状況では、同じ設定を使用して複数のマルチプレクササーバを実行し、DNS ラウンドロビンや負荷分散システムによってこれらのマルチプレクササーバへの接続を管理することができます。

  マルチプレクサは LDAP ディレクトリに格納されている情報を使って各ユーザの Messaging Server を検出します。このため、システム管理者は、ユーザに意識させることなく、ユーザを簡単に新しいサーバに移動することができます。管理者はユーザのメールボックスをある Messaging Server から別の Messaging Server に移動し、その後 LDAP ディレクトリでユーザのエントリを更新することができます。ユーザのメールアドレス、メールボックスアクセス、およびその他のクライアント設定は変更する必要がありません。

- **パフォーマンスの向上**

  メッセージストアの処理量が 1 台のマシンで可能な範囲を超えた場合は、メッセージストアの一部をほかのマシンに移動して負荷を均等にすることができます。

  異なるクラスのユーザを異なるマシンに割り当てることができます。たとえば、重要なユーザを大型の強力なマシンに割り当てることもできます。

  マルチプレクサでは一定のバッファリングが行われるので、ユーザが低速で接続 (モデム経由など) しても Messaging Server の速度が下がることはありません。

- **コストの削減**

  マルチプレクサを使うと複数の Messaging Server を効率的に管理できるので、小型のサーバマシンを数台購入しても超大型マシンを 1 台購入するほどにはコストがかからず、全体のコストを抑えることができます。

- **スケーラビリティの向上**

マルチプレクサを使うと、構成を簡単に拡張できます。パフォーマンスやストレージ容量を強化する必要があれば、既存のシステムを無駄にすることなく、マシンを段階的に追加することができます。

- **最小限のユーザダウンタイム**

  マルチプレクサを使用すると、大規模なユーザベースを多数の小さなストアマシンに振り分けることで、ユーザダウンタイムを抑えることができます。あるサーバが故障しても、影響を受けるのはそのサーバのユーザだけです。

- **セキュリティの強化**

  マルチプレクサがインストールされているサーバマシンをファイアウォールマシンとして使用することができます。クライアント接続をすべてこのマシンにルーティングすることで、外部のコンピュータから内部のメッセージストアマシンへのアクセスを制限することができます。マルチプレクサは、クライアントとの非暗号化通信および暗号化通信をサポートしています。

iPlanet Messaging Multiplexor について

iPlanet Messaging Multiplexor (MMP) は、複数のバックエンド Messaging Server の単一接続ポイントとして機能する特別な Messaging Server です。Messaging Multiplexor を利用すると、大規模なメッセージングサービスプロバイダは、POP および IMAP のユーザメールボックスを多数のマシン間に分散してメッセージストア容量を増やすことができます。すべてのユーザは、単一の Multiplexor サーバに接続します。Multiplexor サーバは、各接続を適切な Messaging Server にリダイレクトします。

多数のユーザに電子メールサービスを提供する場合は、ユーザには複数の Messaging Server が単一のホストであるかのように表示されるよう、Messaging Multiplexor をインストールして設定することができます。

Messaging Multiplexor は iPlanet Messaging Server の一部として提供されます。MMP は Messaging Server やほかの iPlanet サーバと同時にインストールすることも、あとで別途インストールすることもできます。

MMP は以下の機能をサポートします。

- メールクライアントとの非暗号化通信および暗号化 (SSL) 通信
- 証明書に基づくクライアント認証 (75 ページの「証明書に基づくクライアント認証」)
- ユーザの事前認証 (76 ページの「ユーザの事前認証」)
- さまざまな IP アドレスをリッスンし、ユーザ ID にドメイン名を自動的に付与する仮想ドメイン (77 ページの「MMP 仮想ドメイン」)

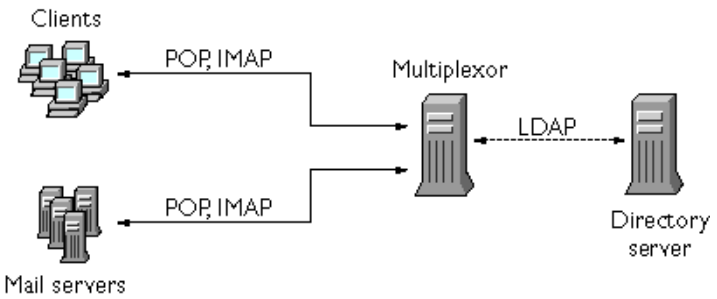
- 複数のマシンへの MMP の複数インストール (1 台に 1 つずつ)。『iPlanet Messaging Server インストールガイド』を参照
- 1 台のサーバマシン上の複数のマルチプレクサインスタンス (78 ページの「複数の Messaging Multiplexor インスタンス」)。複数のインスタンスを使って、仮想ドメインでは処理できない SSL やリッスンポートなどの設定が可能
- 高度な LDAP 検索
- 古いバージョンの POP クライアント用の POP before SMTP サービス。詳細は、428 ページの「POP before SMTP を有効にする」を参照してください。

Messaging Multiplexor のしくみ

iPlanet MMP は、メールユーザを複数のサーバマシンに分散させるマルチスレッドサーバです。MMP は、ユーザメールボックスがあるサーバマシン宛ての受信クライアント接続を処理します。クライアントは MMP に接続します。MMP はユーザの正しいサーバを判断し、そのサーバに接続し、クライアントとサーバとの間でデータの受け渡しを行います。この機能を使用すると、インターネットサービスプロバイダやその他の大規模なインストール環境では、処理能力を上げるためにメッセージストアを複数のマシンに分散しても、ユーザおよび外部クライアントに対しては単一のメールホストであるかのように機能するので、ユーザの効率を向上させ、外部クライアントに対するセキュリティを強化することができます。

図 4-1 に、MMP をインストールした場合のサーバとクライアントの関係を示します。

図 4-1 MMP をインストールした場合のクライアントとサーバ



POP、IMAP、および SMTP クライアントはすべて、Messaging Multiplexor に接続して動作します。MMP は接続を許可し、LDAP ディレクトリ検索を行い、正しい接続先にルーティングします。ほかのメールサーバをインストールした場合と同様、各ユーザは特定の Messaging Server 上の特定のアドレスとメールボックスに割り当てられます。ただし、接続はすべて MMP を経由します。

詳しく説明すると、ユーザ接続は次の手順で確立されます。

1. ユーザのクライアントが MMP に接続し、予備的な認証情報 (ユーザ名) を受け入れます。
2. MMP は Directory Server に照会して、そのユーザのメールボックスがある Messaging Server を判断します。
3. MMP は適切な Messaging Server に接続し、再度認証を行い、接続中は中継パイプとして動作します。

暗号化 (SSL) オプション

iPlanet Messaging Multiplexor は、Messaging Server とメールクライアントの間の暗号化 (SSL) 通信および非暗号化通信をサポートしています。

SSL を有効にすると、MMP の IMAP および SMTP サービスは STARTTLS をサポートします。また、SSL の IMAP、POP、および SMTP 接続で追加ポートをリッスンするように MMP を設定することもできます。

IMAP、POP、または SMTP サービスで SSL を有効にするには、それぞれ `ImapProxyAService.cfg`、`PopProxyAService.cfg`、および `SmtpProxyAService.cfg` の各ファイルを編集します。また、IMAP、POP、または SMTP サーバがセキュアサーバであるかどうかにかかわらず、`AService.cfg` ファイルの `default:ServiceList` オプションを編集し、ファイル内ですべての IMAP、POP および SMTP サーバポートを指定する必要があります。

SSL 設定パラメータはコメントアウトされているため、デフォルト設定では SSL が無効になっています。SSL を有効にするには、SSL サーバ証明書をインストールする必要があります。次にコメントを解除し、SSL パラメータを設定します。SSL パラメータのリストは、『iPlanet Messaging Server リファレンスマニュアル』に記載されています。

証明書に基づくクライアント認証

MMP では証明書マッピングファイル (`certmap`) を使ってクライアントの証明書と Users and Groups Directory Server の正しいユーザを一致させることができます。

証明書に基づくクライアント認証を使用するには、SSL 暗号化も有効にする必要があります。75 ページの「暗号化 (SSL) オプション」を参照してください。

また、ストア管理者も設定する必要があります。メール管理者を使用することもできますが、必要に応じてアクセス権を設定できるように、一意のユーザ ID (`mmpstore` など) を作成することをお勧めします。

MMP は certmap プラグインをサポートしていないことに注意してください。代わりに、certmap.conf ファイルの拡張された DNComps および FilterComps の各プロパティ値エントリを使用できます。これらの拡張されたフォーマットエントリの形式は以下のとおりです。

mapname:DNComps *FROMATTR=TOATTR*
mapname:FilterComps *FROMATTR=TOATTR*

これにより、証明書の subjectDN の *FROMATTR* 値を使って、*TOATTR=value* という要素を含む LDAP クエリを形成することができます。たとえば、証明書の subjectDN が「cn=Pilar Lorca, ou=pilar o=siroe.com」の場合、この証明書を「(uid=pilar)」という LDAP クエリにマップするには、以下の行を使用します。

mapname:FilterComps ou=uid

IMAP サービスに対して証明書に基づく認証を有効にするには、以下の手順に従います。

1. ストア管理者のユーザ ID を決定します。

   メール管理者を使用することもできますが、ストア管理者用に一意のユーザ ID (mmpstore など) を作成することをお勧めします。

2. SSL が有効になっていることを確認します。詳細については、75 ページの「暗号化 (SSL) オプション」を参照してください。

3. 設定ファイルで certmap.conf ファイルの場所を指定し、MMP が証明書に基づくクライアント認証を使用するように設定します。

4. 信頼できる認証局の証明書を少なくとも 1 つはインストールします。詳細については、408 ページの「信頼できる CA の証明書をインストールするには」を参照してください。

ユーザの事前認証

MMP には、受信ユーザとしてディレクトリにバインドし、その結果をログに記録することによってユーザを事前認証するオプションがあります。

注	ユーザの事前認証を有効にすると、サーバのパフォーマンスが低下します。

ログエントリの形式は、以下のとおりです。

date time (sid 0x*hex*) ユーザ *name* 事前認証 - クライアント IP *address*、サーバ IP *address*

date は yyyymmdd 形式、*time* は UTC (標準協定世界時、GMT (グリニッジ標準時) でもある) の hhmmss 形式であり、*hex* は 16 進数のセッション ID (sid) を表します。仮想ドメインがあれば user *name* に含まれており、IP アドレスはドットで 4 つに区切られた形式です。

MMP 仮想ドメイン

MMP 仮想ドメインはサーバの IP アドレスに関連付けられている一連の構成設定です。この機能の主な用途は、サーバ IP アドレスごとに個別のデフォルトドメインを提供することです。

ユーザは、省略形のユーザ ID または完全指定のユーザ ID (user@domain という形式) を使用して、MMP に認証を求めることができます。省略形のユーザ ID が提示されると、MMP は指定があれば DefaultDomain 設定を行います。その結果、複数のホストドメインをサポートするサイトでは、サーバ IP アドレスと MMP 仮想ドメインにそれぞれのホストドメインを関連付けるだけで、省略形のユーザ ID を使用できるようになります。

特定のホストドメインのユーザサブツリーを検索する場合は、そのドメインの LDAP ドメインツリーエントリで inetDomainBaseDN 属性を使用する方法をお勧めします。バックエンドメールストアサーバでも LDAP のユーザを検索する必要があり、さらに仮想ドメインがサポートされないため、MMP で LdapUrl を設定するのは適していません。

仮想ドメインを有効にするには、インスタンスディレクトリで ImapProxyAService.cfg、PopProxyAService.cfg、または SmtpProxyAService.cfg の各ファイルを編集し、VirtualDomainFile 設定で仮想ドメインマッピングファイルへの絶対パスを指定します。

仮想ドメインファイルの各エントリには、以下のシンタックスを使用します。

vdmap *name IPaddr*
name:parameter value

name は IP アドレスと設定パラメータを関連付けるためだけに使用するので任意の名前を使用できます。*IPaddr* はドットで 4 つに区切られた形式で、*parameter* と *value* のペアによって仮想ドメインを構成します。仮想ドメインの設定パラメータ値を設定すると、その値はグローバルな設定パラメータ値よりも優先されます。

仮想ドメインに指定できる設定パラメータは以下のとおりです。

AuthCacheSize および AuthCacheSizeTTL
AuthService
BindDN および BindPass
CertMap

ClientLookup
CRAMs
DefaultDomain
DomainDelim
HostedDomains
LdapCacheSize および LdapCacheTTL
LdapURL
MailHostAttrs
PreAuth
ReplayFormat
StoreAdmin および StoreAdminPass
SearchFormat
TCPAccess
TCPAccessAttr

注	LdapURL が正しく設定されていない場合、BindDN、BindPass、LdapCacheSize、および LdapCacheTTL の設定は無視されます。

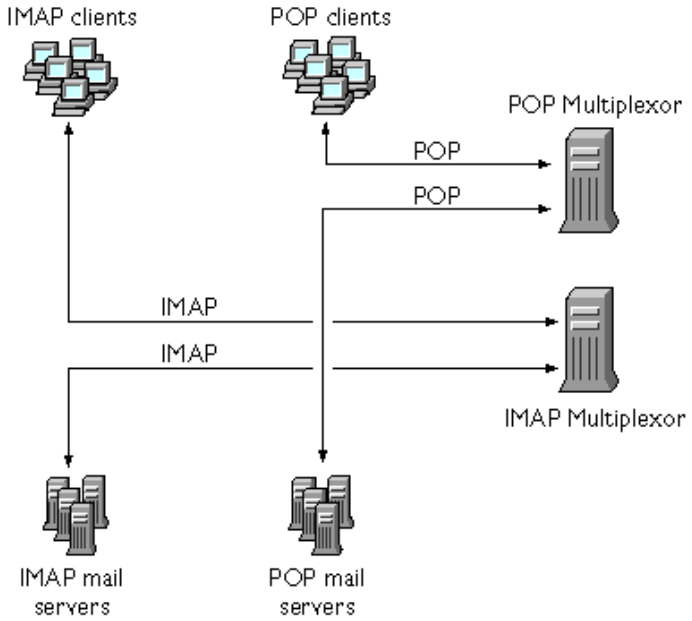
設定パラメータの詳細については、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

複数の Messaging Multiplexor インスタンス

1 台のサーバに複数の MMP インスタンスをインストールできます。それぞれのインスタンスは独立したプロセスとして実行され、異なる設定ファイルを持つことができます。複数のインスタンスは、サーバ IP アドレスやポートごとに異なる設定が必要な場合や、設定を仮想ドメインで変更できない場合に必要です。このような設定の例としては、SSL サーバ証明書があります。

図 4-1 に示すように、POP、IMAP、および SMTP の各プロトコルをすべてサポートする MMP インスタンスを 1 つ設定することもできますし、図 4-2 に示すように、プロトコルごとに個別の MMP インスタンスを作成することもできます。メッセージングサービスを複数のマシンに振り分けることで、各コンピュータのリソースのパフォーマンスを最大限に高めることができます。

図 4-2　プロトコルによって MMP インスタンスを分けた場合



複数の MMP インスタンスを作成する手順については、『iPlanet Messaging Server インストールガイド』を参照してください。

SMTP プロキシについて

MMP には SMTP プロキシが含まれていますが、デフォルトでは無効になっています。大半のサイトでは SMTP プロキシは必要ありません。インターネットメール規格には、SMTP の水平スケーラビリティ機能が十分に備わっているからです。

SMTP プロキシには有用なセキュリティ機能があります。まず、古いバージョンの POP クライアントの一部で必要な POP before SMTP 認証機能を実装するために、SMTP プロキシは POP プロキシに統合されています。詳細は、428 ページの「POP before SMTP を有効にする」を参照してください。

さらに、SMTP プロキシを使用すると、SSL アクセラレータハードウェアを最大限に活用できます。414 ページの「SMTP プロキシを使用した SSL パフォーマンスの最適化方法」を参照してください。

Messaging Multiplexor を構成する

Messaging Multiplexor を構成するには、表 4-1 に示す Messaging Multiplexor 設定ファイルの設定パラメータを手動で編集する必要があります。

表 4-1　Messaging Multiplexor の設定ファイル

ファイル	説明
PopProxyAService.cfg	POP サービス用の設定変数を指定する設定ファイル
PopProxyAService-def.cfg	POP サービスの設定テンプレート。PopProxyAService.cfg ファイルが存在しない場合は、PopProxyAService-def.cfg テンプレートがコピーされ、新しい PopProxyAService.cfg ファイルが作成される
ImapProxyAService.cfg	IMAP サービス用の設定変数を指定する設定ファイル
ImapProxyAService-def.cfg	IMAP サービスの設定テンプレート。ImapProxyAService.cfg ファイルが存在しない場合は、ImapProxyAService-def.cfg テンプレートがコピーされ、新しい ImapProxyAService.cfg ファイルが作成される
AService.cfg	開始するサービス、および POP サービスと IMAP サービスが共有するオプションを指定する設定ファイル
AService-def.cfg	開始するサービス、および POP サービスと IMAP サービスが共有するオプションを指定する設定テンプレート。AService.cfg ファイルが存在しない場合は、AService-def.cfg テンプレートがコピーされ、新しい AService.cfg ファイルが作成される
AService.rc	MMP の開始、停止、再起動、および再読み込みに使用するスクリプト 再起動後に MMP が自動的に起動されるように設定するには、AService.rc スクリプトを /etc/init.d にコピーし、適切な /etc/rc?.d ディレクトリへのシンボリックリンクを作成する (? は任意の一文字を示す)。初期化および終了に使用するスクリプトの詳細は、マニュアルページの init.d を参照

表 4-1　Messaging Multiplexor の設定ファイル (続き)

ファイル	説明
SmtpProxyAService.cfg	SMTP プロキシサービス用の設定変数を指定するオプションの設定ファイル。POP before SMTP を有効にする場合に必要。POP before SMTP を有効にしない場合でも、SSL ハードウェアに対するサポートを最大限にするのに役立つ。POP before SMTP の詳細については、428 ページの「POP before SMTP を有効にする」を参照
SmtpProxyAService-def.cfg	SMTP プロキシサービス用の設定変数を指定する設定テンプレート。SmtpProxyAService.cfg ファイルが存在しない場合は、SmtpProxyAService-def.cfg テンプレートがコピーされ、新しい SmtpProxyAService.cfg ファイルが作成される

Messaging Multiplexor の設定ファイルは、*server_root*/mmp-*hostname* ディレクトリに保存されています。*server_root* は Messaging Server をインストールしたディレクトリ、mmp-*hostname* は MMP インスタンスにちなんで名付けられたサブディレクトリを表します。たとえば、tarpit というマシンにデフォルトのインストールディレクトリを使って Messaging Multiplexor をインストールした場合、設定ファイルは /usr/iplanet/server5/mmp-tarpit に保存されます。

例として、LogDir パラメータおよび LogLevel パラメータは、すべての設定ファイルで使用されています。これらのパラメータは、ImapProxyAService.cfg ファイルでは IMAP 関連イベントのロギングパラメータを設定する目的で使われており、PopProxyAService.cfg ファイルでは POP 関連イベントのロギングパラメータを設定するために使われています。SmtpProxyAService.cfg では、SMTP プロキシ関連イベントのロギングを指定するために使われています。

ただし、AService.cfg ファイルの LogDir パラメータと LogLevel パラメータは、POP、IMAP、または SMTP サービスの起動に失敗した場合など、MMP に関する全般的な問題を記録するために使用されています。

注　MMP をインストールまたはアップグレードした場合、設定テンプレートファイルは上書きされます。

MMP 設定パラメータの詳細については、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

Messaging Multiplexor を起動するには

UNIX システム

UNIX システムで Messaging Multiplexor インスタンスを起動するには、*server_root*/mmp-*hostname* ディレクトリにある AService.rc スクリプトを以下のように指定して実行します。

```
./AService.rc [options]
```

表 4-2 に、AService.rc スクリプトのオプションパラメータを示します。

表 4-2　AService.rc スクリプトのオプションパラメータ

オプション	説明
start	MMP を起動する (別のインスタンスがすでに起動している場合でも可能)
stop	最後に起動した MMP を停止する
restart	最後に起動した MMP を停止してから、MMP を起動する
reload	実行中の MMP が、アクティブな接続を中断せずに設定情報を再読み込みするようにする

Windows NT システム

Windows NT で Messaging Multiplexor のインスタンスを起動するには、Windows NT コントロールパネルの「サービス」で「開始」をクリックします。また、「停止」をクリックすると MMP を停止できます。表 4-3 に、サービスのオプションとその説明を示します。

表 4-3　Windows NT MMP サービスのオプション

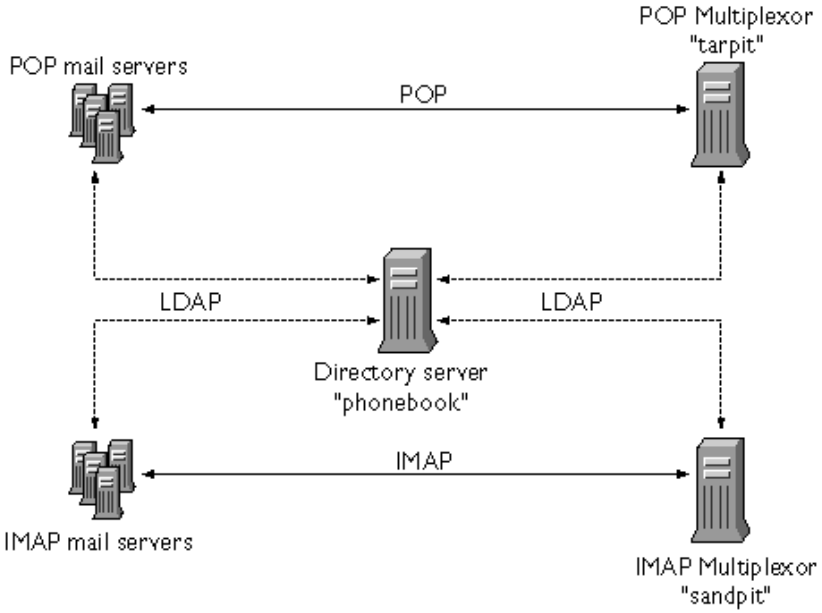
オプション	説明
start	コントロールパネルで MMP を起動するか (すでに実行中の場合も含む)、コマンドラインで AService.exe start コマンドを実行する
stop	コントロールパネルで最後に起動した MMP を停止するか、コマンドラインで AService.exe stop コマンドを実行する
restart	Windows NT で再起動するために、最後に起動した MMP を停止してから、MMP を起動する
reload	MMP を再読み込みするために、mmp-instance ディレクトリに移動し、コマンドラインで AService.exe refresh コマンドを実行する

トポロジの例

Siroe Corporation という会社には Messaging Multiplexor をインストールしたマシンが 2 台あり、どちらのマシンも複数の Messaging Server をサポートしているというシナリオを想定します。POP および IMAP のユーザメールボックスは複数の Messaging Server マシンに振り分けられており、それぞれのサーバは POP 専用または IMAP 専用となっています (クライアントアクセスを POP サービスだけに限定するには、`ServiceList` から `ImapProxyAService` エントリを削除します。同様に IMAP サービスだけに限定するには、`ServiceList` から `PopProxyAService` エントリを削除します)。どちらの Messaging Multiplexor も POP だけ、または IMAP だけしかサポートしません。LDAP ディレクトリサービスは、別の専用マシンに置かれます。

このトポロジを、図 4-3 に示します。

図 4-3　複数の MMP による複数の Messaging Server のサポート



IMAP の構成例

図 4-3 の IMAP Messaging Multiplexor は、2 個のプロセッサを持つ `sandpit` というマシンにインストールされています。この Messaging Multiplexor は、IMAP 接続の標準ポート (143) を待機しています。Messaging Multiplexor はユーザメールボックスの情報を扱うホスト `phonebook` の LDAP サーバと通信し、適切な IMAP サーバに接続をルーティングします。この Multiplexor は、IMAP の capability 文字列を無効にし、仮想ドメインファイルを提供し、SSL 通信をサポートします。

この例の ImapProxyAService.cfg 設定ファイルの内容は、以下のとおりです。

```
default:LdapUrl              ldap://phonebook/o=Siroe.com
default:LogDir               /usr/iplanet/server5/mmp-sandpit/log
default:LogLevel             5
default:BindDN               "cn=Directory Manager"
default:BindPass             secret
default:BacksidePort         143
default:Timeout              1800
default:Capability           "IMAP4 IMAP4rev1 ACL QUOTA LITERAL+ NAMESPACE
UIDPLUS CHILDREN LANGUAGE XSENDER X-NETSCAPE XSERVERINFO AUTH=PLAIN"
default:SearchFormat         (uid=%s)
default:SSLEnable            yes
default:SSLPorts             993
default:SSLSecmodFile        /usr/iplanet/server5/mmp-sandpit/secmod.db
default:SSLCertFile          /usr/iplanet/server5/mmp-sandpit/cert7.db
default:SSLKeyFile           /usr/iplanet/server5/mmp-sandpit/key3.db
default:SSLKeyPasswdFile     ""
default:SSLCipherSpecs       all
default:SSLCertNicknames     Siroe.com Server-Cert
default:SSLCacheDir          /usr/iplanet/server5/mmp-sandpit
default:SSLBacksidePort      993
default:VirtualDomainFile    /usr/iplanet/server5/mmp-sandpit/vdmap.cfg
default:VirtualDomainDelim   @
default:ServerDownAlert      "your IMAP server appears to be temporarily out of
service"
default:MailHostAttrs        mailHost
default:PreAuth              no
default:CRAMs                no
default:AuthCacheSize        10000
default:AuthCacheTTL         900
default:AuthService          no
default:AuthServiceTTL       0
default:BGMax                10000
default:BGPenalty            2
default:BGMaxBadness         60
default:BGDecay              900
default:BGLinear             no
default:BGExcluded           /usr/iplanet/server5/mmp-sandpit/bgexcl.cfg
default:ConnLimits           0.0.0.0|0.0.0.0:20
default:LdapCacheSize        10000
default:LdapCacheTTL         900
default:HostedDomains        yes
default:DefaultDomain        Siroe.com
```

POP の構成例

図 4-3 の POP Messaging Multiplexor は、4 個のプロセッサを持つ tarpit というマシンにインストールされています。この Messaging Multiplexor は POP 接続の標準ポート (110) を待機しています。Messaging Multiplexor はユーザメールボックス情報を扱うホスト phonebook の LDAP サーバと通信し、適切な POP サーバに接続をルーティングします。さらに、この Multiplexor は、スプーフメッセージファイルも提供します。

この例の PopProxyAService.cfg 設定ファイルの内容は、以下のとおりです。

```
default:LdapUrl             ldap://phonebook/o=Siroe.com
default:LogDir              /usr/iplanet/server5/mmp-tarpit/log
default:LogLevel            5
default:BindDN              "cn=Directory Manager"
default:BindPass            password
default:BacksidePort        110
default:Timeout             1800
default:Capability          "IMAP4 IMAP4rev1 ACL QUOTA LITERAL+ NAMESPACE
UIDPLUS CHILDREN LANGUAGE XSENDER X-NETSCAPE XSERVERINFO AUTH=PLAIN"
default:SearchFormat        (uid=%s)
default:SSLEnable           no
default:VirtualDomainFile   /usr/iplanet/server5/mmp-tarpit/vdmap.cfg
default:VirtualDomainDelim  @
default:MailHostAttrs       mailHost
default:PreAuth             no
default:CRAMs               no
default:AuthCacheSize       10000
default:AuthCacheTTL        900
default:AuthService         no
default:AuthServiceTTL      0
default:BGMax               10000
default:BGPenalty           2
default:BGMaxBadness        60
default:BGDecay             900
default:BGLinear            no
default:BGExcluded          /usr/iplanet/server5/mmp-tarpit/bgexcl.cfg
default:ConnLimits          0.0.0.0|0.0.0.0:20
default:LdapCacheSize       10000
default:LdapCacheTTL        900
default:HostedDomains       yes
default:DefaultDomain       Siroe.com
```

Messenger Express Multiplexor について

iPlanet Messenger Express Multiplexor は、HTTP アクセスサービスへの単一の接続ポイントとして機能する特別なサーバです。Messenger Express は、iPlanet Messaging Server HTTP サービスに対するクライアントインタフェースです。すべてのユーザがこのメッセージングプロキシサーバに接続し、ここで該当するメールボックスに転送されます。このため、メールユーザには複数の Messaging Server が単一のホストであるかのように表示されます。

iPlanet Messaging Multiplexor (MMP) は POP および IMAP サーバに接続しますが、Messenger Express Multiplexor は HTTP サーバに接続します。つまり、Messenger Express Multiplexor と Messenger Express との関係は、MMP と POP や IMAP との関係と同じです。

MMP と同様に、Messenger Express Multiplexor でも次の機能をサポートします。

- メールクライアントとの非暗号化通信および暗号化 (SSL) 通信

  SSL の設定に関する詳細は、第 12 章の「セキュリティとアクセス制御」を参照してください。

- ホストドメイン

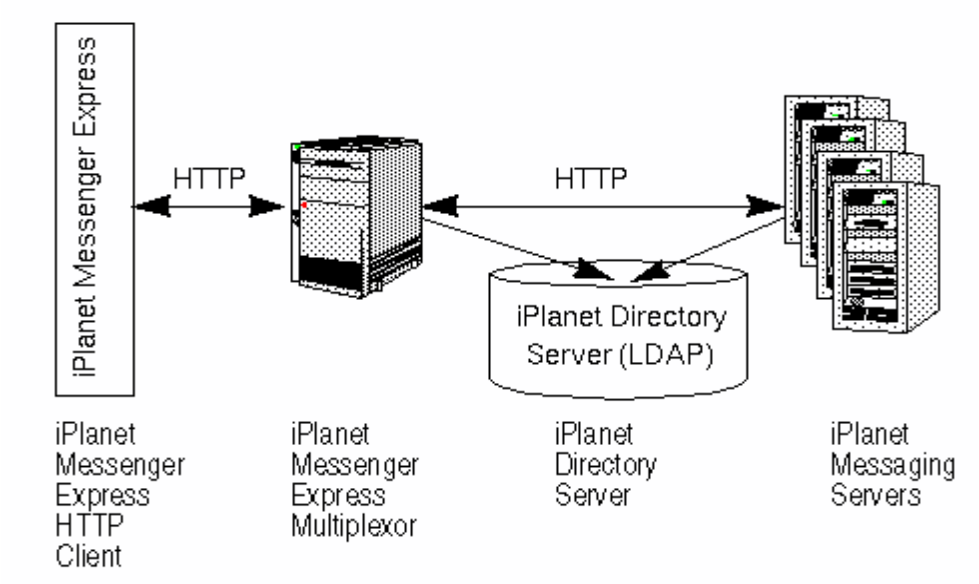
  詳細は、『iPlanet Messaging Server プロビジョニングガイド』を参照してください。

MMP とは異なり、Messenger Express Multiplexor は `mshttpd` サービスに組み込まれているため、ロギングと構成には同じ機能が使用されます。

Messenger Express Multiplexor のしくみ

Messenger Express Multiplexor はマルチプレクサとして機能するプロキシメッセージングサーバで構成されており、ユーザが iPlanet Messaging Server (Messenger Express) の HTTP サービスに接続できるようにします。Messenger Express Multiplexor を使用すると、複数のサーバマシンにメールボックスを分散できるようになります。クライアントは iPlanet Messenger Express にログオンすると Multiplexor に接続します。Multiplexor はユーザの正しいサーバを判断し、そのサーバに接続し、クライアントとサーバとの間でデータの受け渡しを行います。この機能を使用すると、大規模なインストール環境では、処理能力を上げるためにメッセージストアを複数のマシンに分散しても、ユーザおよび外部クライアントに対しては単一のメールホストであるかのように機能するので、ユーザの効率を向上させ、外部クライアントに対するセキュリティを強化することができます。88 ページの図 4-4 に、iPlanet Messaging Server での Messenger Express Multiplexor の位置を示します。

図 4-4　iPlanet Messenger Express Multiplexor の概要



Messenger Express Multiplexor は、iPlanet Messenger Express クライアントと iPlanet Messaging Server の間に入り、両者の接続を許可して正しくルーティングします。ほかのメールサーバをインストールした場合と同様、各ユーザは特定の Messaging Server 上の特定のアドレスとメールボックスに割り当てられます。ただし、HTTP 接続はすべて Messenger Express Multiplexor を経由します。

詳しく説明すると、ユーザ接続は次の手順で確立されます。

1. ユーザのクライアントが Messenger Express Multiplexor に接続し、予備的な認証情報を受け入れます。
2. Messenger Express Multiplexor は Directory Server に照会して、そのユーザのメールボックスがある Messaging Server を判断します。
3. Messenger Express Multiplexor は関連する Messaging Server に接続し、再度認証を行い、接続中は中継パイプとして動作します。

Messenger Express Multiplexor を設定するには

ここでは、Messenger Express Multiplexor の設定手順について説明します。以下の項目があります。

- 89 ページの「プロキシマシンに Messaging Server をインストールする」
- 89 ページの「Multiplexor パラメータを設定するには」
- 91 ページの「Messenger Express Multiplexor を有効にする」

プロキシマシンに Messaging Server をインストールする

まず、Messenger Express Multiplexor になるプロキシマシンに Messaging Server をインストールします。具体的なインストール手順については、『iPlanet Messaging Server インストールガイド』を参照してください。

Messaging Server は、バックエンドメッセージングサーバを指す Users and Groups Directory Server に構成してください。このディレクトリサーバは、Messenger Express Multiplexor を介して、Messenging Server でユーザを認証するために使用します。

Multiplexor パラメータを設定するには

プロキシマシンに Messaging Server をインストールしたら、Messenger Express Multiplexor パラメータを設定します。

1. 必要なバックエンド Messaging Server の情報を集めます。

   バックエンドメッセージングサーバで `configutil` コマンドを実行し、パラメータの値を設定します。パラメータの値については、この節の後半で説明します。設定を正常に行うには、プロキシマシンとバックエンドメッセージングサーバの設定を同じにする必要があります。Multiplexor はプロキシマシンで有効にします。

2. Messenger Express Multiplexor の設定パラメータを設定します。

   設定値を指定するには、プロキシマシンの Messaging Server の *server_root*`/bin/msg-`*instance*`/configutil` ディレクトリで `configutil` コマンドを実行します。設定値がバックエンドメッセージングサーバの値と同じであることを確認します。

   `configutil` コマンドの実行の詳細は、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

以下の節では、Messenger Express Multiplexor の設定に必要な `configutil` パラメータについて説明します。

- 90 ページの「LDAP パラメータ」
- 90 ページの「dcroot」
- 90 ページの「デフォルトドメイン」

- 90 ページの「ログイン区切り」

LDAP パラメータ

Messenger Express Multiplexor を有効にする前に、Directory Server パラメータを正しく指定する必要があります。LDAP パラメータを指定するには、適切なバックエンド Messaging Server のインスタンスディレクトリで次のコマンドを実行します。

- `configutil -o local.ugldaphost`

  バックエンドメッセージングサーバが使用する、ユーザおよびグループの LDAP Directory Server を表すパラメータです。`ldaphost` には、バックエンドメッセージングサーバが使用するものと同じ値、または同じデータを含む複製された LDAP サーバを指定します。

- `configutil -o local.ugldapbinddn`
  `configutil -o local.ugldapbindcred`

  Users and Groups Directory Server 管理者の DN とパスワードを表すパラメータです。`ldapbinddn` も `ldapbindcred` も、バックエンドメッセージングサーバの指定と同じである必要があります。

dcroot

dcroot が正しく指定されていることを確認する必要があります。*dcroot* を指定するには、適切な Messaging Server インスタンスディレクトリで次のコマンドを実行します。

```
configutil -o service.dcroot
```

デフォルトドメイン

Messaging Server のデフォルトドメイン (*defaultdomain*) が正しく指定されていることを確認する必要があります。Messaging Server のデフォルトドメインを指定するには、適切な Messaging Server インスタンスディレクトリで次の `configutil` コマンドを実行します。

```
configutil -o service.defaultdomain
```

ログイン区切り

ログイン区切り (*loginseparator*) は、バックエンドメッセージングサーバで使用するものと同じにします。Messaging Server のログイン区切りを指定するには、適切なバックエンドメッセージングサーバのインスタンスディレクトリで次の `configutil` コマンドを実行します。

```
configutil -o service.loginseparator
```

Messenger Express Multiplexor を有効にする

設定パラメータを指定したら、プロキシマシンの Messenger Express Multiplexor を有効にすることができます。プロキシマシンの Messaging Server インスタンスにある *server_root*/bin/msg-*instance*/configutil ディレクトリで、次の configutil コマンドを実行します。

```
configutil -o local.service.http.proxy -v 1
```

1 を指定すると Messenger Express Multiplexor が有効になります。デフォルトは 0 です。

非ローカルユーザ (ログインしたサーバにメールホストがないユーザ) がログインした場合、local.service.http.proxy の値が 0 であれば、このユーザは自分のホストに転送されます。ユーザには、ホスト名が変更されたことがわかります。したがって、Multiplexor は有効になっていません。

local.service.http.proxy の値が 1 の場合は、Multiplexor が有効になり、ホスト名は変更されず、非ローカルメールユーザからは Messaging Server 全体が 1 台のホストであるかのように見えます。

ローカルユーザ (ログインしたサーバがメールホストであるユーザ) の場合は、local.service.http.proxy のパラメータ値とは無関係にサーバのローカルメッセージストアが使用されます。同じ Messaging Server 上でプロキシユーザとローカルユーザを共存させることもできます。

configutil コマンドの詳細は、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

設定をテストする

ここでは、Messenger Express Multiplexor の設定をテストし、ログファイルのメッセージを検索する方法を説明します。Messenger Express Multiplexor が設定され、有効になっていることを前提にしています。

Messenger Express クライアントにアクセスする

インストール状態をテストするには、Messenger Express 製品についての知識が必要です。また、テストアカウントを作成しておく必要があります。

Messenger Express Multiplexor プロキシをテストするには、次の手順に従います。

1. ブラウザに次のように入力して、Messenger Express Multiplexor を介して Messenger Express に接続します。

   `http://`*msgserver_name*

   たとえば、以下のようになります。

   `http://budgie.sesta.com`

2. 作成済みのテストアカウントを使用して、Messenger Express にログインします。
3. 正しくログインし、バックエンドメッセージングサーバのメッセージにアクセスできる必要があります。
4. Messenger Express を介してログインすると Messaging Server 名が変更される場合は、`local.service.http.proxy` が `1` に設定されており、メッセージングプロキシサーバが再起動されているかどうかを確認してください。Messenger Express Multiplexor が有効であれば、ユーザからは 1 台のメールホストであるかのように見えます。

エラーメッセージ

ユーザ ID とパスワードを入力し「接続」をクリックするとエラーメッセージが表示される場合は、プロキシマシンの HTTP ログファイルを確認してください。エラーメッセージを確認するには、*server_root*`/msg-`*instance*`/log/http/` ディレクトリに移動します。多くの場合、エラーメッセージには問題を解決するための情報が含まれています。問題を解決するための十分な情報が含まれていない場合は、iPlanet のカスタマサポートに連絡してください。

Messenger Express Multiplexor を管理する

ここでは、Messenger Express Multiplexor の基本的な管理機能を説明します。

SSL を設定および管理する

Messenger Express Multiplexor の SSL (Secure Sockets Layer) の設定と管理については、410 ページの「SSL を有効化し符号化方式を選択するには」を参照してください。

複数プロキシサーバの設定

単一の名前でアドレス指定される複数の Messenger Express Multiplexor を設定する場合は、セッション対応の負荷分散デバイスを使用できます。このデバイスにより、任意のクライアントからのすべての要求を特定のサーバにルーティングできます。

バージョンの異なる Messaging Server と Messenger Express Multiplexor を管理する

Messenger Express Multiplexor とバックエンドメールホストで異なるバージョンの iPlanet Messaging Server を使う場合は、Messenger Express の静的ファイルを更新してサーバの互換性を確保する必要があります。

Messenger Express インタフェースを構成する静的ファイルは、ユーザのメールホストではなく Messenger Express Multiplexor から直接提供されます。ファイルは Messenger Express Multiplexor の *server_root*/msg-*instance*/html ディレクトリに格納されています。

サーバの互換性を確保するためにファイルを更新するには、新しいバージョンの Messaging Server にある *server_root*/msg-*instance*/html ディレクトリの内容 (Messenger Express インタフェースを構成する静的ファイルが含まれる) を、古いバージョンの Messaging Server にある同じディレクトリの内容にすべて置き換えます。

たとえば、バックエンドメッセージングサーバで iPlanet Messaging Server 5.1 を使用し、Messenger Express Multiplexor には iPlanet Messaging Server 5.2 をインストールしている場合は、*server_root*/msg-*instance*/html ディレクトリの内容を iPlanet Messaging Server 5.1 を使用するバックエンドサーバの同じディレクトリの内容にすべて置き換える必要があります。最終的に、iPlanet Messaging Server 5.1 から iPlanet Messaging Server 5.2 にアップグレードするときに、Messenger Express Multiplexor サーバの *server_root*/msg-*instance*/html ディレクトリにある静的ファイルも更新することができます。

第 5 章

MTA の概念

この章では、MTA の概念について説明します。この章には、以下の節があります。

MTA の機能

MTA (97 ページの図 5-2) は以下の機能を実行する Messaging Server (96 ページの図 5-1) の構成要素です。

- **メッセージのルーティング** - メッセージを受け取り、A) 別の SMTP ホスト、B) ローカルメッセージストア、または C) 処理プログラム (ウィルスチェックなど) にルーティングする
- **メッセージのブロッキング** - 特定のソースや宛先の IP アドレス、メールアドレス、ポート、チャネル、ヘッダー文字列などに基づいて、メッセージをブロックまたは許可する
- **アドレスの書き換え** - 受信したアドレスの `From:` や `To:` を必要な形式に書き換える
- **メッセージの処理** - さまざまな種類のメッセージを処理する。たとえば、以下のような処理を行う
  - エイリアスのエクスパンド

- SMTP コマンドおよびプロトコルサポートの管理
- SASL サポートの提供
- 指定したアドレス数を超過した場合のメッセージの保留
- ウィルスチェックやメールファイリングプログラムなど、サイト提供プログラムへのメッセージの配信
- メッセージ部分ごとのメッセージ変換の実行
- 配信状況通知メッセージのカスタマイズ

図 5-1　iPlanet Messaging Server の簡易コンポーネント表示 (Messenger Express では表示されない)

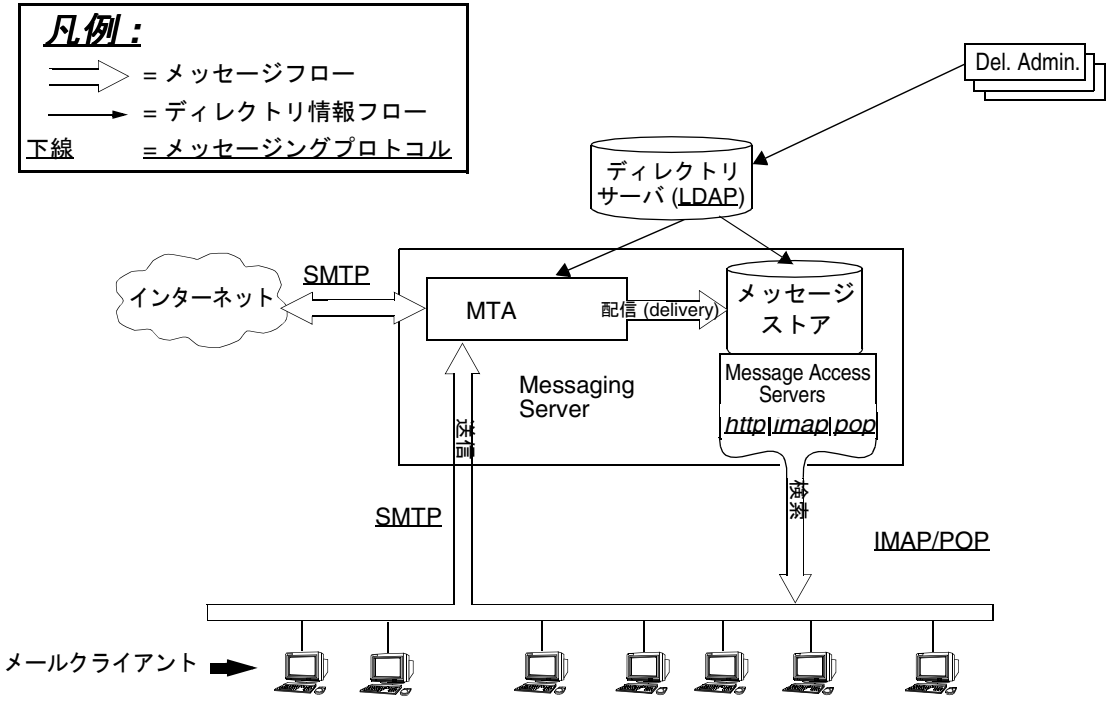
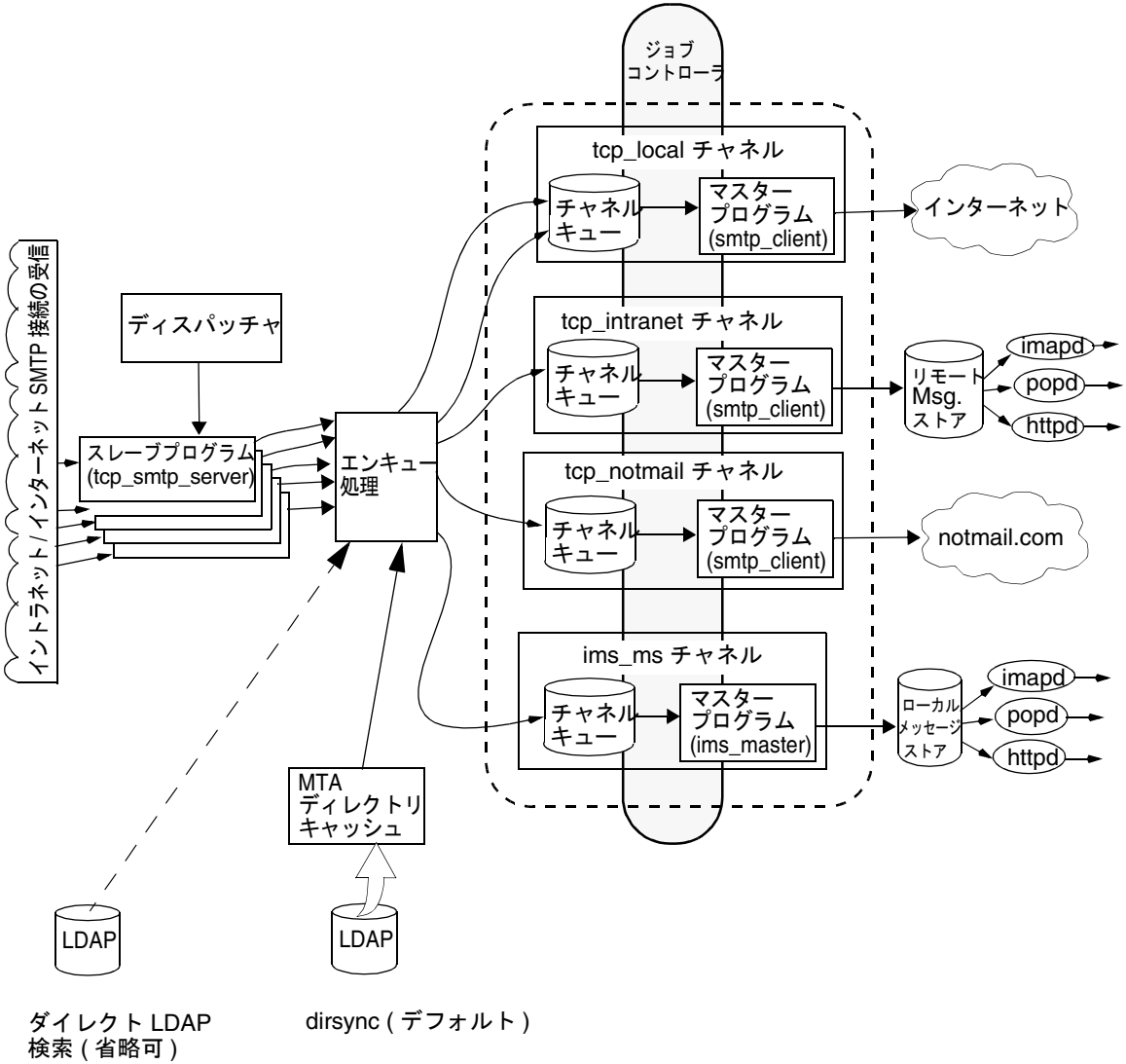


図 5-2　MTA のアーキテクチャ



MTA アーキテクチャとメッセージフローの概要

ここでは、MTA のアーキテクチャとメッセージフローの概要を簡単に説明します (図 5-2)。

ディスパッチャと SMTP サーバ (スレーブプログラム)

SMTP セッションを介して、インターネットまたはイントラネットから MTA にメッセージが届きます。MTA が SMTP 接続要求を受信すると、MTA ディスパッチャ (エージェントを振り分けるマルチスレッド接続) はスレーブプログラム (`tcp_smtp_server`) を実行して SMTP セッションを処理します。さらにセッションが要求されると、ディスパッチャは SMTP サーバプログラムを起動して、それぞれのセッションを処理します。ディスパッチャとスレーブプログラムにより、受信メッセージごとにさまざまな機能が実行されます。次の 3 つの基本機能があります。

- メッセージのブロッキング - 特定の IP アドレス、メールアドレス、ポート、チャネル、ヘッダー文字列などを含むメッセージをブロックする (第 10 章「メールのフィルタリングとアクセス制御」)
- アドレスの変更 - 受信したアドレスの `From:` や `To:` を必要な形式に書き換える
- チャネルのエンキュー処理 - アドレスに書き換え規則を適用し、メッセージを送信するチャネルを決定する

詳細は、100 ページの「ディスパッチャ」を参照してください。

エンキュー

配信のこの段階ではさまざまなタスクが実行されますが、主なタスクは以下のとおりです。

- エイリアスをエクスパンドする
- アドレスに書き換え規則を適用してメッセージをキューに入れるチャネルを決定し、アドレスのドメイン部分を正しい形式または必要な形式に書き換える
- チャネルキーワードを処理する
- メッセージを該当するチャネルキューに送信する

チャネル

チャネルは、メッセージを処理するための基本的な MTA コンポーネントです。チャネルは、ほかのシステム (ほかの MTA、ほかのチャネル、ローカルメッセージストアなど) とのメッセージ接続を表します。メールが届くと、メッセージのソースや宛先によってルーティングや処理方法が異なります。たとえば、ローカルメッセージストアに配信されるメールと、インターネットに配信されるメールと、メールシステムの

別の MTA に配信されるメールは、それぞれ別の方法で処理されます。チャネルは、各接続に必要な処理とルーティングをカスタマイズするしくみを提供します。デフォルトの設定では、メッセージの大半はインターネット、イントラネット、およびローカルのメッセージを扱う 1 本のチャネルに入ります。

特定の状況のための特殊なチャネルを作成することもできます。たとえば、メールの処理が非常に遅いインターネットドメインがあり、このドメイン宛てのメールがあると MTA の処理が停滞するとします。このような場合は、処理が遅いドメイン宛てのすべてのメッセージを処理する特別なチャネルを作成すると、このドメインのボトルネックが解消されます。

アドレスのドメイン部分は、メッセージがどのチャネルのキューに入れられるのかを決定します。ドメインを読み取って適切なチャネルを決定するしくみを、書き換え規則と呼びます (102 ページの「書き換え規則」を参照)。

チャネルは通常、マスタープログラムというチャネル処理プログラムとチャネルキューで構成されています。スレーブプログラムが該当するチャネルキューにメッセージを配信すると、マスタープログラムが必要な処理とルーティングを行います。チャネルの指定と設定は、書き換え規則と同様、`imta.cnf` ファイルで行います。チャネルエントリの例を次に示します。

```
tcp_intranet smtp mx single_sys subdirs 20 noreverse maxjobs 7
SMTP_POOL  maytlsserver allowswitchchannel  saslswitchchannel
tcpauth
tcpintranet-daemon
```

この場合、最初の単語 `tcp_intranet` はチャネル名です。最後の単語はチャネルタグです。チャネル名とチャネルタグの間にある単語はチャネルキーワードで、メッセージの処理方法を表します。さまざまなキーワードを使って、さまざまな方法でメッセージを処理できます。チャネルキーワードの詳しい説明は、『iPlanet Messaging Server リファレンスマニュアル』と第 8 章「チャネル定義を設定する」にあります。

メッセージの配信

メッセージが処理されると、マスタープログラムはメッセージの配信パスに沿って次の送信先にメッセージを送ります。次の送信先が予定した受取人のメールボックスであることもあれば、別の MTA や別のチャネルであることもあります。この図では別のチャネルへの転送は表示されていませんが、そのようなケースもよくあります。

ディスパッチャ

ディスパッチャは、複数のマルチスレッドサーバ処理が SMTP 接続サービスを分担できるようにする、マルチスレッドディスパッチエージェントです。ディスパッチャを使用すると、複数のマルチスレッド SMTP サーバを同時に実行し、同じポートへの接続を処理できるようになります。さらに、それぞれのサーバで 1 つ以上のアクティブな接続が可能になります。

ディスパッチャは、その設定に指定されている TCP ポートの中心的なレシーバとして機能します。定義された各サービスに対して、ディスパッチャは 1 つまたは複数の SMTP サーバプロセスを作成し、確立後の接続を処理します。

通常、ディスパッチャは、定義された TCP ポートの接続を受信すると、そのポートにおけるサービスのワーカープロセスのプールを確認し、その接続用に最適なワーカープロセスを選択します。適当なワーカープロセスがない場合、ディスパッチャはこの接続と後続の接続を処理するための新しいワーカープロセスを作成します。また、ディスパッチャは、今後の受信接続を予測して、新しいワーカープロセスを作成することもできます。ディスパッチャのさまざまなサービスを制御するための設定オプションがいくつかあります。これらの設定オプションは特に、ワーカープロセス数、および各ワーカープロセスが処理できる接続の数を制御するのに使用されます。

詳細については、132 ページの「ディスパッチャ設定ファイル」を参照してください。

サーバプロセスの作成と有効期限

ディスパッチャの自動ハウスキーピング機能により、新規サーバプロセスの作成や、アイドル状態の古いサーバプロセスの有効期限を制御することができます。ディスパッチャの動作を制御する基本的なオプションは、MIN_PROCS と MAX_PROCS です。MIN_PROCS は、受信接続用に一定のサーバプロセス数を待機させることにより、一定レベルのサービスを確実に提供します。一方、MAX_PROCS は、指定したサービスに対して同時にアクティブにできるサーバプロセス数の上限を設定します。

すでに処理可能な最大数の接続を処理しているため、またはプロセスの終了がスケジュールされているために、動作中のサーバプロセスが接続を受け入れられないことがあります。ディスパッチャは、今後の接続に役立つよう追加のプロセスを作成することができます。

MIN_CONNS および MAX_CONNS オプションを使うと、サーバプロセス間で接続を分散できます。MIN_CONNS はサーバプロセスが「十分にビジー」であることを示す接続数を指定し、MAX_CONNS はサーバプロセスが「最高にビジー」な状態となる場合の接続数を指定するものです。

通常、現在のサーバプロセス数が MIN_PROCS 未満である場合、または既存のサーバプロセスがすべて「十分にビジー」(各サーバプロセスに対し、現在アクティブな接続の数が MIN_CONNS 以上である) である場合、ディスパッチャは新しいサーバプロセスを作成します。

たとえば UNIX システムの kill コマンドによってサーバプロセスが突然終了した場合、ディスパッチャは新しい接続ごとに新規サーバプロセスを作成します。

ディスパッチャの設定の詳細については、132 ページの「ディスパッチャ設定ファイル」を参照してください。

ディスパッチャを起動および停止するには

ディスパッチャを起動するには、次のコマンドを実行します。

```
imsimta start dispatcher
```

このコマンドには、ディスパッチャが管理するように設定された MTA のコンポーネントを起動するために以前使用していた、ほかのすべての imsimta start コマンドが組み込まれています。以前のコマンドはすべて無効になっています。特に、imsimta start smtp は使用しないでください。無効になったコマンドを実行しようとすると、MTA によって警告メッセージが表示されます。

ディスパッチャを終了するには、次のコマンドを実行します。

```
imsimta stop dispatcher
```

ディスパッチャの終了時にサーバプロセスがどのように処理されるかは、その基礎となっている TCP/IP パッケージによって決まります。ディスパッチャに適用される MTA の設定やオプションを変更した場合は、ディスパッチャを必ず再起動して新しい設定やオプションを有効にします。

ディスパッチャを再起動するには、次のコマンドを実行します。

```
imsimta restart dispatcher
```

ディスパッチャを再起動すると、実行中のディスパッチャが終了し、新しいディスパッチャが起動します。

書き換え規則

書き換え規則には、以下の目的があります。

- アドレスのドメイン部分を適切な形式や希望の形式に書き換える方法を指定する
- アドレスを書き換えたあとにメッセージをキューに入れるためのチャネルを決定する

書き換え規則にはそれぞれパターンとテンプレートがあります。パターンは、アドレスのドメイン部分と照合する文字列です。テンプレートは、ドメイン部分がパターンと一致した場合に実行するアクションを指定します。テンプレートは、1) アドレスを書き換える方法を指定する指示のセット (一連の制御文字) と、2) メッセージの送信先のチャネル名で構成されます。アドレスの書き換え後、メッセージは予定された受取人に配信するために宛先チャネルに入れられます。

書き換え規則の例を次に示します。

```
siroe.com               $U%$D@tcp_siroe-daemon
```

`siroe.com` はドメインパターンです。アドレスに `siroe.com` を含むメッセージはテンプレートの指示 (`$U%$D`) に基づいて書き換えられます。`$U` は、書き換えられたアドレスでも同じユーザ名を使うように指定します。`%` は、書き換えられたアドレスでも同じドメイン区切り文字を使用するように指定します。`$D` は、パターンと一致したドメイン名を使うように指定します。`@tcp_siroe-daemon` は、書き換えられたアドレスのメッセージがチャネル `tcp_siroe-daemon` に送信されるように指定します。詳細については、第 7 章「書き換え規則を設定する」を参照してください。

書き換え規則の設定の詳細については、111 ページの「MTA 設定ファイル」および第 7 章「書き換え規則を設定する」を参照してください。

チャネル

チャネルは、メッセージを処理するための基本的な MTA コンポーネントです。チャネルは、別のコンピュータシステムまたはシステムグループとの接続を表します。実際のハードウェア接続やソフトウェア転送は、チャネルによって大きく異なることがあります。

チャネルには、以下のような機能があります。

- メッセージをリモートシステムに送信し、その後メッセージをキューから削除する
- リモートシステムからメッセージを受信し、適切なチャネルキューに保存する
- メッセージをローカルのメッセージストアに配信する

- メッセージを特殊処理プログラムに配信する

メッセージは、MTA に入るときにチャネルを介してキューに入れられ、MTA から出るときにキューから取り出されます。通常、メッセージは 1 つのチャネルを介して入り、別のチャネルを介して送り出されます。チャネルは、キューからのメッセージの取り出し、メッセージの処理、別の MTA チャネルのキューへのメッセージの保存などを行います。

マスタープログラムとスレーブプログラム

通常、各チャネルにはマスターとスレーブの 2 つのプログラムがあります。スレーブプログラムは、ほかのシステムからのメッセージを受け取り、そのメッセージをチャネルのメッセージキューに追加します。マスタープログラムは、チャネルからほかのシステムにメッセージを転送します。

たとえば、SMTP チャネルには、メッセージを送信するマスタープログラムと、メッセージを受信するスレーブプログラムがあります。これらは、それぞれ SMTP クライアントおよびサーバに相当します。

通常、マスタープログラムは、MTA が発した送信接続を管理します。マスターチャネルプログラムには、以下のような機能があります。

- ローカルの処理要求に応えて起動する
- チャネルメッセージキューからメッセージを取り出す
- 宛先の形式が、キューにあるメッセージの形式と異なる場合は、必要に応じて、アドレス、ヘッダー、および内容の変換を行う
- メッセージのネットワーク転送を開始する

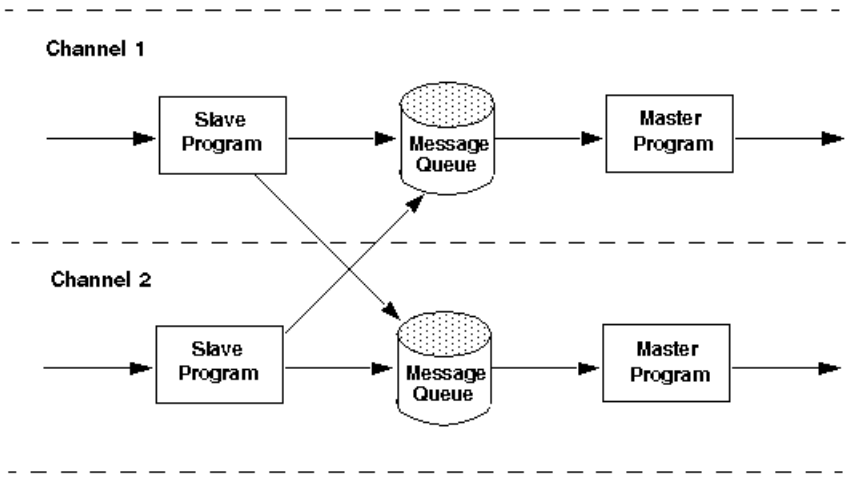
通常、スレーブプログラムは、MTA が外部要求に応答するための受信接続を受け入れます。スレーブチャネルプログラムには、以下のような機能があります。

- 外部イベントまたはローカル要求に応えて起動する
- メッセージをチャネルキューに入れる。宛先チャネルは、書き換え規則でエンベロープアドレスを渡すと決定される

たとえば、図 5-3 では、チャネル 1 とチャネル 2 の 2 つのチャネルプログラムが示されています。チャネル 1 のスレーブプログラムは、リモートシステムからメッセージを受信します。スレーブプログラムは、アドレスを確認して必要な書き換え規則を適用し、書き換えられたアドレスに基づいてメッセージを適切なチャネルメッセージキューに入れます。

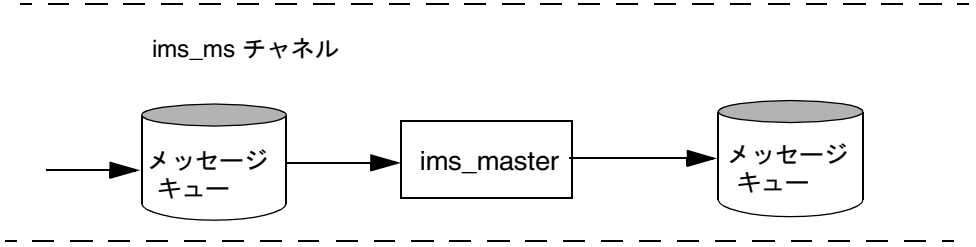
マスタープログラムは、キューからメッセージを取り出し、メッセージのネットワーク転送を開始します。ただし、マスタープログラムは、自分のチャネルキューにあるメッセージしか取り出せません。

図 5-3　マスタープログラムとスレーブプログラム



通常、1 つのチャネルにはマスタープログラムとスレーブプログラムの両方がありますが、スレーブプログラムまたはマスタープログラムしかないチャネルもあります。たとえば、Messaging Server で提供される `ims-ms` チャネルには、マスタープログラムしかありません。このチャネルでは、図 5-4 に示すように、キューからのメッセージの取り出しとローカルメッセージストアへの送信だけを行います。

図 5-4　ims-ms チャネル



チャネルメッセージキュー

すべてのチャネルに、メッセージキューが関連付けられています。メッセージがメッセージングシステムに入ると、スレーブプログラムがメッセージを入れるキューを決定します。キューに入れられたメッセージは、チャネルキューディレクトリのメッセージファイル内に保存されます。デフォルトでは、これらのディレクトリは */server_instance*/imta/queue/*channel*/* に保存されます。

チャネル定義

チャネル定義は MTA 設定ファイル (imta.cnf) の後半で、書き換え規則のあとに記載されています (111 ページの「MTA 設定ファイル」を参照)。設定ファイル内で最初に現れる空白行は、書き換え規則の終了とチャネル定義の開始を表します。

チャネル定義には、チャネル名、チャネルの設定を定義するキーワードのオプションリスト、および一意のチャネルタグが含まれています。チャネルタグは書き換え規則で使用され、メッセージをチャネルにルーティングします。チャネル定義は 1 行の空白行によって区切られています。1 つのチャネル定義の中にコメント行を含めることはできますが、空白行を含めることはできません。

```
[blank line]
! sample channel definition
Channel_Name  keyword1  keyword2
Channel_Tag
[blank line]
```

チャネル定義を総称してチャネルホストテーブルと呼びます。個々のチャネル定義はチャネルブロックと呼ばれます。たとえば、図 5-5 のチャネルホストテーブルには、チャネル定義つまりチャネルブロックが 3 つあります。

図 5-5　簡単な設定ファイル - チャネル定義

```
! test.cnf - 設定ファイルの例。
!
! Rewrite Rules
      .
      .
      .

! BEGIN CHANNEL DEFINITIONS
! FIRST CHANNEL BLOCK
l
local-host

! SECOND CHANNEL BLOCK
a_channel defragment charset7 usascii
a-daemon

! THIRD CHANNEL BLOCK
b_channel noreverse notices 1 2 3
b-daemon
```

典型的なチャネルエントリは次のようなものです。

```
tcp_intranet smtp mx single_sys subdirs 20 noreverse maxjobs 7
SMTP_POOL  maytlsserver allowswitchchannel  saslswitchchannel
tcpauth
tcpintranet-daemon
```

この例の最初の単語 `tcp_intranet` はチャネル名です。また、最後の単語 `tcpintranet-daemon` はチャネルタグです。チャネルタグは、書き換え規則でメッセージを送信するために使用する名前です。チャネル名とチャネルタグの間にある単語はチャネルキーワードで、メッセージの処理方法を表します。さまざまなキーワードを使って、さまざまな方法でメッセージを処理できます。チャネルキーワードの一覧と説明は、『iPlanet Messaging Server リファレンスマニュアル』と第 8 章「チャネル定義を設定する」にあります。

チャネルホストテーブルは、Messaging Server で使用できるチャネルと、各チャネルに関連付けられているシステム名を定義します。

UNIX システムでは、ファイルにある最初のチャネルブロックは必ずローカルチャネル l を表します (ただし defaults チャネルは例外で、ローカルチャネルより先に表示されます)。ローカルチャネルを使ってルーティングを決定し、UNIX メールツールでメールを送信します。

MTA オプションファイル (option.dat) でも、チャネルのグローバルオプションを設定したり、チャネルオプションファイルで特定チャネルのオプションを設定したりできます。オプションファイルの詳細については、134 ページの「オプションファイル」および 131 ページの「TCP/IP (SMTP) チャネルオプションファイル」を参照してください。チャネル設定の詳細については、第 8 章「チャネル定義を設定する」を参照してください。MTA チャネル作成の詳細については、111 ページの「MTA 設定ファイル」を参照してください。

MTA ディレクトリ情報

MTA は、処理する各メッセージに関して、サポートするユーザ、グループ、およびドメインに関するディレクトリ情報にアクセスする必要があります。この情報は、LDAP ディレクトリサービスに保存されています。MTA からは、2 つの方法でこの情報にアクセスできます。1 つは LDAP ディレクトリに直接アクセスする方法です。これはダイレクト LDAP モードと呼ばれます。詳細は付録 B「MTA ダイレクト LDAP 操作」に記載されています。もう 1 つはデフォルトの方法で、ディレクトリキャッシュを介してディレクトリ情報にアクセスします。これは dirsync モードと呼ばれます。

dirsync モードでは、MTA が使用するユーザとグループに関するディレクトリ情報には、多数のファイルおよびデータベース (総称してディレクトリキャッシュと呼ぶ) を介してアクセスします。データ自体は LDAP ディレクトリに格納されていますが、実際の情報にはキャッシュを介してアクセスします。キャッシュのデータは dirsync プログラムを使って更新します。このプログラムでは LDAP ディレクトリへの変更を監視し、変更されたファイルやデータベースがあれば更新します。

dirsync の操作と設定の詳細については、114 ページの「dirsync の設定」を参照してください。

ジョブコントローラ

メッセージがチャネルキューに入れられるたびに、ジョブコントローラはメッセージを配信するためのジョブが実行されていることを確認します。これには、新規ジョブプロセスの開始、スレッドの追加、実行中のジョブの確認などの操作が含まれます。チャネルまたはプールのジョブ範囲を超えたためにジョブを開始できない場合、ジョブコントローラは別のジョブが終了するのを待ち、ジョブ範囲を超えていないことを確認してからジョブを開始します。

チャネルジョブは、ジョブコントローラ内の処理プール内で実行されます。プールは、チャネルジョブが実行される「場所」であると考えることができます。プールは、プール外のジョブとリソースを奪い合うことなく処理できる計算領域です。プールについては、135 ページの「ジョブコントローラファイル」および 240 ページの「チャネル実行ジョブのプールを処理する」を参照してください。

チャネルのジョブ範囲は `maxjobs` チャネルキーワードで決定します。プールのジョブ範囲は、プールの `JOB_LIMIT` オプションで決定します。

通常 Messaging Server は、すべてのメッセージの配信を即座に試行します。最初の試行でメッセージを配信できない場合、メッセージの配信は `backoff` キーワードに指定した時間だけ遅れることになります。メッセージは、`backoff` キーワードに指定した時間が経過するとすぐに配信できる状態になり、必要に応じてチャネルジョブがメッセージの処理を開始します。

ジョブコントローラのメモリ内における処理中メッセージおよび処理待ちメッセージのデータ構造は、ディスクの MTA キュー領域に保存されているすべてのメッセージファイルを反映しています。ただし、ディスク上のメッセージファイルのバックログが大きくなり、ジョブコントローラのメモリ内データ構造のサイズ限界値を超えると、ジョブコントローラはメモリ内でディスク上のメッセージファイルの一部だけをトラッキングします。ジョブコントローラはメモリ内でトラッキング中のメッセージだけを処理します。メモリ内ストレージを開放できるだけの大量のメッセージが配信されると、ジョブコントローラは MTA キュー領域をスキャンしてメッセージリストを更新し、メモリ内ストアを自動的に更新します。その後、ジョブコントローラはディスクから取り出したばかりの新しいメッセージファイルの処理を開始します。ジョブコントローラは、MTA キュー領域のスキャンを自動的に行います。

サイトに大量のメッセージバックログが頻繁にたまる場合は、`MAX_MESSAGES` オプションを使ってジョブコントローラをチューニングすることもできます。`MAX_MESSAGES` オプションの値を大きくすると、ジョブコントローラが使用するメモリが増え、メッセージのバックログがジョブコントローラのメモリ内キャッシュでオーバーフローする回数が減ります。これにより、ジョブコントローラが MTA キューディレクトリをスキャンするための負荷が低減されます。ただし、ジョブコントローラでメモリ内キャッシュを再構築する必要がある場合は、キャッシュが大きく

なるので処理時間も長くなる点に注意してください。ジョブコントローラの起動時または再起動時には必ず MTA キューディレクトリをスキャンする必要があります。このため、メッセージのバックログが大量にある場合は、そのようなバックログがない場合に比べて、ジョブコントローラの起動や再起動に大きな負荷がかかります。

ジョブコントローラは、数多くの定期的なジョブも実行します。これらのジョブは、cron などの一般的な機能を使用せず、ジョブコントローラ設定で設定されるため、ジョブのスケジュールは実行中のジョブコントローラに依存します。これは、フェイルオーバーが考慮される高可用性の設定では、重要なポイントとなります。

ジョブコントローラの設定とプールの詳細については、135 ページの「ジョブコントローラファイル」および 236 ページの「メッセージの処理と配信を設定する」を参照してください。

ジョブコントローラを起動および停止するには

ジョブコントローラを起動するには、次のコマンドを実行します。

```
imsimta start job_controller
```

ジョブコントローラを停止するには、次のコマンドを実行します

```
imsimta stop job_controller
```

ジョブコントローラを再起動するには、次のコマンドを実行します。

```
imsimta restart job_controller
```

ジョブコントローラを再起動すると、実行中のジョブコントローラが終了し、その後すぐに新しいジョブコントローラが起動します。

第 6 章

MTA サービスと設定について

この章では、一般的な MTA サービスと設定について説明します。より具体的で詳細な説明については、ほかの章を参照してください。この章には、以下の節があります。

MTA 設定ファイル

MTA の主要設定ファイルは `imta.cnf` です。デフォルトでは、このファイルは *instance_root*`/imta/config/imta.cnf` にあります。このファイルには、MTA チャネル定義およびチャネル書き換え規則が含まれています。書き換えられた宛先アドレスに関連付けられたチャネルが、宛先チャネルとなります。

この節では、MTA 設定ファイルについて簡単に説明します。書き換え規則と、MTA 設定ファイルを構成するチャネル定義の詳細については、第 7 章「書き換え規則を設定する」および第 8 章「チャネル定義を設定する」を参照してください。

MTA 設定ファイルを変更することにより、サイトで使用されるチャネルを確立し、書き換え規則を介して各チャネルが処理するアドレスの種類を決定することができます。設定ファイルは、使用可能な転送方法 (チャネル) および転送経路 (書き換え規則) を指定し、アドレスの種類を適切なチャネルに関連付けることにより電子メールシステムの設計を定めるファイルです。

設定ファイルは、ドメイン書き換え規則とチャネル定義の 2 つの部分から構成されます。ファイルの最初にドメイン書き換え規則があり、チャネル定義との間には空白行が 1 行あります。チャネル定義はチャネルテーブルと総称されています。個々のチャネル定義がチャネルブロックを構成します。

次の imta.cnf 設定ファイルの例は、書き換え規則を使って適切なチャネルにメッセージをルーティングする方法を示しています。わかりやすくするために、ドメイン名は使用していません。書き換え規則は設定ファイルの前半部分にあり、そのあとにチャネル定義が続いています。

図 6-1　簡単な MTA 設定ファイル

```
! test.cnf - An example configuration file. (1)
!
! これは、単に設定ファイルの例です。It serves
! no useful purpose and should not be used in a real system.
!
! Part I: Rewrite rules
a     $U@a-daemon (2)
b     $U@b-daemon
c     $U%c@b-daemon
d     $U%d@a-daemon
            (3)
! Part II: Channel definitions
l              (4)
local-host

a_channel defragment charset7 usascii (5)
a-daemon

b_channel noreverse notices 1 2 3
b-daemon

</usr/iplanet/server5/msg-tango/table/internet.rules (6)
```

以下に、上記設定ファイルの主な項目 (括弧に入っている太字の番号付き) について説明します。

1. コメント行を示すには、感嘆符 (!) を使用します。感嘆符は行頭に表示されていなければなりません。その他の場所にある感嘆符は、文字として解釈されます。
2. 書き換え規則は設定ファイルの前半部分にあります。書き換え規則に空白行を入れることはできません。コメント行 (行頭に感嘆符が付いている) を入れることはできます。
3. 設定ファイル内で最初に現れる空白行は、書き換え規則の終わりとチャネル定義の始まりを表します。これらの定義は「チャネルホストテーブル」と総称され、MTA が使用できるチャネルと、各チャネルに関連付けられた名前を定義します。
4. 通常、最初のチャネルブロックはローカルチャネル (`l` チャネル) です。各チャネル定義ブロックは、空白行で区切られています (`defaults` チャネルは例外で、このチャネルは `l` チャネルより先に表示されます)。
5. 典型的なチャネル定義は、チャネル名 (`a_channel`)、チャネルの設定を定義するキーワード (`defragment charset7 usascii`)、およびルーティングシステム (`a-daemon`) で構成されます。ルーティングシステムはチャネルタグとも呼ばれます。
6. ほかのファイルの内容を設定ファイルに含めることもできます。行頭に「小なり」(<) の記号があると、その行の残りはファイル名として扱われます。ファイル名は絶対名でフルパスでなければなりません。指定されたファイルが開かれ、設定ファイルのその場所にほかのファイルの内容が入れられます。ファイルの包含は、3 階層までネストすることができます。設定ファイルに含めるファイルは、設定ファイルと同じように、だれでも読み取り可能でなければなりません。

表 6-1 に、上記の設定でアドレスをルーティングする方法の例を示します。

表 6-1 アドレスと関連付けられたチャネル

アドレス	チャネルキュー
`u@a`	`a_channel`
`u@b`	`b_channel`
`u@c`	`b_channel`
`u@d`	`a_channel`

MTA 設定ファイルの詳細については、102 ページの「書き換え規則」、105 ページの「チャネル定義」、および第 7 章「書き換え規則を設定する」を参照してください。

dirsync の設定

Messaging Server のデフォルトのインストールでは、dirsync モードの操作を使用します (付録 B「MTA ダイレクト LDAP 操作」で説明しているダイレクト LDAP モードを使用する方法もあります)。dirsync モードでは、MTA は、メッセージを処理するたびにディレクトリサービスを照会するのではなく、ディレクトリ情報をキャッシュに保存し、データが必要な場合はキャッシュにアクセスします。

ディレクトリサービスに格納されているディレクトリ情報は、dirsync というプログラムによって絶えず更新されています。このため、ディレクトリキャッシュも、ディレクトリサービス内のディレクトリ情報に合わせて定期的に更新 (同期化) する必要があります。同期には 2 種類の方法があります。

- **完全同期** - 既存のキャッシュは新しいキャッシュに置き換えられ、ディレクトリサービスのユーザおよびグループのエントリを使って完全に再構築される。同期後、MTA 設定ファイルは再構築され、MTA が自動的に再起動する
- **増分同期** - 既存のキャッシュは、前回の完全同期または増分同期以降に作成されたユーザおよびグループのエントリを使って定期的に更新される。MTA は再起動しない

デフォルトでは、MTA ディレクトリキャッシュでは、毎日午前 2 時に完全同期が実行され、10 分おきに増分同期が実行されます。

表 6-2 に、完全同期または増分同期で行われる更新を示します。

表 6-2 MTA ディレクトリキャッシュの更新

MTA ディレクトリキャッシュの更新	完全同期	増分同期
新規ユーザエントリの追加	○	○
修正したユーザエントリの更新	○	○
* 削除したユーザエントリの破棄	○	×
既存の配布リストへの新規メンバーの追加	○	○
既存の配布リストから削除したメンバーの破棄	○	○
新規の配布リストの追加	○	○
* 削除した配布リストの破棄	○	×

* 削除したエントリに対して増分同期を実行するには、そのエントリのステータスに削除済みのマークが付いている必要があります。増分同期の実行後、MTA はそのユーザまたはグループが存在しないものとみなします。実際にディレクトリエントリを削除する作業は、増分同期のあとに行ってください。

通常、ディレクトリの同期は自動的に行われます。ただし、必要に応じて、imsimta dirsync コマンドを使って MTA ディレクトリキャッシュを再作成または更新することができます。imsimta dirsync コマンドの詳細については、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

ディレクトリ同期設定パラメータ

表 6-3 に、ディレクトリ同期設定パラメータの一覧を示します。

表 6-3 ディレクトリ同期設定パラメータ

パラメータ	説明
local.imta.ldsearchtimeout	ユーザおよびメーリングリストの情報を検索する際の LDAP 検索のタイムアウトを指定する。デフォルトでは、タイムアウトは発生しない
local.imta.hostnamealiase	LDAP エントリがローカルであるかどうかを確認するためにそのエントリの mailhost または mailRoutingHosts 属性をチェックする際に、dirsync プロセスは local.hostname パラメータを使って比較を行う。さらに、local.imta.hostnamealiases パラメータにより、ホスト名エイリアスのカンマ区切りのリストが提供される。dirsync プロセスは、これらの 2 つのパラメータによって提供されるすべてのホスト名を使って、エントリがローカルであるかどうかを調べる
local.imta.mailaliases	デフォルトでは、mail および mailAlternateAddress という LDAP 属性のみがルーティング可能なメールアドレスとみなされる。または、local.imta.mailaliases パラメータにより、LDAP 属性のカンマ区切りのリストが提供される。このリストはデフォルト属性を上書きする。たとえば、メッセージをルーティングする際に、MTA は次の 4 つの属性を考慮する local.imta.mailaliases=mail,mailAlternateAddress, rfc822mailbox,rfc822mailalias

表 6-3　ディレクトリ同期設定パラメータ (続き)

パラメータ	説明
local.imta.ugfilter	このパラメータは、ユーザやメーリングリストの情報を検索する際に、dirsync が使用する LDAP 検索フィルタを設定する デフォルトのフィルタは (objectClass=inetLocalMailRecipient) たとえば、inetLocalMailRecipient オブジェクトクラスおよび myispSubscriber オブジェクトクラスで LDAP エントリだけを考慮する場合は、このパラメータを次のように設定する local.imta.ugfilter=(&(objectClass=inetLocalMailRecipient)(objectClass=myispSubscriber)) 注 : 増分同期の場合は、このフィルタにタイムスタンプフィルタが追加される。このため、カスタムフィルタを ( ) で囲む必要がある
local.imta.statssamplesize	このパラメータを設定すると、dirsync は、開始時からのユーザとメーリングリストのエントリ数の要約、および平均値 (エントリ数 / 秒) を標準出力で印刷する。ユーザとメーリングリストは、同期が完了したかどうかにかかわらず計算される
local.imta.reverseenabled	リバースデータベースの生成をトリガする。デフォルト値は、yes である。リバースデータベースの実際の使用は、USE_REVERSE_DATABASE オプションで制御される
local.imta.ssrenabled	サーバ側規則 (SSR) データベースの生成をトリガする。デフォルト値は、yes である。SSR データベースの実際の使用は、ssr チャネルキーワードで制御される
local.imta.vanityenabled	バニティドメイン (msgVanityDomain ユーザ LDAP 属性) が有効であるかどうかを制御する。デフォルトは yes である
local.imta.catchallenabled	キャッチオールアドレス (@domain 形式の mail または mailAlternateAddress) が有効であるかどうかを制御する。デフォルトは yes である
local.imta.scope	このパラメータは、dirsync が同期化するエントリを指定する mailhost 属性がローカルホストであるユーザおよびメーリングリストのエントリだけをキャッシュする場合は、値に local を指定する mailhost 属性に関係なくユーザおよびメーリングリストのエントリをキャッシュする場合は、値に domains を指定する。これはパラメータがない場合のデフォルト値である ドメイン、ユーザ、またはメーリングリストをキャッシュしない場合は、値に nobody を指定する

マッピングファイル

MTA コンポーネントの多くは、テーブル検索に基づいた情報を使用します。このタイプのテーブルは、入力文字列を出力文字列に変える (マップする) のに使用されます。このようなテーブルはマッピングテーブルと呼ばれ、通常 2 つのカラムで構成されます。最初 (左側) のカラムにはパターンを照合する入力文字列が、2 番目 (右側) のカラムにはその入力文字列がマップされた (テンプレート) 結果の出力文字列が並んでいます。

MTA データベースのほとんどは、このタイプのテーブルのインスタンスです。これらのデータベースにはさまざまなタイプの MTA データが含まれています。マッピングテーブルとは混同しないでください。ただし、MTA データベースファイルには、ワイルドカード検索機能がありません。データベース全体でワイルドカードに一致するものを検索するのは非効率的だからです。

MTA マッピングファイルは、複数のマッピングテーブルをサポートします。ワイルドカード機能もあり、複数の手順や反復マッピング方法にも対応しています。このアプローチは、データベースを使用する場合に比べ、さらに多くの処理を必要とします。特に、エントリ数が多い場合などはなおさらです。ただし、それに付随して柔軟性が増すため、同等のデータベースにおけるエントリのほとんどを必要としなくなり、全体的にオーバーヘッドが少なくなります。

表 6-4 に、本書で説明するマッピングテーブルの一覧を示します。

表 6-4 iPlanet Messaging Server のマッピングテーブル

マッピングテーブル	ページ	説明
CHARSET-CONVERSION	296	チャネル間における文字セット変換やメッセージフォーマット変換の種類を指定するために使用される
COMMENT_STRINGS	254	アドレスヘッダーのコメント (括弧で囲まれた文字列) を変更するために使用される
CONVERSIONS	280	変換チャネルのメッセージトラフィックを選択するために使用される
"domain lookup"	548	ダイレクト LDAP モードで、エイリアスを検索するツリーのベースを検索するために使用される
FORWARD	149	エイリアスファイルまたはエイリアスデータベースを使用した場合と同様の転送を行う
FROM_ACCESS	306	エンベロープの From アドレスに基づいてメールをフィルタリングする場合に使用する。このテーブルは、To アドレスが不適切な場合に使用する
INTERNAL_IP	318	内部のシステムとサブネットを認識する

表 6-4　iPlanet Messaging Server のマッピングテーブル (続き)

マッピングテーブル	ページ	説明
MAIL_ACCESS	306	SEND_ACCESS テーブルと PORT_ACCESS テーブルを組み合わせた情報に基づいて受信接続をブロックする場合に使用する
NOTIFICATION_LANGUAGE	150	通知メッセージをカスタマイズまたはローカライズする
ORIG_MAIL_ACCESS	306	ORIG_SEND_ACCESS テーブルと PORT_ACCESS テーブルを組み合わせた情報に基づいて受信接続をブロックする場合に使用する
ORIG_SEND_ACCESS	306	エンベロープ From アドレス、エンベロープ To アドレス、ソースおよび宛先チャネルに基づいて、受信接続をブロックする
PERSONAL_NAMES	255	個人名 (角括弧で区切られたアドレスの前にある文字列) を変更するために使用される
PORT_ACCESS	306	IP 番号に基づいて受信接続をブロックする
REVERSE	145	内部形式から公のアドバタイズ形式にアドレスを変換する
SEND_ACCESS	306	エンベロープ From アドレス、エンベロープ To アドレス、ソースおよび宛先チャネルに基づいて、受信接続をブロックする
X-ATT-NAMES	289	マッピングテーブルからパラメータ値を検索するために使用される

マッピングファイルの検索と読み込み

マッピングテーブルは、MTA マッピングファイルに保存されています。これは、MTA テイラーファイルの IMTA_MAPPING_FILE オプションで指定されているファイルで、デフォルトは *server_root*/msg-*instance*/imta/config/mappings です。マッピングファイルの内容は、コンパイルされた設定に取り込まれます。

マッピングファイルは、だれでも読み取り可能でなければなりません。だれでも読み取りおよびアクセスが可能でないと、誤作動をまねくことになります。

マッピングファイルのファイル形式

マッピングファイルは、一連のテーブルで構成されています。各テーブルはその名前で始まり、名前の先頭は必ずアルファベット文字です。テーブル名の次には必ず空白行が続き、そのあとにテーブルのエントリが続きます。エントリには、インテンド行がない場合とある場合があります。各エントリ行は、1 つ以上のスペースまたはタブで区切られた 2 つのカラムから成ります。エントリ内のスペースはすべて、$ 文字で囲む必要があります。各テーブル名のあとおよびテーブル間には空白行が必要ですが、1 つのテーブル内のエントリ間に空白行があってはなりません。コメント行を挿入する場合は、その行の 1 列目を感嘆符 (!) にします。

つまり、ファイル形式は以下のようになります。

```
TABLE1_NAME

   pattern1-1    template1-1
   pattern1-2    template1-2
   pattern1-3    template1-3
      .             .
      .             .
      .             .
   pattern1-n    template1-n

TABLE2_NAME

   pattern2-1    template2-1
   pattern2-2    template2-2
   pattern2-3    template2-3
       .             .
       .             .
       .             .
   pattern2-n    template2-n

           .
           .
           .

TABLE3_NAME

           .
           .
           .
```

TABLE2_NAME マッピングテーブルを使用するアプリケーションは、pattern2-2 文字列を template2-2 で指定された文字列にマップします。各パターンまたはテンプレートには、252 文字までを含めることができます。マッピングテーブルに含まれるエントリの数に制限はありません (ただし、エントリが必要以上に多い場合は、CPU とメモリを大量に消費することになります)。252 文字を超えるの長い行は、円記号 (¥) を行の末尾に置くことで次の行に続けることができます。2 つのカラム間および最初のカラムの前にある空白スペースは削除しないでください。

マッピングファイルでマッピングテーブル名が重複することは許されていません。

マッピングファイルにほかのファイルを含める

マッピングファイルにはほかのファイルを含めることができます。次の形式の行を使用します。

```
<file-spec
```

これによって、マッピングファイル内の file-spec の行が、その実際のファイルに置き換えられます。ファイル指定には、完全なファイルパス (ディレクトリなど) が必要です。この方法で含めるファイルは、だれでも読み取り可能でなければなりません。マッピングファイルに含めるファイルにはコメントを入れることもできます。含めるファイルは 3 段階までネスティングすることができます。含められたファイルは、マッピングファイルと一緒に読み込まれます。オンデマンドで読み込まれるのではないため、ファイルを含めることによってパフォーマンスまたはメモリを節約することはできません。

マッピングの動作

マッピングファイル内のマッピングはすべて一定の方法で適用されます。マッピングごとに異なるのは、入力文字列のソースとマッピング出力の使用目的のみです。

マッピングの動作は、常に入力文字列とマッピングテーブルから始まります。マッピングテーブルのエントリは、テーブルに表示される順に上から下へ 1 つずつスキャンされます。各エントリの左側の部分がパターンとして使用され、入力文字列は大文字 / 小文字の区別なくそのパターンと比較されます。

マッピングエントリのパターン

パターンには、ワイルドカード文字を含めることができます。たとえば、次のような一般的なワイルドカード文字を使用できます。アスタリスク (*) は 0 個以上の文字と一致し、パーセント記号 (%) は 1 文字と一致します。アスタリスク、パーセント記号、スペース、およびタブの前にドル記号 ($) を置くことによって、これらの記号を引用できます。アスタリスクやパーセント記号を引用した場合は、特別な意味は失われます。パターンやテンプレートを正しく認識させるために、その中のスペースやタブは文字として認識させる必要があります。ドル記号を文字として使用するには、2 重のドル記号 ($$) を使用します。この場合、最初のドル記号によって、2 番目のドル記号が文字として認識されるようになります。

表 6-5 マッピングパターンのワイルドカード

ワイルドカード	説明
%	1 文字だけ一致する
*	0 個以上の文字と一致する。左から右への最大限の一致を使用する
後照合	**説明**
$ n*	n 番めのワイルドカードまたはグロブに一致する
修飾子	**説明**
$_	左から右への最低限の一致を使用する
$@	後続のワイルドカードまたはグロブの「保存」をオフにする
$^	後続のワイルドカードまたはグロブの「保存」をオンにする (デフォルト設定)
グロブワイルドカード	**説明**
$A%	A ～ Z または a ～ z の 1 文字と一致する
$A*	A ～ Z または a ～ z の 0 個以上の文字と一致する
$B%	1 つのバイナリ数字 (0 または 1) と一致する
$B*	0 個以上のバイナリ数字 (0 または 1) と一致する
$D%	1 つの 10 進数 (0 ～ 9) と一致する
$D*	0 個以上の 10 進数 (0 ～ 9) と一致する
$H%	1 つの 16 進数 (0 ～ 9 または A ～ F) と一致する
$H*	0 個以上の 16 進数 (0 ～ 9 または A ～ F) と一致する
$O%	1 つの 8 進数 (0 ～ 7) と一致する
$O*	0 個以上の 8 進数 (0 ～ 7) と一致する
$S%	1 つの記号セット文字 (例 : 0 ～ 9、A ～ Z、a ～ z、_、$) と一致する

表 6-5　マッピングパターンのワイルドカード (続き)

$S*	0 個以上の記号セット文字 (例 : 0 ～ 9、A ～ Z、a ～ z、_、$) と一致する
$T%	1 つのタブ、垂直タブ、またはスペース文字と一致する
$T*	0 個以上のタブ、垂直タブ、またはスペース文字と一致する
$X%	$H% と同義
$X*	$H* と同義
$[c]%	文字 c と一致する
$[c]*	任意の数の文字 c と一致する
$[c_1 c_2 ... c_n]%	文字 c_1、c_2、c_n のいずれか 1 つと一致する
$[c_1 c_2 ... c_n]*	文字 c_1、c_2、c_n のいずれかと任意の数一致する
$[c_1 -c_n]%	c_1 から c_n の範囲内の任意の 1 文字と一致する
$[c_1 -c_n]*	c_1 から c_n の範囲内の任意の文字列と一致する
$< IPv4>	ビットは無視し、IPv4 アドレスと一致する
$(IPv4)	プレフィックスビットを維持した状態で、IPv4 アドレスと一致する
${IPv6}	IPv6 アドレスと一致する

グロブ内、つまり `$[...]` 内では、円記号 (¥) は引用符となります。実際のハイフン (-) または右角括弧 (]) をグロブ内で表すには、ハイフンまたは右角括弧に円記号を付ける必要があります。

パターン内のその他の文字はすべて、文字として使用されます。特に、一重引用符や二重引用符、および括弧は、マッピングパターンやテンプレートにおいて特殊な意味を持たず、通常の文字とみなされます。このため、不正なアドレスや部分的なアドレスに対応するエントリの書き出しが簡単になります。

複数の修飾子、または修飾子および後照合を指定するには、シンタックスにドル記号を 1 つだけ使用します。たとえば、最初のワイルドカードを、後照合そのものを保存せずに後照合するには、$@$0 ではなく $@0 を使用します。

マッピングパターンのテスト、特にパターン内のワイルドカードの動作のテストを行うには、`imsimta test -mapping` ユーティリティを使用できます。

アスタリスクのワイルドカードは、パターンを左から右へスキャンすることにより、一致する対象を最大化します。たとえば、文字列 `a/b/c` をパターン `*/*` と比較する場合、左のアスタリスクが「`a/b`」に一致し、右のアスタリスクが残りの `c` に一致します。

$_ 修飾子は、ワイルドカードによる照合を最小にするため、パターンの左から右に向かって、もっとも可能性の少ない一致がその一致とみなされます。たとえば、a/b/c 文字列を $_*/$_* というパターンと比較した場合、左側の $_* は a と、右側の $_* は b/c と一致します。

IP の照合

IPv4 接頭辞の照合では、IP アドレス、またはサブネットを指定し、そのあとにオプションとして、照合比較の際に有効となるスラッシュと接頭辞のビット数を続けます。たとえば、次の例は 123.45.67.0 サブネット内にあるものに一致します。

```
$(123.45.67.0/24)
```

IPv4 照合でビットを無視する場合は、IP アドレスまたはサブネットを指定し、そのあとにオプションとして、照合を確認する際に無視するスラッシュとビット数を続けます。たとえば、次の例は 123.45.67.0 サブネット内にあるものに一致します。

```
$<123.45.67.0/8>
```

次の例は、123.45.67.4 から 123.45.67.7 の範囲内にあるものに一致します。

```
$<123.45.67.4/2>
```

IPv6 照合は、IPv6 アドレスまたはサブネットを照合します。

マッピングエントリのテンプレート

指定したエントリのパターン比較に失敗した場合は、何の動作も行われず、次のエントリのスキャンへ移行します。比較が成功した場合は、エントリの右側の部分がテンプレートとして使用され、出力文字列が生成されます。このテンプレートによって、入力文字列がテンプレートの指示によって構成された出力文字列に置き換えられます。

テンプレート内のほとんどすべての文字が、そのまま出力文字列として生成されますが、ドル記号 ($) は例外です。

ドル記号の後ろにドル記号、スペース、またはタブが続く場合は、出力文字列内にドル記号、スペース、またはタブが生成されます。これらの文字を出力文字列に挿入するには、引用符を付ける必要があります。

ドル記号とそれに続く *n* 桁目は、置換を要求します。ドル記号とそれに続くアルファベット文字は、「メタ文字」と呼ばれます。メタ文字自体は、テンプレートで作成される出力文字には表示されませんが、いくつかの特殊な置換や処理を生成します。特殊な置換および標準処理のメタ文字の一覧については、表 6-6 を参照してください。その他のメタ文字はマッピング特有の用途に制限されています。

テンプレートの照合パターン内に $C、$E、$L または $R のいずれかのメタ文字がある場合、それらはマッピング処理に影響を及ぼし、処理の終了または続行を決定します。つまり、1 つのエントリの出力文字列が別のエントリの入力文字列となるような反復的なマッピングテーブルエントリを設定することができます。テンプレートの照合パターン内に $C、$E、$L、または $R のどのメタ文字も含まれていない場合は、$E (マッピング処理の即時終了) が行われます。

無限ループを避けるために、マッピングテーブル内のパス (文字列が渡されること) の反復回数には制限があります。前回のパスと同じか、それより長いパターンを使用してパスが反復されるたびに、カウンタは 1 増えます。文字列が直前のものより短い場合は、カウンタがゼロにリセットされます。カウンタが 10 に達すると、マッピングの反復要求は受け付けられません。

表 6-6 マッピングテンプレートの置換とメタ文字

置換シーケンス	置き換える内容
$n	左から数えて *n* 番目のワイルドカードフィールド
$#...#	シーケンス番号の置換
$]...[	LDAP による URL 検索。結果的に置換が行われる
$|...|	提供された文字列に指定したマッピングテーブルを適用する
${...}	一般的なデータベース置換
$[...]	サイト提供ルーチンを呼び出す。結果的に置換が行われる
メタ文字	**説明**
$C	次のテーブルエントリからマッピング処理を続行し、このエントリの出力文字列をマッピング処理の新しい入力文字列として使用する
$E	マッピング処理をただちに終了し、このエントリの出力文字列をマッピング処理の最終結果とする
$L	次のテーブルエントリからマッピング処理を続行し、このエントリの出力文字列を新しい入力文字列として使用する。テーブル内のすべてのエントリを照合したら、もう一度最初のテーブルエントリから照合する。後続の照合エントリにメタ文字 $C、$E、または $R がある場合は、それらのエントリが優先される
$R	マッピングテーブルの最初のエントリからマッピング処理を続行し、このエントリの出力文字列をマッピング処理の新しい入力文字列として使用する
$?x?	マッピングエントリが x パーセントの割合で成功する
$¥	後続のテキストを小文字にする
$^	後続のテキストを大文字にする

表 6-6　マッピングテンプレートの置換とメタ文字 (続き)

置換シーケンス	置き換える内容
$_	後続のテキストを元の状態で残す
$:*x*	指定したフラグが設定されている場合にのみ一致する
$;*x*	指定したフラグが解除されている場合にのみ一致する

ワイルドカードフィールドの置換 ($n)

ドル記号の後ろに数字 n が続いている場合、これは、パターン内の *n* 番目のワイルドカードに一致するデータに置き換えられます。ワイルドカードには、0 から順に番号が付けられています。たとえば、次のエントリは入力文字列 PSI%A::B に一致し、その結果 b@a.psi.siroe.com という出力文字列を生成します。

```
PSI$%*::*     $1@$0.psi.siroe.com
```

また、入力文字列 PSI%1234::USER は、出力として生成される USER@1234.psi.siroe.com と照合されます。入力文字列 PSIABC::DEF は、このエントリ内のパターンに一致しないため置換は行われません。つまり、このエントリから出力文字列は生成されません。

テキストの大文字小文字の制御 ($¥、$^、$_)

メタ文字 $¥ は後続のテキストを小文字に変換し、メタ文字 $^ は後続のテキストを大文字に変換するものです。また、メタ文字 $_ は、後続のテキストを元の大文字 / 小文字の状態で残します。たとえば、これらのメタ文字は、マッピングを使って大文字 / 小文字の区別が有効なアドレスを変更する際に役立ちます。

処理制御 ($C、$L、$R、$E)

メタ文字 $C、$L、$R、および $E は、マッピング処理を終了するかどうか、またいつ終了するかなど、マッピング処理に影響を与えます。これらのメタ文字には、以下の効果があります。

- $C は現在のエントリの出力文字列をマッピング処理の新しい入力文字列として使用し、次のエントリからマッピング処理を続行します。
- $L は、現在のエントリの出力文字列をマッピング処理の新しい入力文字列として使用し、次のエントリからマッピング処理を続行します。一致するエントリが見つからない場合には、もう一度そのテーブルの最初のテーブルエントリから照合を開始します。後続の照合エントリにメタ文字 $C、$E または $R がある場合には、それらのエントリが優先されます。

- $R は、現在のエントリの出力文字列をマッピング処理の新しい入力文字列として使用し、テーブルの最初のエントリからマッピング処理を続行します。
- $E はマッピング処理を終了し、このエントリの出力文字列が最終結果となります。デフォルト設定は $E です。

マッピングテーブルのテンプレートは、左から右にスキャンされます。「成功」または「失敗」するエントリ (たとえば、一般データベースの置換またはランダム値で制御されるエントリ) に $C、$L、または $R のフラグを設定するには、メタ文字 $C、$L、または $R を成功または失敗するエントリ部分の左側に配置します。エントリのそれ以外の部分が失敗しても、フラグは表示されません。

ランダムに成功または失敗するエントリ ($?x?)

マッピングテーブルのエントリに $?x? というメタ文字がある場合は、これによって、*x* パーセントの割合でエントリが「成功」します。それ以外の場合、エントリは「失敗」し、マッピングエントリの入力文字列が変更されずにそのまま出力文字列となります (マッピングによっては、エントリが失敗したこととエントリが一致しなかったこととは、必ずしも同義ではありません)。*x* は、成功率を指定するための実数値です。

たとえば、IP アドレスが 123.45.6.78 であるシステムが、自分のサイトに大量の SMTP 電子メールを送信していて、このメールの量を少し減らしたいとします。この場合、PORT_ACCESS マッピングテーブルを次のように使用できます。たとえば、接続の 25 パーセントのみを許可し、残りの 75 パーセントを拒否するとします。次のマッピングテーブル PORT_ACCESS は、$?25? を使用し、$Y のあるエントリを 25 パーセントの割合で成功させます (すなわち、接続を許可します)。残りの 75 パーセントの割合でエントリが失敗すると、そのエントリの最初の $C によって MTA が次のエントリからマッピングを続行するため、接続試行が拒否され、「Try again later (あとでもう一度試してください)」という SMTP エラーメッセージが表示されます。

```
PORT_ACCESS

  TCP|*|25|123.45.6.78|*          $C$?25?$Y
  TCP|*|25|123.45.6.78|*          $N45s$ 4.40$ Try$ again$ later
```

シーケンス番号の置換 ($#...#)

$#...# 置換は、MTA シーケンスファイルに保存されている値を増やし、その値をテンプレート内に入れます。たとえば、マッピングテーブルを使ってファイル名を生成するときなど、マッピングテーブルの出力に固有の修飾子があることが望ましい場合に、シーケンス番号付きの固有文字列を生成することができます。

以下のいずれかのシンタックスを使用できます。

```
$#seq-file-spec|radix|width#
```

```
$#seq-file-spec|radix#
```

```
$#seq-file-spec#
```

必須の引数 *seq-file-spec* は、既存の MTA シーケンスファイルに対する完全なファイル仕様であり、オプションの引数 *radix* および *width* は、それぞれ出力するシーケンス値の基数および桁数を指定するものです。デフォルトの基数は 10 ですが、-36 ～ 36 の範囲内の基数も使用できます。たとえば、基数 36 では 0 ～ 9、A ～ Z の文字からなる値を使用することができます。デフォルトでは、シーケンス値は自然幅で出力されますが、大きな桁数を指定すると、桁数に合わせるために数値の左側に 0 が追加されます。

桁数を明示的に指定する場合は、基数も明示的に指定する必要があります。

上記にあるように、マッピングで参照される MTA シーケンスファイルはすでに存在するものでなければなりません。MTA シーケンスファイルを作成するには、以下のコマンドを使用します。

```
touch seq-file-spec
```

または

```
cat >seq-file-spec
```

マッピングテーブルを使ってアクセスされるシーケンス番号ファイルは、だれでも読み取り可能でないと正常に操作できません。また、このようなシーケンス番号ファイルを使用するには、MTA ユーザアカウント (`imta_tailor` ファイルで `nobody` として設定) を持つことが必要です。

LDAP クエリ URL の置換 $]...[

$]*ldap-url*[の形式の置換は、特別に処理されます。*ldap-url* は LDAP クエリ URL として解釈され、LDAP クエリの結果が置換されます。ホストとポートが省略された標準の LDAP URL が使用されます。ホストとポートは、代わりに LDAP_HOST オプションと LDAP_PORT オプションで指定されます。LDAP URL は次のように指定する必要があります。

ldap:///*dn*[?*attributes*[?*scope*?*filter*]]

上記の角括弧 ([と]) は、URL のオプションの部分を示します。*dn* は検索ベースを指定する名前で、この部分は必須です。URL の *attributes*、*scope*、および *filter* の各オプションを指定すると、より細かい情報が返されます。つまり、*attributes* では、この LDAP クエリに一致する LDAP ディレクトリエントリから返される属性を指定します。*scope* には、base (デフォルト)、one、または sub のいずれかを指定できます。*filter* には一致するエントリの特徴を記述します。

特定の LDAP URL 置換シーケンスは、LDAP クエリ URL 内で使用できます。

マッピングテーブルの置換 ($|...|)

$|*mapping*、*argument*| 形式の置換は、特殊な方法で処理されます。MTA は、MTA マッピングファイル内の *mapping* で指定されている補足的なマッピングテーブルを探し、その補足的なマッピングテーブルで *argument* を入力文字列として使用します。この補足的なマッピングテーブルは既存のものであり、置換が成功した場合にはその出力文字列に $Y フラグを設定しなければなりません。この補足的なマッピングテーブルが存在しなかったり、または $Y フラグを設定しなかった場合には、補足的なマッピングテーブルの置換は失敗し、元のマッピングエントリも失敗とみなされます。元の入力文字列が出力文字列として使用されます。

マッピングテーブルの置換を行うマッピングテーブルエントリで $C、$R、または $L などの処理制御メタ文字を使用する場合は、処理制御メタ文字をマッピングテーブルテンプレート内のマッピングテーブル置換の左側に配置します。そうしないと、マッピングテーブルの置換が「失敗」したときに、処理制御メタ文字が処理されません。

一般データベースの置換 (${...})

${*text*} 形式の置換は、特殊な方法で処理されます。*text* 部分は、一般データベースにアクセスするための鍵として使われます。このデータベースは imsimta crdb ユーティリティにより生成されます。*text* がデータベース内のエントリに一致すると、データベース内の対応するテンプレートがその文字列に置き換えられます。*text* がデータベース内のエントリに一致しない場合は、入力文字列がそのまま出力文字列として使用されます。

一般データベースは、正しい操作が行われるように、だれでも読み取り可能でなければなりません。

一般データベースの置換を行うマッピングテーブルエントリで、$C、$R、または $L などの処理制御メタ文字を使用する場合は、処理制御メタ文字をマッピングテーブルテンプレート内の一般データベース置換の左側に配置します。そうしないと、一般データベースの置換が「失敗」したときに、処理制御メタ文字が処理されないことになります。

サイト提供ルーチンの置換 ($[...])

$[*image*,*routine*,*argument*] 形式の置換は、特殊な方法で処理されます。image、routine、argument の各部分は、カスタマ提供のルーチンを探し、呼び出すために使用されます。UNIX では、MTA は dlopen および dlsym を使って動的に共有ライブラリ *image* から指定した *routine* をロードし、呼び出します。Windows NT のランタイムでは、MTA により *routine* ルーチンがダイナミックリンクライブラリの *image* から呼び出されます。そのとき、その *routine* は、以下の引数を伴った関数として呼び出されます。

```
status = routine (argument, arglength, result, reslength)
```

argument および result は、252 バイトの文字列バッファです。argument および result は、文字列へのポインタ (たとえば、C 言語での char* のように) として渡されます。arglength および reslength は、参照によって渡される符号付きの long 型整数です。入力時、argument にはマッピングテーブルテンプレートからの *argument* 文字列が含まれ、arglength にはその文字列の長さが含まれます。値を返すときには、result に結果文字列が入り、reslength にその長さが入ります。この結果文字列が、マッピングテーブルテンプレート内の $[image,routine,argument] に置き換わります。*routine* は、マッピングテーブルの置換が失敗した場合には 0 を返し、成功した場合には -1 を返します。置換が失敗した場合は、通常、元の入力文字列がそのまま出力文字列として使用されます。

サイト提供ルーチンの置換を行うマッピングテーブルエントリで、$C、$R、または $L などの処理制御メタ文字を使用する場合は、処理制御メタ文字をマッピングテーブルテンプレート内のサイト提供ルーチン置換の左側に配置します。そうしないと、マッピングテーブルの置換が「失敗」したときには、処理制御メタ文字が処理されないことになります。

サイト提供ルーチンの呼び出し機構によって、MTA のマッピング処理はさまざまな方法で拡張することができます。たとえば、マッピングテーブル PORT_ACCESS または ORIG_SEND_ACCESS 内で、ロードモニタサービスへの呼び出しを行い、その結果を使って接続やメッセージを受け入れるかどうかを決定することができます。

image (サイト提供の共有ライブラリイメージ) は、だれでも読み取り可能でなければなりません。

その他の MTA 設定ファイル

imta.cnf ファイルのほかにも、iPlanet Messaging Server には MTA サービスを設定するのに役立ついくつかの設定ファイルがあります。表 6-7 にファイルの一覧を示します。

表 6-7　MTA 設定ファイル

ファイル	説明
自動返信オプションファイル	autoreply プログラムによって使用されるオプション *instance_root*/imta/config/autoreply_option
エイリアスファイル (必須)	ディレクトリにないエイリアスを実行する *instance_root*/imta/config/aliases
TCP/IP (SMTP) チャネルオプションファイル (または SMTP オプションファイル)	チャネル固有のオプションを設定する *instance_root*/imta/config/*channel*_option
変換ファイル	変換チャネルがメッセージ本体部分の変換を制御するのに使用する *instance_root*/imta/config/conversions
Dirsync オプションファイル (dirsync モードで実行する場合のみ必須)	dirsync プログラムによって使用されるオプション *instance_root*/imta/config/dirsync.opt
ディスパッチャ設定ファイル (必須)	ディスパッチャ用の設定ファイル *instance_root*/imta/config/dispatcher.cnf
ジョブコントローラファイル (必須)	ジョブコントローラ が使用する設定ファイル /*instance_root*/imta/config/job_controller.cnf
MTA 設定ファイル (必須)	アドレスの書き換え、ルーティング、およびチャネル定義に使用する /*instance_root*/imta/config/imta.cnf
マッピングファイル (必須)	マッピングテーブルのリポジトリ /*instance_root*/imta/config/mappings
オプションファイル	グローバル MTA オプションのファイル /*instance_root*/imta/config/option.dat
テイラーファイル (必須)	場所といくつかの調整パラメータを指定するファイル /*instance_root*/imta/config/imta_tailor

自動返信オプションファイル

自動返信ファイル autoreply_option は、自動返信または Vacation プログラムのオプションを設定します。詳細については、iPlanet Messaging Server リファレンスマニュアルを参照してください。

エイリアスファイル

エイリアスファイル aliases は、ディレクトリに設定されていないエイリアスを設定します。その例として、ルートのアドレスが挙げられます。このファイルで設定したエイリアスがディレクトリにもある場合は、ファイル内の設定が無視されます。エイリアスおよび aliases ファイルの詳細については、141 ページの「エイリアス」を参照してください。

aliases ファイルに変更を加えた場合は、必ず MTA を再起動してください。

TCP/IP (SMTP) チャネルオプションファイル

TCP/IP チャネルオプションファイルは、TCP/IP チャネルのさまざまな特性を制御します。ファイルには *x*_option という名前を付けてください。ファイル名の *x* はチャネル名となります。たとえば、/*ServerInstance*/config/imta/tcp_local_option のようになります。詳細は、216 ページの「SMTP チャネルオプションを設定する」を参照してください。すべてのチャネルオプションキーワードおよびシンタックスの詳細については、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

変換ファイル

変換ファイル conversions は、MTA を介して送受信されるメッセージの変換チャネルにおける変換方法を指定します。変換には、MTA トラフィックの任意のサブセットを選択できます。また、変換処理を行うには、プログラムまたはコマンドの任意のセットを使用できます。MTA は変換ファイルに基づいて、それぞれのメッセージ本文に対する適切な変換を選択します。

このファイルのシンタックスの詳細については、278 ページの「変換チャネル」を参照してください。

Dirsync オプションファイル

Dirsync オプションファイル `dirsync.opt` は、コマンドラインで設定できない `dirsync` プログラムのオプションを設定します。詳細については、114 ページの「dirsync の設定」と iPlanet Messaging Server リファレンスマニュアルを参照してください。

ディスパッチャ設定ファイル

ディスパッチャ設定ファイル `dispatcher.cnf` は、ディスパッチャの設定情報を指定します。インスール時に作成されたデフォルトの設定ファイルをそのまま使用することができます。ただし、セキュリティやパフォーマンスなどの理由でデフォルトの設定ファイルを変更する場合には、`dispatcher.cnf` ファイルを編集します (詳細については、100 ページの「ディスパッチャ」を参照)。

ディスパッチャ設定ファイルの形式は、ほかの MTA 設定ファイルの形式に似ています。オプションを指定する行は、次の形式で記述されています。

option=value

option はオプション名、*value* はオプションを設定する文字列または整数です。*option* に整数値を指定できる場合は、*b%v* の文字列表記規則を使って基数を指定できます。この場合、*b* は基数 10 で表される基数であり、*v* は基数 *b* で表される実際の値です。これらのオプションの仕様は、次のオプション設定を適用するサービスに対応するセクションにグループ分けされています。各行では、次の形式が使用されます。

`[SERVICE=`*service-name*`]`

service-name はサービスの名前です。最初のオプション仕様、すなわちこのようなセクションタグよりも前に記述されているオプション仕様はすべてのセクションに適用されます。

以下に、ディスパッチャ設定ファイル (dispatcher.cnf) の例を示します。

```
! The first set of options, listed without a [SERVICE=xxx]
! header, are the default options that will be applied to all
! services.
!
MIN_PROCS=0
MAX_PROCS=5
MIN_CONNS=5
MAX_CONNS=20
MAX_LIFE_TIME=86400
MAX_LIFE_CONNS=100
MAX_SHUTDOWN=2
!
! Define the services available to Dispatcher
!
[SERVICE=SMTP]
PORT=25
IMAGE=server_root/msg-instance/imta/lib/tcp_smtp_server
LOGFILE=server_root/msg-instance/imta/log/tcp_smtp_server.log
```

このファイルのパラメータの詳細については、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

マッピングファイル

マッピングファイル mappings は、MTA が入力文字列を出力文字列にマップする方法を定義します。

MTA コンポーネントの多くは、テーブル検索に基づいた情報を使用します。一般に、このタイプのテーブルは、入力文字列を出力文字列に変える (マップする) のに使用されます。このようなテーブルは、マッピングテーブルと呼ばれ、通常 2 つのカラムで構成されます。最初 (左側) のカラムには入力文字列が、2 番目 (右側) のカラムにはその入力文字列に関連付けられた出力文字列が並んでいます。MTA データベースのほとんどは、このタイプのマッピングテーブルです。ただし、MTA データベースファイルには、ワイルドカード検索機能がありません。データベース全体でワイルドカードに一致するものを検索するのは非効率的だからです。

マッピングファイルによって、MTA が複数のマッピングテーブルをサポートできるようになります。さらに、完全なワイルドカード機能もあり、複数の手順や反復マッピング方法にも対応しています。このアプローチは、データベースを使用する場合に比べ、さらに多くの処理を必要とします。特に、エントリ数が多い場合などはなおさらです。ただし、それに付随して柔軟性が増すため、同等のデータベースにおけるエントリのほとんどを必要としなくなり、全体的にオーバーヘッドが少なくなります。

`imsimta test -mapping` コマンドを使ってマッピングテーブルをテストすることができます。マッピングファイルのシンタックスおよび `test -mapping` コマンドの詳細については、117 ページの「マッピングファイル」および『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

オプションファイル

オプションファイル `option.dat` はグローバル MTA オプションを指定します。これはチャネル固有のオプションとは逆のオプションです。

オプションファイルを使って、MTA 全体に適用されるさまざまなパラメータのデフォルト値を無効にすることができます。特に、オプションファイルは、設定ファイルやエイリアスファイルが読み込まれるさまざまなテーブルのサイズを確立するのに使用されます。また、MTA が許可するメッセージのサイズを制御したり、MTA 設定で許可するチャネル数を指定したり、許可する書き換え規則の数を設定したりできます。

オプションファイルのシンタックスの詳細については、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

テイラーファイル

テイラーファイル `imta_tailor` は、さまざまな MTA コンポーネントの場所を設定します。MTA が正常に機能するには、`imta_tailor` ファイルが常に *ServerInstance*`/imta/config` ディレクトリ内になければなりません。

このファイルを編集して特定の設定にその変更を反映させることはできますが、その際には注意が必要です。このファイルを変更した場合は、必ず MTA を再起動してください。MTA が停止しているときに変更を行うのが望ましい方法です。

注	特に必要でないかぎり、このファイルを変更することは避けてください。

このファイルの詳細については、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

ジョブコントローラファイル

ジョブコントローラは、メッセージを配信するためのチャネルジョブを作成および管理します。これらのチャネルジョブは、ジョブコントローラ内の処理プール内で実行されます。プールは、チャネルジョブが実行される「場所」であると考えることができます。プールは、プール外のジョブとリソースを奪い合うことなく処理できる計算領域です。ジョブコントローラの概念とチャネルキーワードの設定については、108 ページの「ジョブコントローラ」、240 ページの「チャネル実行ジョブのプールを処理する」、および 241 ページの「サービスジョブの制限」を参照してください。

ジョブコントローラファイル job_controller.cnf では、以下のチャネル処理情報を指定します。

- さまざまなプールを定義する
- すべてのチャネルに対し、マスタープログラム名とスレーブプログラム名を指定する (該当する場合)

imta.cnf ファイル では、pool キーワードを使ってプロセスプール (job_controller.cnf で定義) の名前を指定できます。たとえば、次のサンプルファイル job_controller.cnf の要素は、プール MY_POOL を定義します。

```
[POOL=MY_POOL]
job_limit = 12
```

次のサンプルファイル imta.cnf の要素は、チャネルブロック内でプール MY_POOL を指定します。

```
channel_x pool MY_POOL
channel_x-daemon
```

デフォルトのプール設定に関連付けられたパラメータを変更したり、プールを追加する場合は、job_controller.cnf ファイルを編集し、ジョブコントローラをいったん終了してから再起動してください。

新しい設定を使用して新規のジョブコントローラプロセスが作成され、それ以降の要求を受信するようになります。古いジョブコントローラプロセスは、キューに残っている要求をすべて処理してから終了します。

ジョブコントローラ設定ファイルの最初のプールは、プール名が指定されていないすべての要求に使用されます。MTA 設定ファイル (imta.cnf) で定義されている MTA チャネルは、後ろにプール名が続く pool チャネルキーワードを使って、特定のプールに処理要求を送ることができます。このプール名は、ジョブコントローラ設定のプール名と一致しなければなりません。ジョブコントローラが要求されたプール名を認識できない場合、その要求は無視されます。

最初の設定で、次のプールを定義します。DEFAULT, LOCAL_POOL, IMS_POOL, SMTP_POOL.

使用例

通常、特定のチャネルの処理を別のチャネルの処理と区別する場合は、ジョブコントローラ設定に付加的なプール定義を追加します。また、特性が異なるプールを使用することもできます。たとえば、チャネルが処理できる同時要求の数を制御する必要があるとします。これを行うには、ジョブ範囲を設定した新規プールを作成し、pool チャネルキーワードを使ってチャネルをより適切なプールに割り当てます。

プール定義のほかに、ジョブコントローラ設定ファイルには、各チャネルの要求を処理するのに必要な MTA チャネルとコマンドのテーブルが含まれています。要求には「マスター」と「スレーブ」の 2 種類があります。通常、チャネルの MTA メッセージキューにメッセージがあると、チャネルマスタープログラムが起動します。マスタープログラムは、メッセージをキューから取り出します。

スレーブプログラムは、チャネルをポーリングし、そのチャンネル内の受信メッセージを取り込むために呼び出されます。マスタープログラムはほぼすべての MTA チャネルにありますが、スレーブプログラムは MTA チャネルにはほとんどなく、必要とされません。たとえば、TCP/IP を介して SMTP を処理するチャネルではスレーブプログラムを使用しません。これは、すべての SMTP サーバからの要求に対して、ネットワークサービスである SMTP サーバが受信 SMTP メッセージを受け取るためです。SMTP チャネルのマスタープログラムは、MTA の SMTP クライアントです。

チャネルに関連付けられた宛先システムが一度に複数のメッセージを処理できない場合は、ジョブ範囲が 1 である新しいタイプのプールを作成する必要があります。

```
[POOL=single_job]
job_limit=1
```

一方、宛先システムで並行処理が可能な場合は、ジョブ範囲の値を増やすことができます。

コード例 6-1 に、ジョブコントローラ設定ファイルの例を示します。表 6-8 に使用できるオプションを示します。

コード例 6-1 ジョブコントローラ設定ファイルの例 (UNIX)

```
!MTA Job Controller configuration file
!
!Global defaults
tcp_port=27442          (1)
secret=never mind
return_job=server_root/bin/msg/imta/bin/return.sh
return_time=00:30/24:00
purge_job=server_root/bin/msg/imta/bin/purge
purge_argv=-num=5
slave_command=NULL      (2)
max_life_age=3600       (3)
!
!
```

コード例 6-1 ジョブコントローラ設定ファイルの例 (UNIX)

```
!Pool definitions
!
[POOL=DEFAULT]          (4)
job_limit=10            (5)
!
[POOL=LOCAL_POOL]
job_limit=10
!
[POOL=IMS_POOL]
job_limit=1
!
[POOL=SMTP_POOL]
job_limit=1
!
!Channel definitions
!
!
[CHANNEL=l]             (6)
master_command=server_root/bin/msg/imta/bin/l_master
!
[CHANNEL=ims-ms]
master_command=server_root/bin/msg/imta/bin/ims_master
!
[CHANNEL=tcp_*]         (7)
anon_host=0
master_command=server_root/bin/msg/imta/bin/tcp_smtp_client
```

以下に、上の例の主な項目 (太字の丸括弧付きの数字がある部分) について説明します。

1. このグローバルオプションは、ジョブコントローラが要求を待機する TCP ポート番号を定義します。
2. そのあとの [CHANNEL] セクションのデフォルト SLAVE_COMMAND を設定します。
3. そのあとの [CHANNEL] セクションのデフォルト MAX_LIFE_AGE を設定します。
4. この [POOL] セクションは、DEFAULT という名前のプールを定義します。
5. このプールの JOB_LIMIT を 10 に設定します。
6. この [CHANNEL] セクションは、l という名前のチャネル (UNIX ローカルチャネル) に適用されます。このセクションに必要な定義は、ジョブコントローラがこのチャネルを実行するために発行する master_command だけです。このチャネル名にはワイルドカードが含まれていないため、チャネル名は完全に一致しなければなりません。
7. この [CHANNEL] セクションは、tcp_* で始まるすべてのチャネル名に適用されます。このチャネル名にはワイルドカードが含まれているため、tcp_ で始まるすべてのチャネルに一致します。

追加プールの例

ジョブコントローラは、メッセージを配信するためのチャネルジョブを作成および管理します。これらのチャネルジョブは、ジョブコントローラ内の処理プール内で実行されます。プールは、チャネルジョブが実行される「場所」であると考えることができます。プールは、プール外のジョブとリソースを奪い合うことなく処理できる計算領域です。ジョブ範囲は、job_controller にプールごとに設定されます。たとえば、SMTP_POOL の job_limit を 10 と定義すれば、このプールで実行できる tcp_smtp クライアントプロセスは常に 10 個だけです。

tcp_* チャネルを追加する必要があることもあります。たとえば、メール処理が非常に遅いサイト用の tcp チャネルなどです。このようなチャネルは別のプールで実行することをお勧めします。理由は、tcp_* チャネルを 10 個作成し、SMTP_POOL ですべてを実行する場合は、tcp_* チャネルごとに常に 1 つの tcp_smtp だけを実行することが可能であるからです (ただし、メールの宛先がすべて tcp_* チャネルであり、SMTP_POOL が 10 個の job_limit で定義されている場合)。システムに大きな負荷があり、どのキューにも複数の tcp_* チャネル宛の待機メッセージがある場合は、十分ではありません。スロットが競合しないように、新しい tcp_* チャネルに別のプールを定義することも考えられます。

たとえば、次の tcp_* チャネルを設定する場合を考えてみます。

```
tcp_yahoo smtp mx pool yahoo_pool keyword keyword keyword
tcp-yahoo-daemon

tcp_aol smtp mx keyword keyword keyword pool aol_pool
tcp-aol-daemon

tcp_hotmail smtp mx pool hotmail_pool keyword keyword keyword
tcp-hotmail-daemon

...

tcp_sun smtp mx pool sun_pool keyword keyword keyword
tcp-sun-daemon
```

新規チャネルごとに 10 個の tcp_smtp_client 処理を追加するには、job_controller.cnf ファイルに次のように追加します。

```
[POOL=yahoo_pool]
job_limit=10

[POOL=aol_pool]
job_limit=10

[POOL=hotmail_pool]
job_limit=10

...

[POOL=sun_pool]
job_limit=10
```

プールについては、240 ページの「チャネル実行ジョブのプールを処理する」を参照してください。ジョブコントローラファイルのシンタックスの詳細については、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

表 6-8 ジョブコントローラ設定ファイルのオプション

オプション	説明
一般的なオプション	**説明**
INTERFACE_ADDRESS=*adapter*	ジョブコントローラがバインドする IP アドレスインタフェースを指定する。値 (アダプタ) には、ANY、ALL、LOCALHOST、または IP アドレスのいずれかを指定できる。デフォルトでは、ジョブコントローラはすべてのアドレスにバインドする (ALL または ANY の指定に相当)。INTERFACE_ADDRESS=LOCALHOST を指定すると、ジョブコントローラは、ローカルマシンからの接続しか受け付けない。これは、ジョブコントローラではマシン間の操作はサポートされていないため、通常の操作には影響はない。ただし、HA エージェントがジョブコントローラの応答をチェックする HA 環境では、不適切な場合がある。Messaging Server の実行しているマシンが HA 環境にあり、「内部ネットワーク」アダプタと「外部ネットワーク」アダプタがある場合で、大きなポート番号への接続をブロックするファイアウォール機能の信頼性が低い場合は、「内部ネットワーク」アダプタの IP アドレスを指定することを推奨

表 6-8　ジョブコントローラ設定ファイルのオプション (続き)

オプション	説明
MAX_MESSAGES=*integer*	ジョブコントローラは、メモリ内構造でメッセージに関する情報を保持する。バックログが大きくなった場合は、この構造のサイズを制限する必要がある。バックログのメッセージ数がこのパラメータ値を超えると、その後のメッセージに関する情報はメモリに保存されない。メールメッセージは常にディスクに書き込まれるため、失われることはないが、ジョブコントローラが認識するメッセージ数の半数になるまで配信されない。この時点では、ジョブコントローラが、imsimta cache -sync コマンドと同じように、キューディレクトリをスキャンする デフォルトは 100000 です。
SECRET=*file_spec*	ジョブコントローラに送信される要求を保護するための共有の秘密情報
SYNCH_TIME=*time_spec*	ジョブコントローラは定期的にディスク上のキューファイルをスキャンしてファイルが不足していないかどうかをチェックする。デフォルトでは 4 時間ごとにスキャンする (ジョブコントローラの起動後 4 時間たってから開始)。*time_spec* の形式は、*HH*:*MM*/*hh*:*mm* または /*hh*:*mm* である。*hh*.*mm* 変数は、イベントの間隔を時間数 (*h*) と分数 (*m*) で示す。*HH*:*MM* 変数は、1 日の中でイベントが最初に発生する時間である。たとえば 15:45/7:15 と指定すると、15:45 にイベントが開始され、その後 7 時間 15 分ごとにイベントが実行される
TCP_PORT=*integer*	ジョブコントローラが要求パケットを待機する TCP ポートを指定する。このオプションは、デフォルト値がシステム内の別の TCP アプリケーションと競合しない限り変更してはならない。このオプションを変更する必要がある場合は、対応する MTA テイラーファイル (*server_root*/msg-*instance*/imta/config/imta_tailor) の IMTA_JBC_SERVICE オプションも同じように変更する必要がある。TCP_PORT オプションはグローバルに適用され、[CHANNEL] セクションまたは [POOL] セクション内にある場合は無視される
TIME=*time_spec*	[PERIODIC_JOB] セクションの定期ジョブを実行する時間と頻度を指定する。デフォルト設定は /4:00 で、ジョブが 4 時間ごとに実行される。*time_spec* の形式は *HH*:*MM*/*hh*:*mm* または /*hh*:*mm* である。*hh.mm* はイベントの間隔の時間数 (*h*) と分数 (*m*)。*HH*:*MM* は、1 日の中でジョブが発生する最初の時間。たとえば 15:45/7:15 と指定すると、15:45 にイベントが開始され、その後 7 時間 15 分ごとにイベントが実行される
プールオプション	**説明**
JOB_LIMIT=*integer*	プールが同時に使用できるプロセスの最大数を指定する。JOB_LIMIT は各プールに個別に適用される。ジョブの最大合計数は、すべてのプールの JOB_LIMIT パラメータの合計数である。この値をセクションの外に設定すると、JOB_LIMIT が指定されていない [POOL] セクションにより、デフォルトとして使用される。[CHANNEL] セクション内では、このオプションは無視される

表 6-8　ジョブコントローラ設定ファイルのオプション (続き)

オプション	説明
チャネルオプション	説明
MASTER_COMMAND=*file_spec*	チャネルを実行し、そのチャネルからメッセージを取り出すために、ジョブコントローラによって作成された UNIX システムプロセスが実行するコマンドのフルパスを指定する。このオプションをセクションの外に設定すると、MASTER_COMMAND が指定されていない [CHANNEL] セクションによりデフォルトとして使用される。[POOL] セクション内では、このオプションは無視される
MAX_LIFE_AGE=*integer*	チャネルマスタージョブに対する最大のライフタイムを秒数で指定する。このパラメータがチャネルに指定されていない場合は、グローバルなデフォルト値が使用される。デフォルト値が指定されていない場合は、1800 (30 分) が使用される
MAX_LIFE_CONNS=*integer*	マスターチャネルの寿命は、最長使用期間パラメータのほか、メッセージがあるかどうかをジョブコントローラに確認する回数によっても制限される。このパラメータがチャネルに指定されていない場合は、グローバルなデフォルト値が使用される。デフォルト値が指定されていない場合は 300 が使用される
SLAVE_COMMAND=*file_spec*	チャネルを実行し、そのチャネルで受信するすべてのメッセージをポーリングするために、ジョブコントローラによって作成された UNIX システムプロセスが実行するコマンドのフルパスを指定する。ほとんどの場合、MTA チャネルには SLAVE_COMMAND がない。その場合は、予約値である NULL を指定する。このオプションをセクションの外に設定すると、SLAVE_COMMAND が指定されていない [CHANNEL] セクションによりデフォルトとして使用される。[POOL] セクション内では、このオプションは無視される

エイリアス

MTA には、ローカルシステムに関連付けられたメールボックス名をサポートする機能である「エイリアス」があります。これは、必ずしも実際のユーザに対応するとは限りません。エイリアスは、メーリングリストの作成、メールの転送、およびユーザの別名の設定に役立ちます。

注　この節では、主に dirsync モードでのエイリアス処理について説明します。ダイレクト LDAP モードでのエイリアス解決については、542 ページの「ダイレクト LDAP 書き換え規則 ($V) でアドレスを解決する」を参照してください。

エイリアスを適用できるのは、l チャネルまたは aliaslocal キーワードの付いたすべてのチャネルに一致するアドレスだけです。MTA のメッセージ送信ロジックが l チャネルまたは aliaslocal キーワードの付いたすべてのチャネルに一致するアドレスを識別するたびに、アドレスに指定されているメールボックス (ユーザ名など) がエイリアスデータベースまたはエイリアスファイル内の各エントリと照合されます。一致するエントリが見つかると、エイリアスアドレスは変換値またはエイリアスで指定された値に置き換えられます。エイリアスは、追加エイリアスまたは実際のアドレスによる任意の組み合わせに変換できます。実際のアドレスが l チャネルまたは aliaslocal キーワードの付いたすべてのチャネルに一致する必要はありません。したがって、エイリアスは、リモートシステムにメールを転送するのに使用することができます。

本当にチャネルに一致するとみなされるアドレスは Envelope To アドレスのみであるため、エイリアスは Envelope To アドレスにしか適用されません。MTA は、アドレスの書き換えが完了したあとにのみエイリアスの変換および拡張を行います。エイリアスによって生成された変換値は、完全に新しいアドレスとして扱われ、最初から処理されます。

エイリアスデータベース

MTA はディレクトリ内の情報を使用し、エイリアスデータベースを作成します。このエイリアスデータベースは、標準のエイリアスファイルが参照されるたびに参照されます。ただし、エイリアスデータベースのエントリが調べられるのは、標準のエイリアスファイルが使用される前です。すなわち、デーベースは、エイリアスファイルが使用される前に実行される、一種のアドレス書き換え機能として動作します。エイリアスデータベースにユーザおよび配布リストのエントリを作成するためのディレクトリ属性については、『iPlanet Messaging Server プロビジョニングガイド』を参照してください。

注	データベースの形式は固有のものです。直接データベースを編集するのではなく、必要な変更はディレクトリ内で行うようにしてください。

エイリアスファイル

aliases ファイルは、ディレクトリで設定されていないエイリアスを設定するのに使用します。例としては、ポストマスターエイリアスがあります。このファイルで設定したエイリアスがディレクトリにもある場合、このファイルの設定は無視されます。変更を有効にするには、MTA を再起動する必要があります。感嘆符 (!) で始まる行は、コメント行として解釈されるため、無視されます。また、空白行も無視されます。

注 Messaging Server には、アドレスリバースデータベースや特殊化されたマッピングテーブルなど、アドレス操作のためのその他の機能もあります。ただし、アドレス操作を実行する可能性がある場合には、常に書き換え規則を使用するようにしてください。それにより、最高のパフォーマンスを得ることができます。第 7 章「書き換え規則を設定する」を参照してください。

このファイルでは、1 行に入力できる文字数が 1024 バイトに制限されています。円記号 (¥) を継続文字として使用すれば、1 つの論理行を複数の行に分割することができます。

ファイル形式は以下のとおりです。

user@domain： <*address*> （ホストドメイン内のユーザ用）

user@domain： <*address*> （ホストドメイン内のユーザ用。例：デフォルトドメイン）

たとえば、以下のようになります。

```
! A /var/mail/ user
inetmail@siroe.com: inetmail@native-daemon

! A message store user
ms_testuser@siroe.com: mstestuser@ims-ms-daemon
```

エイリアスファイルにほかのファイルを含める

プライマリ aliases ファイルには、ほかのファイルを含めることができます。次の行は、MTA に file-spec ファイルを読み込むように指示するためのものです。

```
<file-spec
```

ファイル仕様は、完全なパスを指定したものでなければなりません。また、そのファイルには、プライマリ aliases ファイルと同じ保護が設定されている必要があります (たとえば、だれでも読み込み可能であることなど)。

含まれているファイルの内容は、aliases ファイル内の参照ポイントに挿入されます。含まれているファイルへの参照をそのファイルの実際の内容に置き換えることによっても、同様の効果が得られます。含まれているファイルの形式は、プライマリ aliases ファイルとまったく同じになります。さらに、含まれているファイルにほかのファイルを含めることも可能です。ファイルは、3 階層までネストすることができます。

コマンドラインユーティリティ

iPlanet Messaging Server には、MTA のさまざまな保守、テスト、管理などのタスクを行うためのコマンドラインユーティリティが備わっています。たとえば、MTA の設定、エイリアス、マッピング、セキュリティ、システム全体のフィルタファイル、およびオプションファイルをコンパイルするには、imsimta cnbuild コマンドを使用します。また、MTA ディレクトリキャッシュを再作成または更新するには、imsimta dirsync コマンドを使用します。MTA コマンドラインユーティリティの詳細については、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

SMTP セキュリティとアクセス制御

SMTP セキュリティとアクセス制御については、第 10 章「メールのフィルタリングとアクセス制御」および第 12 章「セキュリティとアクセス制御を設定する」を参照してください。

ログファイル

MTA 専用のログファイルはすべて、MTA ログディレクトリ (*ServerInstance*/log/imta/) に保存されます。このディレクトリには、MTA を介したメッセージトラフィックのログファイル、および特定のマスタープログラムまたはスレーブプログラムの情報を記述したログファイルがあります。

MTA ログファイルの詳細については、第 13 章「ログ記録とログ解析」を参照してください。

内部形式から公的な形式にアドレスを変換するには

アドレスは、アドレスリバースデータベースと REVERSE マッピングテーブルを使用して内部形式から公的なアドバタイズ形式に変換することができます。たとえば、uid@mailhost.siroe.com は、siroe.com ドメイン内では有効なアドレスであっても、外部に公開するには適切なアドレスではない場合があります。この場合は、firstname.lastname@siroe.com のような公式アドレスを使用することをお勧めします。

注	Messaging Server には、aliases ファイルや特殊化されたマッピングテーブルなど、アドレス操作のためのその他の機能もあります。ただし、アドレス操作を実行する可能性がある場合には、常に書き換え規則を使用するようにしてください。それにより、最高のパフォーマンスを得ることができます。第 7 章「書き換え規則を設定する」を参照してください。

リバースデータベースでは、各ユーザの公式アドレスはディレクトリ内のユーザエントリの mail 属性で指定されています。プライベートアドレスや内部アドレスは、mailAlternativeAddress 属性で指定されています。配布リストについても同様です。

リバースデータベースには、有効なアドレスと公式アドレスとの間のマッピングが含まれています。リバースデータベースは、imsimta dirsync コマンドを実行するたびに更新および作成されます。ダイレクト MTA LDAP 操作 (付録 B「MTA ダイレクト LDAP 操作」を参照) を有効にした場合、アドレスリバースデータベースは使われません。

通常、リバースデータベースは MTA データベースディレクトリにあります。このデータベースは、*server_root*/msg-*instance*/imta/config/imta_tailor ファイルの IMTA_REVERSE_DATABASE オプションで名前が指定されているファイルで構成されます。特に設定を変更しないかぎり、これらのファイルは *server_root*/msg-*instance*/imta/db/reversedb.* です。

データベース内でアドレスが見つかった場合は、そのデータベースの対応する右側部分がアドレスとして置き換えられます。アドレスが見つからなかった場合は、マッピングファイルで REVERSE という名前のマッピングテーブルが検索されます。このマッピングテーブルが存在しない場合、またはマッピングテーブル内に一致するエントリがない場合には、置換は行われず、書き換えは通常どおりに終了します。

REVERSE マッピングテーブルがマッピングファイル内にあり、アドレスがマッピングエントリと一致すると、そのエントリが $Y を指定している場合は、結果の文字列によってアドレスが置き換えられます。$N を指定している場合は、マッピングの結果が破棄されます。マッピングエントリが $Y のほかに $D を指定している場合は、結果の文字列を使ってもう一度リバースデータベースがスキャンされます。一致するエントリが見つかった場合は、データベースのテンプレートによってマッピングの結果 (つまりアドレス) が置き換えられます。一般的な REVERSE マッピングテーブルエントリ (すべてのチャネルに適用されるエントリ) の形式は、以下のとおりです。フラグは、新しいアドレスの前または後ろに指定できます。

```
REVERSE

  OldAddress        $Y[Flags]NewAddress
```

チャネル固有のエントリ (特定のチャネルから渡されるメッセージ上でのみ発生するマッピング) の形式は、以下のとおりです。チャネル固有のエントリを機能させるには、option.dat で use_reverse_database を 13 に設定する必要があります。

```
REVERSE

  source-channel|destination-channel|OldAddress  $Y[Flags]NewAddresS
```

REVERSE マッピングテーブルフラグを表 6-9 に示します。

表 6-9　REVERSE マッピングテーブルのフラグ

フラグ	説明
$Y	出力文字列を新規アドレスとして使用する
$N	アドレスは変更されない
$D	出力文字列を使ってリバースデータベースをスキャンする
$A	パターンをリバースデータベースエントリとして追加する
$F	パターンを転送データベースエントリとして追加する
フラグの比較	説明
$:B	ヘッダー (本文) のアドレスのみを照合する
$:E	エンベロープアドレスのみを照合する
$:F	前方を探すアドレスのみを照合する
$:R	後方を探すアドレスのみを照合する
$:I	メッセージ ID のみを照合する

アドレスリバース制御を設定するには

reverse チャネルキーワードと noreverse チャネルキーワード、および MTA の USE_REVERSE_DATABASE オプションと REVERSE_ENVELOPE オプションを使用して、アドレスリバースを適用する時期や方法などの指定を制御できます。デフォルトでは、アドレスリバース操作は、後方を探すアドレスだけではなく、すべてのアドレスに適用されます。

アドレスリバースは、REVERSE_ENVELOPE システムオプションの値を設定することによって (デフォルト : 1-on、0-off)、有効または無効にすることができます。

宛先チャネル上の noreverse は、アドレスリバースがメッセージ内のアドレスに適用されないことを指定します。reverse は、アドレスリバースが適用されることを指定します。詳細は、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

USE_REVERSE_DATABASE は、MTA が置換アドレスとしてアドレスリバースデータベースと REVERSE マッピングを使用するかどうかを制御します。値 0 は、アドレスリバースがどのチャネルでも使われないことを示します。値 5 (デフォルト) は、アドレスリバースが、MTA アドレス書き換えプロセスによる書き換え後に、後方を探すア

ドレスだけではなく、すべてのアドレスに適用されることを指定します。値 13 は、アドレスリバースが、MTA アドレス書き換えプロセスによる書き換え後に、後方を探すアドレスだけではなく、reverse チャネルキーワードを含むアドレスに適用されることを指定します。また、USE_REVERSE_DATABASE オプションのビット値を設定して、アドレスリバース操作の単位を指定することもできます。詳細は、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

REVERSE_ENVELOPE オプションは、メッセージヘッダーアドレスとともにエンベロープ From アドレスにもアドレスリバースを適用するかどうか制御します。

これらの効果の詳細については、iPlanet Messaging Server リファレンスマニュアルの各オプションおよびキーワードの説明を参照してください。

一般的なリバースマッピングの例

一般的なリバースマッピングの例を以下に示します。この例では、siroe.com の内部アドレスの形式が user@mailhost.siroe.com であると仮定しています。ただし、ユーザのネームスペースでは、user@host1.siroe.com と user@host2.siroe.com が siroe.com のすべてのホストで同じユーザを指定しています。以下の REVERSE マッピングは、アドレスリバースデータベースとともに使用できます。

```
REVERSE

  *@*.siroe.com         $0@siroe.com$Y$D
```

この例では、*name*@*anyhost*.siroe.com という形式のアドレスが *name*@siroe.com に変更されています。$D メタ文字では、アドレスリバースデータベースが参照されるようになります。アドレスリバースデータベースには、以下の形式のエントリが含まれています。

```
user@mailhost.siroe.com    first.last@siroe.com
```

チャネル固有のリバースマッピングの例

デフォルトでは、ルーティングの範囲がメールサーバドメインに設定されている場合に、アドレスリバースデータベースが使用されます。チャネル固有の REVERSE マッピングテーブルエントリの例を以下に示します。

```
REVERSE

  tcp_*|tcp_local|binky@macho.siroe.com     $D$YRebecca.Woods@siroe.com
```

このエントリは、MTA に対して、ソースチャネル tcp_* から宛先チャネル tcp_local に送信されるすべてのメールのアドレスの形式を、binky@macho.siroe.com から Rebecca.Woods@siroe.com に変更するように指示します。

注 チャネル固有のリバースマッピングを有効にするは、option.dat の USE_REVERSE_DATABASE オプションを 13 に設定する必要があります (デフォルト =5)。

FORWARD アドレスマッピング

アドレスリバースは、エンベロープ To アドレスには適用されません。これらのアドレスは、メッセージがメールシステム内で処理される際に常に書き換えられ、変更されます。ルーティングの目的は、エンベロープ To アドレスをシステム固有またはメールボックス固有のフォーマットに変換していくことです。アドレスリバースの公認機能は、エンベロープ To アドレスに対して適当ではありません。

エンベロープ To アドレスのさまざまな代替機構によって、リバースデータベースと同等の機能が提供されますが、リバースマッピングと同じ機能はありません。エンベロープ To アドレスのマッピング機能が有用で、望ましい場合もあります。

この不足している機能は、FORWARD マッピングテーブルによって補われます。マッピングファイル内に FORWARD マッピングテーブルがある場合、それは各エンベロープ To アドレスに適用されます。このマッピングテーブルがない場合や一致するエントリがマッピングテーブルにない場合、変更は行われません。

アドレスに一致するマッピングエントリがある場合は、マッピングの結果がテストされます。エントリが $Y を指定している場合は、結果の文字列によってエンベロープ To: アドレスが置き換えられ、エントリが $N を指定している場合は、マッピングの結果が破棄されます。

転送データベースを使用してメールを転送する

転送データベースを使用して、エイリアスファイルやエイリアスデータベースを使用した場合と同様の転送を行うことができます。ただし、エイリアスファイルやエイリアスデータベースを使用できる場合は、転送データベースよりも効率がよいため、それらの使用をお勧めします。

一般に、転送するメッセージのソースによって異なる種類の転送が必要な場合は、転送データベースによるメール転送が適しています。転送データベースによる転送は、USE_FORWARD_DATABASE オプションを使用して、ソース固有の設定を行うことができます。詳細については、iPlanet Messaging Server リファレンスマニュアルを参照してください。

配信ステータス通知メッセージを制御する

配信ステータス通知、すなわち通知メッセージは、MTA が差出人に送信する電子メールステータスメッセージで、ポストマスターに送信することもできます。Messaging Server では、通知メッセージの内容や言語をカスタマイズすることができます。また、配信ステータス (たとえば、FAILED、BOUNCED、TIMEDOUT など) の種類ごとに異なるメッセージを作成することもできます。さらに、特定のチャネルから送信されたメッセージに関して通知メッセージを作成することもできます。

デフォルトでは、通知メッセージは、*server_root*/msg-*instance*/imta/config/locale/C/LC_MESSAGES/ ディレクトリのファイルに保存されています (/*server_root*/msg-*instance*/imta/config/imta_tailor ファイルの IMTA_LANG 設定で指定)。次のような種類があります。

return_bounced.txt, return_delivered.txt, return_header.opt, return_timedout.txt, return_deferred.txt, return_failed.txt, return_prefix.txt, return_delayed.txt, return_forwarded.txt, return_suffix.txt.

これらのファイルには直接変更を加えないでください。これらのファイルは、iPlanet Messaging Server の更新時に上書きされます。ファイルを変更して独自の通知メッセージテンプレートファイル (return_*.txt) として使用する場合は、新しいディレクトリにファイルをコピーし、そちらを編集してください。次に imta_tailor ファイルに IMTA_LANG オプションを設定し、このテンプレートがある新しいディレクトリを指定します。通知ファイルのセットを複数作成する場合は (言語別のセットを作成する場合など)、NOTIFICATION_LANGUAGE マッピングテーブルを設定する必要があります。

通知メッセージを作成および変更するには

通知メッセージは、return_prefix.txt、return_*ActionStatus*.txt、return_suffix.txt の 3 ファイルのセットで構成されています。

通知をカスタマイズまたはローカライズするには、ロケールやカスタマイズごとに return_*.txt ファイルのセットを作成し、別のディレクトリに保存します。たとえば、あるディレクトリにはフランス語の通知ファイルを保存し、別のディレクトリにはスペイン語の通知ファイルを保存し、さらに別のディレクトリには特別な不特定多数宛てメールのチャネル用の通知ファイルを保存することができます。

注	このリリースには、フランス語、ドイツ語、およびスペイン語のサンプルファイルが含まれています。これらのファイルは、必要に応じて変更できます。

通知メッセージの形式と構造は次のとおりです。

1. return_prefix.txt には、該当するヘッダーテキストと本文の導入部分が含まれます。米国英語のロケールのデフォルトは以下のとおりです。

   ```
   Content-type: text/plain; charset=us-asci
   Content-language: EN-US

   This report relates to a message you sent with the following
   header fields: %H
   ```

   US-ASCII 以外の通知メッセージの場合は、charset パラメータと Content-Language ヘッダーを適切な値に変更する必要があります (たとえばフランス語用のファイルでは ISO-8859-1 と fr になります)。%H は、表 6-10 で定義されているヘッダー置換シーケンスです。

2. return_<*ActionStatus*>.txt にはステータス専用のテキストが含まれています。*ActionStatus* は、メッセージの MTA ステータスの種類です。たとえば、デフォルトでは return_failed.txt のテキストは次のようになります。

   ```
   Your message cannot be delivered to the following recipients:
   %R
   ```

   return_bounced.txt のデフォルトのテキストは次のようになります。

   ```
   Your message is being returned.  It was forced to return by
   the postmaster.

   The recipient list for this message was:
   %R
   ```

3. return_suffix.txt には結びのテキストが含まれます。デフォルトでは、このファイルは空白です。

表 6-10　通知メッセージの置換シーケンス

置換	定義
%H	メッセージのヘッダーに展開する
%C	メッセージがキューに入っていた時間の単位[1] に展開する
%L	返送されるまでメッセージがキューに置かれていた時間の単位[1] に展開する
%F	メッセージがキュー内に留まることができる時間の単位[1] に展開する
%S [%s]	以前展開した数値が 1 以外の場合は、S または s に展開する。たとえば、"%C 日 %s" は、メッセージがキューに入っていた日数によって「1 日」または「2 日」などに展開できる
%U [%u]	使用する時間の単位[1] (時間または日) に展開する。たとえば、「%C %U%s」は、メッセージがキューに入っていた時間と MTA オプションの値 RETURN_UNITS によって「2 日」や「1 時間」などに展開できる。RETURN_UNITS=1 (時間) を設定し、サイトでローカライズした英語以外の通知メッセージを使用している場合は、return_delayed.txt と return_timedout.txt を編集し、「日」に相当する単語を時間を表す単語と置き換える必要がある。たとえば、フランス語では、jour(s) を heure(s) と置き換える。ドイツ語では、Tag(e) を Stunde(n) と置き換える。スペイン語では、d 誕 (s) を hora(s) と置き換える
%R	メッセージの受取人のリストに展開する
%%	% (テキストの置換シーケンスは、文字セットに関係なくバイト単位でスキャンされる。2 バイトの文字セットを使用する場合は、意図しない % 記号を確認する必要がある)

1. 単位は時間または日 (デフォルト) で、MTA オプションファイルの RETURN_UNITS オプションで定義される

通知メッセージをカスタマイズおよびローカライズするには

通知メッセージをローカライズして、言語別に異なるユーザにメッセージを返すことができます。たとえば、フランス語を使用しているユーザにフランス語の通知を返すことができます。

通知メッセージのローカライズまたはカスタマイズは、次の 2 つの手順で行います。

1. ローカライズまたはカスタマイズされた return_*.txt メッセージファイルのセットを作成し、別々のディレクトリに保存します。手順については、151 ページの「通知メッセージを作成および変更するには」を参照してください。
2. NOTIFICATION_LANGUAGE マッピングテーブルを設定します。

NOTIFICATION_LANGUAGE マッピングテーブル (*server_root*/*msg*-instance/imta/config/mappings) は、送信元メッセージ (通知を送信するメッセージ) の属性 (言語、国、ドメイン、アドレスなど) に基づいて、ローカライズまたはカスタマイズされた通知メッセージファイルのセットを指定します。

元の差出人のメッセージが解析され、ステータス通知の種類、ソースチャネル、優先言語、返信アドレス、および最初の受取人が決定されます。テーブルの構築方法によって異なりますが、通知ファイルのセットは 1 つ以上の属性によって選択されます。

NOTIFICATION_LANGUAGE マッピングテーブルの形式は次のとおりです。

```
NOTIFICATION_LANGUAGE

 dsn-type-list|source-channel|preferred-language|return-address|first-recipient ¥
$Idirectory-spec
```

dsn-type-list は、配信ステータス通知の種類のカンマ区切りリストです。数の種類を指定する場合はカンマで区切ります。スペースでは区切りません。スペースを使用すると、マッピングテーブルエントリのパターンが終了します。次のような種類があります。

failed - 一般的な、永続的失敗を示すメッセージ (「そのようなユーザはありません」など)。return_failed.txt ファイルが使用される

bounced - ポストマスターが手動で「バウンス」した場合に使用される通知メッセージ。return_bounced.txt ファイルが使用される

timedout - MTA が、指定された配信期間内にメッセージを配信できなかったことを示す。メッセージは送り返される。return_timedout.txt ファイルが使用される

delayed - MTA が、メッセージを配信できなかったが、引き続き配信を試みていることを示す。return_delayed.txt ファイルが使用される

deferred - 「delayed」に類似した配信不能通知。ただし、MTA が配信を試みる期間は表示されない。return_deferred.txt ファイルが使用される

forwarded - このメッセージに対して配信受理が要求されていたが、このメッセージは配信受理がサポートされていないシステムに転送されたことを示す。return_forwarded.txt ファイルが使用される

source-channel は通知メッセージを生成するチャネル、つまり現在メッセージがキューに入っているチャネルです。たとえば、メッセージストアの配信キューの ims-ms、送信用 SMTP キューの tcp_local などがあります。

preferred-language は、処理中のメッセージ (通知を生成中のメッセージ) で使用される言語です。この情報のソースは、第 1 に accept_language フィールドです。このフィールドにない場合は、Preferred-language: ヘッダーフィールドと X-Accept-Language: ヘッダーフィールドが使用されます。標準の言語コード値のリストは、*server_root*/bin/msg/install/templates/msg-inst/msg/imta/config/languages.txt ファイルを参照してください。

このフィールドには、空の場合を除き、メッセージの発信者が Preferred-language: ヘッダー行または X-Accept-language: ヘッダー行で指定したものが使用されます。このため、意味のない文字が使用されることもあります。

return-address は、送信元メッセージのエンベロープ From: *address* です。これは、通知メッセージの送信先となるエンベロープアドレスであり、使用言語の手掛かりになることがあります。

first-recipient はエンベロープ To: アドレス (メッセージが複数の受取人に届かない場合の最初の受取人のアドレス) で、元のメッセージの宛先です。たとえば、「dan@siroe.com へのメッセージは配信されませんでした」という通知では、dan@siroe.com が、報告を受ける、エンベロープ To: アドレスです。

directory-spec は、マッピングテーブルのプローブに一致する場合に使用する return_*.txt ファイルを含むディレクトリです。$I の後ろにディレクトリの指定が続きます。

たとえば、フランス語の通知ファイル (return_*.txt) が /lc_messages/table/notify_french/ ディレクトリにあり、スペイン語の通知ファイル (return_*.txt) が /lc_messages/table/notify_spanish/ ディレクトリにあるサイトでは、次のようなテーブルを使用できます。各エントリは 1 つまたは複数のスペースで始まり、エントリ間には空白行はありません。

```
NOTIFICATION_LANGUAGE

! Preferred-language: header value specified
!
   *|*|fr|*|*     $I/lc_messages/table/notify_french/
   *|*|es|*|*     $IIMTA_TABLE/notify_spanish/
   *|*|en|*|*     $I/imta/lang/
!
! If no Preferred-language value, then select notification based on the
! country code in the domain name.EX:PF=French Polynesia; BO=Bolivia
!
   *|*|*|*.fr|*   $I/imta/table/notify_french/
   *|*|*|*.fx|*   $I/imta/table/notify_french/
   *|*|*|*.pf|*   $I/imta/table/notify_french/
   *|*|*|*.tf|*   $I/imta/table/notify_french/
   *|*|*|*.ar|*   $I/imta/table/notify_spanish/
```

```
*|*|*|*.bo|*   $I/imta/table/notify_spanish/
*|*|*|*.cl|*   $I/imta/table/notify_spanish/
*|*|*|*.co|*   $I/imta/table/notify_spanish/
*|*|*|*.cr|*   $I/imta/table/notify_spanish/
*|*|*|*.cu|*   $I/imta/table/notify_spanish/
*|*|*|*.ec|*   $I/imta/table/notify_spanish/
*|*|*|*.es|*   $I/imta/table/notify_spanish/
*|*|*|*.gp|*   $I/imta/table/notify_spanish/
*|*|*|*.gt|*   $I/imta/table/notify_spanish/
*|*|*|*.gy|*   $I/imta/table/notify_spanish/
*|*|*|*.mx|*   $I/imta/table/notify_spanish/
*|*|*|*.ni|*   $I/imta/table/notify_spanish/
*|*|*|*.pa|*   $I/imta/table/notify_spanish/
*|*|*|*.ve|*   $I/imta/table/notify_spanish/
```

注 デフォルトの mappings.locale ファイルはインストールによって組み込まれます。これは、通知言語マッピングを有効にするために mappings ファイルに組み込まれます。通知言語マッピングを無効にするには、インクルード行を以下のようにコメントアウトします。

```
! <IMTA_TABLE: mappings.locale
```

(ファイル内のコメントを読み、必要に応じて変更してください。)

通知メッセージの追加機能

通知メッセージの設定に必要な手順は前の節で説明したとおりです。ここでは、追加機能について説明します。

サイズの大きいメッセージの内容が戻るのをブロックするには

通常、メッセージがバウンスまたはブロックされる場合は、差出人とローカルドメインのポストマスターに通知メッセージでメッセージの内容が戻されます。サイズの大きいメッセージが何通もそのまま戻されると、リソースに負担がかかります。一定のサイズを超えるメッセージの内容が戻るのをブロックするには、MTA オプションファイルで CONTENT_RETURN_BLOCK_LIMIT オプションを設定します。

通知メッセージのヘッダーから US-ASCII 以外の文字を削除するには

インターネットメッセージヘッダーの本来の形式では US-ASCII 以外の文字は使用できません。メッセージヘッダーに使用されている US-ASCII 以外の文字は「MIME ヘッダーエンコーディング」でエンコードされたものです。「MIME ヘッダーエンコーディング」については RFC 2047 に記述されています。したがって、電子メールメッセージの「件名」行は、実際には次のように表されています。

```
Subject: =?big5?Q?=A4j=AB=AC=A8=B1=AD=B1=B0=D3=F5=A5X=AF=B2?=
```

電子メールクライアントは、ヘッダーを表示する際にエンコーディングを削除する必要があります。

%H テンプレートは通知メッセージの本文にヘッダーをコピーするので、通常はエンコードされたヘッダーが表示されます。ただし、Messaging Server では、件名の文字セット (この場合は big5) が return_prefix.txt の Content-Type ヘッダー文字セットパラメータの文字セットと一致する場合は、エンコーディングが削除されます。一致しない場合は、Messaging Server のエンコーディングはそのまま残ります。

通知メッセージの配信間隔を設定するには

キーワード : notices, nonurgentnotices, normalnotices, urgentnotices

配信不能メッセージは、指定したチャネルキューに一定期間保存したあとで差出人に戻されます。また、Messaging Server が配信を試みている期間に、一連のステータスメッセージや警告メッセージを差出人に戻すこともできます。その期間とメッセージの配信間隔は、notices、nonurgentnotices、normalnotices、urgentnotices のキーワードで指定できます。次に例を示します。

```
notices 1 2 3
```

この例では、すべてのメッセージについて、一時的な失敗の通知メッセージが 1 日目と 2 日目に送信されます。メッセージが 3 日たってもまだ配信されない場合は、差出人に返されます。

```
urgentnotices 2,4,6,8
```

この例では、優先度の高いメッセージについて、一時的な失敗の通知が 2、4、6 日目に送信されます。メッセージが 8 日たってもまだ配信されない場合は、差出人に返されます。

MTA オプションファイルの RETURN_UNITS オプションでは、時間 (1) または日 (0) で単位を指定することができます。デフォルトは日 (0) です。

notices キーワードが指定されていない場合は、デフォルトでは、ローカルの 1 チャネル用の notices 設定が使用されます。ローカルチャネル用の設定がない場合は、デフォルトでは、notices 3, 6, 9, 12 が使用されます。

通知メッセージに代替アドレスを含めるには

キーワード : includefinal, suppressfinal, useintermediate

MTA により通知メッセージ (バウンスメッセージ、配信受理メッセージなど) が生成される場合、元の形式の受取人アドレスと、変更された最終的な形式の受取人アドレスの両方が MTA に提示されることがあります。元の形式の方が通知メッセージの受取人 (通知メッセージの場合は元のメッセージの差出人) によって認識される可能性が高いため、MTA は、常に元の形式を通知メッセージに含めます。

includefinal および suppressfinal チャネルキーワードは、MTA が最終的な形式のアドレスを含めるかどうかを制御するためのものです。外部に対して内部のメールボックス名を隠しているサイトでは、最終的な形式のアドレスを含めずに、元の外部用アドレスだけを通知メッセージに含めることをお勧めします。includefinal はデフォルトで、最終的な形式の受取人アドレスを含めます。suppressfinal は、元の形式のアドレスが存在する場合に、通知メッセージに最終形式のアドレスを含めないようにします。

useintermediate キーワードでは、リストのエクスパンド後、ユーザメールボックス名を生成するまでの間に作成された中間形式のアドレスを使用します。この情報を入手できない場合は、最終形式を使用します。

ポストマスターへの通知メッセージを送信、ブロック、指定するには

デフォルトでは、エラーが返された場合や Errors-to: ヘッダー行またはエンベロープ From: アドレスが空白であるために警告をまったく送信できない場合を除いて、失敗と警告の通知メッセージのコピーがポストマスターに送信されます。ポストマスターへの通知メッセージの詳細は、以後の節および 表 6-11 で説明する多数のチャネルキーワードで制御できます。

返送された配信不能メッセージ

キーワード : sendpost, nosendpost, copysendpost, errsendpost

長期間にわたってサービスが支障をきたしている場合や、アドレスが不正確な場合は、チャネルプログラムがメッセージを配信できないことがあります。このような場合、MTA チャネルプログラムは、配信不能の理由を説明する文と一緒にメッセージを差出人に返送します。さらに、配信できないメッセージのコピーをすべてローカルポストマスターに送るように設定することも可能です。これはメッセージ配信障害を監視するのに便利ですが、ポストマスターにとっては大量のメールを処理しなければならないことにもなります (表 6-11 を参照)。

警告メッセージ

キーワード : `warnpost, nowarnpost, copywarnpost, errwarnpost`

メッセージの返送に加えて、MTA では、配信できないメッセージに関する詳細な情報を記載した警告を送信することができます。通常、この警告メッセージは notices チャネルキーワードが指定するタイムアウトに基づいて送られますが、配信試行に失敗したときに送られることもあります。警告には、問題点の説明と配信試行を継続する時間枠が記載されます。また、多くの場合、該当するメッセージのヘッダーと最初の数行も含まれます。

さらに、警告メッセージのコピーをすべてローカルポストマスターに送るように設定することも可能です。これはメッセージ配信障害を監視するのに便利ですが、ポストマスターにとっては大量のメールを処理しなければならないことにもなります。ポストマスターへの警告メッセージの送信を制御するには、`warnpost`、`copywarnpost`、`errwarnpost`、`nowarnpost` の各キーワードを使用します (表 6-11 を参照)。

空白のエンベロープ返信アドレス

キーワード : `returnenvelope`

`returnenvelope` キーワードは 1 つの整数値をとり、これはビットフラグのセットとして解釈されるビット 0 (値 = 1) は、MTA によって生成された返送通知のエンベロープアドレスを空白にするか、ローカルポストマスターのアドレスを入れるかを指定するものです。このビットを設定した場合は、ローカルポストマスターのアドレスを使用することになり、ビットをクリアすると空白アドレスを使用することになります。

注	RFC 1123 では空白アドレスの使用が義務付けられています。ただし、一部のシステムでは空白エンベロープ From: アドレスを適切に処理できないため、このオプションが必要な場合があります。

ビット 1 (値 = 2) は、MTA がすべての空白のエンベロープアドレスをローカルポストマスターのアドレスに置き換えるかどうかを指定します。これは、RFC 821、RFC 822、あるいは RFC 1123 に準拠しないシステムを扱うために使用されます。

ビット 2 (値 = 4) は構文的に不正な返信アドレスを禁止します。

ビット 3 (値 = 8) は `mailfromdnsverify` キーワードと同じです。

ポストマスター返送メッセージの内容

キーワード : postheadonly, postheadbody

チャネルプログラムまたは定期的なメッセージ返送ジョブがメッセージをポストマスターと差出人の両方に返送する場合は、ポストマスターへのコピーには、メッセージ全体を含めることも、ヘッダーだけを含めることもできます。ポストマスターへのコピーをヘッダーに限定することで、ユーザメールのプライバシーのレベルを高めることができます。ただし、ポストマスターやシステム管理者は一般に root システム権限を使用してメッセージの内容を読むことができるため、このキーワードを使用してもメッセージのセキュリティを完全に保証することにはなりません (表 6-11 を参照)。

チャネルポストマスターアドレスの設定

キーワード : aliaspostmaster, returnaddress, noreturnaddress, returnpersonal, noreturnpersonal

デフォルトでは、MTA がバウンスメッセージや通知メッセージを作成する際に使用するポストマスターの返信アドレスは、postmaster@*local-host* です。*local-host* は、ローカルホストの正式な名前 (ローカルチャネル名) で、ポストマスターの個人名は「MTA e-Mail Interconnect」になります。ポストマスターのアドレスは慎重に選択してください。不正な名前を選択すると、すぐにメッセージのループが発生し、大量のエラーメッセージが発生することがあります。

RETURN_ADDRESS オプションと RETURN_PERSONAL オプションを使用すると、MTA システムでポストマスターのアドレスと個人名をデフォルトに設定できます。また、チャネルごとに制御する必要がある場合は、returnaddress および returnpersonal の各チャネルキーワードを使用できます。returnaddress と returnpersonal は、それぞれポストマスターのアドレスと個人名を指定する引数をとります。noreturnaddress と noreturnpersonal がデフォルトであり、デフォルト値が使用されます。デフォルトは、RETURN_ADDRESS オプションと RETURN_PERSONAL オプションで設定します。これらのオプションが設定されていない場合は、通常のデフォルト値が使用されます。

aliaspostmaster キーワードがチャネルに指定されている場合は、正式なチャネル名におけるユーザ名 postmaster (大文字のみ、小文字のみ、またはその両方) 宛てのすべてのメッセージは、postmaster@*local-host* にリダイレクトされます。*local-host* には、正式なローカルホスト名 (ローカルチャネルの名前) が入ります。インターネット標準規格では、メールを受け付ける DNS のすべてのドメインに、メールを受信する有効なポストマスターアカウントを設定する必要があります。このため、各ドメインに個別のポストマスターアカウントを設定するのではなく、ポストマスターの責務を一元化する場合はこのキーワードが便利です。つまり、returnaddress は、MTA がポストマスターからの通知メッセージを生成する際に使用するポストマスターの返信アドレスを制御し、aliaspostmaster は、MTA がポストマスター宛てのメッセージを処理する方法を制御します。

表 6-11　ポストマスターに送信された通知メッセージと差出人キーワード

キーワード	説明
返送メッセージの内容	**通知のアドレスを指定する**
notices	通知の送信とメッセージの返送を行うまでの時間を指定する
nonurgentnotices	優先度が低いメッセージを配信できない場合に通知を送り、そのメッセージを返送するまでの時間を指定する
normalnotices	優先度が標準のメッセージを配信できない場合に通知を送り、そのメッセージを返送するまでの時間を指定する
urgentnotices	優先度が高いメッセージを配信できない場合に通知を送り、そのメッセージを返送するまでの時間を指定する
返送メッセージ	**配信不能な返送メッセージの処理方法**
sendpost	配信不能メッセージのコピーをすべてポストマスターに送信する
copysendpost	配信不能メッセージの差出人アドレスが空白の場合を除き、配信不能通知のコピーをポストマスターに送信する。この場合、ポストマスターは、バウンスメッセージや通知メッセージ以外のすべての配信不能メッセージのコピーを受け取る
errsendpost	通知を差出人に返すことができない場合に、配信不能通知のコピーをポストマスターに送信する。nosendpost が指定されている場合は、配信不能メッセージがポストマスターに送信されることはない
nosendpost	配信不能メッセージのコピーをポストマスターには一切送信しない
警告メッセージ	**警告メッセージの処理方法**
warnpost	警告メッセージのコピーをすべてポストマスターに送信する。デフォルトでは、Warnings-to: ヘッダーやエンベロープ From: アドレスが空白であるために警告をまったく送信できない場合を除いて、警告のコピーがポストマスターに送信される
copywarnpost	配信不能メッセージの差出人アドレスが空白になっている場合を除き、警告メッセージのコピーがポストマスターに送信される
errwarnpost	通知を差出人に返すことができない場合に、警告メッセージのコピーをポストマスターに送信する
nowarnpost	警告メッセージのコピーをポストマスターには一切送信しない
返送メッセージの内容	**ポストマスターにメッセージの内容をすべて送信するか、ヘッダーだけを送信するかを指定する**

表 6-11　ポストマスターに送信された通知メッセージと差出人キーワード (続き)

キーワード	説明
postheadonly	ポストマスターにヘッダーだけを返送する。ポストマスターへのコピーをヘッダーに限定することで、ユーザメールのプライバシーのレベルを高めることができる。ただし、ポストマスターやシステム管理者は root システム権限を使用してメッセージの内容を読むことができるため、このキーワードを選択してもメッセージのセキュリティを完全に保証することにはならない
postheadbody	ヘッダーとメッセージの内容の両方を返送する
返送メッセージの内容	**通知のアドレスを指定する**
includefinal	配信通知の中にアドレスの最終的な形式を含める (受取人アドレス)
returnenvelope	空白のエンベロープ返信アドレスの使用を制御する。returnenvelope キーワードは 1 つの整数値をとり、これはビットフラグのセットとして解釈される ビット 0 (値 = 1) は、MTA によって生成された返送通知のエンベロープアドレスを空白にするか、ローカルポストマスターのアドレスを入れるかを指定する。このビットを設定した場合は、ローカルポストマスターのアドレスを使用することになり、ビットをクリアすると空白アドレスを使用することになる ビット 1 (値 = 2) は、MTA がすべての空白のエンベロープアドレスをローカルポストマスターのアドレスに置き換えるかどうかを指定する。これは、RFC 821、RFC 822、あるいは RFC 1123 に準拠しないシステムを扱うために使用される ビット 2 (値 = 4) は構文的に不正な返信アドレスを禁止する ビット 3 (値 = 8) は mailfromdnsverify キーワードと同じである
suppressfinal	元の形式のアドレスが存在する場合は、通知メッセージに最終形式のアドレスを含めない
useintermediate	リストのエクスパンド後、ユーザメールボックス名の設定前に作成された中間形式のアドレスを使用する。この情報を入手できない場合は、最終形式を使用する
返送メッセージの内容	**通知のアドレスを指定する**
aliaspostmaster	正式なチャネル名でのユーザ名ポストマスター宛てのメッセージは postmaster@local-host にリダイレクトされる。local-host には、ローカルホスト名 (ローカルチャネルの名前) が入る
returnaddress	ローカルポストマスターの返信アドレスを設定する
noreturnaddress	ポストマスターアドレス名に RETURN_ADDRESS オプション値を使用する
returnpersonal	ローカルのポストマスターに対する個人名を設定する
noreturnpersonal	ポストマスター個人名に RETURN_PERSONAL オプション値を使用する

第 7 章

書き換え規則を設定する

この章では、imta.cnf ファイル内で書き換え規則を設定する方法について説明します。この章を読む前に、第 6 章「MTA サービスと設定について」をお読みください。

この章には、以下の節があります。

Messaging Server のアドレス書き換え機能は、アドレスのホストまたはドメイン部分を操作および変更するのに欠かせない重要な機能です。Messaging Server には、エイリアス、アドレス置き換えデータベース、および特殊化されたマッピングテーブルといったほかの機能もあります。ただし、アドレス操作を実行する可能性がある場合には、常に書き換え規則を使用するようにしてください。それにより、最高のパフォーマンスを得ることができます。

注 imta.cnf ファイル内の書き換え規則を変更する場合は、imsimta start コマンドを使って起動するときに設定データを 1 回だけ読み込むようなプログラムまたはチャネルを再起動する必要があります (例 : SMTP サーバ)。コンパイルされた設定を使用する場合は、設定を再コンパイルしたあとにプログラムを再起動する必要があります。

設定情報のコンパイルおよびプログラムの再起動については、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

書き換え規則の構造

書き換え規則は MTA 設定ファイル imta.cnf の前半にあります。設定ファイルに、各規則が 1 行ごとに記述されています。規則間にコメントを記述することは可能ですが、空白行を挿入することはできません。書き換え規則の最後には空白行が挿入され、その後ろにチャネルの定義が続きます。図 7-1 に、設定ファイル内の書き換え規則を示します。

図 7-1 設定ファイルの例 - 書き換え規則

```
! test.cnf - 設定ファイルの例。
!
! これは、単に設定ファイルの例です。It serves
! no useful purpose and should not be used in a real system.
!
a.com   $U@a-host
b.org   $U@b-host
c.edu   $U%c@b-daemon
d.com   $U%d@a-daemon

! 以下、チャネルの定義が続きます。
```

書き換え規則は、2 つの部分から構成されています。最初にパターン、その後ろに同等の文字列またはテンプレートを指定します。これらの 2 つの部分は空白文字を挿入して区切る必要があります。ただし、パターンやテンプレート自体に空白文字を使用することはできません。書き換え規則の構造は、次のとおりです。

```
pattern  template
```

pattern

検索の対象となるドメイン名内の文字列です。図 7-1 では、パターンは `a.com`、`b.org`、`c.edu`、および `d.com` です。

パターンがアドレスのドメイン部分に一致すると、その書き換え規則がアドレスに適用されます。パターンの後ろには空白文字を挿入して、テンプレート部分を区別できるようにします。パターンのシンタックスについては、166 ページの「書き換え規則のパターンとタグ」を参照してください。

template

以下のいずれかの形式を使って指定します。

UserTemplate`%`*DomainTemplate*`@`*ChannelTag* [*controls*]

UserTemplate`@`*ChannelTag* [*controls*]

UserTemplate`%`*DomainTemplate* [*controls*]

UserTemplate`@`*DomainTemplate*`@`*ChannelTag* [*controls*]

UserTemplate`@`*DomainTemplate*`@`*SourceRoute*`@`*ChannelTag* [*controls*]

UserTemplate	アドレスのユーザ部分をどのように書き換えるかを指定します。元のアドレスの一部またはデータベース検索の結果を表すために置換シーケンスを使用することもできます。置換シーケンスは、アドレスを書き換える際に、それが表す本来の文字列に置き換えられます。図 7-1 では、$U という置換シーケンスが使用されています。詳細は、178 ページの「テンプレートの置換シーケンスと書き換え規則コントロールシーケンス」を参照してください。
DomainTemplate	アドレスのドメイン部分をどのように書き換えるかを指定します。*UserTemplate* と同様に、*DomainTemplate* にも置換シーケンスを含めることができます。
ChannelTag	このメッセージが送信されるチャネルを表します。チャネル定義にはすべて、チャネルタグとチャネル名が必要です。一般に、チャネルタグは書き換え規則とそのチャネル定義に記述されます。
controls	コントロールを使って、規則の適用度を制限できます。コントロールシーケンスの中には、規則の前に置くものと、規則の後ろに置くものとがあります。コントロールについては、178 ページの「テンプレートの置換シーケンスと書き換え規則コントロールシーケンス」を参照してください。

テンプレートのシンタックスについては、170 ページの「書き換え規則のテンプレート」を参照してください。

書き換え規則のパターンとタグ

この節には、以下の項目があります。

一般に、書き換え規則のパターンは、特定のホスト名 (そのホスト名だけに一致) またはサブドメインパターン (サブドメイン全体における任意のホスト / ドメインに一致) のいずれかで構成されます。

たとえば、次の書き換え規則のパターンには、特定のホスト名が含まれています。このパターンは、この指定したホスト名だけに一致します。

```
host.siroe.com
```

次の書き換え規則のパターンには、サブドメインパターンが含まれています。このパターンは、サブドメイン全体における任意のホストまたはドメインに一致します。

```
.siroe.com
```

ただし、このパターンは `siroe.com` というホスト名には一致しません。`siroe.com` も対象に含める場合は、`siroe.com` という別のパターンが必要です。

MTA は、特定のホスト名に一致するものからホスト / ドメイン名を書き換えていき、より不特定のパターンへと処理を進めます。つまり、特定のパターンは、不特定のパターンよりも優先して使用されることになります。たとえば、設定ファイルに、以下の書き換え規則パターンが記述されているとします。

```
hosta.subnet.siroe.com
.subnet.siroe.com
.siroe.com
```

この場合、まず `jdoe@hosta.subnet.siroe.com` というアドレスが `hosta.subnet.siroe.com` という書き換え規則パターンに一致します。その後、`jdoe@hostb.subnet.siroe.com` というアドレスが `.subnet.siroe.com` という書き換え規則パターンに一致し、次に `jdoe@hostc.siroe.com` というアドレスが `.siroe.com` という書き換え規則パターンに一致します。

特に、サブドメインの書き換え規則パターンを含む書き換え規則は、インターネットのサイトでよく使用されます。一般に、そのようなサイトでは、内部のホストやサブネット用に多数の書き換え規則が用意され、トップレベルのインターネットドメインに対する書き換え規則が `internet.rules` (*server-instance*`/imta/config/internet.rules`) ファイル内の設定に含められます。

インターネット宛先 (より特定の書き換え規則を通じて処理されたインターネットホスト宛先を除く) へのメッセージが正しく書き換えられ、送信 TCP/IP チャネルに送られるようにするには、`imta.cnf` ファイルに以下の内容を含めます。

- トップレベルのインターネットドメインに一致するパターンを含んだ書き換え規則
- そのようなパターンに一致するアドレスを書き換える、送信 TCP/IP チャネルに対するテンプレート

```
!    Ascension Island
.AC                                             $U%$H$D@TCP-DAEMON
. [ 簡潔にするため
.    テキストを
.              省略 ]
!    Zimbabwe
.ZW                                             $U%$H$D@TCP-DAEMON
```

同様に、IP ドメインリテラルの場合も階層に基づいて照合が行われます。ただし、左から右ではなく、右から左へ照合が行われます。たとえば、次のパターンは `[1.2.3.4]` という IP リテラルにのみ一致します。

```
[1.2.3.4]
```

次のパターンは `1.2.3.0` サブネット内の任意の IP リテラルに一致します。

```
[1.2.3.]
```

ホストあるいはサブドメイン名を使った一般的な書き換え規則パターンのほかに、特殊なパターンを使用することもできます。これらの特殊なパターンについては、表 7-1 およびそのあとの説明を参照してください。

表 7-1 書き換え規則の特殊パターン

パターン	説明 / 使用方法
$*	任意のアドレスに一致。この規則は、ファイル内のどこに配置されていても最初に試行される (指定されている場合)

表 7-1　書き換え規則の特殊パターン (続き)

パターン	説明 / 使用方法
$%	パーセントハック規則。A%B 形式の任意のホスト / ドメイン仕様に一致する
$!	bang-style 規則。B!A 形式の任意のホスト / ドメイン仕様に一致する
[ ]	IP リテラル全一致規則。任意の IP ドメインリテラルに一致する
.	任意のホスト / ドメイン仕様に一致する。たとえば、`joe@[129.165.12.11]`

これらの特殊パターンに加え、Messaging Server には、書き換え規則のパターン内で使われるタグの概念もあります。これらのタグは、アドレスが複数回にわたって書き換えられる場合に使用されます。この区別は、直前に行われた書き換えに基づき、どの書き換え規則がアドレスに一致するかを制御することによって行います。詳細は、169 ページの「タグ付き書き換え規則セット」を参照してください。

パーセントハックに一致する規則

MTA が `A%B` 形式のアドレスを書き換えようとして失敗した場合は、そのアドレスが `A%B@localhost` 形式のアドレスとして扱われる前に、もう 1 つの規則が適用されます (このようなアドレス形式については、170 ページの「書き換え規則のテンプレート」を参照)。このもう 1 つの規則がパーセントハック規則です。パターンは `$%` で、変わることはありません。この規則は、パーセント記号を含むローカル部分がほかのすべての方法 (あとで説明する全一致規則を含む) で書き換えに失敗した場合にのみアクティブになります。

パーセントハック規則は、パーセントハックアドレスに何らかの特別な意味を持たせる場合に便利です。

bang-style (UUCP) アドレスに一致する規則

MTA が `B!A` 形式のアドレスを書き換えようとして失敗した場合は、そのアドレスが `B!A@localhost` 形式のアドレスとして扱われる前に、もう 1 つの規則が適用されます。この規則が *bang-style* 規則です。パターンは `$!` で、変わることはありません。この規則は、感嘆符を含むローカル部分がほかのすべての方法 (あとで説明するデフォルトの規則を含む) で書き換えに失敗した場合にのみアクティブになります。

bang-style 規則を使用すると、UUCP スタイルのアドレスが UUCP システムおよびルーティングに関する総合的な情報を備えたシステムを経由するように書き換えることができます。

任意のアドレスに一致する規則

特殊パターン「.」(ドット文字) は、ほかに一致する規則がない場合に、任意のホスト / ドメイン仕様に一致します。ただし、そのホスト / ドメイン仕様は、チャネルテーブル内で見つからないものに限ります。つまり、「.」規則は、アドレスの書き換えに失敗する前の最後の手段として使用されます。

注	置換シーケンスについては、全一致規則が一致し、そのテンプレートが展開される場合、$H はホストのフルネームに展開し、$D はドット文字 1 個「.」に展開します。したがって、全一致規則のテンプレートでは、$D の使用が制限されます。

タグ付き書き換え規則セット

書き換えプロセスを実行するにあたり、別の規則セットを追加するとうまくいく場合があります。別の規則セットを追加するには、書き換え規則タグを使用します。現在のタグは、設定ファイルまたはドメインデータベースでパターンが検索される前に、各パターンの前に付けられます。タグは、書き換え規則テンプレート内の $T という置換文字列を使って一致する書き換え規則により変更することができます (後述の説明を参照)。

タグは、1 つのアドレスから抽出されたすべてのホストに対し、連続して適用されます。そのため、タグを使用した場合は、別の規則を指定する際にそれが正しいタグ値から始まるように注意してください。一般に、タグは特殊な目的でしか使用しないため、このことが問題になることはほとんどありません。アドレスの書き換えが完了すると、タグはデフォルトのタグ (空白文字列) にリセットされます。

規則により、すべてのタグ値には、その最後に縦棒 (|) が付けられます。この文字は通常のアドレスには使用されないため、パターンの残りの部分とタグとを区別することができます。

書き換え規則のテンプレート

以下の節では、書き換え規則のテンプレート形式について説明します。表 7-2 にテンプレート形式を示します。

表 7-2　書き換え規則のテンプレート形式

テンプレート	ページ	使用目的
A%B	170	A は新しいユーザ / メールボックス名になり、B は新しいホスト / ドメイン仕様になる。繰り返して書き換える
A@B	170	A%B@B として扱われる
A%B@C	170	A は新しいユーザ / メールボックス名になり、B は新しいホスト / ドメイン仕様になる。ホスト C に関連したチャネルに送る
A@B@C	171	A@B@C@C として扱われる
A@B@C@D	171	A は新しいユーザ / メールボックス名になり、B は新しいホスト / ドメイン名になる。C をソースルートとして挿入し、ホスト D に関連したチャネルに送る

よく使われる書き換えテンプレート : A%B@C または A@B

以下に示すテンプレート形式は、もっともよく使われるものです。規則は、アドレスのユーザ部分とドメイン部分に適用されます。その後、新しいアドレスがメッセージを特定のチャネル (*ChannelTag* で指定されたチャネル) へ送るために使用されます。

UserTemplate%*DomainTemplate*@*ChannelTag* [*controls*]

以下に示すテンプレート形式は、上記のテンプレートと実質的に同じものです。ただし、この形式は、*DomainTemplate* と *ChannelTag* が同じ場合にしか使用できません。

UserTemplate@*ChannelTag* [*controls*]

繰り返し書き換えテンプレート : A%B

以下に示すテンプレート形式は、繰り返して適用する必要がある規則に使用されます。規則適用後は、新しいアドレスで書き換えプロセス全体を繰り返します (ほかのテンプレート形式では、規則を適用すると書き換えプロセスが終了します)。

UserTemplate%*DomainTemplate* [*controls*]

たとえば、以下に示す規則を使うと、.removable というドメイン名で終わるすべてのアドレスから .removable が削除されます。

```
.removable        $U%$H
```

繰り返し規則を使用する場合には、「規則ループ」が生じないよう特別な注意が必要です。そのため、特に必要がないかぎり、繰り返し書き換え規則の使用を控えるようお勧めします。繰り返し規則を使用する際には、`imsimta test -rewrite` コマンドを使って規則をテストするとよいでしょう。`test -rewrite` コマンドについては、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

指定ルート書き換えテンプレート：A@B@C@D または A@B@C

以下に示すテンプレート形式は、一般によく使われる形式 *UserTemplate*`%`*DomainTemplate*`@`*ChannelTag* (最初の区切り文字が異なります) と同じように機能します。ただし、*ChannelTag* はソースルートとしてアドレスに挿入されています。メッセージは *ChannelTag* に送られます。

UserTemplate`@`*DomainTemplate*`@`*Source-Route*
`@`*ChannelTag* [*controls*]

書き換えられたアドレスは `@`*route*`:`*user*`@`*domain* になります。また、次のテンプレートも使用できます。

UserTemplate`@`*DomainTemplate*`@`*ChannelTag* [*controls*]

たとえば、以下に示す規則を使うと、`jdoe@com1` というアドレスが `@siroe.com:jdoe@com1` というソースルートアドレスに書き換えられます。チャネルタグは `siroe.com` になります。

```
com1 $U@com1@siroe.com
```

書き換え規則テンプレートにおける大文字と小文字の区別

書き換え規則内のパターンとは異なり、テンプレートでは大文字と小文字が区別されます。この機能は、大文字と小文字を区別するメールシステムへのインタフェースを提供するような書き換え規則を使用する場合に必要となります。アドレスから抽出された部分の代わりに使われる $U や $D などの置換シーケンスでも、大文字と小文字が区別され、元のアドレスと同じ状態が維持されます。

UNIX システムでメールボックスを小文字にする場合など、代替部分に特定の大文字 / 小文字が使われるようにするには、テンプレートに特殊な置換シーケンスを使用します。たとえば、$¥ は後ろに続く代替部分を小文字にし、$^ は後ろに続く代替部分を大文字にします。また、$_ は元と同じ状態を保ちます。

たとえば、以下の規則を使うと、unix.siroe.com のアドレスに対するメールボックスを小文字にすることができます。

```
unix.siroe.com     $¥$U$_%unix.siroe.com
```

MTA がアドレスに書き換え規則を適用する方法

以下に、MTA が指定アドレスに書き換え規則を適用する手順について説明します。

1. アドレスから最初のホスト仕様またはドメイン仕様を抽出します。

   アドレスには、次のように 1 つ以上のホスト名またはドメイン名が指定されている場合があります。

   ```
   jdoe%hostname@siroe.com.
   ```

2. 最初のホスト名またはドメイン名を識別したあと、そのホスト名またはドメイン名に一致するパターンが含まれている書き換え規則を検索します。
3. 一致する書き換え規則が見つかると、MTA により、その規則のテンプレート部分に従ってアドレスが書き換えられます。
4. 最後に、チャンネルタグと各チャネルに関連するホスト名が比較されます。

   一致するものが見つかると、MTA は関連するチャネルへのメッセージをキューに入れます。一致するものが見つからない場合、書き換えプロセスは失敗に終わります。一致するチャネルがローカルチャネルであれば、エイリアスデータベースとエイリアスファイルを検索して、アドレスの書き換えが追加されることもあります。

これらの動作の詳細については、後続の節を参照してください。

注 既存のどのチャネルにも属さないチャネルタグを使用すると、この規則に一致するアドレスを持つメッセージが戻ってきます。すなわち、一致するメッセージが配信不能となります。

動作 1 最初のホスト / ドメイン仕様を抽出する

アドレスの書き換えプロセスは、アドレスの最初のホストまたはドメイン仕様を抽出することから始まります (以下の説明をより理解するために、RFC 822 アドレス規則について把握しておくことをお勧めします)。アドレス内のホスト / ドメイン仕様が検索される順序は、以下のとおりです。

1. ソースルートのホスト (左から右へ読み取り)
2. 単価記号 (@) の右側にあるホスト
3. 最後のパーセント記号 (%) の右側にあるホスト
4. 最初の感嘆符 (!) の左側にあるホスト

最後の 2 項目の順序は、アドレスの書き換えを行っているチャネルで bangoverpercent キーワードが有効になっているかどうかによって入れ替わります。すなわち、メッセージをキューに入れようとしているチャネルが bangoverpercent チャネルキーワードでマークされているかどうかによって順序が異なります。

表 7-3 に、アドレスと最初に抽出されるホスト名の例を示します。

表 7-3 アドレスと抽出されるホスト名

アドレス	最初のホスト ドメイン仕様	コメント
user@a	a	「短形式」のドメイン名。
user@a.b.c	a.b.c	「完全指定」のドメイン名 (FQDN)。
user@[0.1.2.3]	[0.1.2.3]	「ドメインリテラル」
@a:user@b.c.d	a	短形式のドメイン名を伴った「ルート」と呼ばれるソースルートアドレス
@a.b.c:user@d.e.f	a.b.c	ソースルートアドレス : ルート部分は完全形
@[0.1.2.3]:user@d.e.f	[0.1.2.3]	ソースルートアドレス : ルート部分はドメインリテラル
@a,@b,@c:user@d.e.f	a	a → b → c ルーティングを伴ったソースルートアドレス
@a,@[0.1.2.3]:user@b	a	ルート部分にドメインリテラルを伴ったソースルートアドレス
user%A@B	B	この非標準形のルーティングは「パーセントハック」と呼ばれる

表 7-3　アドレスと抽出されるホスト名 (続き)

アドレス	最初のホストドメイン仕様	コメント
user%A	A	
user%A%B	B	
user%%A%B	B	
A!user	A	「bang-style」のアドレス。UUCP によく使用される
A!user@B	B	
A!user%B@C	C	
A!user%B	B	nobangoverpercent キーワードが有効な場合 (デフォルト)
A!user%B	A	bangoverpercent キーワードが有効な場合

RFC 822 には、アドレスにおける感嘆符 (!) およびパーセント記号 (%) の解釈が含まれていません。慣例上、パーセント記号は単価記号 (@) と同じように解釈されます (単価記号 @ がない場合)。この規則は Messaging Server MTA で採用されています。

パーセント記号をローカルユーザ名の一部として扱うために、繰り返しパーセント記号の解釈が使用されます。これは、外部メールシステムのアドレスを処理するような場合に便利です。感嘆符の解釈は、RFC 976 の「bang-style」アドレス規則に従います。この解釈により、Messaging Server MTA で UUCP アドレスを使用することが可能になります。

これらの解釈の順序については、RFC 822 または RFC 976 のどちらにも指定されていません。そのため、bangoverpercent および nobangoverpercent キーワードを使って、書き換えを行うチャネルによって解釈が適用される順序を制御します。デフォルト設定がより「標準的」ですが、状況によっては代わりの設定を使った方が便利な場合もあります。

注　アドレス内に感嘆符 (!) やパーセント記号 (%) を使用することはお勧めしません。

動作 2 書き換え規則を検索する

アドレスから最初のホストまたはドメイン仕様が抽出されると、MTA は書き換え規則を調べてその仕様の処理方法を明らかにします。ホスト / ドメイン仕様は、各規則のパターン部分 (各規則の左側) と比較されます。その場合、大文字と小文字の区別はありません。大文字と小文字の区別がないことは、RFC 822 で定められています。MTA では、特に大文字と小文字を区別しませんが、可能な限り元の状態が維持されます。

ホスト / ドメイン仕様がどのパターンにも一致しない場合は、ホスト / ドメイン仕様の最初の部分 (最初のドット文字より前の部分、通常はホスト名) がアスタリスク (*) に置き換えられ、その新しいホスト / ドメイン仕様が検索されます。ただし、その場合、検索対象となるのは設定ファイル内の書き込み規則だけで、ドメインデータベースは調べられません。

この試行が失敗に終わると、最初の部分が削除され、プロセスが繰り返されます。この試行も失敗に終わると、次の部分 (通常はサブドメイン) が削除され、再び検索が行われます。最初にアスタリスクを含めて検索が行われ、その後アスタリスクを含めずに検索が行われます。アスタリスクを含んだ検索が行われるのは設定ファイル内の書き換え規則テーブルだけで、ドメインデータベースは調べられません。このプロセスは、一致する規則が見つかるか、またはホスト / ドメイン仕様全体がなくなるまで続けられます。このようなプロセスを使用することにより、より目的に近いドメインを最初に見つけ出し、次により特化したドメインを検索することができます。

このマッチングプロセスのアルゴリズムは、以下のとおりです。

- ホスト / ドメイン仕様が比較文字 `spec_1` と `spec_2` の最初の値として使用される。たとえば、`spec_1 = spec_2 = a.b.c`
- 比較文字列 `spec_1` は、一致するものが見つかるまで、まず設定ファイル内にある各書き換え規則のパターン部分が調べられ、次にドメインデータベース内が調べられる。このマッチングプロセスは、一致するものが見つかった時点で終了する
- 一致するものが見つからなかった場合は、`spec_2` のもっとも左側の部分 (アスタリスク以外) がアスタリスクに変換される。たとえば、`spec_2` が `a.b.c` の場合に一致するものが見つからなければ `*.b.c` に、`spec_2` が `*.b.c` の場合に一致するものが見つからなければ `*.*.c` に変換される。このマッチングプロセスは、一致するものが見つかった時点で終了する
- 一致するものが見つからなければ、比較文字列 `spec_1` の最初の部分はドット文字も含めて削除される。`.c` や `c` のように、`spec_1` に 1 つの部分しかない場合は、文字列は 1 文字のドット文字「.」で置き換えられる。削除後の `spec_1` 文字列の長さがゼロでない場合は、動作 1 に戻る。削除後の新しい文字列の長さがゼロの場合 (たとえば、置換前の文字列が「.」だった場合) は、検索プロセスが失敗に終わり、マッチング手順が終了する

たとえば、アドレス dan@sc.cs.siroe.edu を書き換えるとします。これにより MTA は、指定した順に以下のパターンを検索します。

```
sc.cs.siroe.edu
*.cs.siroe.edu
.cs.siroe.edu
*.*.siroe.edu
.siroe.edu
*.*.*.edu
.edu
*.*.*.*
.
```

動作 3 テンプレートに従ってアドレスを書き換える

ホスト / ドメイン仕様が書き換え規則に一致すると、そのホスト / ドメイン仕様は規則のテンプレート部分を使って書き換えられます。テンプレートには、次の 3 つの仕様があります。

1. アドレスの新しいユーザ名
2. アドレスの新しいホスト / ドメイン仕様
3. メッセージの送信先である既存の MTA チャネルが指定されたチャネルタグ

動作 4 書き換えプロセスを終了する

ホスト / ドメイン仕様が書き換えられると、次の 2 つの動作のうちどちらかが行われます。

- チャネルタグがローカルチャネルまたは routelocal チャネルキーワードでマークされているチャネルのどちらにも関連付けられていない場合、またはアドレス内にほかのホスト / ドメイン仕様がない場合は、書き換え後の指定が抽出された元の指定に置き換えられ、書き換えプロセスが終了する
- チャネルタグがローカルチャネルまたは routelocal でマークされたチャネルに一致し、かつアドレス内にほかのホスト / ドメイン仕様がある場合は、書き換え後のアドレスが破棄され、アドレスから元 (最初) のホスト / ドメイン仕様が削除される。次にそのアドレスから新しいホスト / ドメイン仕様が抽出され、プロセス全体が繰り返される。書き換えプロセスは、すべてのホスト / ドメイン仕様がなくなるか、あるいは非ローカルチャネルまたは非ルートローカルチャネルを

介したルートが見つかるまで続けられる。MTA がソースルーティングをサポートできるのは、この反復メカニズムがあるためで、実際、ローカルシステムまたはルートローカルシステムを介した不必要なルートは、このプロセスによってアドレスから削除される

書き換え規則に一致しなかった場合

ホスト / ドメイン仕様がどの書き換え規則にも一致せず、デフォルトの規則もない場合には、「そのまま」の仕様が使われます。たとえば、元の仕様が新しい仕様およびルーティングシステムになります。アドレスに無意味なホスト / ドメイン仕様が含まれている場合、その仕様は、ルーティングシステムが任意のチャネルに関連付けられたどのシステム名にも一致しないときに検出され、メッセージが戻されます。

書き換え後のシンタックスチェック

書き換え規則が適用されたあとのアドレスに対し、シンタックスチェックは行われません。これは意図的なものです。シンタックスチェックを行わないようにすることで、書き換え規則を使ってアドレスを RFC 822 に準拠しない形式に変換することができます。ただし、設定ファイル内に間違いがあると、MTA から送出されるメッセージに不正なアドレスが含まれる可能性もあります。

ドメインリテラルの処理

ドメインリテラルは、特に書き換えプロセス中に処理されます。アドレスのドメイン部分にあるドメインリテラルが書き換え規則のパターンに一致しない場合、そのリテラルは、角括弧で囲まれ、ドット文字で区切られた文字列の集まりとして解釈されます。そして、もっとも右側にある文字列が削除され、検索が繰り返されます。それでも一致するものが見つからない場合は、角括弧だけが残るまで次々に文字列が削除されていきます。空白の角括弧を使った検索も失敗に終わった場合は、ドメインリテラル全体が削除され、ドメインアドレスの次の部分について書き換え処理が実行されます (次の部分が存在する場合)。ドメインリテラルの内部処理では、アスタリスクが使用されません。ドメインリテラル全体がアスタリスクに置き換えられた場合は、アスタリスクの数とドメインリテラル内の要素の数とが一致します。

通常のホスト / ドメイン仕様の場合と同じように、ドメインリテラルの場合も指定した内容にもっとも近いものから順に検索が行われます。そして、パターンに一致した最初の規則を使って、ホスト / ドメイン仕様の書き換えが行われます。規則リスト内に同じパターンが 2 つある場合は、先に記述されている方の規則が適用されます。

たとえば、dan@[128.6.3.40] というアドレスを書き換えるとします。この場合、まず [128.6.3.40] の検索が行われ、その後、[128.6.3.]、[128.6.]、[128.]、[]、[*.*.*.*]、そして最後に全一致規則 「.」 という順に検索が実行されます。

テンプレートの置換シーケンスと書き換え規則コントロールシーケンス

置換シーケンスは、書き換え後のアドレスに文字列を挿入することにより、ユーザ名またはアドレスを書き換えるために使用します。挿入される文字列は、使用している置換シーケンスによって決まります。この節には、以下の項目があります。

たとえば、以下のテンプレートでは、$U が置換シーケンスです。この置換シーケンスを使用することにより、書き換えられるアドレスのユーザ名部分がテンプレートの出力に挿入されます。したがって、このテンプレートで jdoe@mailhost.siroe.com を書き換えると、その出力は jdoe@siroe.com になります。つまり $U が元のアドレスのユーザ名部分 jdoe に置き換えられます。

```
$U@siroe.com
```

コントロールシーケンスは、書き換え規則を適用するうえで、さらに条件を加えるためのものです。すなわち、書き換え規則のパターン部分がホスト / ドメイン仕様に一致しなければならないだけでなく、書き換えるアドレスのほかの要素がコントロールシーケンスの条件を満たしていなければなりません。たとえば、$E コントロールシーケンスは、書き換えるアドレスがエンベロープアドレスでなければならないことを意味します。また、$F コントロールシーケンスは、そのアドレスが前方を探すアドレスでなければならないことを意味します。以下の書き換え規則は、user@siroe.com 形式の (書き換え) エンベロープの To: アドレスにのみ適用されます。

```
siroe.com $U@mail.siroe.com$E$F
```

ホスト / ドメイン仕様は書き換え規則のパターン部分に一致するが、テンプレート内のコントロールシーケンスに指定されている条件がすべて満たされない場合、その書き換え規則は失敗に終わり、ほかの適用可能な規則が検索されます。

表 7-4 に、テンプレートの置換シーケンスとコントロールシーケンスを示します。

表 7-4 テンプレート置換シーケンスとコントロールシーケンス

置換シーケンス	置き換える内容
$D	一致したドメイン仕様の部分
$H	ホスト / ドメイン仕様の不一致部分。パターン内のドットの左側
$L	ドメインリテラルの不一致部分。パターンリテラル内のドットの右側
$U	元のアドレスのユーザ名
$0U	元のアドレスのローカル部分 (ユーザ名)。サブアドレスは含まない
$1U	元のアドレスのローカル部分 (ユーザ名) にあるサブアドレス (存在する場合のみ)
$$	ドル記号 ($) を挿入
$%	パーセント記号 (%) を挿入
$@	単価記号 (@) を挿入
$¥	小文字にする
$^	大文字にする
$_	元の大文字 / 小文字を使用
$W	ランダムに選択される固有文字列の置換
$] . . . [	LDAP クエリの URL 検索

表 7-4　テンプレート置換シーケンスとコントロールシーケンス (続き)

置換シーケンス	置き換える内容
$ （テキスト）	一般データベースの置換。検索に失敗すると、規則も失敗する
${ . . . }	与えられた文字列に指定マッピングを適用
$[...]	カスタマ提供のルーチンを起動する。結果の置換
$&*n*	一致しなかった (またはワイルドカードを使った) ホストの *n* 番目の部分。0 から始まり、左から右へ数える
$!*n*	一致しなかった (またはワイルドカードを使った) ホストの *n* 番目の部分。0 から始まり、右から左へ数える
$***n*	一致したパターンの *n* 番目の部分。0 から始まり、左から右へ数える
$#*n*	一致したパターンの *n* 番目の部分。0 から始まり、右から左へ数える
$*n*D	一致したドメイン仕様の一部。左側の 0 から n 番目までの部分が残される
$*n*H	一致しなかったホスト / ドメイン仕様の一部。左側の 0 から n 番目までの部分が残される
コントロールシーケンス	**書き換え規則における効果**
$1M	チャネルが内部再処理チャネルの場合にのみ適用される
$1N	チャネルが内部再処理チャネルでない場合にのみ適用される
$1~	保留状態のチャネルの照合チェックを実行する。チェックに失敗した場合、現在の書き換え規則テンプレートの処理は正常に終了する
$A	ホストが単価記号 @ の右側にある場合に適用される
$B	ヘッダー / 本文アドレスにのみ適用される
$C *channel*	*channel* へ送る場合は適用されない
$E	エンベロープアドレスにのみ適用される
$F	前方を探すアドレス (例 : To:) にのみ適用される
$M *channel*	*channel* がアドレスを書き換える場合にのみ適用される
$N *channel*	*channel* がアドレスを書き換える場合は適用されない
$P	ホストがパーセント記号の右側にある場合に適用される
$Q *channel*	*channel* へ送る場合に適用される
$R	後方を探すアドレス (例 : From:) にのみ適用される

表 7-4　テンプレート置換シーケンスとコントロールシーケンス (続き)

置換シーケンス	置き換える内容
$S	ホストがソースルートからのものである場合に適用される
$T*newtag*	書き換え規則タグを newtag に設定する
$V*host*	ホスト名が LDAP ディレクトリ内に (DC ツリー内または仮想ドメインとして) 定義されていない場合は、適用されない。LDAP 検索がタイムアウトになると、書き換え規則パターンのホスト名の直後の文字以降の残りの部分は、MTA オプション文字列 DOMAIN_FAILURE と置き換えられる
$X	ホストが感嘆符の左側にある場合に適用される
$Z*host*	ホスト名が LDAP ディレクトリ内に (DC ツリー内または仮想ドメインとして) 定義されている場合は、適用されない。LDAP 検索がタイムアウトになると、書き換え規則パターンのホスト名の直後の文字以降の残りの部分は、MTA オプション文字列 DOMAIN_FAILURE と置き換えられる
$?*errmsg*	書き換えに失敗した場合は、デフォルトのエラーメッセージの代わりに *errmsg* を返す。エラーメッセージは US ASCII でなければならない
$*number*?*errmsg*	書き換えに失敗した場合は、デフォルトのエラーメッセージではなく *errmsg* を返し、SMTP 拡張エラーコードを *a.b.c* に設定する • *a* は *number* / 1000000 (1 桁目) • *b* は (*number* /1000) 残り 1000 (2 桁目から 4 桁目の値) • *c* は *number* の残り 1000 (最後の 3 桁の値) 以下の例では、エラーコードを 3.45.89 に設定する `$3045089?the snark is a boojum`

ユーザ名とサブアドレスの置換 : $U、$0U、$1U

テンプレート内にある $U はすべて、元のアドレスから抽出されたユーザ名 (RFC 822「ローカル部」) に置き換えられます。この場合、a."b" 形式のアドレスは "a.b" に置き換えられます。RFC 2822 における古いシンタックスの使用は推奨しません。今後、より新しいシンタックスの使用が中心になると考えられます。

テンプレート内にある $0U はすべて、元のアドレスのユーザ名に置き換えられます。ただし、サブアドレスおよびサブアドレスを示す文字 (+) は含まれません。テンプレート内にある $1U はすべて、元のアドレスのサブアドレスおよびサブアドレスを示す文字 (+) に置き換えられます (それらが存在する場合のみ)。$0U と $1U はユーザ名を互いに補う関係にあります。すなわち、$0U$1U と $U とは同じものです。

ホスト / ドメインと IP リテラルの置換 : $D、$H、$nD、$nH、$L

$H はすべて、規則に一致しなかったホスト / ドメイン仕様の部分に置き換えられます。また、$D はすべて、規則に一致したホスト / ドメイン仕様の部分に置き換えられます。$nH および $nD は、通常の $H または $D の部分から左側の 0 から n 番目までの部分を残す変形体です。すなわち、$nH または $nD を使用すると、通常 $H または $D で得られる部分から左端の 1 から n 番目までの部分が省略されます。$0H と $H、および $0D と $D はそれぞれ同じものです。

たとえば、`jdoe@host.siroe.com` というアドレスが以下の規則に一致したとします。

```
host.siroe.com     $U%$1D@TCP-DAEMON
```

この規則が適用されると、出力チャネルに `TCP-DAEMON` を使用する `jdoe@siroe.com` というアドレスが得られます。$D は一致したドメイン全体 (つまり `host.siroe.com`) に置き換えられる置換シーケンスですが、この例で使われている $1D は一致したドメインから部分 1 (`siroe`) を省略した部分 (`siroe.com`) に置き換えられます。

$L は、書き換え規則に一致しなかったドメインリテラルの部分に置き換えられます。

リテラル文字の置換 : $$、$%、$@

通常、$、%、および @ 文字は書き換え規則テンプレートのメタ文字です。これらの文字を挿入する場合は、その文字の前にドル記号 $ を付けます。すなわち、$$ は単一のドル記号 $ に、$% は単一のパーセント記号 % (この場合、パーセントはテンプレートのフィールド区切り文字として解釈されません) に、$@ は単一の単価記号 @ (同様に、フィールド区切り文字として解釈されません) に展開されます。

LDAP クエリ URL の置換 : $]...[

$]ldap-url[形式の置換シーケンスは LDAP クエリ URL として解釈され、LDAP クエリの結果に置き換えられます。標準の LDAP URL では、ホストとポートが省略されます。その代わり、ホストとポートは、msg.conf ファイル (local.ldaphost および local.ldapport 属性) で指定されています。

すなわち、LDAP URL は、以下のように指定されます。ここで、角括弧 [ ] は URL のオプション部分を表しています。

```
ldap:///dn[?attributes[?scope?filter]]
```

dn は検索ベースを指定する名前で、この部分は必須です。URL のオプションである属性 (attributes)、範囲 (scope)、フィルタ (filter) は、戻される情報を指定するためのものです。書き換え規則の場合、戻される情報を指定するための属性として望ましいのは mailRoutingSystem 属性 (または同様の属性) です。範囲は、任意のベース (デフォルト)、one、または sub にすることができます。また、フィルタには、mailDomain の値が書き換えられるドメインに一致するオブジェクトを戻すような要求を指定するとよいでしょう。

LDAP ディレクトリスキーマに mailRoutingSystem および mailDomain 属性が含まれている場合、指定アドレスの送り先となるシステムを決定する書き換え規則は、たとえば次のようになります。この例で、作成された LDAP クエリ内の LDAP URL 置換シーケンス $D は、現在のドメイン名に置き換えられます。

```
.siroe.com ¥
  $U%$H$D@$]ldap:///o=siroe.com?mailRoutingSystem?sub? ¥
  (mailDomain=$D)
```

この例で使われている円記号は、書き換え規則の 1 行が次の行に続いていることを示すためのものです。表 7-5 に LDAP URL 置換シーケンスの一覧を示します。

表 7-5 LDAP URL 置換シーケンス

置換シーケンス	説明
$$	リテラル $ 文字
$~ *account*	ユーザアカウントのホームディレクトリ
$A	アドレス
$D	ドメイン名
$H	ホスト名 (完全なドメイン名の最初の部分)

表 7-5　LDAP URL 置換シーケンス (続き)

置換シーケンス	説明
$L	~ または _ などの特別な先頭文字を除くユーザ名
$S	サブアドレス
$U	ユーザ名

一般データベースの置換 : $(...)

$ (テキスト) 形式の置換シーケンスは、特殊な方法で処理されます。テキスト部分は、特殊な一般データベースにアクセスするためのキーとして使われます。このデータベースは、/imta/config/imta_tailor ファイル内の IMTA_GENERAL_DATABASE オプションで指定されているファイル (通常、/imta/db/generaldb.db ファイル) で構成されています。

このデータベースは、imta crdb ユーティリティを使って作成されます。「テキスト文字列」がデータベース内のエントリに一致すると、データベース内の対応するテンプレートがその文字列に置き換えられます。「テキスト文字列」がデータベース内のどのエントリにも一致しなかった場合は、書き換えプロセスが失敗に終わります。つまり、最初から何も一致しなかったのと同じ状態に戻ります。置き換えがうまくいくと、次にデータベースから抽出されたテンプレートに別の置換シーケンスが含まれていないかどうかが調べられます。ただし、抽出されたテンプレート内に別の $ (テキスト) を含めることは禁じられています。参照ループが発生する可能性があるからです。

例として、次の書き換え規則に jdoe@siroe.siroenet というアドレスが一致した場合を考えてみます。

```
.SIROENET $($H)
```

まず、一般データベースで siroe というテキスト文字列が検索され、その結果 (見つかった場合) が書き換え規則のテンプレートとして用いられます。ここで、siroe の検索結果を $u%eng.siroe.com@siroenet とします。この場合、テンプレートの出力は jdoe@eng.siroe.com (すなわち、ユーザ名 = jdoe、ホスト / ドメイン仕様 = eng.siroe.com) になり、ルーティングシステムは siroenet になります。

一般データベースは、正しい操作を行うためにだれでも読み取り可能でなければなりません。

指定マッピングの適用 : ${...}

${mapping,argument} 形式の置換シーケンスは、MTA マッピングファイルでマッピングを検索し、見つかったマッピングを適用するのに使用します。mapping フィールドにはマッピングテーブルの名前を指定し、argument フィールドにはマッピングへ渡す文字列を指定します。この置換シーケンスを使用するには、指定したマッピングが存在し、かつその出力に $Y フラグが設定されていなければなりません。マッピングが存在しなかったり、$Y フラグが設定されていない場合、書き換えは失敗に終わります。問題なく処置が行われた場合は、マッピングの結果がテンプレート内の同じ位置にマージされたあと、再び展開されます。

このメカニズムにより、さまざまな方法で MTA 書き換えプロセスを展開することができます。たとえば、アドレスのユーザ名部分を選択しながら分析したり変更することができます。通常の MTA 書き換えプロセスに、このような機能はありません。

カスタマ指定ルーチンの置換 : $[...]

$ [*image*,*routine*,*argument*] 形式の置換シーケンスは、カスタマ指定ルーチンを検索し呼び出すのに使用します。UNIX において、MTA は dlopen および dlsym を使って動的に共有ライブラリイメージから指定したルーチンをロードし、呼び出します。そのとき、そのルーチンは以下の引数を伴った関数として呼び出されます。

status: = *routine* (*argument*, *arglength*, *result*, *reslength*)

argument および *result* は、252 バイトの文字列バッファです。UNIX で、*argument* と *result* は文字列へのポインタ (例 : C 言語の char*) として渡されます。*arglength* と *reslength* は、参照によって渡される符号付の long 型整数です。入力時に *argument* には書き込み規則テンプレートからの引数文字列が含まれ、*arglength* にはその文字列の長さが含まれます。値を返すときには、*result* に結果文字列が入り、*reslength* にその長さが入ります。次にこの結果文字列は書き換え規則テンプレートで "$[*image*,*routine*,*argument*]" に置換されます。routine の値として 0 が返された場合には書き換え規則が失敗に終わり、-1 が返された場合には書き換え規則は有効になります。

このメカニズムによって、書き換えプロセスの複雑な展開が可能になります。たとえば、あるタイプの名前サービスに対して呼び出しを実行し、その結果を使って名前を変化させることができます。次の書き換え規則を使って、以下のような前方を探すアドレスのディレクトリサービス検索 (例 : To: アドレス) がホスト siroe.com で実行されることがあります。$F を指定すると、この書き換え規則を前方を探すアドレスだけに使用することができます。詳細は 189 ページの「方向および位置に固有の書き換え規則 ($B、$E、$F、$R)」を参照してください。

```
siroe.com $F$[LOOKUP_IMAGE,LOOKUP,$U]
```

jdoe@siroe.com という前方を探すアドレスがこの規則に一致すると、メモリ内に LOOKUP_IMAGE (UNIX の共有ライブラリ) がロードされ、jdoe を引数パラメータとして持つ LOOKUP ルーチンが呼び出されます。その後、LOOKUP ルーチンは、結果パラメータ内の John.Doe%eng.siroe.com などの別のアドレスと書き換え規則が適用されたことを示す値 (-1) を返します。結果文字列にパーセント記号 (170 ページの「繰り返し書き換えテンプレート : A%B」を参照) が使用されていると、アドレスを書き換えるものとして John.Doe@eng.siroe.com を使った書き換えプロセスが再開されます。

UNIX システムでは、サイト提供の共有ライブラリイメージはだれでも読み取り可能でなければなりません。

単一フィールドの置換 : $&、$!、$*、$#

単一フィールド置換シーケンスは、書き換えるホスト / ドメイン仕様からサブドメイン部分を抽出するためのものです。表 7-6 に、使用可能な単一フィールド置換シーケンスを一覧にして示します。

表 7-6 単一フィールドの置換シーケンス

コントロールシーケンス	使用目的
$&n	ホスト仕様 (ワイルドカードに一致しなかった / 一致した部分) 内の n 番目の要素を表す (n=0,1,2,..,9,)。要素はドット文字で区切られており、もっとも左にあるものが「要素 0」となる。要求された要素が存在しない場合は、書き換えは失敗する
$!n	ホスト仕様 (ワイルドカードに一致しなかった / 一致した部分) 内の n 番目の要素を表す (n=0,1,2,..,9,)。要素はドット文字で区切られており、もっとも右にあるものが「要素 0」となる。要求された要素が存在しない場合は、書き換えは失敗する
$*n	ドメイン仕様 (パターンで指定されているテキストに一致した部分) 内の n 番目の要素を表す (n=0,1,2,...,9,)。要素はドット文字で区切られており、もっとも左にあるものが「要素 0」となる。要求された要素が存在しない場合は、書き換えは失敗する
$#n	ドメイン仕様 (パターンで指定されているテキストに一致した部分) 内の n 番目の要素を表す (n=0,1,2,...,9,)。要素はドット文字で区切られており、もっとも右にあるものが「要素 0」となる。要求された要素が存在しない場合は、書き換えは失敗する

jdoe@eng.siroe.com というアドレスが次の書き換え規則に一致したとします。

```
*.SIROE.COM       $U%$&0.siroe.com@mailhub.siroe.com
```

この場合、テンプレートからは「mailhub.siroe.com をルーティングシステムとして使った jdoe@eng.siroe.com」という結果が得られます。

固有文字列の置換

$W コントロールシーケンスは、大文字の英数字からなる繰り返し不可能な固有のテキスト文字列に挿入します。$W は、繰り返されないアドレス情報を作成するような場合に便利です。

ソースチャネル固有の書き換え規則 ($M、$N)

特定のソースチャネルに関してのみ動作する書き換え規則を作成することができます。これは、短形式の名前に 2 つの意味が含まれるような場合に便利です。

1. 名前が 1 つのチャネルに届くメッセージ内にある場合
2. 名前が別のチャネルに届くメッセージ内にある場合

ソースチャネル固有の書き換えは、使用中のチャネルプログラムと、rules や norules というチャネルキーワードに関連しています。書き換えを実行する MTA コンポーネントに関連付けられたチャネルに norules が指定されている場合、チャネル固有の書き換え規則チェックは行われません。そのチャネルに rules 指定されている場合は、チャネル固有の書き換え規則チェックが行われます。デフォルトのキーワードは rules です。

ソースチャネル固有の書き換えは、指定されたアドレスに一致するチャネルとは関係がありません。このタイプの書き換えは、書き換えを実行する MTA コンポーネントとそのコンポーネントのチャネルテーブルエントリにのみ依存します。

チャネル固有の書き換え規則チェックは、規則のテンプレート部分に $N または $M コントロールシーケンスがある場合に実行されます。$N や $M に続く文字は、単価記号 (@)、パーセント記号 (%) または、後続の $N、$M、$Q、$C、$T、または $? までチャネル名と解釈します。

たとえば、$M*channel* を使用したときに *channel* が現在書き換えを行っているチャネルでない場合は、規則が適用されません。また、$N*channel* を使用したときに *channel* が書き換えを行っている場合も、規則が適用されません。複数の $M および $N 句を指定することもできます。複数の $M 句を使用した場合は、そのうちの 1 つでも一致すれば、規則が適用されます。複数の $N 句を使用している場合は、そのうちの 1 つでも一致すれば、規則の適用は失敗に終わります。

宛先チャネル固有の書き換え規則 ($C、$Q)

メッセージをキューに入れる依存する書き換え規則を作成することができます。これは、あるホストに対して名前が 2 つあるような場合に便利です。つまり、1 つのホストグループに認識されている名前と、別のホストグループに認識されている名前とが異なる場合です。異なるチャネルを使って各グループにメールを送ることにより、各グループに知られている名前を使ってホストを参照するようにアドレスを書き換えることができます。

宛先チャネル固有の書き換えは、メッセージを取り出して処理するチャネルと、そのチャネルに関する `rules` および `norules` キーワードに関連しています。宛先チャネルに `norules` が指定されている場合、チャネル固有の書き換え規則チェックは行われません。宛先チャネルに `rules` 指定されている場合は、チャネル固有の書き換え規則チェックが行われます。デフォルトのキーワードは `rules` です。

宛先チャネル固有の書き換えは、指定されたアドレスに一致するチャネルとは関係がありません。このタイプの書き換えは、メッセージのエンベロープ `To:` アドレスにのみ依存します。メッセージがキューに入ると、まずそのエンベロープ `To:` アドレスが書き換えられ、メッセージの送り先チャネルが決定されます。エンベロープ `To:` アドレスの書き換え中、`$C` コントロールシーケンスや `$Q` コントロールシーケンスはすべて無視されます。エンベロープ `To:` アドレスが書き換えられて宛先チャネルが決まると、その後、メッセージに関連するほかのアドレスが書き換えられるときに、`$C` および `$Q` コントロールシーケンスが考慮されます。

宛先チャネル固有の書き換え規則チェックは、規則のテンプレート部分に `$C` または `$Q` コントロールシーケンスがあると実行されます。`$C` または `$Q` に続く文字は、単価記号 (`@`) やパーセント記号 (`%`) または、後続の `$N`、`$M`、`$C`、`$Q`、`$T`、または `$?` までチャネル名と解釈します。

たとえば、`$Q`*channel* を使用したときに *channel* が宛先チャネルでない場合は、規則が適用されません。また、`$C`*channel* を使用したときに *channel* が宛先である場合にも、規則は適用されません。複数の `$Q` および `$C` 句を指定することもできます。複数の `$Q` 句を使用した場合は、そのうちの 1 つでも一致すれば、規則が適用されます。複数の `$C` 句を指定した場合は、そのうちの 1 つでも一致すれば、規則の適用は失敗に終わります。

方向および位置に固有の書き換え規則 ($B、$E、$F、$R)

エンベロープアドレスにのみ適用される書き換え規則、またはヘッダーアドレスにのみ適用される書き換え規則を指定したい場合があります。$E コントロールシーケンスを使うと、書き換えるアドレスがエンベロープアドレスでない場合、書き換えを実行することができなくなります。$B コントロールシーケンスを使うと、書き換えるアドレスがメッセージのヘッダーまたは本文からのものでない場合、書き換えを実行することができなくなります。これらのシーケンスはこのような効果を得る目的でのみ使用され、書き換え規則テンプレート内の任意の場所に含めることができます。

アドレスは、方向によって分類することもできます。前方を探すアドレスは、To:、Cc:、Resent-to:、または宛先を参照するほかのヘッダー行またはエンベロープ行に関して生じるアドレスです。また、後方を探すアドレスは、From:、Sender:、または Resent-From: といったソースを参照するものです。$F コントロールシーケンスを使うと、前方を探すアドレスである場合に書き換え規則が適用されます。$R コントロールシーケンスを使うと、リバースポインティングを探すアドレスである場合に書き換え規則が適用されます。

ホスト名の位置に固有の書き換え ($A、$P、$S、$X)

アドレス内のホスト名の位置に基づいて適用されるような規則を必要とする場合があります。アドレス内のホスト名は、以下の位置にくることが考えられます。

- ソースルート内
- 単価記号 (@) の右側
- ローカル部分のパーセント記号 (%) の右側
- ローカル部分の感嘆符 (!) の左側

通常ホスト名は、それがどこに位置するかに関係なく、同じように処理されます。ただし、特別な処理を必要とする場合もあります。

アドレス内のホスト名の位置に基づいてマッチング動作を制御するには、以下の 4 つのコントロールシーケンスを使用できます。

- 規則をソースルートから抽出されたホストに一致させるには、$S を使用します。
- 規則を単価記号 @ の右側にあるホストに一致させるには、$A を使用します。
- 規則を % 記号の右側にあるホストに一致させるには、$P を使用します。
- 規則を感嘆符 (!) の左側にあるホストに一致させるには、$X を使用します。

ホスト名が指定した位置にない場合は、規則の適用が失敗に終わります。これらのシーケンスは、1 つの書き換え規則内で組み合わせることもできます。たとえば、$S と $A を指定すると、規則はソースルート内のホスト名または単価記号 @ の右側にあるホスト名のいずれかに一致します。これらのシーケンスをすべて指定したのと、どれも指定しないのとは同じことです。すなわち、規則はホスト名の位置に関係なく一致します。

現在のタグ値の変更 ($T)

現在の書き換え規則タグを変更するには、$T コントロールシーケンスを使用します。書き換え規則タグはすべての書き換え規則パターンの先頭に付けられ、その後、設定ファイルやドメインデータベースで書き換え規則パターンの検索が行われます。$T の直後から単価記号 @、パーセント記号 %、$N、$M、$Q、$C、$T、または $? までの間のテキストが新しいタグとして扱われます。

タグは、特定のコンポーネントが検出されたときにアドレスの特性全体が変わるような、特殊なアドレス形式を処理する場合に便利です。たとえば、internet という特別なホスト名が見つかったときに、そのホスト名をアドレスから削除し、削除後のアドレスを強制的に TCP-DAEMON チャネルにマッチングするとします。

これは、以下のような規則を使って実行できます (ローカルホストの正式な名前を localhost とします)。

```
internet                $S$U@localhost$Tmtcp-force|

mtcp-force|.            $U%$H@TCP-DAEMON
```

最初の規則は、ソースルート内で internet という特別なホスト名が見つかった場合、そのホスト名に一致します。その後、ローカルチャネルと internet とのマッチングが行われ、アドレスから internet が削除されます。そして、書き換えタグが設定されます。書き換えプロセスは続けられますが、タグに対して通常の規則が一致することはありません。最後に、デフォルトの規則がタグとともに試され、2 番目の規則に移ります。この規則では、ほかの条件に関係なく、アドレスが強制的に TCP-DAEMON チャネルに対してマッチングされます。

書き換えに関連するエラーメッセージの制御 ($?)

MTA には、書き換えとチャネルの照合に失敗したときに表示されるデフォルトのエラーメッセージがあります。これらのメッセージは、特定の条件下で変更することができます。たとえば、だれかが Ethernet ルーターボックスにメールを送信しようとした場合などは、「不正なホスト / ドメインが指定されています」というより「ルーターがメールを受け入れられません」というメッセージを表示した方がより適切です。

特殊なコントロールシーケンスを使って、規則の適用に失敗した場合に印刷されるエラーメッセージを変更することができます。エラーメッセージを指定するには、$? シーケンスを使用します。$? の直後から単価記号 @、% 記号、$N、$M、$Q、$C、$T、または $? までの間のテキストがエラーメッセージのテキストとして扱われます。このエラーメッセージは、書き換えの結果がどのチャネルにも一致しなかった場合に印刷されます。エラーメッセージの設定は記憶され、書き換えプロセスを通じて有効となります。

$? を含む規則もほかの規則と同じように動作します。特別なケースとして、$? だけを含む規則には注意してください。この場合、アドレスのメールボックスまたはホスト部分は変更されずに書き換えプロセスが終了し、ホストがそのままチャネルテーブル内で検索されます。この検索は失敗に終わり、その結果としてエラーメッセージが返されます。

たとえば、MTA 設定ファイル内に、次に示すような最終的な書き換え規則があるとします。

```
. $?Unrecognized address; contact postmaster@siroe.com
```

この例で、認識されないホスト / ドメイン仕様は、その失敗のプロセスにおいて、「`Unrecognized address; contact postmaster@siroe.com`」というエラーメッセージを生成します。

多数の書き換え規則を扱う

MTA は常に `imta.cnf` ファイルからすべての書き換え規則を読み取り、メモリ内のハッシュテーブルにそれらの規則を保存します。コンパイルした設定を使用すると、情報が必要になるたびに設定ファイルを読み取るという作業を省くことができます。この場合でも、メモリ内にすべての書き込み規則を保存するためにハッシュテーブルが使われます。この方法は、書き換え規則があまり多くない場合に適しています。サイトによっては 10,000 個以上の書き換え規則が必要になる場合もあります。このような場合には、かなり多くのメモリを費やさなければなりません。

MTA では、補助的なインデックス付きデータファイルに多数の書き換え規則を保存するオプションの機能を使って、この問題を解決することができます。通常の設定ファイルが読み取られるたびに、MTA はドメインデータベースがあるかどうかを調べます。データベースがある場合は、設定ファイルの規則が照合に失敗するたびにそのデータベースが開かれ、その内容が調べられます。ドメインデータベースが調べられるのは、指定された規則が設定ファイル内に見つからなかったときだけです。そのため、規則はいつでも設定ファイルに追加することができ、それによってデータベース内の規則が無効になります。デフォルトでは、ドメインデータベースはホストドメインに関連する書き換え規則を保存するために使用されます。
IMTA_DOMAIN_DATABASE 属性は `imta_tailor` ファイルに保存されています。このデータベースのデフォルトの場所は *server-instance*`/imta/db/domaindb.db` です。

注	このファイルは手作業で編集しないでください。Directory Server でホストドメインが作成されると、`dirsync` プロセスが既存のドメインデータベースを上書きします。そのため、カスタム編集した内容は失われてしまいます。

書き換え規則をテストする

書き換え規則をテストするには `imsimta test -rewrite` コマンドを使用します。`-noimage` 修飾子を使うと、新しい設定をコンパイルする前に、設定ファイルに加えた変更内容をテストすることができます。

このユーティリティと `-debug` 修飾子を使って少数のアドレスを書き換えると便利かもしれません。この場合、ステップバイステップ形式でアドレスの書き換えが行われます。たとえば、以下のコマンドを実行します。

```
% imsimta test -rewrite -debug joe@siroe.com
```

`imsimta test -rewrite` ユーティリティの詳細については、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

書き換え規則の例

以下に、書き換え規則の例とそれらの規則によってサンプルアドレスがどのうように書き換えられるかを示します。

SC.CS.SIROE.EDU システムの設定ファイルに、図 7-2 に示す書き換え規則が含まれているとします。

図 7-2 書き換え規則の例

```
sc                      $U@sc.cs.siroe.edu
sc1                     $U@sc1.cs.siroe.edu
sc2                     $U@sc2.cs.siroe.edu
*                       $U%$&0.cs.siroe.edu
*.cs                    $U%$&0.cs.siroe.edu
*.cs.siroe              $U%$&0.cs.siroe.edu
*.cs.siroe.edu          $U%$&0.cs.siroe.edu@ds.adm.siroe.edu
sc.cs.siroe.edu         $U@$D
sc1.cs.siroe.edu        $U@$D
sc2.cs.siroe.edu        $U@$D
sd.cs.siroe.edu         $U@sd.cs.siroe.edu
.siroe.edu              $U%$H.siroe.edu@cds.adm.siroe.edu
.edu                    $U@$H$D@gate.adm.siroe.edu
[]                      $U@[$L]@gate.adm.siroe.edu
```

表 7-7 に、サンプルアドレスと、それらの書き換え結果およびルートを示します。

表 7-7 サンプルアドレスと書き換え結果

最初のアドレス	書き換え後	ルート
user@sc	user@sc.cs.siroe.edu	sc.cs.siroe.edu
user@sc1	user@sc1.cs.siroe.edu	sc1.cs.siroe.edu
user@sc2	user@sc2.cs.siroe.edu	sc2.cs.siroe.edu
user@sc.cs	user@sc.cs.siroe.edu	sc.cs.siroe.edu
user@sc1.cs	user@sc1.cs.siroe.edu	sc1.cs.siroe.edu
user@sc2.cs	user@sc2.cs.siroe.edu	sc2.cs.siroe.edu
user@sc.cs.siroe	user@sc.cs.siroe.edu	sc.cs.siroe.edu
user@sc1.cs.siroe	user@sc1.cs.siroe.edu	sc1.cs.siroe.edu
user@sc2.cs.siroe	user@sc2.cs.siroe.edu	sc2.cs.siroe.edu
user@sc.cs.siroe.edu	user@sc.cs.siroe.edu	sc.cs.siroe.edu
user@sc1.cs.siroe.edu	user@sc1.cs.siroe.edu	sc1.cs.siroe.edu
user@sc2.cs.siroe.edu	user@sc2.cs.siroe.edu	sc2.cs.siroe.edu
user@sd.cs.siroe.edu	user@sd.cs.siroe.edu	sd.cs.siroe.edu
user@aa.cs.siroe.edu	user@aa.cs.siroe.edu	ds.adm.siroe.edu
user@a.eng.siroe.edu	user@a.eng.siroe.edu	cds.adm.siroe.edu
user@a.cs.sesta.edu	user@a.cs.sesta.edu	gate.adm.siroe.edu —route inserted

表 7-7　サンプルアドレスと書き換え結果 (続き)

最初のアドレス	書き換え後	ルート
user@b.cs.sesta.edu	user@b.cs.sesta.edu	gate.adm.siroe.edu —route inserted
user@[1.2.3.4]	user@[1.2.3.4]	gate.adm.siroe.edu —route inserted

基本的に、これらの書き換え規則の内容は次のとおりです。ホスト名が短形式の名前 (sc、sc1、または sc2) の 1 つである場合、またはフルネーム (sc.cs.siroe.edu など) の 1 つである場合は、その名前をフルネームに展開し、ユーザに送る。cs.cmu.edu を 1 つの部分からなる短形式の名前に追加し、再試行する。.cs が後ろに続く 1 つの部分を .cs.siroe.edu が後ろに続く 1 つの部分に変換し、もう一度試行する。また、.cs.siroe も .cs.siroe.edu に変換し、もう一度試行する。

名前が sd.cs.siroe.edu (ユーザが直接接続するシステム) である場合は、それを書き換えて、そこに送る。ホスト名が .cs.siroe.edu サブドメイン内のほかのものである場合は、それを ds.cs.siroe.edu (.cs.siroe.edu サブドメインのゲートウェイ) に送る。ホスト名が siroe.edu サブドメイン内のほかのものである場合は、それを cds.adm.siroe.edu (siroe.edu サブドメインのゲートウェイ) に送る。ホスト名が .edu トップレベル内のほかのものである場合は、それを gate.adm.siroe.edu (メッセージを適切な宛先に送ることが可能) に送る。ドメインリテラルが使用されている場合は、それも gate.adm.siroe.edu に送る。

上記の例のように、書き換え規則によってアドレスのユーザ名 (またはメールボックス) 部分が変更されることはほとんどありません。アドレスのユーザ名部分を変更する機能は、MTA が RFC 822 に準拠しないメールソフトウェア (ホスト / ドメイン仕様をアドレスのユーザ名部分に詰め込む必要があるメールソフトウェア) へのインタフェースとして使われる場合に使用されます。この機能を使用する際には、十分な配慮が必要です。

第 8 章

チャネル定義を設定する

この章では、MTA 設定ファイル `imta.cnf` でのチャネルキーワード定義の使用方法について説明します。この章を読む前に、第 6 章「MTA サービスと設定について」、および 105 ページの「チャネル定義」と 111 ページの「MTA 設定ファイル」をお読みください。この章には、以下の節があります。

- チャネルキーワードの一覧 (アルファベット順)
- 機能別チャネルキーワード
- チャネルのデフォルトを設定する
- SMTP チャネルを設定する
- メッセージの処理と配信を設定する
- アドレス処理を設定する
- ヘッダー処理を設定する
- 添付と MIME 処理
- メッセージのサイズ制限、ユーザ制限容量、権限
- MTA キュー領域でのファイル作成
- メールボックスフィルタファイルの場所を指定する
- ログ記録とデバッグを設定する
- その他のキーワード

注　imta.cnf 内のチャネル定義を変更する場合は、imsimta start コマンドを使って起動するときに設定データを 1 回だけ読み込むようなプログラムまたはチャネルを再起動する必要があります (例 : SMTP サーバ)。コンパイルされた設定を使用する場合は、設定を再コンパイルしてからプログラムを再起動する必要があります。設定情報のコンパイルおよびプログラムの再起動については、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

チャネルキーワードの一覧 (アルファベット順)

次の表にキーワードの一覧をアルファベット順に示します。

表 8-1　チャネルキーワード (アルファベット順)

キーワード	ページ	キーワード	ページ	キーワード	ページ	キーワード	ページ
733	246	822	246	addreturnpath	253	addrsperfile	268
aliaslocal	256	aliaspostmaster	159	allowetrn	220	allowswitchchannel	232
authrewrite	234	backoff	239	bangoverpercent	248	bangstyle	246
bidirectional	238	blocketrn	220	blocklimit	267	cacheeverything	228
cachefailures	228	cachesuccesses	228	channelfilter	272	charset7	223
charset8	223	charsetesc	223	checkehlo	219	commentinc	254
commentmap	254	commentomit	254	commentstrip	254	commenttotal	254
connectalias	249	connectcanonical	249	copysendpost	157	copywarnpost	158
daemon	232	datefour	261	datetwo	261	dayofweek	261
defaulthost	250	defaultmx	230	defaultnameservers	231	deferred	238
defragment	264	dequeue_removeroute	257	destinationfilter	272	disableetrn	220
domainetrn	220	domainvrfy	221	dropblank	252	ehlo	219
eightbit	223	eightnegotiate	223	eightstrict	223	errsendpost	157
errwarnpost	158	expandchannel	244	expandlimit	244	exproute	248
fileinto	272	filesperjob	241	filter	272	forwardcheckdelete	229
forwardchecknone	229	forwardchecktag	229	header_733	246	header_822	246
header_uucp	246	headerlabelalign	262	headerlinelength	262	headerread	259

表 8-1　チャネルキーワード (アルファベット順) (続き)

キーワード	ページ	キーワード	ページ	キーワード	ページ	キーワード	ページ
headertrim	259	holdexquota	268	holdlimit	244	identnone	229
identnonelimited	229	identnonenumeric	229	identnonesymbolic	229	identtcp	229
identtcplimited	229	identtcpsymbolic	229	ignoreencoding	264	immnonurgent	
improute	248	includefinal	157	indenttcpnumeric	229	inner	259
innertrim	259	interfaceaddress	228	interpretencoding	264	language	263
lastresort	231	linelength	266	linelimit	267	localvrfy	221
logging	270	loopcheck	271	mailfromdnsverify	222	master	238
master_debug	270	maxblocks	265	maxheaderaddrs	262	maxheaderchars	262
maxjobs	241	maxlines	265	maxprocchars	262	maysaslserver	233
maytls	235	maytlsclient	235	maytlsserver	235	missingrecipientpolicy	251
msexchange	234	multiple	268	mustsaslserver	233	musttls	235
musttlsclient	235	musttlsserver	235	mx	230	nameservers	231
noaddreturnpath	253	nobangoverpercent	248	noblocklimit	267	nocache	228
nochannelfilter	272	nodayofweek	261	nodefaulthost	250	nodeferred	238
nodefragment	264	nodestinationfilter	272	nodropblank	252	noehlo	219
noexproute	248	noexquota	268	nofileinto	272	nofilter	272
noheaderread	259	noheadertrim	259	noimproute	248	noinner	259
noinnertrim	259	nolinelimit	267	nologging	270	noloopcheck	271
nomailfromdnsverify	222	nomaster_debug	270	nomsexchange	234	nomx	230
nonrandomemx	230	nonurgentbackoff	239	nonurgentblocklimit	243	nonurgentnotices	156
noreceivedfor	253	noreceivedfrom	253	noremotehost	250	norestricted	252
noreturnaddress	159	noreturnpersonal	159	noreverse	252	normalbackoff	239
normalblocklimit	243	normalnotices	156	norules	257	nosasl	233
nosaslserver	233	nosaslswitchchannel	233	nosendetrn	220	nosendpost	157
noservice	245	noslave_debug	270	nosmtp	219	nosourcefilter	272
noswitchchannel	232	notices	156	notls	235	notlsclient	235
notlsserver	235	novrfy	221	nowarnpost	158	nox_env_to	260
percentonly	248	percents	246	personalinc	255	personalmap	255

表 8-1　チャネルキーワード (アルファベット順) (続き)

キーワード	ページ	キーワード	ページ	キーワード	ページ	キーワード	ページ
personalomit	255	personalstrip	255	pool	240	port	228
postheadbody	159	postheadonly	159	randommx	230	receivedfor	253
receivedfrom	253	remotehost	250	restricted	252	returnaddress	159
returnenvelope	158	returnpersonal	159	reverse	252	routelocal	249
rules	257	rules	257	saslswitchchannel	233	sendetrn	220
sendpost	157	sensitivitycompanyconfidential	263	sensitivitynormal	263	sensitivitypersonal	263
sensitivityprivate	263	service	245	sevenbit	223	silentetrn	220
single	268	single_sys	232	slave	238	slave_debug	270
smtp	219	smtp_cr	219	smtp_crlf	219	smtp_crorlf	219
smtp_lf	219	sourceblocklimit	267	sourcecommentinc	254	sourcecommentmap	254
sourcecommentomit	254	sourcecommentstrip	254	sourcecommenttotal	254	sourcefilter	272
sourcepersonalinc	255	sourcepersonalmap	255	sourcepersonalomit	255	sourcepersonalstrip	255
sourceroute	246	streaming	224	subaddressexact	256	subaddressrelaxed	256
subaddresswild	256	subdirs	269	submit	271	suppressfinal	157
switchchannel	232	threaddepth	243	tlsswitchchannel	235	unrestricted	252
urgentbackoff	239	urgentblocklimit	243	urgentnotices	156	useintermediate	157
user	271	uucp	246	viaaliasoptional	258	viaaliasrequired	258
vrfyallow	221	vrfydefault	221	vrfyhide	221	warnpost	158
x_env_to	260						

機能別チャネルキーワード

次の表に分類したキーワードの一覧を示します。

表 8-2　機能別チャネルキーワード (**太字**はデフォルト)

キーワード	ページ	定義
アドレス処理		
733	246	エンベロープで % ルーティングを使用する。percents と同義
822	246	エンベロープでソースルートを使用する。sourceroute と同義
addreturnpath	253	このチャネルにキューを入れる際に、メッセージに Return-Path: ヘッダーを追加する
aliaslocal	256	書き換えられたアドレスをエイリアスファイルとエイリアスデータベースで検索する
authrewrite	234	認証された差出人の情報がある場合は MTA がヘッダーに含めるようにするために、ソースチャネルで使用する
bangoverpercent	248	A!B%C を A!(B%C) としてグループ化する
bangstyle	246	エンベロープで UUCP! ルーティングを使用する。uucp と同義
defaulthost	250	アドレスを完成させるためにドメイン名を指定する
dequeue_removeroute	257	エンベロープの To: アドレスからソースルートを削除する
exproute	248	アドレスをリモートシステムに渡す際に明示的なルーティングを要求する
holdlimit	244	エンベロープ受取人アドレス数がこの制限を越えた場合、メッセージを保留する
improute	248	このチャネルのアドレスに対して黙示的なルーティングを実行する
missingrecipientpolicy	251	受取人ヘッダーがないメッセージを有効にする (どのヘッダーに追加するか指定する) ポリシーを設定する
noaddreturnpath	253	メッセージをキューに入れる際に Return-Path: ヘッダーを追加しない
nobangoverpercent	248	A!B%C を (A!B)%C としてグループ化する
nodefaulthost	250	アドレスを完成させるために使用する、ドメイン名を指定しない
noexproute	248	このチャネルのアドレスに対して明示的なルーティングを実行しない

表 8-2　機能別チャネルキーワード (**太字**はデフォルト) (続き)

キーワード	ページ	定義
noimproute	248	このチャネルのアドレスに対して黙示的なルーティングを実行しない
noreceivedfrom	253	元のエンベロープの From: アドレスを含めずに Received: ヘッダー行を作成する
noremotehost	250	アドレスを完成させるために、ローカルホストのドメイン名をデフォルトのドメイン名として使用する
norestricted	252	unsrestricted と同じ
noreverse	252	メッセージのアドレスを、アドレスリバース処理から外すことを指定する
norules	257	このチャネル固有の書き換え規則を確認しない
percentonly	248	bang パスを無視する。エンベロープで % ルーティングを使用する
percents	246	エンベロープで % ルーティングを使用する。733 と同義
remotehost	250	アドレスを完成させるために、リモートホストの名前をデフォルトのドメイン名として使用する
restricted	252	チャネルは、エンコーディングを必要とするメールシステムに接続する
reverse	252	アドレスリバースデータベースまたは REVERSE マッピングのアドレスを確認
routelocal	249	アドレスをチャネルに書き換える際に、MTA にアドレスのすべての明示的ルーティングを短絡化しようとする
rules	257	このチャネル固有の書き換え規則を確認する
sourceroute	246	822 と同義
subaddressexact	256	エントリの一致の確認中に特別なサブアドレスの処理を行わない。エイリアスが一致するとみなされるためには、サブアドレスを含むメールボックス全体が一致する必要がある
subaddressrelaxed	256	完全一致と「名前 +*」の形式一致を検索したあと、MTA で名前の部分のみの一致を検索する
subaddresswild	256	サブアドレス全体を含む完全一致を検索したあと、MTA で「名前 +*」の形式のエントリを検索する
unrestricted	252	RFC 1137 エンコーディングとデコーディングを実行するように MTA に指示する
uucp	246	エンベロープで UUCP! ルーティングを使用する。bangstyle と同義

表 8-2　機能別チャネルキーワード (**太字**はデフォルト) (続き)

キーワード	ページ	定義
viaaliasoptional	258	チャネルに一致する最終受取人のアドレスを、エイリアスで作成する必要がない
viaaliasrequired	258	チャネルに一致する最終受取人アドレスを、エイリアスで作成する必要がある
添付と MIME 処理		
defragment	264	このチャネルのキューに入っている部分メッセージは、デフラグメンテーションチャネルのキューに移動する
ignoreencoding	264	受信メッセージの Encoding: ヘッダーを無視する
interpretencoding	264	受信メッセージの Encoding: ヘッダーを必要に応じて解釈する
nodefragment	264	デフラグメンテーションを無効にする
文字セットと 8 ビットデータ		
charset7	223	7 ビットのテキストメッセージに関連付けるデフォルトの文字セット
charset8	223	8 ビットのテキストメッセージに関連付けるデフォルトの文字セット
charsetesc	223	エスケープ文字を含む 7 ビットのテキストに関連付けるデフォルトの文字セット
eightbit	223	チャネルが 8 ビット文字をサポートする
eightnegotiate	223	チャネルが 8 ビット転送の使用をネゴシエートする (可能な場合)
eightstrict	223	ネゴシエーションが行われていない 8 ビットデータを含むメッセージを拒否する
sevenbit	223	8 ビット文字をサポートしない。8 ビット文字はエンコードされなければならない
MTA キュー領域でのファイル作成		
addrsperfile	268	チャネルのキューにある 1 つのメッセージファイルに関連付けられる受取人の最大数を制限する
expandchannel	244	expandlimit の適用による遅延拡張を実行するチャネルを指定する
expandlimit	244	アドレスの数がこの制限を超えた場合、受信メッセージを「オフライン」で処理する
multiple	268	メッセージファイル内の受取人数を制限しない。ただし SMTP チャネルのデフォルトは 99 である

表 8-2　機能別チャネルキーワード (**太字**はデフォルト) (続き)

キーワード	ページ	定義
single	268	チャネル上の各宛先アドレス用にメッセージのコピーが 1 つずつ作成される
single_sys	268	各宛先システム用にメッセージのコピーを 1 つずつ作成する
subdirs	269	チャネルキューのメッセージを拡散するサブディレクトリの数を指定する
ヘッダー		
authrewrite	234	認証された差出人の情報がある場合は MTA がヘッダーに含めるようにするために、ソースチャネルで使用する
commentinc	254	メッセージのヘッダー行内のコメントをそのままにする
commentmap	254	COMMENT_STRINGS マッピングテーブルを通じて、メッセージヘッダー行でコメント文字列を実行する
commentomit	254	メッセージのヘッダー行内のコメントを削除する
commentstrip	254	メッセージのヘッダー行内にある問題を起こす文字を削除する
commenttotal	254	Received: ヘッダー行以外のすべてのヘッダー行から ( ) に入っているコメントを削除する。ただし、推奨しない
datefour	261	すべての年表示フィールドを 4 桁に展開する
datetwo	261	4 桁の日付表示から先頭の 2 桁を削除する。2 桁の日付表示を要求するメールシステムとの互換性を提供するための機能なので、その他の目的のために使用しないこと
dayofweek	261	曜日情報を残し、曜日情報がない場合にはその情報を日付 / 時刻ヘッダーに追加する
defaulthost	250	アドレスを完成させるためにドメイン名を指定する
dropblank	252	受信メッセージから不正な空白ヘッダーを削除する
header_733	246	メッセージヘッダーで % ルーティングを使用する
header_822	246	メッセージヘッダーでソースルートを使用する
headerlabelalign	262	このチャネルのキューに入れられたメッセージヘッダーの配置ポイントを制御する。整数値の引数をとる
headerlinelength	262	このチャネルのキューに入れられたヘッダー行の長さを制御する
headerread	259	メッセージがキューに入れられたときに、元のメッセージヘッダーが作成される前にオプションファイルからそのメッセージのヘッダーにトリミングの規則を適用する (注意して使用すること)
headertrim	259	元のメッセージヘッダーが作成されたあとで、オプションファイルからそのメッセージのヘッダーにトリミングの規則を適用する

表 8-2　機能別チャネルキーワード (**太字**はデフォルト) (続き)

キーワード	ページ	定義
header_uucp	246	ヘッダーで ! ルーティングを使用する
inner	259	メッセージを解析して、内部ヘッダーを書き換える
innertrim	259	内部のメッセージヘッダーに、オプションファイルからのヘッダートリミング規則を適用する (注意して使用すること)
language	263	ヘッダーにデフォルトの言語を指定する
maxheaderaddrs	262	1 行に表示できるアドレスの数を指定する
maxheaderchars	262	1 行に表示できる文字数を指定する
missingrecipientpolicy	251	受取人ヘッダーがないメッセージを有効にする (どのヘッダーに追加するか指定する) ポリシーを設定する
nodayofweek	261	日付 / 時刻ヘッダーから曜日情報を削除する。この情報が処理できないメールシステムとの互換性を提供するための機能なので、その他の目的のために使用しないこと
nodefaulthost	250	アドレスを完成させるために使用する、ドメイン名を指定しない
nodropblank	252	受信メッセージから不正な空白ヘッダーを削除しない
noheaderread	259	オプションファイルからのヘッダートリミング規則を適用しない
noheadertrim	259	オプションファイルからのヘッダートリミング規則を適用しない
noinner	259	内部のメッセージヘッダー行を書き換えない
noinnertrim	259	内部のメッセージヘッダーにヘッダートリミング規則を適用しない
noreceivedfor	253	エンベロープ受取人情報を含めずに Received: ヘッダー行を作成する
noreceivedfrom	253	元のエンベロープの From: アドレスを含めずに Received: ヘッダー行を作成する
noremotehost	250	アドレスを完成させるために、ローカルホストのドメイン名をデフォルトのドメイン名として使用する
noreverse	252	チャネルのキューに入れられたメッセージのアドレスを、アドレスリバース処理から外す
norules	257	このチャネル固有の書き換え規則を確認しない
nox_env_to	260	X-Envelope-to ヘッダー行を削除する
personalinc	255	メッセージのヘッダー行にある個人名のフィールドをそのままにする

表 8-2　機能別チャネルキーワード (**太字**はデフォルト) (続き)

キーワード	ページ	定義
personalmap	255	PERSONAL_NAMES マッピングテーブルを通じて、個人名を実行する
personalomit	255	メッセージのヘッダー行にある個人名のフィールドを削除する
personalstrip	255	ヘッダー行にある個人名のフィールドから問題になる文字を削除する
receivedfor	253	メッセージの宛先になっているエンベロープ受取人アドレスが 1 つだけの場合は、そのエンベロープの To: アドレスを Received: ヘッダー行に含める
receivedfrom	253	MTA がエンベロープの From: アドレスを変更する場合は、受信メッセージの Received: ヘッダー行を作成するときに元のエンベロープの From: アドレスを含める
remotehost	250	アドレスを完成させるために、リモートホストの名前をデフォルトのドメイン名として使用する
restricted	252	チャネルは、このエンコーディングを必要とするメールシステムに接続する
reverse	252	アドレスリバースデータベースまたは REVERSE マッピングのアドレスを確認
rules	257	このチャネル固有の書き換え規則を確認する
sensitivitycompanyconfidential	263	Companyconfidential が、受け付けるメッセージの重要度の上限である
sensitivitynormal	263	Normal が、受け付けるメッセージの重要度の上限である
sensitivitypersonal	263	Personal が、受け付けるメッセージの重要度の上限である
sensitivityprivate	263	Private が、受け付けるメッセージの重要度の上限である
sourcecommentinc	254	受信メッセージのヘッダー行にコメントを残す
sourcecommentmap	254	ソースチャネルを通じて、ヘッダー行のコメント文字列を実行する
sourcecommentomit	254	受信メッセージの To:、From:、Cc: などのヘッダー行からコメントを削除する
sourcecommentstrip	254	受信メッセージのヘッダー行内にある問題を起こす文字を削除する
sourcecommenttotal	254	受信メッセージから、( ) 内に入っているコメントを削除する
sourcepersonalinc	255	メッセージのヘッダー行にある個人名のフィールドをそのままにする

表 8-2　機能別チャネルキーワード (**太字**はデフォルト) (続き)

キーワード	ページ	定義
sourcepersonalmap	255	ソースチャネルを通じて個人名を実行する
sourcepersonalomit	255	メッセージのヘッダー行にある個人名のフィールドを削除する
sourcepersonalstrip	255	メッセージのヘッダー行にある個人名のフィールドから、問題になる文字を削除する
unrestricted	252	RFC 1137 エンコーディングとデコーディングを実行するように MTA に指示する
x_env_to	260	X-Envelope-to ヘッダー行の生成を有効にする
受信チャネルの一致と切り替え		
allowswitchchannel	232	switchchannel チャネルからこのチャネルへの切り替えを許可する
nosaslswitchchannel	233	SASL 認証に成功した場合、このチャネルへの切り替えは許可されない
noswitchchannel	232	チャネルへの切り替えを行わない
switchchannel	232	サーバチャネルから送信元のホストに関連付けられたチャネルに切り替える
saslswitchchannel	233	クライアントが SASL の使用に成功した場合、受信接続が指定のチャネルに切り替えられる
tlsswitchchannel	235	TLS のネゴシエートが成功した場合に、ほかのチャネルに切り替える
ログ記録とデバッグ		
logging	270	キューに対するメッセージの出入りがログに記録され、特定チャネルのログ機能を有効にする
loopcheck	271	MTA が MTA 自体と通信しているかどうかを確認するために、SMTP EHLO 応答見出しに文字列を配置する
master_debug	270	チャネルのマスタープログラム出力内にデバッグ出力を作成する
nologging	270	キューに対するメッセージの出入りをログに記録しない
noloopcheck	271	SMTP EHLO 応答見出しに文字列がない
nomaster_debug	270	チャネルのマスタープログラム出力内にデバッグ出力を行わない
noslave_debug	270	スレーブのデバッグ出力を生成しない
slave_debug	270	スレーブのデバッグ出力を生成する
長いアドレスリストやヘッダー		

表 8-2　機能別チャネルキーワード (**太字**はデフォルト) (続き)

キーワード	ページ	定義
expandchannel	244	expandlimit の適用による遅延拡張を実行するチャネルを指定する
expandlimit	244	アドレスの数がこの制限を超えた場合、受信メッセージを「オフライン」で処理する
holdlimit	244	アドレスの数がこの制限を越えた場合、メッセージを保留する
maxprocchars	262	処理や書き換えができるヘッダーの最大長
メールボックスフィルタ		
channelfilter	272	チャネルフィルタファイルの場所。destinationfilter と同じ
destinationfilter	272	送信するメッセージに提供されるチャネルフィルタの場所
fileinto	272	メールボックスフィルタ fileinto の操作が適用されたときの、アドレスに対する効果を指定する
filter	272	ユーザフィルタファイルの場所を指定する
nochannelfilter	272	送信メッセージに対するチャネルフィルタリングを行わない。nodestinationfilter とも呼ばれる
nodestinationfilter	272	送信メッセージに対するチャネルフィルタリングを実行しない
nofileinto	272	メールボックスフィルタ fileinto の演算子が効果を発揮しない
nofilter	272	ユーザメールボックスのフィルタリングを実行しない
nosourcefilter	272	受信メッセージに対してチャネルフィルタリングを実行しない
sourcefilter	272	受信メッセージ用のチャネルフィルタの場所を指定する
通知メッセージとポストマスターメッセージ (完全な通知手順については 150 ページを参照)		
aliaspostmaster	159	正式なチャネル名でのユーザ名がポストマスターのメッセージは postmaster@ ローカルホストにリダイレクトされる。ローカルホストには、ローカルホスト名 (ローカルチャネルの名前) が入る
copysendpost	157	メッセージの差出人アドレスが空白になっている場合を除き、配信不能メッセージのコピーがポストマスターに送信される
copywarnpost	158	配信不能メッセージの差出人アドレスが空白になっている場合を除き、警告メッセージのコピーがポストマスターに送信される
errsendpost	157	通知を差出人に返すことができない場合に、配信不能通知のコピーをポストマスターに送信する
errwarnpost	158	通知を差出人に返すことができない場合に、警告メッセージのコピーをポストマスターに送信する
includefinal	157	配信通知の中に受取人アドレスの最終的な形式を含める

表 8-2　機能別チャネルキーワード (**太字**はデフォルト) (続き)

キーワード	ページ	定義
nonurgentnotices	156	優先度が低いメッセージを配信できない場合に通知を送り、そのメッセージを返送するまでの時間を指定する
noreturnaddress	159	ポストマスターアドレス名に RETURN_ADDRESS オプション値を使用する
noreturnpersonal	159	ポストマスター個人名に RETURN_PERSONAL オプション値を使用する
normalnotices	156	優先度が標準のメッセージを配信できない場合に通知を送り、そのメッセージを返送するまでの時間を指定する
nosendpost	157	配信不能メッセージのコピーをポストマスターには一切送信しない
notices	156	通知を送り、メッセージを返すまでの時間を指定する
nowarnpost	158	警告メッセージのコピーをポストマスターには一切送信しない
postheadbody	159	ヘッダーとメッセージの内容の両方を返送する
postheadonly	159	ポストマスターにヘッダーだけを返送する
returnaddress	159	ローカルポストマスターの返信アドレスを設定する
returnenvelope	158	空白のエンベロープ返信アドレスの使用を制御する
returnpersonal	159	ローカルのポストマスターに対する個人名を設定する
sendpost	157	配信不能メッセージのコピーをすべてポストマスターに送信する
suppressfinal	157	元の形式のアドレスが存在する場合は、通知メッセージに最終形式のアドレスを含めない
urgentnotices	156	優先度が高いメッセージを配信できない場合に通知を送り、そのメッセージを返送するまでの時間を指定する
useintermediate	157	リストのエクスパンド後、ユーザメールボックス名の設定前に作成された中間形式のアドレスを使用する
warnpost	158	警告メッセージのコピーをすべてポストマスターに送信する。
処理制御とジョブ送信 (より大きい機能単位については 236 ページの表 8-7 を参照)		
backoff	239	配信不能メッセージを再配信する回数。normalbackoff、nonurgentbackoff、urgentbackoff キーワードで置き換え可能
bidirectional	238	マスターとスレーブの両方のプログラムによって処理されるチャネル
deferred	238	Deferred-delivery: ヘッダー行を認識および処理する

表 8-2　機能別チャネルキーワード (**太字**はデフォルト) (続き)

キーワード	ページ	定義
expandchannel	244	expandlimit の適用による遅延拡張を実行するチャネルを指定する
expandlimit	244	アドレスの数がこの制限を超えた場合、受信メッセージを「オフライン」で処理する
filesperjob	241	1 つのジョブで処理できるキューエントリの数
immnonurgent		優先度にかかわらず、送信後すべてのメッセージの配信を即座に開始する
master	238	マスタープログラムによって処理されるチャネル (master)
maxjobs	241	1 つのチャネルに対して同時実行できるジョブの最大数
nodeferred	238	Deferred-delivery: ヘッダー行が許可されないように指定する
nonurgentbackoff	239	優先度が低いメッセージの配信試行頻度
nonurgentblocklimit	243	指定値以上のサイズを持つメッセージの優先度を「低」以下 (2 番目の優先度) に設定する。該当するメッセージは次の定期ジョブまで処理されない
normalbackoff	239	優先度が標準であるメッセージの配信試行頻度
normalblocklimit	243	指定値以上のサイズを持つメッセージの優先度を「低」に設定する
noservice	245	このチャネルで受信するメッセージのサービス変換は CHARSET-CONVERSION を使用して有効にする
pool	240	チャネル用のプールを指定する。この後ろに、現在のチャネルの配信ジョブのプール先となるプール名を指定する
service	245	CHARSET-CONVERSION エントリにかかわらず、無条件でサービスを有効にする
slave	238	マスタープログラムによって処理されるチャネル (slave)
threaddepth	243	マルチスレッド SMTP クライアントに対して新しいスレッドをトリガするために必要なメッセージ数
urgentbackoff	239	優先度が高いメッセージの配信試行頻度
urgentblocklimit	243	指定値以上のサイズを持つメッセージの優先度を「標準」に設定する
user	271	pipe チャネルでどのユーザ名で実行するかを示すのに使用される
重要度の上限		

表 8-2　機能別チャネルキーワード (**太字**はデフォルト) (続き)

キーワード	ページ	定義
sensitivitycompanyconfidential	263	受け付けるメッセージの重要度の上限
sensitivitynormal	263	Normal が、受け付けるメッセージの重要度の上限である
sensitivitypersonal	263	Personal が、受け付けるメッセージの重要度の上限である
sensitivityprivate	263	Private が、受け付けるメッセージの重要度の上限である
メッセージのサイズ制限、ユーザ制限容量、権限		
blocklimit	267	メッセージ当たりの許可されている MTA ブロックの最大数
holdexquota	268	制限容量を超過したユーザに対するメッセージを保留する
holdlimit	244	アドレスの数がこの制限を越えた場合、受信メッセージを保留する
linelength	266	チャネルごとに許される最大のメッセージ行の長さを制限する
linelimit	267	1 つのメッセージに対して許可される最大行数
maxblocks	265	1 つのメッセージに許可するブロックの最大数を指定する
maxlines	265	1 つのメッセージに許可する最大行数を指定する
noblocklimit	267	メッセージ当たりに許可される MTA ブロックの数に制限はない
noexquota	268	制限容量を超過したユーザに対し、すべてのメッセージを差出人に送り返す
nolinelimit	267	メッセージ当たりに許可される行数に制限はない
nonurgentblocklimit	243	指定値以上のサイズを持つメッセージの優先度を「低」以下 (2 番目の優先度) に設定する。該当するメッセージは次の定期ジョブまで処理されない
normalblocklimit	243	指定値以上のサイズを持つメッセージの優先度を「低」に設定する
sourceblocklimit	267	メッセージ当たりの許可されている MTA ブロックの最大数
urgentblocklimit	243	指定値以上のサイズを持つメッセージの優先度を「標準」に設定する
SMTP 認証、SASL および TLS (より大きい機能単位については 233 を参照)		
authrewrite	234	認証された差出人の情報がある場合は MTA がヘッダーに含めるようにするために、ソースチャネルで使用する
maysaslserver	233	クライアントが SASL 認証を使用することを許可する
maytls	235	MTA は TLS 使用の接続を受け入れ、送信接続にも TLS を使用しようと試みる

表 8-2　機能別チャネルキーワード (**太字**はデフォルト) (続き)

キーワード	ページ	定義
maytlsclient	235	MTA SMTP クライアントは TLS をサポートする SMTP サーバにメッセージを送信する際に TLS を使用する
maytlsserver	235	MTA SMTP サーバが STARTTLS 拡張をサポートすることを通知し、メッセージを受信する際に TLS を使用することを許可する
msexchange	234	TCP/IP チャネルで使用して、MTA にこれが MS Exchange ゲートウェイとクライアントとの通信を行うチャネルであることを指示する
mustsaslserver	233	SMTP サーバは、リモートクライアントが認証に成功しないかぎり、メッセージを受け付けない
musttls	235	MTA は送受信接続に必ず TLS を使用する
musttlsclient	235	MTA SMTP クライアントは、メッセージの送信に必ず TLS を使用する (MTA は STARTTLS コマンドを発行し、このコマンドは必ず成功する必要がある)
musttlsserver	235	MTA SMTP サーバが STARTTLS 拡張をサポートすることを通知し、メッセージを受信する際に TLS を使用する
nomsexchange	234	デフォルト
nosasl	233	SASL 認証は許可されない。試行もされない
nosaslserver	233	SASL 認証は許可されない
notls	235	TLS 認証は許可されない。試行もされない
notlsclient	235	送信接続時に MTA SMTP クライアントは TLS を使用することがない (送信接続時に STARTTLS コマンドが発行されない)
notlsserver	235	受信接続時に MTA SMTP サーバは TLS の使用を許可しない (SMTP サーバもコマンド自体も STARTTLS 拡張に通知しない)
saslswitchchannel	233	クライアントが SASL の使用に成功した場合、受信接続が指定のチャネルに切り替えられる
tlsswitchchannel	235	クライアントが TLS ネゴシエーションに成功した場合、受信接続が指定のチャネルに切り替えられる。このキーワードには、切り替え先のチャネルを指定する必要がある
SMTP コマンドとプロトコル (より大きい機能単位については 217 ページの表 8-4 を参照)		
allowetrn	220	ETRN コマンドを処理する
blocketrn	220	ETRN コマンドをブロックする
checkehlo	219	SMTP 応答の見出しを確認して、EHLO と HELO のどちらを使用するか決定する

表 8-2　機能別チャネルキーワード (**太字**はデフォルト) (続き)

キーワード	ページ	定義
disableetrn	220	ETRN SMTP コマンドのサポートを無効にする
domainetrn	220	ドメインを指定する ETRN コマンドだけを処理する
domainvrfy	221	完全なアドレスを使用して VRFY コマンドを発行する
ehlo	219	初期接続に SMTP EHLO コマンドを使用する
eightbit	223	チャネルが 8 ビット文字をサポートする
eightnegotiate	223	チャネルが 8 ビット転送の使用をネゴシエートする (可能な場合)
eightstrict	223	ネゴシエーションが行われていない 8 ビットデータを含むメッセージを拒否する
localvrfy	221	ローカルアドレスを使用して VRFY コマンドを発行する
mailfromdnsverify	222	MAIL FROM: コマンドに使用されているドメインが DNS に存在するかどうかを確認する
noehlo	219	EHLO コマンドを使用しない
nomailfromdnsverify	222	MAIL FROM: コマンドに使用されているドメインが DNS に存在するかどうかを確認しない
nosendetrn	220	ETRN コマンドを発行しない
nosmtp	219	SMTP プロトコルをサポートしない。デフォルトでは、このキーワードが使用される
novrfy	221	VRFY コマンドを発行しない
sendetrn	220	ETRN コマンドを発行する
sevenbit	223	8 ビット文字をサポートしない。8 ビット文字はエンコードされなければならない
silentetrn	220	チャネル情報をエコーせずに ETRN コマンドを処理する
smtp	219	SMTP プロトコルをサポートする。smtp キーワードはすべての SMTP チャネルで必須 (このキーワードは smtp_crorlf と同等)
smtp_cr	219	ラインフィーダ (LF) なしの、キャリッジリターン (CR) のみが改行記号として受け入れられる
smtp_crlf	219	キャリッジリターン (CR) + ラインフィーダ (LF) のシーケンスのみがで改行記号として認識される
smtp_crorlf	219	キャリッジリターン (CR) 、ラインフィーダ (LF) のシーケンス、または完全な CRLF が改行記号として使用可能である
smtp_lf	219	キャリッジリターン (CR) なしの、ラインフィーダ (LF) のみを使用できる

表 8-2　機能別チャネルキーワード (**太字**はデフォルト) (続き)

キーワード	ページ	定義
streaming	224	チャネルに関連付けられたプロトコルのストリーミングの程度を制御する
vrfyallow	221	VRFY コマンドに対して詳細な情報を提供する応答を出す
vrfydefault	221	チャネルの HIDE_VERIFY オプションの設定に従い、VRFY コマンドに対してデフォルトの応答を提供する
vrfyhide	221	SMTP VRFY コマンドに対してあいまいな応答を出す
TCP/IP 接続および DNS 検索サポート (より大きい機能単位については 225 ページの表 8-5 を参照)		
cacheeverything	228	すべての接続情報をキャッシュする
cachefailures	228	接続失敗に関する情報だけをキャッシュする
cachesuccesses	228	接続成功に関する情報だけをキャッシュする
connectalias	249	受取人のアドレスに書かれているホストに配信する
connectcanonical	249	MTA が接続するシステムのホストエイリアスに接続する
daemon	232	エンベロープアドレスにかかわらず特定のホストシステムに接続する
defaultmx	230	チャネルが、ネットワークから MX 検索を実行するかどうかを決定する
defaultnameservers	231	TCP/IP スタックが選択したネームサーバを照合する
forwardcheckdelete	229	リバース DNS 検索のあとに正引き検索を行い、リバース DNS 検索で返された名前の正引き検索がオリジナルの接続の IP 番号に一致するかどうかを確認する。一致しない場合、リバース DNS 検索で返された名前は削除され、IP アドレスが使用される
forwardchecknone	229	DNS リバース検索のあとに正引き検索を実行しない
forwardchecktag	229	リバース DNS 検索が実行して返された名前を正引き検索して、IP 番号がオリジナルの接続の IP 番号に一致するかどうかを確認する。一致しなければ名前に「*」を付ける
identnone	229	IDENT 検索を実行しない。IP からホスト名への変換を実行し、Received: ヘッダーにホスト名と IP アドレスを含める
identnonelimited	229	IDENT 検索を実行しない。IP からホスト名への変換を実行し (ただしチャネルの切り替えを行う際にはホスト名を使用しない)、Received: ヘッダーにホスト名と IP アドレスの両方を含める
identnonenumeric	229	IDENT 検索および IP からホスト名への変換を実行しない
identnonesymbolic	229	IDENT 検索を実行しない。IP からホスト名への変換を実行し、Received: ヘッダーにホスト名だけを含める

表 8-2　機能別チャネルキーワード (**太字**はデフォルト) (続き)

キーワード	ページ	定義
identtcp	229	受信 SMTP 接続での IDENT 検索と IP からホスト名への変換を実行し、Received: ヘッダーにホスト名と IP アドレスの両方を含める
identtcplimited	229	受信 SMTP 接続での IDENT 検索と IP からホスト名への変換を実行する。チャネルの切り替えを行う際にはホスト名を使用しない。Received: ヘッダーにホスト名と IP アドレスを含める
indenttcpnumeric	229	受信 SMTP 接続で IDENT 検索を実行する。IP からホスト名への変換を実行しない
identtcpsymbolic	229	受信 SMTP 接続での IDENT 検索と IP からホスト名への変換を実行し、Received: ヘッダーにホスト名だけを含める
interfaceaddress	228	指定された TCP/IP インタフェースアドレスにバインドする
lastresort	231	最後のホストを指定する
mailfromdnsverify	222	MAIL FROM: コマンドに使用されているドメインが DNS に存在するかどうかを確認する
mx	230	TCP/IP ネットワークおよびソフトウェアが MX レコード検索をサポートする
nameservers	231	TCP/IP スタックが選択したネームサーバの代わりに照合するネームサーバのリストを指定する。nameservers には、空白文字で区切られたネームサーバの IP アドレスのリストが必要
nocache	228	接続情報をキャッシュしない
nomailfromdnsverify	222	MAIL FROM: コマンドに使用されているドメインが DNS に存在するかどうかを確認しない
nomx	230	TCP/IP ネットワークが MX 検索をサポートしない
nonrandomemx	230	MX 検索を実行するが、返されたエントリを同等の優先度でランダム化しない
port	228	SMTP 接続用のデフォルトポート番号を指定する。標準ポートは 25
randommx	230	MX 検索を実行し、返されたエントリを同等の優先度でランダム化する
single	232	チャネル上の各宛先アドレス用にメッセージのコピーが 1 つずつ作成されるように指定する
single_sys	232	各宛先システム用にメッセージのコピーを 1 つずつ作成する
threaddepth	243	マルチスレッド SMTP クライアントに対して新しいスレッドをトリガするために必要なメッセージ数

表 8-2　機能別チャネルキーワード (**太字**はデフォルト) (続き)

キーワード	ページ	定義
その他		
submit	271	チャネルを送信専用のチャネルに指定する
user	271	pipe チャネルでどのユーザ名で実行するかを示すのに使用される

チャネルのデフォルトを設定する

設定ファイルにはさまざまなチャネルキーワードが繰り返し記述されていることがあります。このような設定を管理するには時間がかかり、エラーの原因にもなります。複数のチャネルに対してまとめてデフォルトのキーワードを指定すると、設定を簡素化することができます。

たとえば、以下の行を設定ファイルに追加すると、行中で指定したキーワードがそれ以降のすべてのチャネルブロックに適用されます。

```
defaults keyword1 keyword2 keyword3 ...
```

defaults 行はチャネルを特定せずにデフォルトのキーワードを変更するための特殊なチャネルブロックだと考えられます。また、defaults 行にほかのチャネルブロック情報を指定する必要はありません (指定しても無視されます)。

1 つのファイルに使用できる defaults 行の数に上限はありません。複数の defaults 行を指定した場合、ファイルの下へ行くほど (あとで追加した行ほど) 優先度が高くなります。

設定ファイル内のある位置 (たとえば、外部ファイルのチャネルブロックの独立したセクションの冒頭など) 以降には無条件に defaults 行が適用されないように設定しておく方がよい場合もあります。そのためには、nodefaults 行を使用します。たとえば、以下の行を設定ファイルに挿入すると、それ以前の部分で defaults を使って指定した設定がすべて無効になり、defaults を使用していないのと同じ状態に戻ります。

```
nodefaults
```

ほかのチャネルブロックと同様に、defaults や nodefaults チャネルブロックを使用する場合も、ブロック間の区切りには空白行を使用します。設定ファイル内でローカルチャネルの前に記述できるチャネルブロックは、defaults と nodefaults のみです。ただし、ほかのチャネルブロックと同様、書き換え規則の前に記述することはできません。

SMTP チャネルを設定する

インストールの種類によっては、Messaging Server のインストール時に数種の SMTP チャネルが提供されます (以下の表を参照)。このようなチャネルは TCP/IP の上位プロトコルとして SMTP を実装します。マルチスレッド TCP SMTP チャネルには、ディスパッチャ制御下のマルチスレッド SMTP サーバが含まれる。送信された SMTP メールは、必要に応じてジョブコントローラの制御下で動作し、チャネルプログラム `tcp_smtp_client` によって処理されます。

表 8-3 SMTP チャネル

チャネル	定義
`tcp_local`	リモート SMTP ホストからのメールを受信する。メールを送信する場合は、スマートホスト / ファイアウォール設定が使われているかどうかによって、直接リモート SMTP ホストに送るか、またはスマートホストファイアウォールシステムに送る
`tcp_intranet`	イントラネット内のメールを送受信する
`tcp_auth`	tcp_local のスイッチチャネルとして使用される。認証されたユーザは、リレーブロックの制約を回避するため tcp_auth チャネルに移される
`tcp_submit`	送信されたメッセージ (通常の場合はユーザエージェントからのメッセージ) を予約されている送信ポート 587 で受け入る (RFC 2476 を参照)
`tcp_tas`	Unified Messaging を使用するサイト用の特殊な IA チャネル

この節で説明するチャネルキーワードを追加したり削除することで、これらのチャネルの定義を変更したり、新規チャネルを作成することも可能です。また、オプションファイルは、TCP/IP チャネルのさまざまな特徴を制御するために使用されます。このようなオプションファイルは、MTA 設定ディレクトリ (*ServerRoot*`/msg-`*instance*`/imta/config`) に保存し、*x_option* という名前を付けなければなりません。この「*x*」はチャネルの名前です。詳細については『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

この節には、以下の項があります。

- SMTP チャネルオプションを設定する
- SMTP コマンドとプロトコルのサポート
- TCP/IP 接続と DNS 検索のサポート

- SMTP 認証、SASL、TLS
- ヘッダー内の SMTP AUTH から認証済みアドレスを使用する
- ヘッダー内の SMTP AUTH から認証済みアドレスを使用する
- Microsoft Exchange ゲートウェイチャネルを指定する
- Transport Layer Security

SMTP チャネルオプションを設定する

TCP/IP チャネルオプションファイルは、TCP/IP チャネルのさまざまな特性を制御します。ファイルには *x*_option という名前を付けてください。ファイル名の *x* はチャネル名となります。たとえば、/*ServerInstance*/imta/config/tcp_local_option のようになります。

オプションファイルは、1 つまたは複数のキーワードとその関連値によって構成されています。たとえば、サーバのメーリングリストのエクスパンドを無効にするには、オプションファイルに DISABLE_EXPAND キーワードを追加し、値を 1 に設定します。

その他のオプションファイルキーワードを使用すると、以下の制御を行うことができます。

- メッセージ当たりの宛先数を制限する (ALLOW_RECIPIENTS_PER_TRANSACTION)
- 接続当たりのメッセージ数を制限する (ALLOW_TRANSACTIONS_PER_SESSION)
- MTA ログファイルに記録される情報のタイプを微調整する (LOG_CONNECTION、LOG_TRANSPORTINFO)
- クライアントチャネルプログラムが許可できる同時送信接続の最大数を指定する (MAX_CLIENT_THREADS)

チャネルオプションキーワードとシンタックスの詳細については、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

SMTP コマンドとプロトコルのサポート

SMTP チャネルが EHLO、ETRN、VRFY などの SMTP コマンドをサポートするように指定することができます。また、チャネルが DNS ドメイン確認をサポートするかどうかや、どの文字を改行記号として受け入れるかなどを指定することも可能です。この項では、以下の内容について説明します。

- チャネルプロトコル選択と改行記号
- EHLO コマンドのサポート

- ETRN コマンドのサポート
- VRFY コマンドのサポート
- DNS ドメイン確認
- 文字セットのラベルと 8 ビットデータ
- プロトコルストリーミング

表 8-4 に、この節で説明されているキーワードのリストを示します。

表 8-4 SMTP コマンドとプロトコルのキーワード

チャネルキーワード	説明
プロトコル選択と改行記号	**チャネルが SMTP プロトコルをサポートするかどうかを指定し、改行記号として受け入れる文字シーケンスを指定する**
smtp	SMTP プロトコルをサポートする。smtp キーワードはすべての SMTP チャネルで必須 (このキーワードは smtp_crorlf と同等)
nosmtp	SMTP プロトコルをサポートしない。デフォルトでは、このキーワードが使用される
smtp_cr	ラインフィーダ (LF) なしの、キャリッジリターン (CR) のみが改行記号として受け入れられる
smtp_crlf	キャリッジリターン (CR) + ラインフィーダ (LF) のシーケンスのみがで改行記号として認識される
smtp_lf	キャリッジリターン (CR) なしの、ラインフィーダ (LF) のみを使用できる
smtp_crorlf	キャリッジリターン (CR) 、ラインフィーダ (LF) のシーケンス、または完全な CRLF が改行記号として使用可能である
EHLO キーワード	**チャネルによる EHLO コマンドの処理方法を指定**
ehlo	初期接続に SMTP EHLO コマンドを使用する
checkehlo	SMTP 応答の見出しを確認して、EHLO と HELO のどちらを使用するか決定する
noehlo	EHLO コマンドを使用しない
ETRN キーワード	**チャネルによる ETRN コマンド (キュー処理の要求) の処理方法を指定する**
allowetrn	ETRN コマンドを処理する
blocketrn	ETRN コマンドをブロックする
domainetrn	ドメインを指定する ETRN コマンドだけを処理する

表 8-4　SMTP コマンドとプロトコルのキーワード (続き)

チャネルキーワード	説明
silentetrn	チャネル情報をエコーせずに ETRN コマンドを処理する
sendetrn	ETRN コマンドを発行する
nosendetrn	ETRN コマンドを発行しない
VRFY キーワード	**チャネルによる VRFY コマンドの処理方法を指定**
domainvrfy	完全なアドレスを使用して VRFY コマンドを発行する
localvrfy	ローカルアドレスを使用して VRFY コマンドを発行する
novrfy	VRFY コマンドを発行しない
vrfyallow	VRFY コマンドに対して詳細な情報を提供する応答を出す
vrfydefault	チャネルの HIDE_VERIFY オプションの設定に従い、VRFY コマンドに対してデフォルトの応答を提供する
vrfyhide	SMTP VRFY コマンドに対してあいまいな応答を出す
DNS ドメイン確認	**チャネルが DNS ドメイン確認を行うかどうかを指定**
mailfromdnsverify	MAIL FROM: コマンドに使用されているドメインが DNS に存在するかどうかを確認する
nomailfromdnsverify	MAIL FROM: コマンドに使用されているドメインが DNS に存在するかどうかを確認しない
文字セットと 8 ビットデータ	**チャネルによる 8 ビットデータの処理方法を指定する。 注 : これらのキーワードは主に SMTP チャネルで使用されるが、その他のチャネルで使用されることもある**
charset7	7 ビットのテキストメッセージに関連付けるデフォルトの文字セット
charset8	8 ビットのテキストメッセージに関連付けるデフォルトの文字セット
charsetesc	エスケープ文字を含む 7 ビットのテキストに関連付けるデフォルトの文字セット
eightbit	チャネルが 8 ビット文字をサポートする
eightnegotiate	チャネルが 8 ビット転送の使用をネゴシエートする (可能な場合)
eightstrict	チャネルがネゴシエーションが行われていない 8 ビットデータを含むメッセージを拒否するように指定する
sevenbit	チャネルは 8 ビット文字をサポートしない。8 ビット文字はエンコードされなければならない

表 8-4　SMTP コマンドとプロトコルのキーワード (続き)

チャネルキーワード	説明
プロトコルストリーミング	プロトコルストリーミングチャネルが使用するプロトコルストリーミングの程度を指定
streaming	チャネルに関連付けられたプロトコルのストリーミングの程度を制御する

チャネルプロトコル選択と改行記号

キーワード : smtp、nosmtp、smtp_crlf、smtp_cr、smtp_crorlf、smtp_lf

smtp および nosmtp キーワードは、チャネルが SMTP プロトコルをサポートするかどうかを指定するものです。smtp (またはその変形) は、すべての SMTP チャネルに対して必須のキーワードです。

smtp_crlf、smtp_cr、smtp_crorlf、および smtp_lf は、MTA が改行記号として受け入れる文字シーケンスの種類を指定するために、SMTP チャネルに対して使用されます。smtp_crlf キーワードを使用すると、キャリッジリターン (CR) + ラインフィーダ (LF) のシーケンスのみが改行記号として認識されます。smtp_lf または smtp キーワードでは、CR なしの LF のみを使用できます。また、smtp_cr を使用すると、CR のみのターミネータが受け入れられます。これらのオプションは、受信データにしか適用されません。

SMTP では 改行記号として CRLF が要求されるため、MTA は常に CRLF シーケンスを生成します。各種の smtp キーワードは、MTA がその他の非標準的な改行記号を受け入れるかどうかを指定するだけのものです。たとえば、MTA が規定どおりの SMTP メッセージだけを受け入れ、非標準的な改行記号を含むメッセージを拒否するように指定するには、stmp_crlf を使います。

EHLO コマンドのサポート

キーワード : ehlo、noehlo、checkehlo

SMTP プロトコルは、その他のコマンドの使用のネゴシエーションを行うことができるよう拡張されています (RFC 1869)。これを利用するには、RFC 821 規定の HELO コマンドの代わりに、新しい EHLO コマンドを使用します。EHLO コマンドを受け取った拡張 SMTP サーバはサポートする拡張内容のリストを返します。拡張をサポートしないサーバにこのコマンドを発行した場合は、不明なコマンドエラーのメッセージが返され、エラーメッセージを受け取ったクライアントは折り返し HELO コマンドを送ります。

このフォールバックは、サーバが拡張されているかどうかにかかわらず機能します。ただし、サーバが RFC 821 に準拠した SMTP を実装していない場合は、問題が発生する可能性があります。特に、認識できないコマンドを受け取ると接続を遮断してしまうサーバもあります。

EHLO コマンドを受け取ったサーバが接続を遮断した場合、SMTP クライアントは HELO コマンドを発行して再接続を試みます。ただし、EHLO を受け取ったリモートサーバが接続を遮断するだけでなく、その他の問題を併発する場合は、クライアントが再接続できないこともあります。

ehlo、noehlo、および checkehlo チャネルキーワードは、このような情況に対処するためのキーワードです。ehlo キーワードは、1 回目の接続試行に EHLO コマンドを使用するよう MTA に指示を出します。noehlo キーワードは EHLO コマンドの使用をすべて無効にします。checkehlo キーワードでは、リモート SMTP サーバから返された応答見出しに「ESMTP」文字列があるかどうかが確認されます。この文字列がある場合は EHLO が、ない場合は HELO が使用されます。デフォルトでは、最初の接続試行に対する応答の見出しに「fire away」文字列が含まれている場合は HELO を使用し、それ以外の場合は EHLO を使用するように設定されています。このデフォルト設定は ehlo キーワードと checkehlo キーワードの中間的な効果を得るものであり、この設定を指定するためのキーワードは存在しないことに注意してください。

ETRN コマンドのサポート

キーワード : allowetrn、blocketrn、disableetrn、domainetrn、silentetrn、sendetrn、nosendetrn、novrfy

RFC 1985 で規定されている ETRN コマンドは SMTP サービスの拡張を可能にするものです。このコマンドによって SMTP サーバがクライアントとの通信に基づいてメッセージキューの処理を開始し、指定のホストにメッセージを配信できるようになります。

SMTP クライアントは ETRN を使用して、自分宛てのメッセージキューの処理を開始するようリモート SMTP サーバに要求できます。つまり、ETRN は、自分のシステムに入ってくるメッセージのためにリモート SMTP システムをポーリングする方法を提供します。これは、一過性の接続しか持たないシステム間 (たとえば、ダイアルアップ以外の方法ではインターネットに接続できないサイト用に二次的な MX ホストとして設定されているサイトなど) に対して有用です。このコマンドを有効にすることで、ダイアルアップ接続を行うリモートサーバもメール配信の要求を送ることができるようになります。

SMTP クライアントは、SMTP ETRN コマンドラインでメッセージの送信先となるシステム名 (通常、その SMTP クライアントシステムの名前) を指定します。リモート SMTP サーバが ETRN コマンドをサポートする場合、サーバは指定のシステムに別途接続し、そのシステム宛てのメッセージの配信を開始するためのプロセスがトリガされます。

ETRN コマンドへの応答

allowetrn、blocketrn、domainetrn、および silentetrn キーワードは、SMTP クライアントが ETRN コマンドを発行して MTA キュー内のメッセージを配信するよう要求した際に、MTA がどのように対応するかを指定するキーワードです。

デフォルト設定では allowetrn キーワードが有効になっているため、MTA はすべての ETRN コマンドを処理します。MTA が ETRN コマンドを拒否するように指定するには、チャネル定義に blocketrn キーワードを使用します。

MTA がすべての ETRN コマンドに従い、かつドメインによって確認されたチャネル名をエコーしないように指定するには、silentetrn キーワードを使用します。ETRN コマンドがドメインを指定している場合にのみ MTA がそのコマンドを処理するように指定するには、domainetrn キーワードを使用します。また、このキーワードを使用すると、MTA はドメインによって確認されたチャネル名をエコーしません。

disableetrn では、ETRN コマンドに対するサポートが完全に無効となります。SMTP サーバで、ETRN はサポートされているコマンドとしてアドバタイズされません。

ETRN コマンドを送信する

sendetrn および nosendetrn チャネルキーワードは、MTA が SMTP 接続開始時に ETRN コマンドを送るかどうかを指定するためのものです。デフォルト設定では nosendetrn が有効になっているため、MTA は ETRN コマンドを送りません。リモート SMTP サーバが ETRN コマンドをサポートする場合にのみ MTA が ETRN を発行するように指定するには、sendetrn キーワードを使用します。sendetrn キーワードの後ろには、メッセージの配信先となるシステムの名前を記述する必要があります。

VRFY コマンドのサポート

キーワード : domainvrfy、localvrfy、vrfyallow、vrfydefault、vrfyhide

VRFY コマンドは、SMTP クライアントが特定のユーザ名に宛てられたメールが存在するかどうかを確認するよう SMTP サーバに要求するためのコマンドです。VRFY コマンドは、RFC 821 で定義されています。

サーバは、ユーザがローカルであるかどうか、メールが転送されるかどうかなどの情報を返します。250 という応答はユーザ名がローカルであることを意味し、251 はローカルではないがメッセージの転送は可能であることを意味します。サーバの応答には、メールボックス名が含まれます。

VRFY コマンドを送信する

通常の環境では、SMTP ダイアログの一部として VRFY コマンドを発行する必要はありません。SMTP RCPT TO コマンドに VRFY コマンドと同じ効果があり、必要に応じて適切なエラーを返すためです。ただし、サーバの中には、RCPT TO コマンドを受け取った場合にはコマンドが指定するアドレスをいったん受理してから返送し、VRFY コマンドを受け取った場合はより広範なチェックを実行するものもあります。

デフォルト設定では novrfy キーワードが有効になっているため、MTA は VRFY コマンドを発行しません。

MTA が SMTP VRFY コマンドを発行するように指定するには、チャネル定義に domainvrfy または localvrfy キーワードを挿入します。domainvrfy キーワードを使用すると、完全なアドレス (user@host) を引数とする VRFY コマンドが発行されます。localvrfy キーワードを使用すると、アドレスのローカル部分 (user) だけを引数とする VRFY コマンドが発行されます。

VRFY コマンドに応答する

vrfyallow、vrfydefault、および vrfyhide キーワードは、送信側の SMTP クライアントから SMTP VRFY コマンドを出したときの SMTP サーバの応答を制御します。

MTA が詳細な情報を含む応答を返すように指定するには、vrfyallow キーワードを使用します。HIDE_VERIFY=1 チャネルオプションが指定されていないかぎり、MTA が詳細な情報を含む応答を返すよう指定するには、vrfydefault キーワードを使用します。MTA があいまいな応答を返すよう指定するには、vrfyhide キーワードを使用します。これらのキーワードを使用すると、VRFY コマンドに対する応答をチャネルごとに制御できます。一方、HIDE_VERIFY オプションは、1 つの SMTP サーバを介して処理されるすべての受信 TCP/IP チャネルに適用されます。

DNS ドメイン確認

キーワード : mailfromdnsverify、nomailfromdnsverify

mailfromdnsverify を受信 TCP/IP チャネルに対して設定すると、MTA は SMTP MAIL FROM コマンドで指定されているドメインのエントリが DNS に存在するかどうかを確認し、エントリが存在しない場合にはメッセージを拒否します。デフォルト設定では nomailfromdnsverify が有効になっているため、この確認は行われません。ただし、返信アドレスに対して DNS 確認を行うと、許可されるべきメッセージも拒否されてしまう可能性があることに注意してください (たとえば、正規のサイトでもそのドメイン名がまだ登録されていない場合や、DNS が適切に動作していない場合など)。これは、RFC 1123 の「Requirements for Internet Hosts (インターネットホストの必要条件)」で規定されている電子メール受信の心得に反する行為です。ただし、存在しないドメインから不特定多数宛てのメール (UBE) が送られる場合は、この確認を行った方がよい場合もあります。

文字セットのラベルと 8 ビットデータ

キーワード : charset7、charset8、charsetesc、sevenbit、eightbit、eightnegotiate、eightstrict

文字セットのラベル

MIME 仕様は、プレーンテキストのメッセージで使用される文字セットにラベルを付けるしくみを提供します。特に、Content-type: ヘッダー行の一部として charset= パラメータを指定することができます。MIME には、US-ASCII (デフォルト)、ISO-8859-1、ISO-8859-2 のようなさまざまな文字セット名が定義されており、その後さらに定義されたものも多数あります。

既存のシステムやユーザエージェントの中には、これらの文字セットラベルを生成するしくみを提供しないものもあり、その結果、プレーンテキストメッセージの中には適切にラベル付けされていないものもあります。charset7、charset8、および charsetesc チャネルキーワードは、文字セットのラベルが欠如しているメッセージヘッダーに文字セット名を挿入するメカニズムをチャネルごとに提供するキーワードです。これらのキーワードを使用する場合は、単一の文字セット名を引数として指定する必要があります。文字セット名が正しいかどうかの確認は行われません。文字セットの変換は、MTA テーブルディレクトリ内の文字セット定義ファイル charsets.txt で定義されている文字セットに対してのみ可能であることに注意してください。できるだけこのファイルで定義されている名前を使用することをお勧めします。

メッセージに含まれるのが 7 ビットデータのみの場合は charset7 を、8 ビットデータが含まれる場合は charset8 を使用します。charsetesc は、メッセージに 7 ビットデータおよびエスケープ文字が含まれる場合に使用します。適切なキーワードが指定されていない場合は、Content-type: ヘッダー行には文字セット名が挿入されません。

charset8 キーワードでは、メッセージヘッダーの 8 ビット文字の MIME エンコーディングも制御されます (メッセージヘッダーでは、8 ビットのデータは常に不正です)。MTA では通常、メッセージヘッダーにあるすべての不正な 8 ビットデータが MIME でエンコードされ、charset8 の値が指定されていない場合は「UNKNOWN」文字セットとしてラベルされます。

これらの文字セット指定が既存のラベルより優先されることはありません。メッセージにすでに文字セットラベルが含まれている場合やメッセージがテキストでない場合、これらのキーワードは効果をもたらしません。通常、MTA のローカルチャネルは次のようにラベル付けされます。

```
l ... charset7 US-ASCII charset8 ISO-8859-1 ...
hostname
```

Content-type ヘッダーがメッセージにない場合は、このヘッダーが追加されます。また、このキーワードは、MIME-version: ヘッダー行がない場合にも追加します。

charsetesc キーワードは、特に日本語や韓国語の文字セットを使用し、エスケープ文字を含むラベルのないメッセージを受信するチャネルに便利です。

8 ビットデータ

127 (10 進) 以上の序数値を持つ文字の使用は制限される場合があります。特に、SMTP サーバの中には、高ビットを切り捨てるために 8 ビット領域の文字を含むメッセージの文字化けの原因となるものもあります。

Messaging Server は、そのようなメッセージを自動的にエンコードし、8 ビットデータがメッセージに直接表示されないようにする機能を備えています。特定のチャネルのキューに入れられるすべてのメッセージにエンコードを適用するには、sevenbit キーワードを使用します。そのような制約がない場合は、eightbit を使用します。

リモート SMTP サーバが 8 ビットをサポートすると明示していないかぎり、SMTP プロトコルは 8 ビットを許可しません。ただし、拡張 SMTP など、転送形式によっては、8 ビットの文字を転送できるかどうかを判断するためのネゴシエーション形式をサポートするものもあります。ネゴシエートが失敗した場合に備えて、eightnegotiate キーワードを使用し、チャネルがメッセージをエンコードするよう指定しておくことを強くお勧めします。デフォルト設定ではすべてのチャネルに対してこのキーワードが有効になっているため、ネゴシエーションをサポートしないチャネルは 8 ビットデータの転送が可能であるという仮定のもとに動作します。

Messaging Server がネゴシエーションされていない 8 ビットデータを含むメッセージをすべて拒否するように設定するには、eightstrict キーワードを使用します。

プロトコルストリーミング

キーワード : streaming

メールプロトコルによっては、ストリーミングをサポートするものもあります。ストリーミングがサポートされている場合は、MTA が一度に複数の要求を発行し、それぞれに対する応答をバッチで受け取ることができます。streaming は、チャネルに関連付けられたプロトコルのストリーミングの程度を制御するキーワードです。このキーワードには整数値のパラメータが必要です。パラメータの解釈は、プロトコルによって異なります。

通常の環境では、ストリーミングサポートが可能な範囲は SMTP パイプライン拡張でネゴシエートされます。このキーワードは、通常の環境で使用されることがありません。

ストリーミング値の範囲は 0 から 3 までです。値が 0 の場合はストリーミングが指定されず、値が 1 の場合は RCPT TO コマンドグループがストリーミングされ、2 の場合は MAIL FROM/RCPT TO が、3 の場合は HELO/MAIL FROM/RCPT TO または RSET/MAIL FROM/RCPT TO がストリーミングされます。デフォルト値は 0 です。

TCP/IP 接続と DNS 検索のサポート

サーバによる TCP/IP 接続およびアドレス検索の処理方法を指定することができます。この項では、以下の内容について説明します。

- TCP/IP ポート番号とインタフェースアドレス
- チャネル接続情報のキャッシング
- リバース DNS 検索
- IDENT 検索
- TCP/IP MX レコードのサポート
- ネームサーバ検索
- 最後のホスト
- 受信メール用代替チャネル (切り替えチャネル)
- ターゲットホストの選択

表 8-5 に、この項で説明されている TCP/IP 接続および DNS 検索に関連するキーワードのリストを示します。

表 8-5 TCP/IP 接続と DNS 検索のキーワード

チャネルキーワード	説明
ポート選択とインタフェースのアドレス	**SMTP 接続用のデフォルトポート番号とインタフェースのアドレスを指定する**
`port`	SMTP 接続用のデフォルトポート番号を指定する。標準ポートは 25
`interfaceaddress`	指定された TCP/IP インタフェースアドレスにバインドする
キャッシュキーワード	**接続情報のキャッシュ方法を指定**
`cacheeverything`	すべての接続情報をキャッシュする
`cachefailures`	接続失敗に関する情報だけをキャッシュする
`cachesuccesses`	接続成功に関する情報だけをキャッシュする

表 8-5　TCP/IP 接続と DNS 検索のキーワード (続き)

チャネルキーワード	説明
nocache	接続情報をキャッシュしない
リバース DNS 検索	**受信 SMTP 接続に対するリバース DNS 検索の処理方法を指定する**
forwardcheckdelete	リバース DNS 検索のあとに正引き検索を行い、リバース DNS 検索で返された名前の正引き検索がオリジナルの接続の IP 番号に一致するかどうかを確認する。一致しない場合、リバース DNS 検索で返された名前は削除され、IP アドレスが使用される
forwardchecknone	DNS リバース検索のあとに正引き検索を実行しない
forwardchecktag	リバース DNS 検索が実行して返された名前を正引き検索して、IP 番号がオリジナルの接続の IP 番号に一致するかどうかを確認する。一致しなければ名前に「*」を付ける
IDENT 検索 /DNS リバース検索	**受信 SMTP 接続に対する IDENT 検索および DNS リバース検索の処理方法を指定する**
identnone	IDENT 検索を実行しない。IP からホスト名への変換を実行し、Received: ヘッダーにホスト名と IP アドレスを含める
identnonelimited	IDENT 検索を実行しない。IP からホスト名への変換を実行し (ただしチャネルの切り替えを行う際にはホスト名を使用しない)、Received: ヘッダーにホスト名と IP アドレスを含める
identnonenumeric	IDENT 検索および IP からホスト名への変換を実行しない
identnonesymbolic	IDENT 検索を実行しない。IP からホスト名への変換を実行し、Received: ヘッダーにホスト名だけを含める
identtcp	受信 SMTP 接続での IDENT 検索と IP からホスト名への変換を実行し、Received: ヘッダーにホスト名と IP アドレスの両方を含める
identtcplimited	受信 SMTP 接続での IDENT 検索と IP からホスト名への変換を実行する。チャネルの切り替えを行う際にはホスト名を使用しない。Received: ヘッダーにホスト名と IP アドレスを含める
indenttcpnumeric	受信 SMTP 接続で IDENT 検索を実行する。IP からホスト名への変換を実行しない
identtcpsymbolic	受信 SMTP 接続での IDENT 検索と IP からホスト名への変換を実行し、Received: ヘッダーにホスト名だけを含める
MX レコードのサポートと TCP/IP ネームサーバ	**チャネルが MX レコード検索をサポートするかどうか、およびどのように処理するかを指定する**

表 8-5　TCP/IP 接続と DNS 検索のキーワード (続き)

チャネルキーワード	説明
mx	TCP/IP ネットワークおよびソフトウェアが MX レコード検索をサポートする
nomx	TCP/IP ネットワークが MX 検索をサポートしない
defaultmx	チャネルが、ネットワークから MX 検索を実行するかどうかを決定する
randommx	MX 検索を実行し、返されたエントリを同等の優先度でランダム化する
nonrandomemx	MX 検索を実行するが、返されたエントリを同等の優先度でランダム化しない
nameservers	TCP/IP スタックが選択したネームサーバの代わりに照合するネームサーバのリストを指定する。nameservers には、空白文字で区切られたネームサーバの IP アドレスのリストが必要
defaultnameservers	TCP/IP スタックが選択したネームサーバを照合する
lastresort	最後のホストを指定する
switch キーワード	**メールを受信する代替チャネルのリストを制御**
allowswitchchannel	switchchannel チャネルからこのチャネルへの切り替えを許可する
noswitchchannel	サーバチャネルの使用を継続し、送信元ホストに関連付けられているチャネルに切り替えない。また、ほかのチャネルからこのチャネルへの切り替えを許可しない
switchchannel	サーバチャネルから送信元のホストに関連付けられたチャネルに切り替える
tlsswitchchannel	TLS のネゴシエートが成功した場合に、ほかのチャネルに切り替える
saslswitchchannel	SASL 認証が成功した場合にほかのチャネルへ切り替える
ターゲットホストの選択とメッセージコピーのストレージ	**ターゲットホストシステムと、メッセージコピーのストレージ方法を指定する**
daemon	エンベロープアドレスにかかわらず特定のホストシステムに接続する
single	チャネル上の各宛先アドレス用にメッセージのコピーが 1 つずつ作成されるように指定する
single_sys	各宛先システム用にメッセージのコピーを 1 つずつ作成する

TCP/IP ポート番号とインタフェースアドレス

キーワード : port、interfaceaddress

通常、SMTP 実装 TCP/IP チャネルは、ポート 25 に接続してメッセージを送信します。SMTP 実装 TCP/IP チャネルがその他のポートを使用するように指定するには、port キーワードを使用します。このキーワードは、PORT ディスパッチャオプション (SMTP 接続を受け入れるために MTA がリッスンするポートを制御するオプション) を補足するものです。

interfaceaddress キーワードは、TCP/IP チャネルが送信時にソースアドレスとしてバインドするアドレスを制御します。つまり、複数のインタフェースアドレスが存在するシステム上で、MTA が SMTP メッセージを送信する際にどのアドレスをソース IP アドレスとして使用するかを制御するキーワードです。このキーワードは、INTERFACE_ADDRESS ディスパッチャオプション (接続およびメッセージを受け入れるために TCP/IP チャネルがリッスンするインタフェースアドレスを制御するオプション) を補足するものです。

チャネル接続情報のキャッシング

キーワード : cacheeverything、nocache、cachefailures、cachesuccesses

SMTP プロトコルを使用するチャネルは、過去の接続試行の履歴を含むキャッシュを管理しています。このキャッシュは、アクセスできないホストに繰り返し接続しようとして時間を浪費し、ほかのメッセージの配信が遅延されることを回避するために使用されます。このキャッシュは送信 SMTP チャネルが動作中の間のみ維持され、動作が終了するたびに削除されます。

通常、キャッシュには、成功した接続試行と失敗した接続試行の両方に関する情報が記録されます (成功した試行は、その後失敗する試行を相殺するために記録されます。すなわち、一度接続に成功したホストがその後失敗しても、はじめて試行する接続や以前失敗した接続ほど次の接続試行が遅れることはありません)。

ただし、MTA が使用するキャッシング方法がすべての状況に適しているというわけではありません。そこで、チャネルキーワードを使用して MTA キャッシュを調整します。

cacheeverything キーワードは、すべての形式のキャッシングを有効にします。デフォルト設定ではこのキーワードが使用されます。nocache キーワードは、すべてのキャッシングを無効にします。

cachefailures キーワードは、失敗した接続のキャッシングだけを有効にします。このキーワードを使用すると、次の試行は cacheeverything を使用した場合より多くの制約を受けることになります。cachesuccesses は成功した接続だけをキャッシュします。このキーワードは、SMTP チャネルに対する nocache キーワードと同等のものです。

リバース DNS 検索

キーワード : forwardchecknone、forwardchecktag、forwardcheckdelete

forwardchecknone、forwardchecktag、および forwardcheckdelete チャネルキーワードは、リバース DNS 検索の影響を修正します。これらのキーワードは、MTA が DNS リバース検索によって検出された IP 名の正引き検索を実行するかどうか、および実行する場合には正引き検索の結果がオリジナルの接続の IP 番号と一致しなかった場合にどのように対処するかを制御します。

デフォルト設定では forwardchecknone キーワードが有効になっているため、正引き検索は実行されません。forwardchecktag キーワードは、リバース検索が行われるたびに正引き検索を実行し、検出された番号が最初の接続の番号と一致しない場合は IP 名にアスタリスク (*) を付けるように指定します。forwardcheckdelete キーワードは、リバース検索が行われるたびに正引き検索を行い、リバース検索で返された名前の正引き検索がオリジナルの接続の IP アドレスに一致しなかった場合はリバース検索で返された名前を無視 (削除) するように、MTA に指示します。

注 複数の IP アドレスに「一般的な」IP 名が使用されているサイトの場合、正引きの結果が最初の IP アドレスと一致しないのは比較的頻繁に見られる現象です。

IDENT 検索

キーワード : identnone、identnonelimited、identtnonnumeric、identnonesymbolic、identtcp、identtcpnumeric、identtcpsymbolic、identtcplimited

IDENT キーワードは、MTA が IDENT プロトコルを使用して接続や検索を処理する方法を制御します。IDENT プロトコルは、RFC 1413 で規定されています。

identtcp、identtcpsymbolic、および identtcpnumeric キーワードは、MTA が接続や検索に IDENT プロトコルを使用するように指定するものです。IDENT プロトコルから入手した情報 (通常、SMTP 接続を使用しているユーザの ID) は、次のようにメッセージの Received: ヘッダー行に挿入されます。

- identtcp は受信した IP 番号に呼応するホスト名 (DNS リバース検索で検出された名前) および IP 番号そのものを挿入します。
- identtcpsymbolic は受信した IP 番号に呼応するホスト名 (リバース DNS 検索で検出された名前) を挿入します。IP 番号そのものは Received: ヘッダーに含まれません。
- identtcpnumeric は受信した IP 番号を挿入します。リバース DNS 検索は実行されません。

注	identtcp、identtcpsymbolic、または identtcpnumeric による IDENT 検索が役に立つのは、リモートシステムで IDENT サーバが稼働している場合です。

IDENT クエリの試行でパフォーマンスヒットが発生する場合があります。そうすると、ルーターは認識できないポートへの接続試行を次第に「ブラックホール化」するようになります。IDENT 検索でこのような状況が発生した場合は、接続がタイムアウトするまで MTA には応答が返されません (通常、このタイムアウトは TCP/IP スタックが制御するもので、1、2 分ほどかかります)。

別のパフォーマンスの問題が、identtcp、indenttcplimited、あるいは identtcpsymbolic を identtcpnumeric とを比較するときにも発生します。identtcp、identtcplimited、または identtcpsymbolic によって DNS リバース検索が実行された場合、よりユーザフレンドリーなホスト名を返すにはより長い時間が必要になります。

identnone キーワードは IDENT 検索を無効にしますが IP からホスト名への変換は行われます。メッセージの Received: ヘッダーには IP 番号とホスト名が含まれます。デフォルトでは、このキーワードが使用されます。

identnonesymbolic キーワードは IDENT 検索を無効にしますが、IP からホスト名への変換は行われます。メッセージの Received: ヘッダーにはホスト名だけが含まれます。

identnonenumeric キーワードは IDENT 検索を無効にし、DNS リバース検索の IP 番号からホスト名への変換を禁止します。また、Received: ヘッダーにユーザフレンドリーでないホスト名を使用するため、パフォーマンスの向上につながる可能性もあります。

identtcplimited および identnonelimited キーワードは、IDENT 検索、リバース DNS 検索、Received: ヘッダーに表示する情報などに関し、identtcp および identnone と同様の効果をもたらします。ただし、異なる点として、identtcplimited および identnonelimited の場合は、switchchannel キーワードの影響で、DNS リバース検索によってホスト名が検出されたかどうかにかかわらず常に IP リテラルアドレスがチャネルスイッチのベースとして使用されます。

TCP/IP MX レコードのサポート

キーワード : mx、nomx、defaultmx、randommx、nonrandommx

TCP/IP ネットワークには、MX (メール転送) レコードの使用をサポートするものとしないものとがあります。MTA システムの接続先であるネットワークから提供される MX レコードだけを使用するように設定できる TCP/IP チャネルプログラムもあります。mx、nomx、defaultmx、randommx、nonrandommx キーワードは MX レコードのサポートを制御します。

randommx キーワードは、MX 検索を実行し、同等の優先順位を持つ MX レコード値を順不同で処理するように指定するものです。nonrandommx キーワードは、MX 検索を実行し、同等の優先順位を持つ MX レコード値を受信したとおりの順番で処理するように指定するものです。

現在のところ、mx キーワードは nonrandommx キーワードと同じものですが、将来のリリースでは randommx と同じになるように変更される可能性もあります。nomx キーワードは MX 検索を無効にします。defaultmx キーワードは、ネットワークが MX レコードをサポートする場合に mx を使用するように指定します。MX 検索をサポートするチャネルではすべて defaultmx キーワードがデフォルトとして設定されています。

ネームサーバ検索

キーワード : nameservers、defaultnameservers

ネームサーバ検索が実行される際、TCP/IP スタックが選択したネームサーバの代わりに nameservers チャネルキーワードを使ってネームサーバのリストを指定することができます。nameservers キーワードには、空白文字で区切られたネームサーバの IP アドレスのリストが必要です。以下の例を参照してください。

```
nameservers 1.2.3.1 1.2.3.2
```

デフォルト設定では defaultnameservers が有効になっているため、TCP/IP スタックの選択によるネームサーバが使用されます。

UNIX でネームサーバ検索を禁止するには、nsswitch.conf ファイルを編集します。NT の場合は、TCP/IP 設定を変更します。

最後のホスト

キーワード : lastresort

lastresort キーワードは、「最後のホスト」つまりほかのホストへの接続試行がすべて失敗した場合に最終的な接続先となるホストを指定します。このキーワードは、事実上の最終手段的 MX レコードとして動作します。このキーワードは、SMTP チャネルに対してのみ効果があります。

このキーワードでは、「最終手段的システム」の名前を指定する単一のパラメータが必要です。たとえば、以下のようになります。

```
tcp_local single_sys smtp mx lastresort mailhub.siroe.com
TCP-DAEMON
```

受信メール用代替チャネル (切り替えチャネル)

キーワード : switchchannel、allowswitchchannel、noswitchchannel
233 ページの「saslswitchchannel」および 235 ページの「tlsswitchchannel」も参照してください。

次の各キーワード switchchannel、allowswitchchannel、および noswitchchannel は受信メール用代替チャネルの選択を制御するものです。

MTA がリモートシステムから受信接続を受け付ける場合、MTA はその接続に関連付けるチャネルを選ぶ必要があります。通常、使用するチャネルは転送形式に基づいて決定されます。たとえば、TCP/IP を介する受信 SMTP 接続は、自動的に tcp_local チャネルに関連付けられます。

ただし、異なる性質を持つ複数の送信チャネルが複数のシステムに対して同時に使用される場合は、受信と送信がそれぞれ異なるチャネルで行われるため、対応するチャネルの性質がリモートシステムに関連付けられません。

この問題は、switchchannel キーワードを使用することにより解決できます。サーバが最初に使用するチャネルに switchchannel を指定すると、送信元ホストの IP アドレスがチャネルテーブルに照合され、一致した場合はソースチャネルがそれに合わせて切り替えられます。一致するものがない場合、または最初のデフォルト受信チャネルに一致するものが検出された場合は、MTA が リバース DNS 検索によって検出したホスト名に一致するエントリを見つけようと試みる場合もあります。ソースチャネルは switchchannel または allowswitchchannel にマークされているチャネルに切り替えられます (デフォルト)。noswitchchannel キーワードは、チャネルの切り替えを行わないように指定するためのものです。

デフォルトでは、サーバが関連付けられているチャネル以外のチャネルに switchchannel を使用しても効果はありません。現在のところ、switchchannel を使用できるのは SMTP チャネルに対してのみですが、いずれにしても SMTP チャネル以外に switchchannel を使用すべきではありません。

ターゲットホストの選択

キーワード : daemon、single、single_sys

daemon キーワードの解釈と使用は、適用するチャネルの種類によって異なります。

daemon キーワードは、SMTP チャネル上でターゲットホストの選択を制御するために使用します。

通常、ホストへの接続に使用されているチャネルは、メッセージのエンベロープアドレスに表示されます。daemon キーワードは、エンベロープアドレスにどのチャネルが表示されているかにかかわらず、チャネルがファイヤウォールやメールハブシステムなど特定のリモートシステムに接続するように設定します。実際のリモートシステム名は、以下の例に示すように daemon キーワードの直後に表示されます。次に例を示します。

```
tcp_firewall smtp mx daemon firewall.acme.com
TCP-DAEMON
```

daemon キーワードの後ろの引数が完全なドメイン名ではない場合、引数は無視され、チャネルは正規ホストに接続します。ファイヤウォールやゲートウェイシステムを正規ホスト名として指定する場合、以下の例に示すように daemon キーワードに与えられる引数は、一般的にルーターとして指定されます。

```
tcp_firewall smtp mx daemon router
firewall.acme.com
TCP-DAEMON
```

また、関連するキーワードとして、single および single_sys があります。single キーワードは、各宛先アドレス用にメッセージのコピーを 1 つずつ作成するように指定します。single_sys キーワードは、各宛先システム用にメッセージのコピーを 1 つずつ作成します。どのキーワードを使用しても、メッセージがキューに入れられる各チャネルごとに最低 1 つずつメッセージのコピーが作成されることに注意してください。

SMTP 認証、SASL、TLS

キーワード : maysaslserver、mustsaslserver、nosasl、nosaslserver、saslswitchchannel、nosaslswitchchannel)

Messaging Server が SASL (Simple Authentication and SecurityLayer) を使用した SMTP サーバの認証をサポートするかどうかを指定できます。SASL は RFC 2222 で定義されています。SASL、SMTP 認証、セキュリティの詳細については、第 12 章「セキュリティとアクセス制御を設定する」を参照してください。

maysaslserver、mustsaslserver、nosasl、nosaslserver, itchchannel、および saslswitchchannel チャネルキーワードは、SMTP プロトコルが使用される際に、TCP/IP チャネルなどの SMTP チャネルによって SASL (SMTP AUTH) が使用されるように設定するためのものです。

デフォルト設定では nosasl が有効になっているため、SASL 認証は許可または試行されません。このキーワードは nosaslserver を包括するため、SASL 認証の使用はすべて禁止されます。maysaslserver を指定すると、SMTP サーバは、クライアントが SASL 認証の使用を試行することを許可します。mustsaslserver を指定すると、SMTP サーバは、クライアントが SASL 認証を使用することを要求します。SMTP サーバは、リモートクライアントが認証に成功しないかぎり、メッセージを受け付けません。

クライアントが SASL の使用に成功したときに受信接続を指定のチャネルに切り替えるには、saslswitchchannel を使います。このキーワードには、切り替え先のチャネルを指定する必要があります。

ヘッダー内の SMTP AUTH から認証済みアドレスを使用する

キーワード : authrewrite

MTA が認証された差出人の情報をヘッダーに含めるようにするために、authrewrite チャネルキーワードをソースチャネルに使用することもできます。FROM_ACCESS マッピングによって無視されることもありますが、通常は SMTP AUTH 情報が使用されます。表 8-6 にあるように、authrewrite キーワードは必須の整数値をとります。

表 8-6　authrewrite の整数値

値	使用目的
1	AUTH 差出人を含む Resent-from: や Resent-sender: がすでに存在していれば、Sender: ヘッダーまたは Resent-sender: ヘッダーを追加する
2	AUTH 差出人を含む Sender: ヘッダーを追加する

Microsoft Exchange ゲートウェイチャネルを指定する

キーワード : msexchange、nomsexchange

msexchange チャネルキーワードは TCP/IP チャネルで使用して、MTA にこれが Microsoft Exchange ゲートウェイとクライアントとの通信を行うチャネルであることを指示できます。SASL 対応の (maysaslserver キーワード、または mustsaslserver キーワードを使用) 受信 TCP/IP チャネルで配置されると、MTA の

SMTP サーバが「不正な」形式 (オリジナルの ESMTP AUTH 仕様に基づく。この仕様は新しい適切な AUTH 仕様ではなく、適切な ESMTP 形式と互換性を持たない) を使って AUTH をアドバタイズするようになります。たとえば、Microsoft Exchange クライアントの中には、適切な AUTH 形式を認識せず、不正な AUTH 形式のみを認識するものがあります。

msexchange チャネルキーワードでも、破損した TLS コマンドをアドバタイズ (および認識) するようになります。

デフォルトは nomsexchange です。

Transport Layer Security

キーワード : maytls、maytlsclient、maytlsserver、musttls、musttlsclient、musttlsserver、notls、notlsclient、notlsserver、tlsswitchchannel

maytls、maytlsclient、maytlsserver、musttls、musttlsclient、musttlsserver、notls、notlsclient、notlsserver、および tlsswitchchannel チャネルキーワードは、TCP/IP チャネルなどの SMTP ベースのチャネルが SMTP プロトコルを使用するときに TLS をどのように処理するかを設定するためのキーワードです。

デフォルト設定では notls が有効になっているため、TLS は許可または試行されません。このキーワードは notlsclient (MTA SMTP クライアントは送信接続に TLS を使用しない。送信接続時に STARTTLS コマンドは発行されない) および notlsserver (MTA SMTP サーバは受信接続時に TLS の使用を許可しない。SMTP サーバもコマンド自体も STARTTLS 拡張に通知しない) を包括しています。

maytls が設定されている場合、MTA は TLS 使用の接続を受け入れ、送信接続にも TLS を使用しようと試みます。このキーワードは、maytlsclient (メッセージを送信する際に TLS をサポートする SMTP サーバに送信するのであれば、MTA SMTP クライアントは TLS を使用する) および maytlsserver (MTA SMTP サーバが STARTTLS 拡張をサポートすることを通知し、メッセージを受信する際に TLS を使用できる) を包括しています。

musttls キーワードを指定すると、MTA がは送受信接続に必ず TLS を使用します。TLS の使用をネゴシエーションを行うことができなかったリモートシステムとの電子メールの交換は許可されません。このキーワードは、musttlsclient (MTA SMTP クライアントはメッセージの送信に必ず TLS を使用し、TLS の使用のネゴシエーションが成功しない SMTP サーバにはメッセージを送らない。MTA 発行の STARTTLS コマ

ンドは必ず成功しなければならない) および musttlsserver (MTA SMTP サーバが STARTTLS 拡張をサポートすることを通知し、TLS 使用のメッセージを受け入れる際には必ず TLS を使用する。TLS の使用のネゴシエーションが成功しないクライアントからのメッセージは拒否される) を包括しています。

tlsswitchchannel キーワードは、クライアントが TSL 使用のネゴシエートに成功した場合、受信した接続を指定のチャネルに切り替えるためのキーワードです。このキーワードには、切り替え先のチャネルを指定する必要があります。

メッセージの処理と配信を設定する

サーバが特定の条件に基づいてメッセージの配信を試みるように指定できます。また、サービスジョブの処理制限や、新しい SMTP チャネルスレッドを作成するタイミングなど、ジョブ処理に関するパラメータを指定することも可能です。この項では、以下の内容について説明します。

- 238 ページの「チャネルの方向性を設定する」
- 238 ページの「指定配信日を実行する」
- 239 ページの「配信失敗メッセージの再配信回数を指定する」
- 240 ページの「チャネル実行ジョブのプールを処理する」
- 241 ページの「サービスジョブの制限」
- 243 ページの「サイズに基づくメッセージの優先度」
- 243 ページの「SMTP チャネルスレッド」
- 244 ページの「複数アドレスの拡張」
- 245 ページの「サービス変換を有効にする」

メッセージの処理と配信の詳細については、108 ページの「ジョブコントローラ」および 135 ページの「ジョブコントローラファイル」を参照してください。

表 8-7 に、この節で説明されているキーワードのリストを示します。

表 8-7 メッセージの処理と配信のキーワード

キーワード	定義
即時配信	**メッセージの即時配信に関する設定を定義**
immnonurgent	優先度にかかわらず、送信後すべてのメッセージの配信を即座に開始する
遅延配信	**遅延ジョブの配信に関する設定を定義**

表 8-7　メッセージの処理と配信のキーワード (続き)

キーワード	定義
backoff	遅延メッセージの配信試行頻度を指定する。normalbackoff、nonurgentbackoff、urgentbackoff で置き換え可能
deferred	Deferred-delivery: ヘッダー行の認識と処理を行う
nodeferred	デフォルト。Deferred-delivery: ヘッダー行が許可されないように指定する
nonurgentbackoff	優先度が低いメッセージの配信試行頻度
normalbackoff	優先度が標準であるメッセージの配信試行頻度
urgentbackoff	優先度が高いメッセージの配信試行頻度
サイズに基づくメッセージの優先度	**サイズに基づいてメッセージの優先度を定義**
nonurgentblocklimit	指定値以上のサイズを持つメッセージの優先度を「低」以下 (2 番目の優先度) に設定する。該当するメッセージは次の定期ジョブまで処理されない
normalblocklimit	指定値以上のサイズを持つメッセージの優先度を「低」に設定する
urgentblocklimit	指定値以上のサイズを持つメッセージの優先度を「標準」に設定する
チャネル実行ジョブの処理プール	**優先度やジョブ期日が異なる処理プールを指定する**
pool	チャネルが動作するプールを指定する
after	チャネルが動作するまでの遅延時間を指定する
サービスジョブの制限	**サービスジョブ数、および 1 つのジョブで処理できるメッセージファイル数を指定する**
maxjobs	1 つのチャネルに対して同時実行できるジョブの最大数を指定する
filesperjob	1 つのジョブで処理できるキューエントリの数を指定する
SMTP チャネルスレッド	
threaddepth	マルチスレッド SMTP クライアントに対して新しいスレッドをトリガするために必要なメッセージ数
複数アドレス拡張	**複数の受取人を持つメッセージ処理を定義する**
expandlimit	アドレスの数がこの制限を超えた場合、受信メッセージを「オフライン」で処理する

表 8-7　メッセージの処理と配信のキーワード (続き)

キーワード	定義
expandchannel	expandlimit の適用による遅延拡張を実行するチャネルを指定する
holdlimit	アドレスの数がこの制限を越えた場合、受信メッセージを保留する
配信不能メッセージ通知	**配信不能メッセージ通知を送るタイミングを指定**
notices	通知を送り、メッセージを返すまでの時間を指定する
nonurgentnotices	優先度が低いメッセージを配信できない場合に通知を送り、そのメッセージを返送するまでの時間を指定する
normalnotices	優先度が標準のメッセージを配信できない場合に通知を送り、そのメッセージを返送するまでの時間を指定する
urgentnotices	優先度が高いメッセージを配信できない場合に通知を送り、そのメッセージを返送するまでの時間を指定する

チャネルの方向性を設定する

キーワード : master、slave、bidirectional

チャネルを処理するプログラムは、マスタープログラム (master)、スレーブプログラム (slave)、あるいは両方のプログラム (bidirectional) という 3 つのキーワードで指定されます。これらのどのキーワードも指定されていない場合のデフォルトは bidirectional です。これらのキーワードによって、チャネルのキューにメッセージが入れられたときに MTA が配信活動を開始するかどうかが決まります。

これらのキーワードを使用すると、対応するチャネルプログラムの特徴が反映されるようになります。これらのキーワードをいつ、どこで使用すべきかについては、MTA がサポートする各種チャネルの説明を参照してください。

指定配信日を実行する

キーワード : deferred、nodeferred

deferred チャネルキーワードは、Deferred-delivery: ヘッダー行の認識と処理を行います。未来の deferred 指定配信日が付いているメッセージは、有効期限が切れて返されるか、あるいは指定配信日がくるまでチャネルのキューに保管されます。Deferred-delivery: ヘッダー行の形式と操作の詳細については、RFC 1327 を参照してください。

デフォルトのキーワードは nodeferred です。RFC 1327 では配信日指定によるメッセージ処理のサポートが義務付けられていますが、実際にそれを効果的に行えば、人々がディスク制限容量の拡張手段としてメールシステムを使用できるようになります。

配信失敗メッセージの再配信回数を指定する

キーワード : backoff、nonurgentbackoff、normalbackoff、urgentbackoff、notices

デフォルトでは、配信に失敗したメッセージの再配信回数はメッセージの優先度によって異なります。以下にデフォルトの再配信間隔を分単位で示します。優先度に続いて数字が示されていますが、最初の数字は最初に配信に失敗してから再配信を試みるまでの時間 (分) です。

優先度が高い : 30、60、60、120、120、120、240
優先度が標準 : 60、120、120、240、240、240、480
優先度が低い : 120、240、240、480、480、480、960

高い優先度では、最初の失敗から 30 分後に最初の再配信を試み、最初の再配信から 60 分後に 2 回目の再配信を、2 回目の再配信から 120 分後に 3 回目の再配信を試みます。最後に示した配信後は同じ間隔で再配信が試みられます。優先度が高いメッセージの場合では 240 分ごとに再配信が試みられます。

再配信が行われるのは、notices、nonurgentnotices、normalnotices、または urgentnotices キーワードで指定された期間内です。期間内に配信が成功しなければ、配信失敗通知が作成され、メッセージは差出人に返送されます。notices キーワードの詳細については、156 ページの「通知メッセージの配信間隔を設定するには」を参照してください。

backoff キーワードを使うと、優先度ごとにメッセージ再配信間隔を設定することができます。nonurgentbackoff は優先度が低いメッセージの再配信間隔を指定します。normalbackoff は優先度が標準のメッセージの再配信間隔を指定します。urgentbackoff は優先度が高いメッセージの再配信間隔を指定します。backoff のどのキーワードも指定されていなければ、優先度とは無関係に再配信間隔が指定されます。

次に例を示します。

```
urgentbackoff "pt30m" "pt1h" "pt2h" "pt3h" "pt4h" "pt5h" "pt8h"
"pt16h"
```

これは優先度の高いメッセージの再配信の場合です。最初の配信失敗から 30 分後に最初の再配信を試み、その 1 時間後 (最初の失敗から 1 時間半後) に 2 回目の再配信、2 時間後に 3 回目、3 時間後に 4 回目、4 時間後に 5 回目、5 時間後に 6 回目、8 時間後に 7 回目、16 時間後に 8 回目の再配信をそれぞれ試みます。その後は notices キーワードで指定した期間内まで 16 時間ごとに再配信を試みます。期間内に配信が成功しなければ、配信失敗通知が作成され、メッセージは差出人に返送されます。間隔のシンタックスは ISO 8601P に記述されており、『iPlanet Messaging Server リファレンスマニュアル』でも説明されています。

次に、優先度が標準のメッセージの例を示します。

```
normalbackoff "pt30m" "pt1h" "pt8h" "p1d" "p2d" "p1w"
```

最初の配信失敗から 30 分後に最初の再配信を試み、その 1 時間後に 2 回目の再配信、8 時間後に 3 回目、1 日後に 4 回目、2 日後に 5 回目、1 週間後に 6 回目の再配信をそれぞれ試みます。その後は notices キーワードで指定した期間内まで毎週、再配信を試みます。期間内に配信が成功しなければ、配信失敗通知が作成され、メッセージは差出人に返送されます。

最後に、優先度によらない、すべての配信失敗メッセージの例を示します。

```
backoff "pt30m" "pt120m" "pt16h" "pt36h" "p3d"
```

nonurgentbackoff、normalbackoff、および urgentbackoff で置き換えなければ、どのメッセージも、最初の配信失敗から 30 分後に最初の再配信を試み、その 120 分後に 2 回目の再配信、16 時間後に 3 回目、36 時間後に 4 回目、3 日後に 5 回目の再配信をそれぞれ試みます。その後は notices キーワードで指定した期間内まで 3 日ごとに再配信を試みます。期間内に配信が成功しなければ、配信失敗通知が作成され、メッセージは差出人に返送されます。

チャネル実行ジョブのプールを処理する

キーワード : pool

複数のチャネルが 1 つのプール内で動作するように設定すると、複数のチャネルが同じプールのリソースを共有できるようになります。特定のチャネル専用に指定されているプール内でほかのチャネルが動作するように設定することも可能です。各プール内のメッセージは優先度に基づいて自動的に適切な処理キューに割り当てられます。優先度の高いメッセージは優先度が低いメッセージよりも先に処理されます。(243 ページの「サイズに基づくメッセージの優先度」を参照)

pool キーワードを使用すると、ジョブが作成されるプールをチャネルごとに指定できます。pool キーワードの後ろには、現在のチャネルの配信ジョブのプール先となるプール名を指定する必要があります。プール名の長さの上限は 12 バイトです。

ジョブコントローラの概念と設定については、135 ページの「ジョブコントローラファイル」、108 ページの「ジョブコントローラ」、および 241 ページの「サービスジョブの制限」を参照してください。

サービスジョブの制限

キーワード : maxjobs、filesperjob

メッセージがチャネルキューに入れられるたびに、ジョブコントローラはメッセージを配信するためのジョブが実行されていることを確認します。これには、新規ジョブプロセスの開始、スレッドの追加、実行中のジョブの確認などの操作が含まれます。しかし、1 つのサービスジョブではすべてのメッセージを手際よく配信できない場合もあります。ジョブコントローラの概念と設定については、135 ページの「ジョブコントローラファイル」、240 ページの「チャネル実行ジョブのプールを処理する」、および 108 ページの「ジョブコントローラ」を参照してください。

メッセージ配信のために開始されるプロセスやスレッドの数には、妥当な制限があります。このプロセスやスレッド数の上限は、プロセッサの数、ディスクの速度、接続の性質などによって決定されます。MTA 設定ファイルでは、以下のものを制御することができます。

- 1 つのチャネルに対して開始できるプロセス数の上限 (maxjobs チャネルキーワード)
- 1 つのチャネルセットに対して開始できるプロセス数の上限 (ジョブコントローラ設定ファイルの該当するプールセクションに設定されている JOB_LIMIT パラメータ)
- 新しいスレッドまたはプロセスを開始する前に受信したキュー内のメッセージ数 (threaddepth チャネルキーワード)
- チャネルによっては、特定の配信プログラム内で実行するスレッド数の上限 (チャネル オプションファイル内の max_client_threads パラメータ)

1 つのチャネルに対して開始されるプロセス数の上限は、そのチャネルに対して設定されている maxjobs、またはチャネルが動作しているプールに対して設定されている JOB_LIMIT の最小値に当たります。

あるメッセージに処理が必要だとします。一般に、ジョブコントローラは次の場合に新しい処理を開始します。

- チャネルに対してプロセスが実行されておらず、プールのジョブ数が制限に達していない場合は、新しいプロセスを開始します。

- チャネルプログラムがシングルスレッドの場合、またはスレッド数が制限に達していて threaddepth で指定されている以上のバックログがあり、かつチャネルとプールのジョブ数がともに制限に達していない場合は、新しいプロセスを開始します。
- チャネルプログラムがマルチスレッドで、スレッド数が制限に達しておらず、かつ threaddepth で指定されている以上のバックログがある場合は、新しいスレッドが開始されます。

特に、SMTP チャネルに対しては、異なるホスト宛てのメッセージがキューに入るにつれて新しいスレッドやプロセスが開始されます。ジョブコントローラは、SMTP チャネルに対し、以下の基準に基づいて新しいプロセスを開始します。あるメッセージに処理が必要だとします。

- SMTP チャネルに対してプロセスが実行されておらず、プールが制限に達していない場合、ジョブコントローラは新しいプロセスを開始します。
- スレッド数が制限 (MAX_CLIENT_THREADS) に達していて、サービス待ち状態のホスト宛のメッセージがキューに入っており、チャネル数 (maxjobs) もプールジョブ (JOB_LIMIT) も制限に達していなければ、新しいプロセスが開始されます。
- スレッド数が制限に達しておらず、サービス待ち状態のホスト宛てのメッセージがキューに入った場合は、新しいスレッドが開始されます。
- スレッド数が制限に達しておらず、メッセージがキューに入ったためにそのホスト宛てのメッセージのバックログが threaddepth で指定されている以上の数になった場合は、新しいスレッドが開始されます。

243 ページの「SMTP チャネルスレッド」も参照してください。

filesperjob キーワードを使うと、MTA に追加のサービスジョブを作成するよう指示することもできます。このキーワードには、正の整数を 1 つパラメータとして設定する必要があります。この整数は、チャネルへ送られるべきキューエントリ (ファイル) の数を指定するもので、その後それらのファイルを処理するために複数のサービスジョブが作成されます。パラメータに 0 またはそれ以下の値を指定した場合は、1 つのサービスジョブだけがキューに入れられます。キーワードを指定しないと、パラメータの値は 0 に指定されます。このキーワードの影響は最大化されます。すなわち、算出された大きな方の数値が実際に作成されるサービスジョブの数となります。

filesperjob キーワードは、実際のキューエントリ (ファイル) 数を与えられた値で割って作成するジョブ数を算出します。各メッセージのキューエントリ数は、single や single_sys キーワード、メーリングリストのヘッダー修正アクション、そのほかさまざまな要素によって決定されます。

maxjobs キーワードは、同時実行可能な合計ジョブ数を制限します。このキーワードの後ろには、整数値を指定する必要があります。算出されたサービスジョブ数がこの値より大きい場合には、maxjobs ジョブだけが作成されます。maxjobs が使用されていない場合のデフォルト値は 100 に設定されています。通常、maxjobs には、そのチャネルが使用するプールまたはサービスプールで同時実行が可能な合計ジョブ数と同じ値、またはそれ以下の値を使用します。

サイズに基づくメッセージの優先度

キーワード : urgentblocklimit、normalblocklimit、nonurgentblocklimit

urgentblocklimit、normalblocklimit、および nonurgentblocklimit キーワードは、サイズに基づいてメッセージの優先度を下げるように MTA に指定するためのものです。これらのキーワードは、ジョブコントローラがメッセージ処理時に適用する優先度に影響を及ぼします。

SMTP チャネルスレッド

キーワード : threaddepth

マルチスレッドの SMTP クライアントは、メッセージを宛先ごとにそれぞれ異なるスレッドに割り当てるために、送信メッセージを並べ替えます。threaddepth キーワードは、マルチスレッドの SMTP クライアントが 1 つのスレッドに割り当てられるメッセージの数を制限し、それ以上のメッセージがある場合には別のスレッドに割り当てるよう指定します。通常、同じ宛先へのメッセージはすべて 1 つのスレッドによって処理されますが、このキーワードを指定すると、それらのメッセージが複数のスレッドによって処理されるようになります。

threaddepth キーワードは、チャネルの接続先の SMTP サーバが複数の接続を同時に処理できる場合に、デーモンルーター TCP/IP チャネル (ある特定の SMTP サーバに接続する TCP/IP チャネル) 上でマルチスレッドを確立する際に便利です。

チャネルに対するバックログが threaddepth で指定されている以上の数に達すると、ジョブコントローラはより多くのリソースをそのチャネルのキューにあるメッセージの処理に割り当てようとします。チャネルがマルチスレッドの場合、ジョブコントローラはメッセージを処理するジョブがそのチャネルに対して新しくスレッドを開始するように指示し、すべてのジョブのスレッド数がそのチャネルの制限に達している場合 (tcp_* チャネルの MAX_CLIENT_THREADS オプション) は、新しいプロセスを開始するように指示します。シングルスレッドのチャネルに対しては、新しいプロセスを開始するように指示します。ただし、チャネルのジョブ数 (maxjobs) またはプールのジョブ数 (JOB_LIMIT) が制限に達している場合、新しいジョブは開始されません。

複数アドレスの拡張

キーワード: expandlimit、expandchannel、holdlimit

大部分のチャネルは複数の宛先アドレスを持つメッセージを受け入れますが、1 つのメッセージに複数の宛先アドレスが指定されていると、配信処理に遅延 (オンライン遅延) が生じます。遅延時間が長いとネットワークのタイムアウトが発生し、メッセージの重複送信やその他の問題が発生する可能性があります。

MTA は、1 つのメッセージに特定数以上のアドレスが指定されている場合に配信を遅らせて処理 (オフライン処理) することができます。この方法によって、オンライン遅延を大きく軽減することが可能です。処理のオーバーヘッドを遅らせることはできますが、遅延を完全に回避することはできません。

この機能を有効にするには、たとえば一般的な reprocessing チャネルと expandlimit キーワードを使用します。expandlimit キーワードには、オフライン処理を開始するまでにチャネルから受け入れることのできるメッセージのアドレス数の上限を示す整数の引数をとります。expandlimit キーワードが設定されていない場合のデフォルトは無限大です。引数の値を 0 にすると、そのチャネルで受信したすべてのメッセージがオフラインで処理されます。

expandlimit キーワードは、ローカルチャネルおよび reprocessing チャネルには使用できません。使用すると、予測できない事態が発生する可能性があります。

オフライン処理を行うチャネルを指定するには、expandchannel キーワードを使用します。特に設定を変更しないかぎり、expandchannel が設定されていない場合は reprocessing チャネルが使用されますが、特別な目的のためにはその他の reprocessing チャネルまたは processing チャネルを設定することもできます。expandchannel を使ってオフライン処理を行うチャネルを指定する場合、reprocessing チャネルまたは processing チャネル以外のチャネルを使用することはできません。その他のチャネルを使用すると、予測できない事態が発生する可能性があります。

expandlimit キーワードを適切に機能させるには、reprocessing チャネル (またはオフライン処理を実行するその他のチャネル) を MTA 設定ファイルに追加する必要があります。ただし、MTA 設定ユーティリティによって生成された設定ファイルを使用しているのであれば、その必要はありません。

非常に多くの宛先アドレスが指定されているのは、不特定多数宛てメールの特徴です。holdlimit キーワードは、MTA が特定数以上の宛先アドレスを持つメッセージを受信した場合、そのメッセージを .HELD メッセージとして reprocess チャネル (または expandchannel キーワードが指定するチャネル) のキューに入れるように指示します。メッセージは MTA ポストマスターが手動で介入するまで reprocess キュー内で未処理のまま待機します。

サービス変換を有効にする

キーワード: service、noservice

service キーワードは、CHARSET-CONVERSION エントリにかかわらず、無条件でサービスを有効にします。noservice キーワードが設定されている場合、チャネルで受信するメッセージのサービス変換は、CHARSET-CONVERSION で有効にします。

アドレス処理を設定する

この節ではアドレス処理を行うキーワードを説明します。この章には、以下の節があります。

- 258 ページの「エイリアスからアドレスを指定する」

アドレスのタイプと規則

キーワード : `822`、`733`、`uucp`、`header_822`、`header_733`、`header_uucp`

このキーワードのグループでは、チャネルでサポートするアドレスのタイプが制御されます。転送レイヤ (メッセージエンベロープ) に使われるアドレスとメッセージヘッダーに使われるアドレスとは区別されます。

822 (sourceroute)

ソースルートのエンベロープアドレス。このチャネルでは、ソースルートを含む、完全な RFC 822 形式のエンベロープアドレス規則がサポートされます。`sourceroute` キーワードは、822 と同義で使用できます。ほかのエンベロープアドレスタイプのキーワードが指定されていない場合、これがデフォルトになります。

733 (percents)

パーセント記号のエンベロープアドレス。このチャネルでは、ソースルートを除く、完全な RFC 822 形式のエンベロープアドレスがサポートされます。ソースルートは、パーセント記号の規則を使用して、書き換える必要があります。`percents` キーワードは、733 と同義で使用できます。

注	SMTP チャネルで 733 アドレス規則を使用すると、SMTP エンベロープの転送レイヤのアドレスでもこれらの規則が使われるようになります。これは、RFC 821 に違反する場合があるため、必要時以外は 733 を使用しないようにします。

uucp (bangstyle)

Bang スタイルのエンベロープアドレス。このチャネルでは、エンベロープの RFC 976 の bang スタイルアドレス規則に準拠するアドレスが使用されます (たとえば、UUCP チャネル)。`bangstyle` キーワードは、uucp と同義で使用できます。

header_822

ソースルートのヘッダーアドレス。このチャネルでは、ソースルートを含む、完全な RFC 822 形式のヘッダーアドレス規則がサポートされます。ほかのヘッダーアドレスタイプのキーワードが指定されていない場合、これがデフォルトになります。

header_733

パーセント記号のヘッダーアドレス。このチャネルでは、ソースルートを除く、完全な RFC 822 形式のヘッダーアドレスがサポートされます。ソースルートは、パーセント記号の規則を使用して、書き換える必要があります。

注	メッセージヘッダーで 733 アドレス規則を使用すると、RFC 822 と RFC 976 に違反する場合があります。このキーワードは、チャネルがソースルートアドレスを処理できないシステムに接続することが確実な場合以外は使用しないようにします。

header_uucp

UUCP または bang スタイルのヘッダーアドレス。このキーワードの使用はお勧めしません。使用すると RFC 976 に違反することになります。

! と % を使用するアドレスを解釈する

キーワード : `bangoverpercent`、`nobangoverpercent`、`percentonly`

アドレスは常に RFC 822 と RFC 976 に準拠して解釈されます。ただし、これらの規格で扱われていない複合アドレスの処理方法については、あいまいな部分があります。特に、`A!B%C` という形式のアドレスは次のどちらにも解釈できます。

- `A` がルーティングホストで、`C` が最終的な宛先ホスト

または

- `C` がルーティングホストで、`A` が最終的な宛先ホスト

RFC 976 では、メールプログラムが後者の規則を使ってアドレスを解釈できるという旨が示唆されていますが、そのような解釈が要求されるとは書かれていません。状況によっては、前者の解釈方法を使ったほうがよい場合があるかもしれません。

`bangoverpercent` キーワードを使うと、前者の `A!(B%C)` のように解釈されます。`nobangoverpercent` キーワードを使うと、後者の `(A!B)%C` のように解釈されます。`nobangoverpercent` がデフォルトです。

注	このキーワードは、`A!B@C` 形式のアドレス処理に影響を与えません。これらのアドレスは、常に `(A!B)@C` として扱われます。このような処理は RFC 822 と RFC 976 の両方で義務付けられています。

`percentonly` キーワードで、bang パスが無視されます。このキーワードが設定されている場合、パーセントはルーティング用に解釈されます。

アドレスにルーティング情報を追加する

キーワード : exproute、noexproute、improute、noimproute

MTA が扱うアドレスモデルは、すべてのシステムがほかのすべてのシステムのアドレスを知っていて、それらのアドレスにどのように到達するかを知っているものと想定しています。しかし、このような理想は、世界に知られていない 1 つ以上のシステムにチャネルが接続する (たとえば、プライベートな TCP/IP ネットワーク内にあるマシン) 場合など、どのような場合にも当てはまるとはかぎりません。このチャネルにあるシステムのアドレスは、サイトの外にあるリモートのシステムからは見ることができないようになっているのかもしれません。このようなアドレスに応答したい場合は、ローカルマシンを通ってメッセージをルーティングするようリモートのシステムに指示するソースルートを含んでいなければなりません。そうすれば、ローカルマシンは (自動的に) これらのマシンにルーティングすることができます。

exproute キーワード (explicit routing の略) は、アドレスがリモートのシステムに渡されるときに、関連するチャネルが明示的なルーティングを要するということを MTA に指示します。このキーワードがチャネルに指定されている場合、MTA により、ローカルシステムの名前 (またはローカルシステムの現在のエイリアス) を含むルーティング情報が、チャネルに一致するすべてのヘッダーアドレスとすべてのエンベロープの From: アドレスに追加されます。noexproute はデフォルトで、ルーティング情報を追加しないことを指定します。

EXPROUTE_FORWARD オプションは、後方を探すアドレスに対する exproute の動作を制限するために使用できます。MTA が適切なルーティングを独自に実行することができないチャネルを通して相手システムに接続する場合には、別の状況が発生します。この場合、ほかのチャネルに関連するアドレスはすべて、能力のないシステムに接続するチャネルに送られたメール内で使用されるときに、ルーティング指定を必要とします。

この状況を処理するには、黙示的なルーティングと improute キーワードが使用されます。MTA は、ほかのチャネルに合致するすべてのアドレスが improute マークの付いたチャネルに送られたメールの中で使用されるときにルーティングを必要とすることを知っています。デフォルトの noimproute は、指定されたチャネルに送られるメッセージのアドレスにルーティングの情報を加えないことを指定するものです。IMPROUTE_FORWARD オプションは、後方を探すアドレスに対する improute の動作を制限するために使用できます。

exproute および improute キーワードは慎重に使用するようにしてください。これらのキーワードは、アドレスを長く、より複雑にし、相手側のシステムで使用されているインテリジェントなルーティング機能を妨害する可能性があります。明示的ルーティングと黙示的ルーティングを、指定ルートと混同しないようにしてください。指定ルートは、書き換え規則からアドレスにルーティング情報を挿入するときに使用されます。これは、特殊な A@B@C 書き換え規則テンプレートによってアクティブになります。

指定ルートは、アクティブになったときに、ヘッダーとエンベロープ内のすべてのアドレスに適用されます。指定ルートは特定の書き換え規則によってアクティブになるもので、通常、現在使用中のチャネルとは関係がありません。一方、明示的ルーティングと黙示的ルーティングはチャネルごとに制御され、挿入されるルートアドレスは常にローカルシステムのものです。

明示的なルーティングアドレスの書き換えを無効にする

キーワード : routelocal

routelocal チャネルキーワードでは、アドレスをチャネルに書き換える際に、MTA にアドレスのすべての明示的ルーティングを短絡化しようとします。明示的にルーティングされたアドレス (!、%、または @ の文字を使用) は簡略化されています。

このキーワードを内部 TCP/IP チャネルなどの「内部」チャネルに使用すると、SMTP リレーブロッキングの設定を簡単にすることができます。

ただし、明示的 % やその他のルーティングを必要とする可能性があるチャネルには、このキーワードを使用してはいけません。

メッセージがキューから取り出されるときのアドレス書き換え

キーワード : connectalias、connectcanonical

通常、MTA はチャネルのキューにメッセージを入れるときにアドレスを書き換えます。メッセージがキューから取り出されるときに、さらに書き換えが行われることはありません。したがって、ホスト名が変更されたときにチャネルのキュー内に元のホスト名宛てのメッセージがまだ残っていても、問題は生じません。

connectalias キーワードは、受取人のアドレスに書かれているホストに配信するように、MTA に指示します。デフォルトでは、このキーワードが使用されます。connectcanonical キーワードは、MTA が接続するシステムのホストエイリアスに接続するように指示します。

不完全なアドレスを修正する際に使用するホスト名を指定する

キーワード : remotehost、noremotehost、defaulthost、nodefaulthost

MTA は、間違って設定された、あるいは標準に準拠しないメーラーや SMTP クライアントから、ドメイン名を含まないアドレスを受け取ることがよくあります。MTA は、そのようなメッセージを通過させる前に、アドレスを有効な形式にしようと試みます。MTA は、アドレスにドメイン名を付け加える (たとえば、@siroe.com を mrochek に付け加える) ことによってそれを行います。

エンベロープ To: アドレスにドメイン名がない場合、MTA では常にローカルホスト名を追加するものと仮定します。From: アドレスなどのその他のアドレスの場合、MTA SMTP サーバには、ドメイン名に関して少なくとも 2 つのオプションが考えられます。それらのオプションとは、ローカル MTA ホスト名と、クライアント SMTP でレポートされたリモートホスト名です。また場合によっては、そのチャネルで受信するメッセージに特定のドメイン名を追加するという、3 つめのオプションが考えられる可能性もあります。最初の 2 つのオプションは、どちらもある程度の頻度で発生することが考えられるため、適切なものと考えられます。不適切に構成された SMTP クライアントを扱う場合には、リモートホストのドメイン名を使用することが適切です。メッセージを掲示するために SMTP を使う POP や IMAP クライアントのように軽量級のリモートメールクライアントを扱う場合には、ローカルホストのドメイン名を使用することが適切です。また、(POP や IMAP などの) 軽量級のリモートメールクライアントの場合は、各クライアントにはローカルホスト以外の専用の特定ドメイン名があります。この場合には、その他の特定ドメイン名の追加が適当な場合もあります。MTA がとれる最善の策は、チャネルごとに選択できるようにすることです。

noremotehost チャネルキーワードはローカルホストの名前が使用されるように指定するものです。デフォルトのキーワードは noremotehost です。

defaulthost チャネルキーワードを使用して、受信側のユーザ ID に追加する特定のホスト名を指定します。このキーワードの後ろには、チャネルで受信するアドレスを完成させるためのドメイン名 (エンベロープ From: 内とヘッダー内) を追加します。送信チャネルの場合は、defaulthost キーワードの最初の引数もエンベロープ To: アドレスに影響します。省略可能な 2 番目のドメイン名 (中に少なくとも 1 つのピリオドが含まれている) を指定してエンベロープ To: アドレスを完成させることもできます。nodefaulthost はデフォルトです。

switchchannel キーワードは、前のセクション「受信メール用代替チャネル (切り替えチャネル)」で説明されているとおり、受信 SMTP 接続を特定のチャネルに関連付けるために使用することができます。この機能は、リモートのメールクライアントを、適切な処理を受けることができるチャネルにグループ化するために使用することができます。代わりの方法として、(標準に準拠しないクライアントが多数に使用されていたとしても) 標準に準拠するリモートメールクライアントを配備する方が、MTA ホストでネットワーク全体の問題を解決しようとするより簡単です。

Recipient ヘッダー行がないメッセージを有効にする

キーワード : missingrecipientpolicy

RFC 822 (Internet) メッセージには、受取人ヘッダー行である To:、Cc:、または Bcc: ヘッダー行が必要です。そのようなヘッダー行がないメッセージは無効になります。しかし、うまく稼働していないユーザエージェントやメーラー (たとえば、古いバージョンの sendmail) は、無効なメッセージを受け入れます。

missingrecipientpolicy キーワードは、そのようなメッセージを扱うときに使用すべきアプローチを指定する整数値をとります。このキーワードが明示的に表現されていない場合は、デフォルト値の 0 が使用され、To: ヘッダーにエンベロープ To: アドレスが使用されます。

表 8-8　missingrecipientpolicy の値

値	動作
0	To: ヘッダー行にエンベロープ To: 受取人を使用する
1	変更せずに無効なメッセージを通過させる
2	To: ヘッダー行にエンベロープ To: 受取人を使用する
3	単一の Bcc: ヘッダー行にすべてのエンベロープ To: 受取人を使用する
4	グループのコンストラクタ (たとえば ;) を To: ヘッダー行にし、To: 受取人は指定しない
5	空白の Bcc: ヘッダー行を生成する
6	メッセージを拒否する

MISSING_RECIPIENT_POLICY オプションは、MTA システムがデフォルトでこの動作をするように設定するためのものであることに注意してください。初期の Messaging Server 設定では、MISSING_RECIPIENT_POLICY が 1 に設定されます。

不正な空白の受取人ヘッダーを削除する

キーワード : dropblank、nodropblank

RFC 822 (インターネット) メッセージでは、To:、Resent-To:、Cc:、Resent-Cc: ヘッダーにはアドレスが少なくとも 1 つ必要です。空白値は使用できません。ただし、一部のメーラーでは、このような不正なヘッダーが生成されることがあります。ソースチャネルに dropblank チャネルキーワードが指定されている場合、MTA により受信メッセージからこれらの不正な空白ヘッダーが削除されます。

チャネル固有のリバースデータベースの使用を有効にする

キーワード : reverse、noreverse

reverse キーワードは、チャネルのキューに入れられたメッセージ内のアドレスを、アドレスリバースデータベースまたは REVERSE マッピング (存在する場合) のいずれかに対して照合し、必要に応じて変更するように指示するものです。また、noreverse は、チャネルのキューに入れられたメッセージのアドレスを、アドレスリバース処理から外すことを指定するものです。デフォルトのキーワードは reverse です。詳細については、145 ページの「内部形式から公的な形式にアドレスを変換するには」を参照してください。

制限されたメールボックスのエンコーディングを有効にする

キーワード : restricted、unrestricted

メールシステムの中には、RFC 822 で許されるアドレスのすべての形式を扱うことができないものもあります。もっとも一般的に見られる例は、設定ファイルが不適切に設定された sendmail ベースのメーラーです。引用されたローカルパート (あるいはメールボックス仕様) が頻繁に見られる問題の原因です。

```
"smith, ned"@siroe.com
```

これは大きな問題なので、この問題を回避するための方策が RFC 1137 に記載されています。基本的なアプローチは、アドレスから引用を取り除き、引用を要する文字を、アトムに許可する文字にマップする変換規則を適用することです (ここで使われているアトムという語の定義については RFC 822 を参照)。たとえば、上記のアドレスは次のようになります。

```
smith#m#_ned@siroe.com
```

restricted チャネルキーワードでは、MTA に、このチャネルがこのエンコーディングを必要とするメールシステムに接続することを示します。すると MTA は、メッセージがチャネルに書かれるときに、ヘッダーとエンベロープアドレスの両方において引用されたローカルパートをエンコードします。そのチャネルの受信メールのアドレスは自動的にデコードされます。unrestricted キーワードは、RFC 1137 エンコーディングとデコーディングを実行するように MTA に指示します。デフォルトは unrestricted キーワードです。

注 restricted キーワードは、引用されたローカルパートを受け入れることができないシステムに接続するチャネルに対して適用します。引用されたローカルパートを実際に生成するチャネルには適用しないでください。(そのようなアドレスを生成することができるチャネルは、そのようなアドレスを処理することができると想定されるからです。)

Return-path: ヘッダー行を生成する

キーワード : addreturnpath、noaddreturnpath

通常、Return-path: ヘッダー行の追加は、最終的な配信を実行するチャネルが行います。ただし、ims-ms チャネルなどの一部のチャネルでは、MTA で Return-path: ヘッダー行を追加する方が、チャネルで追加するよりも効率的です。addreturnpath キーワードでは、このチャネルのキューにメッセージを入れる際に、MTA により Return-path: ヘッダーが追加されます。

エンベロープ To: アドレスと From: アドレスから Received: ヘッダー行を作成する

キーワード : receivedfor、noreceivedfor、receivedfrom、noreceivedfrom

receivedfor キーワードは、メッセージの宛先になっているエンベロープ受取人アドレスが 1 つだけの場合は、そのエンベロープの To: アドレスを Received: ヘッダー行に含めるように MTA に指示します。デフォルトのキーワードは receivedfor です。noreceivedfor キーワードは、エンベロープアドレス情報を含めずに、Received: ヘッダー行を作成するよう MTA に指示します。

receivedfrom キーワードは、たとえばメーリングリストの拡大などのために MTA がエンベロープ From: アドレスを変更した場合、受信メッセージの Received: ヘッダー行を作成する際は MTA に元のエンベロープの From: アドレスを含めるように指示します。receivedfrom はデフォルトのキーワードです。noreceivedfrom キーワードは、元のエンベロープ From: アドレスを使わずに Received: ヘッダー行を作成するよう MTA に指示します。

アドレスヘッダー行内のコメントを処理する

キーワード : commentinc、commentmap commentomit、commentstrip、commenttotal、sourcecommentinc、sourcecommentmap、sourcecommentomit、sourcecommentstrip、sourcecommenttotal

MTA は必要なときだけヘッダー行の内容を解釈します。ただし、省略形のアドレスを書き換えてなくすために (それ以外の場合は、有効なアドレスに変換するために)、アドレスを含むすべての登録されたヘッダー行を解析しなければなりません。この処理の途中では、コメント (括弧で囲まれた文字列) が抽出され、ヘッダー行が再構成されるときに変更されるか、あるいは除外されることがあります。

この動作は、commentinc、commentmap、commentomit、commentstrip、および commenttotal キーワードを使用して制御されます。commentinc キーワードは、ヘッダー行内のコメントを残すように MTA に指示します。デフォルトでは、このキーワードが使用されます。commentomit キーワードは、アドレスヘッダー、たとえば To:、From:、あるいは Cc: ヘッダー行からコメントを取り除くよう MTA に指示します。

commenttotal キーワードは、MTA にすべてのヘッダー行 (Received: ヘッダー行を除く) からコメントを削除するように指示します。このキーワードは通常特に使い道はなく、お勧めもしません。commentstrip は MTA にすべてのコメントフィールドから、すべての非原子的文字列を削除するように指示します。commentmap キーワードは、COMMENT_STRINGS マッピングテーブルを通じてコメント文字列を実行します。

ソースチャネルでは、この動作は sourcecommentinc、sourcecommentmap、sourcecommentomit、sourcecommentstrip、および sourcecommenttotal の各キーワードを使用して制御されます。sourcecommentinc キーワードは、MTA にヘッダー行のコメントを維持するように指示します。デフォルトでは、このキーワードが使用されます。sourcecommentomit キーワードは、MTA にアドレスヘッダー (To:、From:、Cc: などのヘッダー) からすべてのコメントを削除するように指示します。commenttotal キーワードは、MTA にすべてのヘッダー行 (Received: ヘッダー行を除く) からコメントを削除するように指示します。このキーワードは通常特に使い道はなく、お勧めもしません。最後に、sourcecommentstrip キーワードは MTA に、すべてのコメントフィールドから非原子的文字を削除するように指示します。sourcecommentmap キーワードは、ソースチャネルを通じてコメント文字列を実行します。

これらのキーワードはどのチャネルにも適用できます。

COMMENT_STRINGS マッピングテーブルのシンタックスは、次のとおりです。

（*comment_text*） | *address*

エントリテンプレートに $Y フラグが設定されている場合、元のコメントは指定したテキスト (閉じる括弧を含むこと) に置き換えられます。

アドレスヘッダー行内の個人名を処理する

キーワード : personalinc、personalmap、personalomit、personalstrip、sourcepersonalinc、sourcepersonalmap、sourcepersonalomit、sourcepersonalstrip

書き換えプロセスの際には、省略形のアドレスを書き換えてなくすために (それ以外の場合は、有効なアドレスに変換するために)、アドレスを含むすべてのヘッダー行を解析しなければなりません。このプロセスの際に、個人名 (角括弧で区切られたアドレスの前にある文字列) が抽出されますが、これはヘッダー行を再構築するときに変更したり除外したりできます。

この動作は、personalinc、personalmap、personalomit、および personalstrip キーワードの使用によって制御されます。キーワード personalinc は、ヘッダー内の個人名を残すよう MTA に指示します。デフォルトでは、このキーワードが使用されます。personalomit キーワードは、MTA にすべての個人名を削除するように指示します。personalstrip キーワードは、MTA にすべての個人名フィールドから、すべての非原子的文字を削除するように指示します。personalmap キーワードは、MTA に PERSONAL_NAMES マッピングテーブルを通じて個人名を実行するように示します。

ソースチャネルでは、この動作は sourcepersonalinc、sourcepersonalmap、sourcepersonalomit、または sourcepersonalstrip キーワードを使用して制御されます。sourcepersonalinc キーワードは、MTA にヘッダーの個人名を維持するように指示します。デフォルトでは、このキーワードが使用されます。sourcepersonalomit キーワードは、MTA にすべての個人名を削除するように指示します。最後に、sourcepersonalstrip キーワードは MTA に、すべての個人名フィールドから非原子的文字を削除するように指示します。sourcepersonalmap キーワードは、MTA にソースチャネルを通じて個人名を実行するように示します。

これらのキーワードはどのチャネルにも適用できます。

PERSONAL_NAMES マッピングテーブルのシンタックスは、次のとおりです。

personal_name | *address*

テンプレートで $Y フラグが設定されている場合、元の個人名は指定したテキストで置き換えられます。

エイリアスファイルとエイリアスデータベースプローブを指定する

キーワード : aliaslocal

通常、ローカルチャネル (UNIX の 1 チャネル) に書き換えられるアドレスのみが、エイリアスファイルとエイリアスデータベースで検索されます。aliaslocal キーワードをチャネルに使用すると、そのチャネルに書き換えられるアドレスも、エイリアスファイルとエイリアスデータベースで検索するようにできます。作成される検索プローブの形式は、ALIAS_DOMAINS オプションで制御されます。

サブアドレスを処理する

キーワード : subaddressexact、subaddressrelaxed、subaddresswild

サブアドレスの概念の背景として、ネイティブと ims-ms のチャネルでは + 記号がアドレスのローカル部分 (メールボックスの部分) として解釈されます。特に、*name+subaddress@domain* の形式のアドレスでは、MTA はプラス記号の後ろのメールボックス部分をサブアドレスとみなします。ローカルチャネルでは、サブアドレスを追加の余分な情報とみなして、サブアドレスを考慮せず実際にアカウント名への配信を行います。ims-ms チャネルでは、サブアドレスを配信先のフォルダ名と解釈します。

また、サブアドレスはローカルチャネル (UNIX の L チャネル) によるエイリアスの検索、aliaslocal キーワードでマークされたすべてのチャネルによるエイリアスの検索、およびディレクトリチャネルによるメールボックスの検索に影響を与えます。これらの検索に対するサブアドレスの処理については、設定可能です。アドレスをエントリと比較する場合、MTA では必ず最初に完全一致の検索にサブアドレスを含むメールボックス全体を確認します。追加のチェックを実行するかどうかは、設定可能です。

subaddressexact キーワードは、MTA にエントリの一致の確認中に、特別なサブアドレスの処理を行わないように指示します。エイリアスが一致するとみなされるためには、サブアドレスを含むメールボックス全体が一致しなければなりません。その他の比較 (特に、ワイルドカードによる比較や、サブアドレスを削除した比較) は行われません。subaddresswild キーワードは、MTA に、サブアドレスを含む完全な一致を検索したあと、「名前 +*」の形式のエントリを検索するように指示します。subaddressrelaxed キーワードは MTA に、完全一致と「名前 +*」の形式の一致を検索したあと、名前の部分のみの一致を検索するように指示します。

subaddressrelaxed では、次の形式のエイリアスエントリが、名前か「名前 + サブアドレス」に一致し、名前を新規の名前に、「名前 + サブアドレス」を「新規の名前 + サブアドレス」に変換します。デフォルトのキーワードは subaddressrelaxed です。

```
name:   newname+*
```

このように、subaddresswild キーワードや subaddressrelaxed キーワードは、エイリアスやディレクトリが使用されていて、ユーザが任意のサブアドレスを使用してメールの受信を希望する場合に便利です。これらのキーワードを使用することにより、アドレスの各サブアドレスに独立のエントリを作成する必要がなくなります。

これらのキーワードは、ローカルチャネル (UNIX の L チャネル) とディレクトリチャネル、および aliaslocal キーワードでマークされたチャネルにかぎり使用できます。

標準の Messaging Server 設定では、実際に subaddressrelaxed キーワード (ほかのキーワードが明示的に使用されていない場合のデフォルト) を指定した L チャネルでリレーします。

チャネル固有の書き換え規則チェックを有効にする

キーワード : rules、norules

rules キーワードは、MTA にこのチャネルにおけるチャネル固有の書き換え規則のチェックを強制するように指示します。デフォルトでは、このキーワードが使用されます。norules キーワードは、MTA にこのチャネルをチェックしないように指示します。これらの 2 つのキーワードは、通常デバッグに使用され、実際のアプリケーションで使用されることはほとんどありません。

ソースルートを削除する

キーワード : dequeue_removeroute

dequeue_removeroute キーワードは、メッセージがキューから取り出されると、エンベロープの To: アドレスからソースルートを削除します。現在、このキーワードは tcp-* チャネルだけに実装されています。ソースルートを正しく処理しないシステムにメッセージを転送する場合に便利なキーワードです。

エイリアスからアドレスを指定する

キーワード : `viaaliasoptional`、`viaaliasrequired`

`viaaliasrequired` は、チャネルに一致する最終受取人アドレスをエイリアスで作成するように指定するキーワードです。最終受取人アドレスとは、関連するエイリアス拡張を行ったあとで一致するアドレスです。アドレスを受取人アドレスとして MTA に直接渡すことはできません。チャネルに書き換えただけでは十分ではないからです。チャネルに書き換えたアドレスをエイリアスを使って拡張し、間違いなくそのチャネルと一致していることを確認する必要があります。

たとえば、ローカルチャネルで `viaaliasrequired` キーワードを使って、任意のアカウント (たとえば UNIX システム上のネイティブな任意の Berkeley メールボックス) に配信させないようにすることができます。

デフォルトは `viaaliasoptional` であり、そのチャネルに一致する最終受取人アドレスはエイリアスで作成する必要がありません。

ヘッダー処理を設定する

この節ではヘッダーとエンベロープ情報を扱うキーワードを説明します。この章には、以下の節があります。

埋め込まれたヘッダーを書き換える

キーワード : noinner、inner

ヘッダー行の内容は必要なときにだけ解釈されます。ただし、メッセージの中にメッセージを埋め込むことができる能力 (メッセージ /RFC822) があるために、MIME メッセージには複数のメッセージヘッダーが含まれていることもあります。通常、MTA は一番外側のメッセージヘッダーだけを解釈し、書き換えます。オプションとして、メッセージの内部ヘッダーに書き換え規則を適用するように指示することも可能です。

この動作は、noinner および inner キーワードを使用して制御できます。キーワード noinner は、内部ヘッダー行を書き換えないように MTA に指示するものです。デフォルトでは、このキーワードが使用されます。キーワード inner は、メッセージを解析して、内部ヘッダーを書き換えるように MTA に指示します。これらのキーワードはどのチャネルにも適用できます。

メッセージヘッダー行を選択して削除する

キーワード : headertrim、noheadertrim、headerread、noheaderread、innertrim noinnertrim

MTA には、メッセージから特定のメッセージヘッダー行をトリミングする (取り除く)、チャネル単位の機能があります。これは、チャネルキーワードと関連する 1 つまたは 2 つのヘッダーオプションファイルの組み合わせによって行われます。headertrim キーワードは、チャネルに関連するヘッダーオプションファイルを作成し、元のメッセージヘッダーが処理されたあと、チャネルのキューに入れられたメッセージのヘッダーをそれに基づいてトリムするよう MTA に指示します。noheadertrim キーワードは、ヘッダートリミングを行いません。デフォルトは noheadertrim キーワードです。

innertrim キーワードは、埋め込まれた MESSAGE/RFC822 部分のような、内部メッセージ部分にヘッダートリミングを実行するよう MTA に指示します。noinnertrim キーワードはデフォルトで、内部メッセージ部分のどのヘッダーにもトリミングを実行しないよう MTA に指示します。

headerread キーワードは、元のメッセージヘッダーが処理される前に、そのチャネルに関連しているヘッダーオプションファイルを参照して、そのソースチャネルによってキューに入れられているメッセージのヘッダーをトリムするよう MTA に指示します。一方、headertrim ヘッダートリミングはメッセージが処理されたあとに適用され、ソースチャネルではなく宛先チャネルになります。noheaderread キーワードは、キューに入っているメッセージのヘッダートリミングを行いません。noheaderread がデフォルトです。

headeromit および headerbottom キーワードとは異なり、headertrim および headerread キーワードはどのチャネルにも適用できます。ただし、重要なヘッダー情報をメッセージから取り除くと MTA が正常に動作しなくなることもあるので、注意してください。取り除くヘッダーまたは制限するヘッダーを選ぶ際には、十分な配慮が必要です。この機能があるのは、特定のヘッダー行を取り除いたり、制限したりしなければならないような状況が発生することがあるからです。

警告 ヘッダー情報をメッセージから取り除くと、MTA が正常に動作しなくなることもあります。取り除くヘッダーまたは制限するヘッダーを選ぶ際には、配慮が必要です。これらのキーワードは、特定のヘッダー行を取り除いたり、制限したりしなければならないような稀な状況で指定します。ヘッダー行を取り除く前に、そのヘッダー行の用途を十分に理解し、それを取り除いた場合の結果を考慮してください。

headertrim および innertrim キーワードのヘッダーオプションファイルには、*channel*_headers.opt という形式の名前があります。このチャネルには、ヘッダーオプションファイルが関連付けられているチャネルの名前が入ります。同じように、headerread キーワードのヘッダーオプションファイルには、channel_read_headers.opt の形式で名前があります。これらのファイルは MTA の設定ディレクトリ (*server_root*/msg-*instance*/imta/config/) に保存されます。

X-Envelope-to: ヘッダー行の生成と削除

キーワード: x_env_to、nox_env_to

x_env_to および nox_env_to キーワードは、特定のチャネルのキューに入れられたメッセージのコピーに X-Envelope-to ヘッダー行を生成するかどうかを制御します。single キーワードでマークされているチャネルでは、x_env_to はこれらのヘッダーの生成を有効にし、nox_env_to はキュー内のメッセージからこれらのヘッダーを削除します。デフォルトは nox_env_to です。

x_env_to キーワードには、有効にするための single キーワードが必要です。

日付表示を 2 桁から 4 桁に変換する

キーワード : datefour、datetwo

オリジナルの RFC 822 仕様では、メッセージヘッダーの日付フィールドに 2 桁の年表示を使用することが規定されています。これはあとで RFC 1123 により 4 桁に変更されました。しかし、古いメールシステムの中には、4 桁の日付を受け入れないものもあります。また、新しいメールシステムの中には、2 桁の日付を受け入れなくなったものもあります。

注 両方の形式を扱うことができないシステムは規格に違反しています。

datefour および datetwo キーワードは、MTA によるメッセージヘッダー内の日付フィールド処理を制御するものです。datefour キーワードがデフォルトで、すべての年表示フィールドを 4 桁に展開するように MTA に指示します。値が 50 以下の 2 桁の日付表示には 2000 が加えられ、50 より大きいものには 1900 が付け加えられます。

警告 datetwo キーワードは、4 桁の日付表示から先頭の 2 桁を取り去るように MTA に指示します。これは、2 桁の日付表示を要求する、標準に準拠していないメールシステムとの互換性を提供する目的で行われます。その他の目的のために使用してはなりません。

日付の曜日を指定する

キーワード : dayofweek、nodayofweek

RFC 822 仕様では、メッセージヘッダー内の日付フィールドにおいて、日付の前に曜日を付けることができます。ただし、システムの中には曜日情報を受け入れられないものもあります。そのため、ヘッダーに含めると便利な情報であるにもかかわらず、曜日情報を含めないシステムもあります。

dayofweek および nodayofweek キーワードは、MTA による曜日情報処理を制御するものです。dayofweek キーワードがデフォルトで、これは曜日情報を残し、曜日情報がない場合にはその情報を月日 / 時間ヘッダーに追加するよう MTA に指示します。

警告 nodayofweek キーワードは、月日 / 時間ヘッダーから先頭の曜日情報を取り除くよう MTA に指示します。これは、この情報を適切に処理することができない、標準に準拠していないメールシステムとの互換性を提供する目的で行われます。その他の目的のために使用してはなりません。

長いヘッダー行を自動分割する

キーワード: maxheaderaddrs、maxheaderchars

メッセージ転送形式、特に sendmail の実装の中には、長いヘッダー行を適切に処理できないものがあります。これは、ヘッダーが破壊されるだけでなく、誤ったメッセージ拒否の原因になりがちです。これは重大な規格違反ですが、よく発生する問題です。

MTA には、長いヘッダー行を複数の独立したヘッダー行に分割するチャネルごとの機能があります。maxheaderaddrs キーワードは 1 つの行にいくつのアドレスを含められるかを制御し、maxheaderchars キーワードは 1 行に何バイト分の文字を含められるかを制御します。どちらのキーワードにも、限度を指定する 1 つの整数引数が必要です。デフォルトでは、ヘッダー行の長さもアドレスの数も制限されていません。

ヘッダーの配置と折り返し

キーワード: headerlabelalign、headerlinelength

headerlabelalign キーワードは、このチャネルのキューに入れられたメッセージヘッダーの配置ポイントを制御するものです。整数値の引数をとります。配置ポイントとは、ヘッダーの内容を揃えるためのマージンです。たとえば、配置ポイントが 10 のヘッダー行は次のようになります。

```
To:       joe@siroe.com
From:     mary@siroe.com
Subject:  Alignment test
```

デフォルトの headerlabelalign は 0 で、ヘッダーは揃えられません。
headerlinelength キーワードは、このチャネルのキューに入れられたメッセージヘッダー行の長さを制御します。これよりも長い行は、RFC 822 の折り返し規則に基づいて折り返されます。

これらのキーワードは、メッセージキュー内にあるメッセージのヘッダー形式を制御するだけのものです。実際のヘッダーの表示は、通常、ユーザエージェントによって制御されます。さらに、ヘッダーはインターネットを転送されるときに何度もリフォーマットされるため、メッセージヘッダーをフォーマットしない単純なユーザエージェントといっしょに使用された場合には、これらのキーワードの効果が見られないこともあります。

ヘッダーの最大長を指定する

キーワード: maxprocchars

たくさんのアドレスを含む長いヘッダー行の処理には、多くのシステムリソースを費やすことがあります。maxprocchars キーワードは、MTA が処理して書き換えることができるヘッダーの最大長を指定するために使用されます。これよりも長いヘッダーを持つメッセージも受け入れられて配信されますが、異なる点は、長いヘッダー行は書き換えられないということです。このキーワードには、1 つの整数引数が伴います。デフォルトでは、どのような長さのヘッダーも処理されます。

機密度チェック

キーワード: sensitivitynormal、sensitivitypersonal、sensitivityprivate sensitivitycompanyconfidential

機密度チェックのキーワードは、チャネルが受け入れられる機密度の上限を設定するものです。デフォルトは sensitivitycompanyconfidential で、どの機密度レベルのメッセージも通過を許されます。Sensitivity: ヘッダーのないメッセージは、通常のメッセージ、つまり、機密度のもっとも低いメッセージとみなされます。このようなキーワードで指定された機密度よりも高い機密度が指定されたメッセージがチャネルのキューに入れられると、次のようなエラーメッセージが表示され、拒否されます。

message too sensitive for one or more paths used（使用されている 1 つ以上のパスに対してメッセージの機密度が高すぎます。）

MTA では、受取人ごとではなく、メッセージごとに機密度のチェックが行われます。1 人の受取人の宛先チャネルが機密度チェックに失敗した場合、そのチャネルに関連付けられた受取人だけでなく、すべての受取人のメッセージが返送されます。

ヘッダーのデフォルト言語を設定する

キーワード: language

ヘッダーのエンコードされた単語には、特定言語を含ませることが可能です。デフォルトの言語は、language キーワードで指定されます。

添付と MIME 処理

この節では添付と MIME 処理を扱うキーワードを説明します。この章には、以下の節があります。

Encoding: ヘッダー行を無視する

キーワード : `ignoreencoding`、`interpretencoding`

MTA は、`Yes CHARSET-CONVERSION` を使用して、さまざまな非標準のメッセージ形式を `MIME` に変更することができます。特に、RFC 1154 形式では非標準の `Encoding:` ヘッダー行が使用されます。しかし、ゲートウェイの中には、ヘッダー行に対して誤った情報を出すものもあり、その結果、このヘッダー行を無視したほうがいい場合もあります。`ignoreencoding` キーワードは、`Encoding:` ヘッダー行をすべて無視するよう MTA に指示するものです。

注	MTA の `CHARSET-CONVERSION` が有効になっていないかぎり、このようなヘッダーはいずれにしても無視されます。`interpretencoding` キーワードは、特にほかの設定が行われている場合を除き、MTA にすべての `Encoding:` ヘッダー行に注目するように指示します。これはデフォルトです。

メッセージあるいは部分メッセージの自動再組立

キーワード : `defragment`、`nodefragment`

MIME 規格には、メッセージをより小さな部分に分割するための message/partial コンテンツタイプがあります。これはメッセージがサイズ制限のあるネットワークを通過する場合、または信頼性の低いネットワークを通過する場合に便利です。メッセージの断片化により、ある種の「チェックポイント」が提供され、メッセージの転送中にネットワークエラーが発生した場合でも、操作の不要な繰り返しを防ぐことができます。メッセージが宛先に到着したときに自動的に再組み立てが行われるように、それぞれの部分に情報が含まれています。

MTA では、defragment チャネルキーワードと再組立チャネルを使うことによって、メッセージの再組み立てを行うことができます。チャネルが defragment でマークされていれば、このチャネルのキューに入れられるメッセージまたは部分メッセージはすべて、代わりに再組立チャネルのキューに入れられます。すべての部分が到着したら、メッセージは再構築されて本来の宛先に送られます。nodefragment は、このような特別な処理を無効にするものです。デフォルトのキーワードは nodefragment です。

大きなメッセージの自動断片化

キーワード : maxblocks、maxlines

電子メールシステムまたはネットワーク転送形式の中には、特定のサイズを超えるメッセージを処理できないものがあります。MTA には、チャネルごとにそのような制限を課す機能があります。設定されたサイズよりも大きなメッセージは自動的に複数の、より小さなメッセージに分割 (断片化) されます。このような断片に使用されるコンテンツタイプは message/partial で、同じメッセージの各部分が互いに関連付けられ、受信先のメーラーによって自動的に再組立されるように固有 ID の引数が付け加えられます。

maxblocks および maxlines キーワードは、自動断片化の対象となるサイズ制限枠を課すために使用されます。これらのキーワードの後ろには 1 つの整数値が続きます。maxblocks キーワードは、1 つのメッセージに許可するブロックの最大数を指定します。1 つの MTA ブロックは通常 1024 バイトで、これは MTA オプションファイルにある BLOCK_SIZE オプションを使用して変更することができます。maxlines キーワードは、1 つのメッセージに許可する最大行数を指定します。これらの 2 つの制限は、必要に応じて同時に課すことができます。

メッセージヘッダーは、ある程度メッセージのサイズに含まれています。メッセージヘッダーを複数のメッセージに分割することはできないにもかかわらず、それ自体が指定されたサイズ制限を超えてしまうこともあるので、メッセージヘッダーのサイズを管理するためにかなり複雑なしくみが使われます。この論理は、MTA オプションファイルにある MAX_HEADER_BLOCK_USE と MAX_HEADER_LINE_USE オプションによって制御されます。

MAX_HEADER_BLOCK_USE は、0 から 1 までの間の実数を指定するために使用されます。デフォルト値は 0.5 です。この場合、メッセージのヘッダーは、(maxblocks キーワードで指定された) 1 つのメッセージが占めることができる合計のブロック数の半分を占めることができます。メッセージヘッダーがそれより大きい場合、MTA は MAX_HEADER_BLOCK_USE と maxblocks の積を、* MAX_HEADER_BLOCK_USE ヘッダーのサイズ (ヘッダーサイズは、実際のヘッダーサイズと maxblocks より小さいものとみなされる) としてとります。

たとえば、maxblocks が 10 で MAX_HEADER_BLOCK_USE がデフォルトの 0.5 である場合、5 ブロックより大きいメッセージヘッダーは 5 ブロックのヘッダーとして取り扱われ、メッセージのサイズが 5 あるいはそれ以下のブロックの場合、断片化されません。0 を指定すると、メッセージのサイズ制限をあてはめる場合にヘッダーは無視されます。

1 を指定すると、利用可能なサイズのすべてをヘッダーに使うことができます。それぞれの断片は、サイズ制限を超えたかどうかにかかわらず、常に最低 1 行のメッセージ行を含みます。MAX_HEADER_LINE_USE および maxlines キーワードも、同様に動作します。

メッセージ行の長さを制限する

キーワード: linelength

SMTP 仕様では、1000 バイトまでのテキスト行が許可されています。しかし、転送形式の中には、行長に制限を課すものもあります。linelength キーワードは、チャネルごとに許される最大のメッセージ行の長さを制限するしくみを提供します。特定のチャネルのキューに入れられたメッセージの中で、そのチャネルに指定された行長を超えるメッセージは自動的にエンコードされます。

MTA にはさまざまなエンコーディング方式が用意されており、エンコーディングの結果、行長は常に 80 バイト以下になります。エンコーディングが行われた元のメッセージは、適切なデコーディングのフィルタを通すことによって元の状態に戻すことができます。

注	エンコーディングは、行長を 80 バイトより短くするだけです。行長に 80 バイトより短い値を指定しても、指定された制限より短い行にできるとはかぎりません。

linelength キーワードでは、データのエンコーディングで転送用のソフト改行が実行されます。このエンコーディングは、通常受信側でデコードされるため、元の長い行が復元されます。ハード改行については、「Record, text」
CHARSET-CONVERSION を参照してください。

メッセージのサイズ制限、ユーザ制限容量、権限

この節では、メッセージのサイズ制限、ユーザ制限容量、権限を設定するキーワードについて説明します。この章には、以下の節があります。

絶対的なメッセージサイズ制限を指定する

キーワード:blocklimit、noblocklimit、linelimit、nolinelimit、sourceblocklimit

メッセージは断片化によって自動的に小さな部分に分割されますが、場合によっては、管理者が指定した制限より大きいメッセージを拒否しなければならないこともあります (たとえば、サービス拒否のアタックを回避するためなど)。

blocklimit、linelimit、および sourceblocklimit キーワードは、絶対的なサイズ制限を実施するために使用されます。これらのキーワードの後ろには、それぞれ 1 つの整数値が必要です。

blocklimit キーワードは、1 つのメッセージに許可するブロックの最大数を指定します。MTA は、これよりも多いブロックを含むメッセージがチャネルのキューに入れられるのを拒否します。1 つの MTA ブロックは通常 1024 バイトで、これは MTA オプションファイルにある BLOCK_SIZE オプションを使用して変更することができます。

sourceblocklimit キーワードは、受信メッセージに許可するブロックの最大数を指定します。MTA は、これよりも多いブロックを含むメッセージがチャネルのキューに入れられるのを拒否します。つまり、blocklimit は宛先チャネルに、sourceblocklimit はソースチャネルに適用されます。1 つの MTA ブロックは通常 1024 バイトで、これは MTA オプションファイルにある BLOCK_SIZE オプションを使用して変更することができます。

linelimit キーワードは、1 つのメッセージに許可する最大行数を指定します。MTA は、この数以上の行を含むメッセージがチャネルのキューに入れられるのを拒否します。これらの 2 つのキーワード (blocklimit と linelimit) は、必要に応じて同時に指定することができます。

同じ制限をすべてのチャネルに課すためには、LINE_LIMIT および BLOCK_LIMIT オプションを使用します。これらの制限は、すべてのチャネルに適用できるという利点があります。したがって、MTA サーバは、メッセージ受信情報を得る前に、それをメールクライアントに知らせることができます。この効果によって、メッセージ拒否の処理を簡略化できるプロトコルもあります。

nolinelimit および noblocklimit チャネルキーワードはデフォルトであり、LINE_LIMIT や BLOCK_LIMIT MTA オプションで適用されている全体的な制限以外の制限がないことを意味します。

制限容量超過ユーザへのメール配信を処理する

キーワード : holdexquota、noexquota

noexquota および holdexquota キーワードは、Berkeley メールボックスユーザ (UNIX) 宛てのメッセージの処理を制御します。ここでいうメッセージとは、ディスク制限容量を超過しているユーザがローカルチャネルのユーザ ID に配信したメッセージです。

noexquota は MTA に、制限容量を超過したユーザ宛てのメッセージを、差出人に返送するように指示します。holdexquota は MTA に、制限容量超過ユーザ宛てのメッセージを保留にするように指示します。これらのメッセージは、配信可能になるまで、またはタイムアウトになってメッセージ返送ジョブによって返送されるまで、MTA キュー内に保持されます。

MTA キュー領域でのファイル作成

この節では、MTA キュー領域でのファイル作成を指定してディスクリソースを制御するキーワードを説明します。この章には、以下の節があります。

- 268 ページの「複数のアドレスを処理する方法を制御する」
- 269 ページの「複数のサブディレクトリにチャネルメッセージキューを拡散する」

複数のアドレスを処理する方法を制御する

キーワード : multiple、addrsperfile、single、single_sys

MTA では、キューに入れられたそれぞれのメッセージに複数の宛先アドレスを使用できるようになっています。チャネルプログラムの中には、1 つの受取人を持つメッセージ、限定された数の受取人を持つメッセージ、あるいは 1 つのメッセージコピーにつき 1 つの宛先システムを持つメッセージしか処理できないものもあります。たとえば、SMTP チャネルのマスタープログラムは、(1 つのチャネルがすべての SMTP トラフィックのために使用されるのにもかかわらず) 1 つのトランザクションで 1 つのリモートホストとの接続を確立するため、そのホストへのアドレスのみが処理されます。

もう 1 つの例として、SMTP サーバの中には、1 度に処理できる受取人の数を制限し、このタイプのエラーを処理できないものもあります。

multiple、addrsperfile、single、および single_sys キーワードは、複数のアドレスを処理する方法を制御するために使用できます。single キーワードは、各宛先アドレス用にメッセージのコピーを 1 つずつ作成するように指定します。single_sys キーワードは、各宛先システム用にメッセージのコピーを 1 つずつ作成します。multiple キーワードは、デフォルトではチャネル全体のメッセージのコピーを 1 つ作成します。

注	どちらのキーワードを使用しても、メッセージがキューに入れられる各チャネルごとに最低 1 つずつメッセージのコピーが作成されることに注意してください。

addrsperfile キーワードは、チャネルのキューにある 1 つのメッセージファイルに関連付けられる受取人の最大数に制限を付けるために使用されます。これによって、1 つの操作で処理される受取人の数が制限されます。このキーワードは、1 つのメッセージファイルに許可する受取人アドレスの最大数を指定する 1 つの整数引数を必要とします。この数に達すると MTA は自動的にそれらを処理するために追加のメッセージファイルを作成します。(一般に、デフォルトの multiple キーワードはメッセージファイル内の受取人数に制限を課さないことを意味します。ただし SMTP チャネルのデフォルトは 99 です。)

複数のサブディレクトリにチャネルメッセージキューを拡散する

キーワード : subdirs

デフォルトでは、チャネルのキューに入れられたすべてのメッセージは、ディレクトリ /imta/queue/*channel-name* にあるファイルとして格納されます。この「*channel-name*」はチャネルの名前です。ただし、TCP/IP チャネルのように、たくさんのメッセージを処理し、処理を待つメッセージファイルをたくさん格納しがちなチャネルの場合は、それらのメッセージファイルを複数のサブディレクトリに拡散するようなファイルシステムを使った方が処理能力が向上する可能性があります。この機能を提供するのが subdirs チャネルキーワードです。チャネルのメッセージを拡散するサブディレクトリの数を指定する整数を、このキーワードの後ろに付けます。

```
tcp_local single_sys smtp subdirs 10
```

ログ記録とデバッグを設定する

この節では、ログ記録とデバッグのキーワードについて説明します。

ログ記録のキーワード

キーワード: `logging`、`nologging`

MTA は、メッセージがキューに出し入れされるたびにログを作成することができます。`logging` および `nologging` キーワードは、チャネルごとのメッセージログの作成を制御します。デフォルト設定では、すべてのチャネルに対してログが作成されます。特定のチャネルに対してログの作成を無効にするには、チャネル定義で logging の代わりに `nologging` キーワードを設定します。

ログ記録については、第 13 章「ログ記録とログ解析」を参照してください。

デバッグのキーワード

キーワード: `master_debug`、`slave_debug`、`nomaster_debug`、`noslave_debug`

チャネルプログラムによっては、デバッグ目的のためにより詳細な診断出力を生成するオプションコードがあるものもあります。このチャネルごとのデバッグとの出力の生成機能を有効にするためのチャネルキーワードには 2 種類あります。`master_debug` キーワードはマスタープログラムのデバッグ出力を有効にし、`slave_debug` キーワードはスレーブプログラムのデバッグ出力を有効にします。デフォルトでは `nomaster_debug` および `noslave_debug` が有効になっているため、デバッグ出力は生成されません。

デバッグを有効にすると、デバッグ出力は各チャネルプログラムに関連付けられているログファイルに記述されます。ログファイルの場所はプログラムによって異なりますが、通常はログディレクトリにあります。マスタープログラムのログファイル名は、通常 *x*`_master.log` の形式をとります。ここで *x* はチャネル名です。また、スレーブプログラムのログファイル名は、通常 *x*`_slave.log` の形式をとります。

UNIX では、`master_debug` と `slave_debug` が `1` チャネルに対して有効になっている場合は、ユーザが MTA デバッグ情報を含む `imta_sendmail.log-`*uniqueid* ファイルを、現在のディレクトリに受信できます (ディレクトリに書き込み権がある場合。書き込み権がない場合はデバッグにより stdout に出力)。

Loopcheck を設定する

キーワード : `loopcheck`、`noloopcheck`

`loopcheck` キーワードは、MTA が MTA 自身と通信しているかどうかを確認するために、SMTP EHLO 応答見出しに文字列を入れます。`loopcheck` が設定されている場合、SMTP サーバでは XLOOP 拡張がアドバタイズされます。

XLOOP をサポートする SMTP サーバと通信する場合、MTA の SMTP クライアントにより、アドバタイズされた文字列と MTA の値が比較され、クライアントが SMTP サーバと通信している場合は、メッセージがただちに返送されます。

その他のキーワード

この節では、その他のキーワードを説明します。この章には、以下の節があります。

チャネル動作のタイプ

キーワード : `submitsubmit`

Messaging Server は、RFC 2476 規定のメッセージ送信プロトコルをサポートしています。チャネルを送信専用に設定するには、`submit` キーワードを使用します。これは通常、特別なポートで実行され、メッセージを送信する目的だけに使用される SMTP サーバなどの TCP/IP チャネルに便利です。RFC 2476 では、このようなメッセージ送信に使用するためにポート 587 を確立します。

pipe チャネル

キーワード : `user`

`user` キーワードは、pipe チャネルでどのユーザ名で実行するかを示すのに使用されます。

`user` の引数は、通常小文字に変換されますが、引数に引用符が付けられている場合は、元の大文字と小文字が維持されます。

メールボックスフィルタファイルの場所を指定する

キーワード: filter、nofilter、channelfilter、nochannelfilter、destinationfilter nodestinationfilter、sourcefilter、nosourcefilter、fileinto、nofileinto

filter キーワードは、そのチャネル用のユーザフィルタファイルの場所を指定するために、ネイティブチャネルと ims-ms チャネルに対して使用します。このキーワードは、フィルタファイルの場所を示す URL を引数としてとります。nofilter がデフォルトで、ユーザメールボックスフィルタがそのチャネルに対して有効にならないことを示します。

一般的な MTA チャネルにチャネルレベルのフィルタを指定するには、受信と送信のメッセージに対してそれぞれ sourcefilter と destinationfilter のキーワードを使用します。これらのキーワードは、チャネルフィルタファイルの場所を示す URL を引数としてとります。nosourcefilter と nodestinationfilter がデフォルトで、チャネルのどちらの方向にもチャネルメールボックスフィルタが無効になります。

旧バージョンの channelfilter キーワードと nochannelfilter キーワードは、それぞれ destinationfilter と nodestinationfilter と同義です。

fileinto キーワードは、現在 ims-ms チャネルに対してのみサポートされており、fileinto メールボックスフィルタ演算子が適用された場合、アドレスをどのように変更するかを指定します。ims-ms チャネルの場合、通常の使用方法は以下のとおりです。

```
fileinto $U+$S@$D
```

上の例では、最初のサブアドレスの代わりに、フォルダ名をサブアドレスとして元のアドレスに挿入するように指定しています。

第 9 章

定義済みチャネルを使用する

チャネルによっては iPlanet Messaging Server をインストールした時点ですでに定義されているものもあります (表 9-1 を参照)。この章では、MTA の定義済みチャネルの使い方を説明します。

この章を読む前に、第 6 章「MTA サービスと設定について」をお読みください。`imta.cnf` ファイルの書き換え規則を設定する方法については、第 7 章「書き換え規則を設定する」を参照してください。

この章には、以下の節があります。

表 9-1 定義済みチャネル

チャネル	定義
`l`	**UNIX 専用。**ルーティングの決定および UNIX メールツールを使用したメール送信に使用する
`ims-ms`	メールをローカルストアに配信する
`native`	**UNIX のみ。**`/var/mail` にメールを配信する (Messaging Server は `/var/mail` へのアクセスをサポートしない。ユーザが `/var/mail` ストアのメールにアクセスするには、UNIX ツールを使う必要がある)

表 9-1　定義済みチャネル (続き)

チャネル	定義
pipe	サイト提供のプログラムやスクリプトを介してメールを配信するために使用される。この pipe チャネルによって実行されるコマンドは、管理者が imsimta プログラムのインタフェースを通じて管理する。詳細は、275 ページの「パイプチャネルを使用してメッセージをプログラムに配信するには」を参照してください。
reprocess プロセス (process)	遅延メッセージのオフライン処理に使用されるチャネル。通常、reprocess チャネルはソースまたは宛先チャネルとして公にされない。process チャネルは、ほかの MTA チャネルと同様、公にされる
defragment	断片化された MIME メッセージの修復方法を提供する
変換	MTA を通じて配信されるメッセージを本文部分ごとに変換する
bitbucket	破棄するメッセージに使用される
inactive/deleted	ディレクトリ内でのステータスが非アクティブまたは削除済みになっているユーザへのメッセージの処理に使用される。通常、受信したメッセージを差出人に送り返し、カスタム返送メッセージを送る
hold	ユーザへのメッセージを保留する。ユーザがあるメールサーバから別のサーバに移行された場合などに使用される
autoreply	自動返信および vacation 通知の要求を処理するために使用される
tcp_local tcp_intranet tcp_auth tcp_submit tcp_tas	TCP/IP の上位プロトコルとして SMTP を実装する。マルチスレッド TCP SMTP チャネルには、ディスパッチャ制御下のマルチスレッド SMTP サーバが含まれる。送信された SMTP メールは、必要に応じてジョブコントローラの制御下で動作し、チャネルプログラム tcp_smtp_client によって処理される。 tcp_local はリモート SMTP ホストからのメールを受信する。メールを送信する場合は、スマートホスト / ファイアウォール設定が使われているかどうかによって、直接リモート SMTP ホストに送るか、またはスマートホストファイアウォールシステムに送る tcp_intranet はイントラネット内のメールを送受信する tcp_auth は tcp_local のスイッチチャネルとして使用される。認証されたユーザは、リレーブロックの制約を回避するため tcp_auth チャネルに移される tcp_submit は、送信されたメッセージ (通常の場合はユーザエージェントからのメッセージ) を予約されている送信ポート 587 で受け入る (RFC 2476 を参照) tcp_tas は Unified Messaging を使用するサイト用の特殊なチャネルである

パイプチャネルを使用してメッセージをプログラムに配信するには

メールをメールボックスで受信する代わりにプログラムに転送することができます。たとえば、受け取ったメールをメール保存用プログラムに転送したり、不在通知のような自動応答エージェントに転送することができます。pipe チャネルはサイト提供のユーザごとのプログラムを使用してメッセージを配信します。

プログラムへの配信を行うには、まず pipe チャネルが呼び出せるプログラムを登録する必要があります。登録は imsimta program ユーティリティを使って行います。このユーティリティにより、pipe チャネルが呼び出せる登録コマンドごとに固有の名前が設定されます。これによってエンドユーザが mailprogramdeliveryinfo LDAP 属性の値としてプログラム名を指定できるようになります。

たとえば、UNIX の myprocmail コマンドをユーザが呼び出せるプログラムとして追加するには、imsimta program ユーティリティを使用して以下の例のようにこのコマンドを登録します。この例では、-d *username* という引数を使用して procmail プログラムをユーザとして実行する myprocmail プログラムが登録されます。

```
imsimta program -a -m myprocmail -p procmail -g "-d %s" -euser
```

programs ディレクトリ (*server-instance*/imta/programs) に実行ファイルが存在し、「others」に対して実行権が設定されていることを確認してください。

ユーザがプログラムにアクセスするためには、そのユーザの LDAP エントリに以下の属性および値が含まれている必要があります。

```
maildeliveryoption: program
mailprogramdeliveryinfo:myprocmail
```

imsimta program ユーティリティについては、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

その他の配信プログラムを使用する場合は、そのプログラムが次の終了コードおよびコマンドラインの引数に関する条件を満たしていることを確認してください。

終了コード条件：pipe チャネルが呼び出す配信プログラムは、チャネルがメッセージをキューから出すか、あとで処理するために配信するか、または返送するかを判断できるように、適切なエラーコードを返さなくてはなりません。

サブプロセスが終了コード 0 (EX_OK) で終了した場合は、メッセージが適切に配信されたと認識され、MTA のキューから削除されます。終了コード 71、74、75、または 79 (EX_OSERR、EX_IOERR、EX_TEMPFAIL、または EX_DB) で終了した場合は、一時的なエラーが発生したとみなされ、メッセージの配信は延期されます。その他のコードが返されると、メッセージは配信不能として差出人に返送されます。終了コードは、システムヘッダーファイル sysexits.h 内で定義されています。

コマンドラインの引数：可変引数 (%s) を含め、配信プログラムが使用できる引数の数に上限はありません。可変引数は、ユーザが実行するプログラムの場合はユーザ名を、ポストマスター「inetmail」が実行するプログラムの場合はユーザ名 + ドメイン名を示します。たとえば、次のコマンドラインは `procmail` プログラムを使用してメールを受取人に配信します。

```
/usr/lib/procmail -d %s
```

ネイティブ (/var/mail) チャネルを設定するには

オプションファイルは、ローカルチャネルのさまざまな機能を制御するために使用されます。このローカルチャネルのオプションファイルは MTA の設定ディレクトリに保存し、`native_option` という名前を付けなければなりません (例 : *server_root*`/msg-`*instance*`/imta/config/native_option`)。

オプションファイルは複数の行から構成されており、各行にはそれぞれ 1 つのオプション設定が含まれています。オプション設定は、次の形式で記述されています。

option=value

「*value*」は、オプションの要件に基づいて文字列または整数のいずれかとなります。

表 9-2　ローカルチャネルのオプション

オプション	説明
`FORCE_CONTENT_LENGTH` (0 または 1。UNIX のみ)	`FORCE_CONTENT_LENGTH=1` の場合、MTA によりローカルチャネルに配信されるメッセージに Content-length: ヘッダー行が追加され、「From」が行の最初にある場合、チャネルで ">From" シンタックスが使用されなくなります。これによって、ローカルの UNIX メールが Sun のより新しいメールツールとの互換性を持つようになりますが、ほかの UNIX メールツールとの互換性がなくなることもあります。
`FORWARD_FORMAT` (文字列)	ユーザの `.forward` ファイルの場所を指定します。%u 文字列は、この部分が各ユーザ ID で置換されることを示します。%h 文字列は、この部分が各ユーザのホームディレクトリで置換されることを示します。このオプションが明示的に指定されていない場合、デフォルトの動作は次と同様になります。 `FORWARD_FORMAT=%h/.forward`

表 9-2　ローカルチャネルのオプション (続き)

オプション	説明
REPEAT_COUNT (整数) SLEEP_TIME (整数)	MTA が新しいメールを配信しようとするときに、ユーザの新しいメールファイルがほかのプロセスによってロックされている場合、これらのオプションによって、ローカルプログラムが試行すべき再試行の回数と頻度を制御することができます。指定された回数の再試行が行われてもファイルを開くことができなかった場合、メッセージはローカルのキューに残され、次にローカルのチャネルが新しいメッセージを配信するときに再試行されます。 The REPEAT_COUNT オプションは、メールファイルを開こうとする試行が何回行われるかを制御します。REPEAT_COUNT のデフォルトは 30 (30 回の試行) です。 SLEEP_TIME オプションは、チャネルプログラムが何秒間隔で試行を繰り返すかを制御します。SLEEP_TIME は 2 (2 秒の間隔で再試行) にデフォルト設定されています。
SHELL_TIMEOUT (整数)	.forward を完成するために、チャネルがユーザのシェルコマンドを待機する時間 (秒数) を制御します。この時間が経過すると、「user の command を完了するシェルコマンドのタイムアウト」という旨のメッセージとともに、元の差出人にエラーメッセージが返送されます。デフォルトは 600 (10 分) です。
SHELL_TMPDIR (ディレクトリ固有)	シェルコマンドに配信を行う際に、ローカルチャネルが一時ファイルを作成する場所を制御します。デフォルトでは、一時ファイルはユーザのホームディレクトリに作成されます。このオプションを使用すると、管理者は一時ファイルを別の (単一の) ディレクトリに作成するように選択できます。たとえば、以下のようになります。 SHELL_TMPDIR=/tmp

hold チャネルを使って一時的にメッセージを保留するには

hold チャネルは、一時的に受信不能になっている宛先へのメッセージを保留するためのチャネルです。一時的な受信不能の原因としては、ユーザ名が変更されている最中であったり、メールボックスが別のホストやドメインに移行されている最中であることが考えられます。原因はほかにもありますが、この 2 つがもっとも一般的なものです。

hold チャネルにメッセージを保留するには、hold にユーザの maildeliveryoption 値の 1 つを設定します。その他の maildeliveryoption 値はすべて無視され (maildeliveryoption は複数値を持つ属性)、そのユーザへのメッセージは hold チャネルにルーティングされます。

ほかのチャネルとは異なり、hold チャネルのマスタープログラムは自動的に起動するように設定されていません。hold チャネルのキュー内のメッセージは、管理者が hold_master プログラムを呼び出すまでそのままの状態で待機します。

変換チャネル

conversion チャネルを使うと、MTA を通じて配信されるメッセージで指定する本文部分ごとの変換を任意に行うことができます。(本文部分とメッセージは違います。メッセージには複数の本文部分が含まれることがあります。たとえば添付ファイルにも本文部分があります。) 変換処理は、サイトが提供した任意のプログラムやコマンド手順で行うことができます。処理内容には、テキストや画像形式の変換、ウィルススキャン、言語変換などがあります。MTA で通信するさまざまなメッセージ形式を変換することができ、特定の処理やプログラムをメッセージの本文部分に指定することができます。

この章を利用するには、チャネルの概念を理解している必要があります (102 ページの「チャネル」を参照)。conversion チャネルを使ったウィルススキャンの補足情報は、iPlanet Messaging Server マニュアルの Web サイトの下部にある iPlanet Messaging Server のテクニカルノートを参照してください。

変換チャネルの実行には、A) 処理するメッセージ通信を選択し、B) 処理するメッセージの不一致の状態を特定する、という 2 つの手順があります。以下に詳細を説明します。

注 デフォルトの変換チャネルは MTA 設定ファイル内 (imta.cnf) に自動的に作成されます。このチャネルはそのままの状態で使用することができます。変更する必要はありません。

MIME の概要

変換チャネルは MIME (Multipurpose Internet Mail Extension) ヘッダー行を幅広く利用します。このため、メッセージ構築と MIME ヘッダーフィールドに関する知識が必要です。MIME の詳細については、RFC 1806、2045 ～ 2049、2183 を参照してください。ここでは、MIME について簡単に説明します。

メッセージの構築

メッセージは基本的にヘッダーと本文で構成されています。ヘッダーはメッセージの最初にあり、日付、件名、差出人、受取人など、一定の制御情報を含んでいます。ヘッダーの後ろに空白行が入り、その後ろはすべて本文です。MIME では、複数の本文部分を持つさらに複雑なメッセージを作成する方法を指定します。本文部分を入れ子にすることもできます。このようなメッセージは複数部分メッセージと呼ばれ、すでに説明したように、メッセージの本文部分ごとに変換チャネルで変換されます。

MIME ヘッダー

MIME 仕様では、本文部分のヘッダー行が定義されています。ヘッダー行には、`MIME-Version`、`Content-type`、`Content-Transfer-Encoding`、`Content-ID`、および `Content-disposition` があります。変換チャネルでよく使用されるヘッダーは `Content-type` と `Content-disposition` です。以下に MIME ヘッダー行の例を示します。

```
Content-type: APPLICATION/wordperfect5.1;name=Poem.wpc
Content-transfer-encoding:BASE64
Content-disposition: attachment; filename=Poem.wpc
Content-description: "Project documentation Draft1 wordperfect format"
```

Content-type ヘッダー

MIME `Content-Type` ヘッダーは本文部分の内容を表します。`Content-Type` ヘッダー形式と実際の例を次に示します。

`Content-type:` *type/subtype; parameter1=value; parameter2=value...*

type は本文部分の内容の種類を表します。種類には、`Text`、`Multipart`、`Message`、`Application`、`Image`、`Audio`、`Video` などがあります。

subtype は コンテンツタイプをさらに詳しくしたものです。Content-type にはそれぞれ独自のサブタイプがあります。たとえば次のようなものがあります。text/plain、application/octet-stream、image/jpeg。MIME メールの Content Subtype は IANA (Internet Assigned Numbers Authority) で割り当てられ、一覧表示されています。割り当て一覧は http://www.isi.edu/in-notes/iana/assignments/media-types/media-types で参照することができます。

parameter は Content-type/subtype の組み合わせに固有のものです。たとえば、charset および name パラメータは以下のようになります。

```
Content-type:text/plain; charset=us-ascii
Content-type:application/msword; name=temp.doc
```

charset パラメータでは、テキスト形式メッセージの文字セットを指定します。name パラメータでは、データをファイルに書き込む場合に使用するファイル名を指定します。

注 Content-Type 値、subtypes、およびパラメータ名では大文字と小文字が区別されます。

Content-disposition ヘッダー

MIME Content-disposition ヘッダーで本文部分のプレゼンテーション情報がわかります。通常、添付ファイルに追加され、添付ファイルの本文部分を表示するのか (inline)、コピーするファイル名として表示するのか (attachment) を指定します。Content-disposition ヘッダーの形式は次のとおりです。

```
Content-disposition: disposition_type; parameter1=value;parameter2=value...
```

disposition_type は通常 inline (本文部分を表示) または attachment (保存ファイルとして表示) です。attachment には通常パラメータ filename があり、ここでファイル保存で推奨される名前を指定します。

Content-disposition ヘッダーの詳細については、RFC 2183 を参照してください。

変換処理のトラフィックを選択する

MTA チャネルとは異なり、通常、変換チャネルはアドレスや MTA 書き換え規則では指定されていません。代わりに、メッセージは CONVERSIONS マッピングテーブル (imta_tailor ファイルの IMTA_MAPPING_FILE パラメータで指定される) を使って変換チャネルに送られます。テーブルへのエントリには次のような形式があります。

```
IN-CHAN=source-channel;OUT-CHAN=destination-channel;CONVERT Yes/No
```

MTA はそれぞれのメッセージを処理する際、CONVERSIONS マッピングテーブルがあれば使用します。*source-channel* がメッセージを発信するチャネルで、*destination-channel* がメッセージの宛先となるチャネルであるとすれば、CONVERT に続くアクションが実行されます (Yes を選択すると、MTA はメッセージを *destination-channel* から変換チャネルに変換します。一致するものがなければ、メッセージは通常の宛先チャネルのキューに入ります)。

注	user@conversion.*localhostname* または user@conversion という形式のアドレスは、CONVERSIONS マッピングテーブルにかかわらず、変換チャネルを通してルーティングされます。

以下の例では、発信元も宛先もインターネットである非内部メッセージをすべて変換チャネルにルーティングします。

```
CONVERSIONS
   IN-CHAN=tcp_local;OUT-CHAN=*;CONVERT   Yes
   IN-CHAN=*;OUT-CHAN=tcp_local;CONVERT     Yes
```

最初の行は tcp_local チャネルから受信するメッセージを処理します。次の行は tcp_local チャネルに送信するメッセージを処理します。tcp_local チャネルはインターネットで送受信するメッセージをすべて処理します。デフォルトでは変換チャネルを経由しないので、ほかのメッセージが変換チャネルを通ることはありません。

これは基本テーブルです。複数のインターネット送信用 tcp_* チャネルを使う場合や、複数のインターネット受信用 tcp_* チャネルを使う場合など、カスタマイズされた設定のサイトでは不十分な場合もあります。

変換処理を制御するには

メッセージは変換チャネルに送信されると、本文部分ごとに処理されます。処理は MTA conversions ファイルで制御されます。このファイルは imta_tailor ファイル (デフォルトのディレクトリは *server_root*/msg-*instance*/imta/conversions) の IMTA_CONVERSION_FILE オプションファイルで指定します。エントリを構成する conversions ファイルで、どの形式の本文部分をどのように処理するかを制御します。

各エントリは 1 つまたは複数の行で構成され、各行には 1 つまたは複数の name=*value* パラメータ句が含まれています。パラメータ句の値は MIME 規則に一致しています。最終行以外のすべての行は、セミコロン (;) で終了する必要があります。このファイルでは、1 行に入力できる文字数が 252 バイトに制限されています。円記号 (¥) を継続文字として使用すれば、1 つの論理行を複数の行に分割することができます。エントリは、セミコロンで終了していない行や空白行が 1 行以上挿入されているところで終了します。

conversion ファイルエントリの簡単な例を次に示します。

コード例 9-1 conversion ファイルエントリ

```
out-chan=ims-ms; in-type=application; in-subtype=wordperfect5.1;
  out-type=application; out-subtype=msword; out-mode=block;
  command="/usr/bin/convert -in=wordp -out=msword 'INPUT_FILE' ¥
'OUTPUT_FILE'"
```

out-chan=ims-ms; in-type=application; in-subtype=wordperfect5.1 は本文部分を表します。つまり変換される部分の種類を指定しています。各部分のヘッダーが読み取られ、Content-Type: とその他のヘッダー情報が抽出されます。次に conversion ファイルのエントリが最初から最後まで順番にスキャンされます。その際、in-* パラメータや OUT-CHAN パラメータがあればチェックされます。すべてのパラメータが処理中の本文部分に対応する情報と一致すれば、command= や delete= 句で指定した変換が実行され、out-* パラメータが設定されます。

一致するものがなければ、その本文部分は次の conversions ファイルエントリと照合されます。本文部分がすべてスキャンされ処理されると、一致するものがあった場合は、メッセージは次のチャネルに送られます。一致するものがなければ、何も処理されないまま、メッセージは次のチャネルに送られます。

out-chan=ims-ms は、ims-ms チャネル宛のメッセージ部分だけを変換するように指定します。in-type=application および in-subtype=wordperfect5.1 により、メッセージ部分の MIME Content-type ヘッダーは application/wordperfect5.1 に指定されます。

メッセージ部分に in-* パラメータを追加すると詳細に指定することができます (表 9-5 を参照)。このエントリは、次のような MIME ヘッダー行を持つメッセージ部分の変換アクションをトリガします。

```
Content-type:APPLICATION/wordperfect5.1;name=Draft1.wpc
Content-transfer-encoding:BASE64
Content-disposition:attachment; filename=Draft1.wpc
Content-description:"Project documentation Draft1 wordperfect format"
```

3 つの conversion ファイルがコード例 9-1 のパラメータを指定したら、次の 2 つのパラメータ out-type=application および out-subtype=msword は置換 MIME ヘッダー行を「処理済み」の本文部分に添付するよう指定します。out-type=application および out-subtype=msword は、送信メッセージの MIME Content-type/subtype が application/msword となるように指定します。

in-type と out-type は同じパラメータなので out-type=application は必要ありません。変換チャネルのデフォルトは送信本文部分の元の MIME ラベルであるからです。送信本文部分の MIME ラベルを追加するには、出力パラメータを指定します。

out-mode=block (コード例 9-1) は、サイト提供のプログラムが返すファイル形式を指定します。つまり、ファイルの保存方法と、変換チャネルが返されたファイルを読み取る方法を指定します。たとえば、html ファイルはテキストモードで保存されますが、.exe プログラムや zip ファイルはブロック / バイナリモードで保存されます。モードは、読み取り中のファイルが一定の保存形式にあることを表しています。

コード例 9-1 の最後のパラメータ

```
command="/usr/bin/convert -in=wordp -out=msword 'INPUT_FILE'
'OUTPUT_FILE'"
```

は、本文部分でのアクションを指定します。command= パラメータは、本文部分でプログラムが実行されることを指定します。/usr/bin/convert は架空のコマンド名です。-in=wordp および -out=msword は入力テキストと出力テキストの形式を指定する架空のコマンドライン引数です。INPUT_FILE および OUTPUT_FILE は、元の本文部分を持つファイルと変換後の本文部分を保存するプログラムがあるファイルとを指定する変換チャネル環境パラメータ (284 ページの「変換チャネル環境変数の使い方」を参照) です。

本文部分でコマンドを実行する代わりに、command パラメータの場所に DELETE=1 を使えばメッセージ部分を簡単に削除することができます。

変換チャネルの情報フロー

情報フローは次のようになります。本文部分を含むメッセージが変換チャネルに入ってきます。変換チャネルはメッセージをパースして、本文部分を 1 つずつ処理します。次に変換チャネルは本文部分が適格であるかどうかを判断します。つまり、MIME ヘッダー行を指定パラメータと比較して処理するかどうかを決定します。本文部分が適格であると判断されれば、変換処理が始まります。MIME や本文部分の情報を変換スクリプトに渡す場合は、「情報引き渡しパラメータ」で指定した環境変数 (表 9-3) に保存します。

この時点で、「アクションパラメータ」で指定したアクションを本文部分に実行します。一般的には、本文部分を削除するか、スクリプトで囲んだプログラムに渡します。本文部分はスクリプトで処理されると変換チャネルに戻され、処理後のメッセージに組み込まれます。スクリプトは、変換チャネルの「出力オプション」を使って情報を変換チャネルに送信することもできます。この情報には、出力本文部分に追加する新しい MIME ヘッダー行、メッセージの差出人に返送するエラーテキスト、MTA にメッセージのバウンス、削除、保留などのアクション開始を指示する命令などがあります。

最後に、変換チャネルは「出力パラメータ」で指定されたように出力本文部分のヘッダー行を置き換えます。

変換チャネル環境変数の使い方

メッセージ本文部分を処理する場合、MIME ヘッダー行情報や本文部分全体をサイト提供のプログラムとやり取りすると便利なことがあります。たとえば、あるプログラムでメッセージ本文部分以外に `Content-type` と `Content-disposition` ヘッダー行情報が必要であるとします。一般にサイト提供のプログラムに入力されているのは、主にファイルから読み取るメッセージ本文部分です。プログラムで本文部分が処理されると、変換チャネルが読み取りファイルに書き込まれます。このような情報の受け渡しは、変換チャネル環境変数を使って行われます。

環境変数は、`parameter-symbol-*` パラメータや定義済みの変換チャネル環境変数のセット (288 ページの表 9-4 を参照) を使って、`conversions` ファイルで作成することができます。

次の `conversions` ファイルエントリと受信ヘッダーでは、サイト提供のプログラムに環境変数を使って MIME 情報を渡す方法が示されています。

conversions ファイルエントリ :

```
in-channel=*; in-type=application; in-subtype=*;
  parameter-symbol-0=APPARENT_NAME; parameter-copy-0=*;
  dparameter-symbol-0=APPARENT_FILENAME; dparameter-copy-0=*;
  message-header-file=2; original-header-file=1;
  override-header-file=1; override-option-file=1;
  command="/bin/viro-scan500.sh 'INPUT_FILE' 'OUTPUT_FILE'"
```

受信ヘッダー :

```
Content-type:APPLICATION/msword; name=Draft1.doc
Content-transfer-encoding:BASE64
Content-disposition:attachment; filename=Draft1.doc
Content-description:"Project documentation Draft1 msword format"
```

in-channel=*; in-type=application; in-subtype=* は、application 形式の任意の入力チャネルから受信したメッセージ本文部分が処理されることを示します。

parameter-symbol-0=APPARENT_NAME は、最初の Content-type パラメータの値 (この例では Draft1.doc) が APPARENT_NAME という環境変数に保存されることを示します。

parameter-copy-0=* は、入力本文部分の Content-type パラメータがすべて出力本文部分にコピーされることを示します。

dparameter-symbol-0=APPARENT_FILENAME は、最初の Content-disposition パラメータの値 (この例では Draft1.doc) が APPARENT_FILENAME という環境変数に保存されることを示します。

dparameter-copy-0=* は、入力本文部分の Content-disposition パラメータがすべて出力本文部分にコピーされることを示します。

message-header-file=2 は、メッセージの元のヘッダー全体 (最初と最後のメッセージヘッダー) が環境変数 MESSAGE_HEADERS で指定したファイルに書き込まれることを示します。

original-header-file=1 は、封入する MESSAGE/RFC822 部分の元のヘッダーが環境変数 INPUT_HEADERS で指定したファイルに書き込まれることを示します。

override-header-file=1 は、MIME ヘッダーが環境変数 OUTPUT_HEADERS で指定したファイルから読み取られ、封入する MIME 部分の元のヘッダーを無視することを示します。$OUTPUT_HEADERS は、変換実行中に作成される実行時テンポラリファイルです。このファイルはサイト提供のプログラムで使用され、変換処理中に変更されたヘッダーが保存されます。本文部分が変換チャネルで再構築される際に、このファイルからヘッダー行が読み取られます。

override-option-file=1 は、変換チャネルが OUTPUT_OPTIONS 環境変数によって名前が付けられたファイルから変換チャネルのオプションを読み取ることを表します。287 ページの「変換チャネル出力オプションを使用するには」を参照してください。

command="SERVER_ROOT/msg-INSTANCE/bin/viro-scan500.sh" は、メッセージ本文部分で実行するコマンドを示します。

表 9-3 変換チャネル環境変数

環境変数	説明
INPUT_ENCODING	元の本文部分に存在するエンコーディング
INPUT_FILE	元の本文部分を含むファイルの名前。サイト提供のプログラムはこのファイルを読み取る
INPUT_HEADERS	本文部分の元のヘッダー行を含むファイルの名前。サイト提供のプログラムはこのファイルを読み取る
INPUT_TYPE	入力メッセージ部分の MIME Content-type
INPUT_SUBTYPE	入力メッセージ部分の MIME コンテンツサブタイプ
INPUT_DESCRIPTION	入力メッセージ部分の MIME content-description
INPUT_DISPOSITION	入力メッセージ部分の MIME content-disposition
MESSAGE_HEADERS	封入するメッセージ (本文部分だけに限らない) の元の一番外側のヘッダーまたは本文部分がすぐに封入する MESSAGE/RFC822 部分のヘッダーを含むファイル名。サイト提供のプログラムはこのファイルを読み取る
OUTPUT_FILE	サイト提供のプログラムで出力結果を保存するファイル名。サイト提供のプログラムはこのファイルを作成して書き込む
OUTPUT_HEADERS	サイト提供のプログラムで封入部分の MIME ヘッダー行を保存するファイル名。サイト提供のプログラムはこのファイルを作成して書き込む。ファイルには option=value 行ではなく実際のヘッダー行が含まれ、最後の行は空白行となる
OUTPUT_OPTIONS	サイト提供のプログラムで変換チャネルのオプションを読み取るファイル名。287 ページの「変換チャネル出力オプションを使用するには」を参照

変換チャネル出力オプションを使用するには

変換チャネル出力オプション (表 9-4) は動的な変数で、変換スクリプトから変換チャネルに情報と特定の指示を渡します。たとえば、本文部分の処理中にメッセージをバウンスさせてスクリプトから変換チャネルに指示を出し、返送メッセージに「このメッセージにはウィルスが含まれている」というエラーテキストを追加させることができます。

出力オプションは、指定した変換エントリに OVERRIDE-OPTION-FILE=1 を設定すると開始されます。次に必要に応じて出力オプションはがスクリプトで設定され、環境変数ファイル OUTPUT_OPTIONS に保存されます。このスクリプトが本文部分の処理を終了すると、変換チャネルは OUTPUT_OPTIONS ファイルからオプションを読み取ります。

OUTPUT_OPTION 変数は、変換チャネルがオプションを読み取るファイル名です。通常、この変数は情報を渡す実行時テンポラリファイルとして使用されます。以下に、出力オプションを使ってウィルスを送信した差出人にエラーメッセージを返すスクリプトの例を示します。

```
/usr/local/bin/viro_screen2k $INPUT_FILE   # run the virus screener

if [ $? -eq 1 ]; then
   echo "OUTPUT_DIAGNOSTIC='Virus found and deleted.'" > $OUTPUT_OPTIONS
   echo "STATUS=178029946" >> $OUTPUT_OPTIONS
else
   cp $INPUT_FILE $OUTPUT_FILE # Message part is OK
fi
```

この例では、$OUTPUT_OPTIONS で定義されたファイルにシステム診断メッセージとステータスコードが追加されます。$OUTPUT_OPTIONS テンポラリファイルを読み出すと、次のように表示されます。

```
OUTPUT_DIAGNOSTIC="Virus found and deleted."
STATUS=178029946
```

OUTPUT_DIAGNOSTIC='Virus found and deleted' の行は、メッセージに「Virus found and deleted」というテキストを追加するように変換チャネルに指示していることを表します。

178029946 は pmdf_err.h ファイルごとの PMDF__FORCERETURN ステータスで、server-root/bin/msg/imtasdk/include/pmdf_err.h に格納されています。このステータスコードは、差出人にメッセージを返送するように変換チャネルに指示しています。特定の指示の使い方について詳細は 290 ページの「変換チャネル出力を使ってメッセージのバウンス、削除、保留を行うには」を参照してください。

出力オプションのリストを以下に示します。

表 9-4　変換チャネル出力オプション

オプション	説明
OUTPUT_TYPE	出力メッセージ部分の MIME コンテンツタイプ
OUTPUT_SUBTYPE	出力メッセージ部分の MIME コンテンツサブタイプ
OUTPUT_DESCRIPTION	出力メッセージ部分の MIME コンテンツ説明
OUTPUT_DIAGNOSTIC	変換チャネルによってメッセージが強制的にバウンスされる場合、差出人に送信するメッセージの一部に含まれるテキスト
OUTPUT_DISPOSITION	出力メッセージ部分の MIME content-disposition
OUTPUT_ENCODING	MIME content transfer encoding で、出力メッセージ部分で使用される
OUTPUT_MODE	変換チャネルが出力メッセージ部分を書き出す際に使用する MIME Mode で、受取人が出力メッセージ部分を読む取る際に使用するモード
STATUS	コンバータの終了ステータス。変換チャネルで何らかのアクションを開始する場合の典型的な指示。指示の全リストは server-root/bin/msg/imtasdk/include/pmdf_err.h で参照できる

封入する MESSAGE/RFC822 部分のヘッダー

メッセージ部分で変換を実行する場合、変換チャネルは封入する MESSAGE/RFC822 部分のヘッダーにアクセスします。封入された MESSAGE/RFC822 部分がない場合は、メッセージヘッダーにアクセスします。ヘッダーの情報はサイト提供のプログラムに役立つことがあります。

ORIGINAL-HEADER-FILE=1 を含むエントリが選択されると、封入する MESSAGE/RFC822 部分の元のヘッダー行はすべて OUTPUT_HEADERS 環境変数で表したファイルに書き込まれます。OVERRIDE-HEADER-FILE=1 であれば、変換チャネルは OUTPUT_HEADERS 環境変数で表したファイルの内容を読み取り、封入された部分のヘッダーとして使用します。

変換エントリからマッピングテーブルに呼び出すには

out-parameter-* 値は、任意に名前を設定したマッピングテーブルに保存したり、検索したりすることができます。この機能は、クライアントが送信する添付ファイル名を変更する場合に便利です。クライアントが送信する場合は、添付ファイルの種類 (postscript、msword、text など) にかかわらず、att.dat のような汎用名が使用されるからです。ほかのクライアント (たとえば Outlook) が拡張子を読み取ってその部分が開けるように、その部分の名前を変更する一般的な方法です。

マッピングテーブルからパラメータ値を検索するシンタックスは次のとおりです。

'mapping-table-name:mapping-input[*$Y,$N*]'

$Y はパラメータ値を返します。何も見つからなかった場合や一致するものとして $N が返された場合、変換ファイルのエントリ内のパラメータは、無視されるか空白文字列として扱われます。一致するものがない場合や $N の場合は、変換エントリ自体が強制終了します。

次のようなマッピングテーブルがあるとします。

```
X-ATT-NAMES

   postscript        temp.PS$Y
   wordperfect5.1    temp.WPC$Y
   msword            temp.DOC$Y
```

このマッピングテーブルの変換エントリは次のとおりで、添付ファイルの指定ファイル名を汎用ファイル名に置換します。

```
out-chan=tcp_local; in-type=application; in-subtype=*;
   in-parameter-name-0=name; in-parameter-value-0=*;
   out-type=application; out-subtype='INPUT-SUBTYPE';
   out-parameter-name-0=name;
   out-parameter-value-0="'X-ATT-NAMES:¥¥'INPUT_SUBTYPE¥¥''";
   command="cp  'INPUT_FILE'  'OUTPUT_FILE'"
```

この例で out-chan=tcp_local; in-type=application; in-subtype=* は、処理するメッセージが tcp_local チャネルからのもので、application/* の content-type ヘッダーが含まれていることを示します (* は任意のサブタイプ)。

また in-parameter-name-0=name; in-parameter-value-0=* は、メッセージに最初のパラメータ形式として name=* が含まれていることを示します (* は任意のパラメータ値)。

out-type=application; は、メッセージ処理後の MIME Content-type パラメータが application であることを示します。

out-subtype='INPUT-SUBTYPE'; は、本文部分処理後の MIME subtype パラメータが INPUT-SUBTYPE 環境変数であることを示しています。これは入力 subtype のオリジナル値です。このように、変更できます。

```
Content-type:application/xxxx; name=foo.doc
```

から

```
Content-type:application/msword; name=foo.doc
```

に変更する場合は、次のようにします。

```
out-type=application; out-subtype=msword
```

out-parameter-name-0=name; は、出力本文部分の最初の MIME Content-type パラメータが name= 形式であることを示します。

out-parameter-value-0='X-ATT-NAMES:¥¥'INPUT_SUBTYPE¥¥''; は、最初の MIME subtype パラメータ値をとり、マッピングテーブル X-ATT-NAMES で subtype と一致するものを検索します。一致するものがあれば、name パラメータは X-ATT-NAMES マッピングテーブルで指定された新しい値を受け取ります。つまりパラメータの形式が msword であれば、name パラメータは temp.DOC になります。

変換チャネル出力を使ってメッセージのバウンス、削除、保留を行うには

この節では、変換チャネルのオプションを使ってメッセージのバウンス、削除、保留を行う方法を説明します。基本手順は次のとおりです。

1. 該当する変換ファイルエントリに OVERRIDE-OPTION-FILE=1 を設定します。変換チャネルで OUTPUT_OPTIONS ファイルの出力オプションを読み取ります。
2. 変換スクリプトを使い、特定のメッセージ本文部分に必要なアクションを決定します。
3. スクリプトで、OUTPUT_OPTIONS ファイルに STATUS=*directive_code* オプションを記述しアクションに対する指示を指定します。

指示の全リストは server_root/bin/msg/imtasdk/include/pmdf_err.h に記載されています。以下に、変換チャネルでよく使用される指示を示します。

名前	16 進数値	10 進数値
PMDF__FORCEHOLD	0x0A9C86AA	178030250
PMDF__FORCERETURN	0x0A9C857A	178029946
PMDF__FORCEDELETE	0x0A9C8662	178030178

指示の関数を例を用いて説明します。

メッセージをバウンスさせるには

変換チャネルを使ってメッセージをバウンスさせるには、該当する conversions ファイルエントリに OVERRIDE-OPTION-FILE=1 を設定し、変換スクリプトに次の行を追加します。

```
echo "STATUS=178029946" >> $OUTPUT_OPTIONS
```

バウンスさせるメッセージに短いテキスト文字列を追加する場合は、変換スクリプトに次の行を追加します。

```
echo OUTPUT_DIAGNOSTIC=text-string >> $OUTPUT_OPTIONS
```

次にテキスト文字列の例を示します。「お使いのマシンから送信されたメッセージにはウィルスが含まれていましたが、削除されました。電子メールの添付ファイルを実行する場合は注意してください。」

メッセージ部分を条件付きで削除するには

メッセージ部分は、含まれている内容によって条件付きで削除すると便利な場合があります。これは出力オプションで実行できます。逆に、DELETE=1 変換パラメータ句を使うとメッセージ部分が無条件に削除されます。

出力オプションを使ってメッセージ部分を削除するには、該当するファイルエントリに OVERRIDE-OPTION-FILE=1 を設定し、変換スクリプトに次の行を追加します。

```
echo "STATUS=178030178" >> $OUTPUT_OPTIONS
```

メッセージを保留にするには

メッセージは、含まれている内容によって条件付きで保留にすると便利な場合があります。出力オプションを使ってメッセージ部分を削除するには、該当するファイルエントリに OVERRIDE-OPTION-FILE=1 を設定し、変換スクリプトに次の行を追加します。

```
echo "STATUS=178030250" >> $OUTPUT_OPTIONS
```

これにより、変換チャネルキューに .HELD ファイルとしてメッセージを保留にするように、変換チャネルに指定します。

変換チャネルの例

以下の例にある CONVERSIONS マッピングと変換規則のセットを使うと、架空のチャネル tcp_docuprint に送られた GIF、JPEG、BITMAP ファイルが自動的に PostScript に変換されます。変換の際には架空の /usr/bin/ps-converter.sh が使用されることもあります。この例には、WordPerfect 5.1 ファイルを Microsoft Word ファイルに変換する規則も含まれています。

```
CONVERSIONS

      IN-CHAN=*;OUT-CHAN=tcp_docuprint;CONVERT    Yes
```

```
!
out-chan=ims-ms; in-type=application; in-subtype=wordperfect5.1;
  out-type=application; out-subtype=msword; out-mode=block;
  command="/bin/doc-convert -in=wp -out=msw   'INPUT_FILE'  'OUTPUT_FILE'"

out-chan=tcp_docuprint; in-type=image; in-subtype=gif;
  out-type=application; out-subtype=postscript; out-mode=text;
  command="/bin/ps-convert -in=gif -out=ps   'INPUT_FILE'  'OUTPUT_FILE'"

out-chan=tcp_docuprint; in-type=image; in-subtype=jpeg;
  out-type=application; out-subtype=postscript; out-mode=text;
  command="/bin/ps-convert -in=jpeg -out=ps   'INPUT_FILE'  'OUTPUT_FILE'"

out-chan=tcp_docuprint; in-type=image; in-subtype=bitmap;
  out-type=application; out-subtype=postscript; out-mode=text;
  command="/bin/ps-convert -in=bmp -out=ps   'INPUT_FILE'  'OUTPUT_FILE'"
```

表 9-5　変換パラメータ

パラメータ	説明
指定用パラメータ (変換する前にメッセージを照合するパラメータを指定)	
OUT-CHAN, OUT-CHANNEL	変換用に照合するチャネルを出力する (ワイルドカード使用可)。このエントリで指定した変換は、メッセージが指定したチャネルに送信される場合にのみ実行される
IN-CHAN, IN-CHANNEL	変換用に照合するチャネルを入力する (ワイルドカード使用可)。このエントリで指定した変換は、メッセージが指定したチャネルから送信される場合にのみ実行される
IN-TYPE	変換用に照合する MIME タイプを入力する (ワイルドカード使用可)。このエントリで指定した変換は、このフィールドが本文部分の MIME タイプに一致した場合にのみ実行される
IN-SUBTYPE	変換用に照合する MIME サブタイプを入力する (ワイルドカード使用可)。このエントリで指定した変換は、このフィールドが本文部分の MIME サブタイプに一致した場合にのみ実行される
IN-PARAMETER-NAME-*n*	変換用に照合する MIME Content-Type パラメータ名を入力する。*n* = 0, 1, 2.... である。このパラメータを IN-PARAMETER-VALUE-*n* とともに使用すると、名前と値でパラメータを特定できる
IN-PARAMETER-VALUE-*n*	対応する IN-PARAMETER-NAME の MIME Content-Type パラメータ値を入力して変換用に照合する。このエントリで指定した変換は、このフィールドが本文部分の Content-Type パラメータリストの対応するパラメータに一致した場合にのみ実行される (ワイルドカード使用可)
IN-PARAMETER-DEFAULT-*n*	パラメータがない場合に、MIME Content-Type パラメータのデフォルト値を入力する。本文部分に IN-PARAMETER-VALUE-*n* が指定されていない場合に、IN-PARAMETER-VALUE-n テストのデフォルト値として使用される
IN-DISPOSITION	変換用に照合する MIME Content-Disposition を入力する
IN-DPARAMETER-NAME-*n*	変換用に照合する MIME Content-Disposition パラメータ名を入力する。*n* = 0, 1, 2.... である。このパラメータを IN-DPARAMETER-VALUE-*n* とともに使用すると、名前と値でパラメータを特定できる
IN-DPARAMETER-VALUE-*n*	対応する IN-DPARAMETER-NAME の MIME Content-Disposition パラメータ値を入力して変換用に照合する。このエントリで指定した変換は、このフィールドが本文部分の Content-Disposition: パラメータリストの対応するパラメータに一致した場合にのみ実行される (ワイルドカード使用可)

表 9-5　変換パラメータ (続き)

パラメータ	説明
IN-DPARAMETER-DEFAULT-*n*	パラメータがない場合に、MIME Content-Disposition パラメータのデフォルト値を入力します。本文部分に IN-DPARAMETER-VALUE-*n* が指定されていない場合に、IN-DPARAMETER-VALUE- n テストのデフォルト値として使用される
IN-DESCRIPTION	変換用に照合する MIME Content-Description を入力する
IN-SUBJECT	封入する MESSAGE/RFC822 部分から件名を入力する
出力パラメータ (本文部分の変換後の出力設定を指定)	
OUT-TYPE	出力 MIME タイプが入力 MIME タイプと異なる場合に、MIME タイプを出力する
OUT-SUBTYPE	出力 MIME サブタイプが入力サブタイプと異なる場合に、MIME サブタイプを出力する
OUT-PARAMETER-NAME-*n*	MIME Content-Type パラメータ名を出力する。*n* = 0, 1, 2, ...。
OUT-PARAMETER-VALUE-*n*	OUT-PARAMETER-NAME-*n* に対応する 出力 MIME Content-Type パラメータの値を出力する
PARAMETER-COPY-*n*	本文入力部分の Content-Type パラメータリストから本文出力部分の Content-Type: パラメータリストにコピーする Content-Type パラメータのリスト。*n*=0, 1, 2, ...。IN-PARAMETER-NAME-*n* 句で一致した MIME パラメータ名と同じパラメータ名を使用する
OUT-DISPOSITION	出力 MIME Content-Description が入力 MIME Content-Disposition と異なる場合に、MIME Content- Disposition を出力する
OUT-DPARAMETER-NAME-*n*	MIME Content-Disposition パラメータ名を出力する。*n*=0, 1, 2...
OUT-DPARAMETER-VALUE-*n*	OUT-DPARAMETER-NAME-*n* に対応する MIME Content-Disposition パラメータの値を出力する
DPARAMETER-COPY-*n*	本文入力部分の Content-Type: パラメータリストから本文出力部分の Content-Type: パラメータリストにコピーする Content-Type: パラメータのリスト。*n*=0, 1, 2, ...。IN-PARAMETER-NAME-*n* 句で一致した MIME パラメータ名をコピーする引数とする。引数にはワイルドカードを使用することができる。特に、* という引数は、元の Content-Disposition: パラメータをすべてコピーすることを示す
OUT-DESCRIPTION	出力 MIME Content-Description が入力 MIME Content-Description と異なる場合に、MIME Content-Description を出力する
OUT-MODE	変換ファイルを読み取って保存するモード。BLOCK (バイナリ形式および実行型形式) と TEXT がある

表 9-5　変換パラメータ (続き)

パラメータ	説明
OUT-ENCODING	メッセージが再組立される場合、変換されたファイルに適用するエンコーディング
アクションパラメータ (メッセージ部分のアクションを指定する)	
COMMAND	変換を実行するためのコマンド。変換を実行するためのコマンド。これは必須パラメータで、コマンドを指定しないとこのエントリは無視される
DELETE	0 または 1 に設定します。このフラッグが設定されている場合は、メッセージ部分が削除される (メッセージにこの部分しかない場合は、1 つの空白のテキスト部分に置き換えられる)
RELABEL	RELABEL=1 では、Output パラメータで指定した MIME ラベルに変更される。Relabel=0 では何も変更されない。通常、ラベルの変更は間違ったラベルに対して行う (たとえば Content-type: application/octet-stream から Content-type: application/msword への変更)。ユーザがその部分をファイルに保存してプログラムで開くのではなく、その部分をダブルクリックして開けるようにするものである
SERVICE-COMMAND	SERVICE-COMMAND=command は、MIME メッセージ全体 (MIME ヘッダーと内容本文部分) で動作するサイト提供の手順を実行する。また、ほかの CHARSET-CONVERSION 操作や conversion チャネルの操作とは異なり、サービスコマンドは独自で MIME 逆アセンブリ、デコード、再エンコード、および再アセンブリを行う。このフラグが付いていると、変換チャネルの処理中にエントリが無視される。その代わり、SERVICE-COMMAND エントリは文字セット変換の処理中に実行される
TAG	メールリスト CONVERSION_TAG パラメータで設定されているタグを入力する
情報引き渡しパラメータ (サイト提供プログラムと情報のやりとりを行う)	
DPARAMETER-SYMBOL-*n*	Content-disposition パラメータ値が存在する場合に保存される環境変数。*n* = 0, 1, 2... それぞれの DPARAMETER-SYMBOL-*n* は、Content-Disposition: パラメータリストから順番に (たとえば n=0 は最初のパラメータ、n=2 は 2 番目のパラメータ) 抽出され、指定した環境変数に使用してサイト提供のプログラムを実行する
PARAMETER-SYMBOL-*n*	Content-Type パラメータ値が存在する場合に保存される環境変数。*n* = 0, 1, 2...。それぞれの PARAMETER-SYMBOL-*n* は、Content-Type: パラメータリストから順番に (たとえば n=0 は最初のパラメータ、n=2 は 2 番目のパラメータ) 抽出され、同じ名前の環境変数に使用してサイト提供のプログラムを実行する。IN-PARAMETER-NAME-*n* 句で一致した MIME パラメータ名に変換する変数名を引数とする

表 9-5　変換パラメータ (続き)

パラメータ	説明
MESSAGE-HEADER-FILE	環境変数 MESSAGE_HEADERS で指定したファイルに対してメッセージの元のヘッダーをすべてまたは一部書き込む。書き込まない場合もある。1 に設定するとすぐに本文部分を封入する元のヘッダーは環境変数 MESSAGE_HEADERS で指定したファイルに書き込まれる。2 に設定すると、メッセージの元のヘッダー全体 (最初と最後のメッセージヘッダー) がファイルに書き込まれる
ORIGINAL-HEADER-FILE	0 または 1 に設定する。1 に設定した場合は、封入する MESSAGE/RFC822 部分の元のヘッダー (本文部分ではない) が環境変数 OUTPUT_HEADERS で表されるファイルに書き込まれる。
OVERRIDE-HEADER-FILE	0 または 1 に設定する。1 に設定した場合は、MIME ヘッダー行は変換チャネルによって環境変数 OUTPUT_HEADERS から読み取られ、封入する MIME 部分の元のヘッダー行を無視する
OVERRIDE-OPTION-FILE	OVERRIDE-OPTION-FILE=1 の場合、変換チャネルは OUTPUT_OPTIONS 環境変数のオプションを読み取る
PART-NUMBER	ドット文字を伴った整数で *a. b. c...* のように表示される。MIME 本文部分の番号を示す

文字セット変換とメッセージの再フォーマット

Messaging Server の基本的なマッピングテーブルの 1 つに、文字セット変換テーブルがあります。CHARSET-CONVERSION という名のこのテーブルは、チャネル間における文字セット変換やメッセージフォーマット変換の種類を指定するために使用されます。

多くのシステムでは、文字セットおよびメッセージフォーマットの変換は不必要なため、このテーブルが使われることはありません。しかし、文字セット変換の必要性が生じる場合もあります。

CHARSET-CONVERSION マッピングテーブルは、メッセージフォーマットを変換するためにも使用され、多数の非 MIME フォーマットを MIME に変換することができます。MIME エンコードおよび構造に変更を加えることもできます。これらのオプションは、MIME または MIME のサブセットだけをサポートするシステムにメッセージを送る際に使用されます。また、場合によっては、MIME フォーマットから非 MIME フォーマットへの変換も可能です。

MTA は 2 つの方法によって CHARSET-CONVERSION マッピングテーブルをプローブします。1 回目のプローブは、MTA がメッセージフォーマットを変換すべきか、また変換する場合はどのフォーマットオプションを使用すべきかを決定するために実行されます。(フォーマット変換が指定されていない場合、特定の文字セットへの変換に関するチェックは行われません)。このプローブには、以下のような形式の入力文字列が使用されます。

IN-CHAN=*in-channel*;OUT-CHAN=*out-channel*;CONVERT

in-channel はソースチャネル (メッセージの送信元)、*out-channel* は宛先チャネル (メッセージの送信先) を示します。一致するソースチャネルおよび宛先チャネルがある場合は、その結果がカンマで区切られたキーワードリストの文字列として表示されます。表 9-6 にキーワードの一覧を示します。

表 9-6 CHARSET-CONVERSION マッピングテーブルのキーワード

キーワード	説明
Always	*out-channel* に送信する前にメッセージが変換チャネルを通過する場合でも、変換を実行する
Appledouble	Appledouble フォーマット以外の MacMIME フォーマットを Appledouble フォーマットに変換する
Applesingle	Applesingle フォーマット以外の MacMIME フォーマットを Applesingle フォーマットに変換する
BASE64	MIME エンコーディングを BASE64 に切り替える
Binhex	Binhex フォーマット以外の MacMIME フォーマット、または Macintosh タイプおよび Mac クリエータ情報を含む部分を Binhex フォーマットに変換する
Block	MacMIME フォーマット部分からデータフォークのみを抽出する
Bottom	message/rfc822 本文部分 (転送メッセージ) をメッセージ内容部分とヘッダー部分に「フラット化」する
Delete	message/rfc822 本文部分 (転送メッセージ) をメッセージ内容部分に「フラット化」し、転送ヘッダーを削除する
Level	重複するマルチパートレベルをメッセージから削除する
Macbinary	Macbinary フォーマット以外の MacMIME フォーマット、または Macintosh のタイプや Mac クリエータ情報を含む部分を Macbinary フォーマットに変換する
No	変換を無効にする
QUOTED-PRINTABLE	MIME エンコードを QUOTED-PRINTABLE に切り替える

表 9-6　CHARSET-CONVERSION マッピングテーブルのキーワード (続き)

キーワード	説明
Record,Text	テキスト部分を 80 バイトのところで折り返す
Record,Text= n	テキスト / プレーン部分を n バイトのところで折り返す
RFC1154	メッセージを RFC 1154 フォーマットに変換する
Top	message/rfc822 本文部分 (転送メッセージ) をヘッダー部分とメッセージ内容部分に「フラット化」する
UUENCODE	MIME エンコードを X-UUENCODE に切り替える
Yes	変換を有効にする

文字セットの変換

プローブを行い、メッセージフォーマットを変換する必要があると判断した場合、MTA はメッセージにおける各部分のチェックを開始します。テキスト部分はすべて検出され、その文字セットのパラメータは 2 回目のプローブに使用されます。ただし、変換が必要であると判断されるまで 2 回目のプローブは行われません。2 回目のプローブを行うための入力文字列は以下のとおりです。

IN-CHAN=*in-channel* ; OUT-CHAN=*out-channel* ; IN-CHARSET=*in-char-set*

in-channel と *out-channel* の部分は前述の例と同じです。*in-char-set* は該当する部分の文字セット名を示します。この 2 回目のプローブで一致するものがない場合、文字セットの変換は行われません (ただし、フォーマットの変換、たとえば MIME 構造への変換などは、最初のプローブで一致したキーワードに基づいて行われます)。一致するものが見つかった場合は、以下の文字列が返されます。

OUT-CHARSET=*out-char-set*

この場合、out-char-set は *in-char-set* が示す文字セットに変換されます。これらの文字セットは、MTA テーブルディレクトリに含まれる文字セット定義テーブル charsets.txt 内で定義されているものでなくてはなりません。文字セットがこのファイル内で適切に定義されていないと、変換は行われません。しかし、このファイルの中には現在もっとも利用度の高い数百種の文字セットが定義されているため、特に心配する必要はないでしょう。charsets.txt ファイルの詳細については、imsimta chbuild (UNIX および NT) ユーティリティの説明を参照してください。

すべての条件が満たされると、MTA は文字セットマッピングを作成し、変換を実行します。変換されたメッセージ部分のラベルは、変換後の文字セット名に変更されます。

メッセージフォーマットの変換

前述したように、CHARSET-CONVERSION マッピングテーブルは MIME フォーマットと数種のメーカー独自のメールフォーマット間における添付ファイルの変換にもかかわりがあります。

以下の各項では、CHARSET-CONVERSION マッピングテーブルによって可能なその他のメッセージフォーマット変換の例を紹介します。

非 MIME バイナリ添付ファイルの変換

メッセージの処理にかかわるチャネルで CHARSET-CONVERSION が有効になっている場合、MIME 以外の非標準フォーマットを使用しているメール、たとえば Microsoft Mail (MSMAIL) SMTP ゲートウェイからのメールは、自動的に MIME フォーマットに変換されます。tcp_local チャネルが存在する場合は通常、このチャネルが Microsoft Mail SMTP ゲートウェイからのメッセージを受信します。以下の例は、ローカルユーザ宛てのメッセージのフォーマット変換を有効にするものです。

```
CHARSET-CONVERSION

   IN-CHAN=tcp_local;OUT-CHAN=ims-ms;CONVERT          Yes
```

すべてのチャネルに対してフォーマット変換を有効にするには、OUT-CHAN=ims-ms を OUT-CHAN=* に変更します。ただし、こうすると tcp_local チャネルからのメールがすべてチェックされることになるため、特定のチャネルに限定する場合より、処理時間が長くなる可能性があります。

さらに、このように無差別な変換を設定すると、エンベロープおよび関連する転送情報部分のみを変換すべきメッセージ (たとえばシステムを通過するだけのメッセージなど) に対してまで広範な変換処理を行うことになりかねません。

MIME を Microsoft Mail SMTP ゲートウェイが理解できるフォーマットに変換するには、MTA 設定ファイルで Microsoft Mail SMTP ゲートウェイ専用のチャネル (tcp_msmail など) を設定し、マッピングファイルに以下の内容を追加します。

```
CHARSET-CONVERSION

   IN-CHAN=*;OUT-CHAN=tcp_msmail;CONVERT        RFC1154
```

MIME ヘッダーのラベル変更

ユーザエージェントやゲートウェイによっては、より正確な MIME ヘッダーを作成するために十分な情報があるにもかかわらず、比較的無益な MIME ヘッダーを作成するものもあります。もっとも良い方法はそのようなエージェントやゲートウェイの設定を適切に変更することですが、それが不可能な場合には有用な MIME ヘッダーを構築するように MTA を設定します。

最初のプローブの際に CHARSET-CONVERSION マッピングテーブルが Yes または Always キーワードを返した場合、MTA は conversions ファイルが存在するかどうかを確認します。ファイルが存在する場合、MTA はそのファイルをチェックして RELABEL=1 という記述があるかどうかを確認し、ある場合はそのエントリの指定に従って MIME ラベルを変換します。

たとえば、以下のような CHARSET-CONVERSION テーブルと MTA conversions ファイルのエントリの組み合わせなら、メッセージは tcp_local チャネルから ims-ms チャネルにルーティングされます。さらに、受信時の MIME ラベルが application/octet-stream でファイル名パラメータの拡張子が ps または msw の場合には、それぞれ application/postscript または application/msword という新しいラベルが付けられます (より正確なこのラベルは、元のユーザエージェントやゲートウェイがメッセージに付けておくべきものです)。

CHARSET CONVERSION テーブル

```
CHARSET-CONVERSION

   IN-CHAN=tcp_local;OUT-CHAN=mr_local;CONVERT            Yes
```

MTA CONVERSIONS ファイル エントリ

```
out-chan=ims-ms; in-type=application; in-subtype=octet-stream;
  in-parameter-name-0=name; in-parameter-value-0=*.ps;
  out-type=application; out-subtype=postscript;
     parameter-copy-0=*; relabel=1

out-chan=ims-ms; in-type=application; in-subtype=octet-stream;
  in-parameter-name-0=name; in-parameter-value-0=*.msw;
  out-type=application; out-subtype=msword;
     parameter-copy-0=* relabel=1
```

MacMIME フォーマットの変換

Macintosh ファイルには、Macintosh 特有の情報を含むリソースフォークと、ほかのプラットフォームで使用できるデータを含むデータフォークの 2 つの部分があります。さらに、Macintosh ファイルの転送には一般に 4 種類の異なるフォーマットが使用されるため、Macintosh ファイルを転送するにはより複雑な処理が必要となります。

Applesingle、Binhex、および Macbinary フォーマットは、Macintosh リソースフォークと Macintosh データフォークを 1 つにエンコードしたものから成り立っています。Appledouble フォーマットの場合は、リソースコードとデータフォークがそれぞれ独立した部分として存在しています。このため、Macintosh 以外のプラットフォームでは、リソースフォーク部分を無視してデータフォーク部分のみを使用できる Appledouble がもっとも便利です。逆に、Macintosh への送信には、ほかの 3 種類のフォーマットが便利です。

MTA は、これらの Macintosh フォーマット間の変換を実行することができます。MTA は `CHARSET-CONVERSION` キーワードである `Appledouble`、`Applesingle`、`Binhex`、および `Macbinary` によって MacMIME フォーマット部分をそれぞれ multipart/appledouble、application/applefile、application/mac-binhex40、または application/macbinary の MIME フォーマットに変換します。さらに、`Binhex` または `Macbinary` キーワードは、MIME Content-type: ヘッダーに `X-MAC-TYPE` および `X-MAC-CREATOR` パラメータを含む特定の MacMIME 以外のフォーマットへの変換も要求します。`CHARSET-CONVERSION` キーワードの `Block` は、MTA に対し、MacMIME フォーマット部分のデータフォークのみを抽出し、リソースフォークを破棄するよう要求します (ただし、このキーワードを使用すると一部の情報が失われるため、`Appledouble` キーワードの使用をお勧めします)。

たとえば、以下の `CHARSET-CONVERSION` テーブルは `ims-ms` チャネルにメッセージを配信する場合に Appledouble フォーマットへの変換を MTA に指示します。

```
CHARSET-CONVERSION

  IN-CHAN=*;OUT-CHAN=l;CONVERT            Appledouble
```

この場合、すでに MacMIME フォーマットが使用されている部分のみが Appledouble フォーマットに変換されます。

Appledouble または Block フォーマットへの変換には、元の Macintosh ファイルに含まれる Macintosh クリエータおよびタイプ情報に基づいて Appledouble または Block フォーマットの部分のデータフォークに付ける MIME ラベルを指定するために、`MAC-TO-MIME-CONTENT-TYPES` マッピングテーブルが使用されることもあります。このテーブルのプローブには、「フォーマット | タイプ | クリエータ | ファイル名」形式が使用されます。フォーマットの値には SINGLE、BINHEX、MACBINARY のどれかが指定され、タイプの値には Macintosh タイプ情報 (16 進)、クリエータの値には Macintosh クリエータ情報 (16 進)、そしてファイル名の値には実際のファイル名が指定されます。

たとえば、ims-ms チャネルにメッセージを送る場合に Appledouble フォーマットに変換し、MACBINARY または BINHEX 部分から MS Word または PostScript に変換されたドキュメントに特定の MIME ラベルを付けるには、以下のテーブルが適切です。

```
CHARSET-CONVERSION

  IN-CHAN=*;OUT-CHAN=ims-ms;CONVERT     Appledouble

MAC-TO-MIME-CONTENT-TYPES

! PostScript
    MACBINARY|45505346|76677264|*     APPLICATION/POSTSCRIPT$Y
    BINHEX|45505346|76677264|*        APPLICATION/POSTSCRIPT$Y
! Microsoft Word
    MACBINARY|5744424E|4D535744|*     APPLICATION/MSWORD$Y
    BINHEX|5744424E|4D535744|*        APPLICATION/MSWORD$Y
```

マッピングエントリのテンプレート (右側) に $Y フラグが設定されていない場合、指定したラベルは付けられません。MTA テーブルディレクトリ内の mac_mappings.sample ファイルには、その他の種類の添付ファイルに関するサンプルエントリが記載されています。

MacMIME 以外のフォーマットが使用されている部分を Binhex または Macbinary フォーマットに変換するには、X-MAC-TYPE および X-MAC-CREATOR MIME Content-type: パラメータ値が必要です。通常これらのパラメータ値を持たない部分にそれを強要するために MIME ラベルの変換を実行することも可能です。

サービス変換

MTA の変換サービス機能をサイト提供のプロシージャと一緒に使用すると、新しい形式のメッセージを作成することができます。前述の CHARSET-CONVERSION や conversion チャネルの場合は個別の MIME メッセージ部分を操作しますが、変換サービスはすべての MIME メッセージ部分 (MIME ヘッダーと内容) および MIME メッセージ全体を操作します。また、ほかの CHARSET-CONVERSION 操作や conversion チャネルの操作とは異なり、変換サービスは独自で MIME 逆アセンブリ、デコード、再エンコード、および再アセンブリを行います。

ほかの CHARSET-CONVERSION 操作と同様に、変換サービスは CHARSET-CONVERSION マッピングテーブルを通じて有効化されます。CHARSET-CONVESION マッピングテーブルを最初にプローブした結果が Yes または Always キーワードの場合、MTA は conversions ファイルが存在するかどうかをチェックします。conversions ファイルが存在する場合は、ファイル内に SERVICE-COMMAND を指定するエントリがあるかどうかを確認し、ある場合はそれを実行します。conversions ファイルのエントリの形式は以下のとおりです。

```
in-chan=channel-pattern;
  in-type=type-pattern; in-subtype=subtype-pattern;
  service-command=command
```

ここでコマンド文字列に注目してください。これは、たとえばドキュメントコンバータを呼び出すなどのサービス変換を行うために必要なコマンドです。このコマンドが実行されると、変換を必要とするメッセージを含む入力ファイルが処理され、新しいメッセージテキストを含む出力ファイルが生成されます。UNIX では、コマンドが成功した場合には 0、失敗した場合にはその他の値で終了する必要があります。

入力ファイル名、出力ファイル名、メッセージのエンベロープ受取人アドレスを含むファイルの名前などを伝達するためには、環境変数が使われます。これらの 3 つの環境変数は以下のとおりです。

- INPUT_FILE - 処理する入力ファイルの名前
- OUTPUT_FILE - 生成する出力ファイルの名前
- INFO_FILE - エンベロープ受取人アドレスを含むファイルの名前

これらの環境変数の値は、通常の方法でコマンドラインに代入することができます。UNIX では、変数名の前に「$」記号を挿入します。

第 10 章

メールのフィルタリングとアクセス制御

この章では、メールサービスへのアクセス制御方法、およびマッピングテーブルと SSR (サーバ側規則) を使ったメールのフィルタリング方法について説明します。

システムレベルで特定の差出人または宛先のメールを拒否したり、特定のユーザ間のメッセージトラフィックに複雑な規制を設けたり、あるいはユーザ自身が受信メッセージのフィルタリング (メッセージヘッダーの内容に基づくメッセージ拒否など) を設定することができます。

エンベロープレベルの制御が望ましい場合には、マッピングテーブルを使ってメールをフィルタリングできます。ヘッダーベースの制御が望ましい場合、またはユーザによる独自の制御設定には、サーバ側規則を使った一般的なメールのフィルタリングアプローチが適切です。

この章は、以下の 2 つの部分から構成されています。

第 1 部 マッピングテーブル

第 2 部 メールボックスフィルタ

第 1 部 マッピングテーブル

第 1 部には以下の節があります。

- マッピングテーブルを使ってアクセスを制御する
- アクセス制御はいつ適用されるのか
- アクセス制御マッピングをテストするには
- SMTP リレーを追加するには
- SMTP リレーブロッキングを設定する
- 多数のアクセスエントリを処理する

- アクセス制御マッピングテーブルのフラグ

マッピングテーブルを使ってアクセスを制御する

メールサービスへのアクセスを制御するには、一定のマッピングテーブルを使用します。マッピングテーブル (表 10-1) を使用することにより、だれがメールを送信または受信できるのか、あるいは送受信できるのかを制御することができます。マッピングファイルの一般的な情報および使用方法については、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

表 10-1 に、この節で説明するマッピングテーブルの一覧を示します。

表 10-1 アクセス制御マッピングテーブル

マッピングテーブル	説明
SEND_ACCESS (307 ページを参照)	エンベロープ From アドレス、エンベロープ To アドレス、ソースおよび宛先チャネルに基づいて、受信接続をブロックする。書き換えやエイリアス展開などの処理が行われてから、To アドレスを調べる
ORIG_SEND_ACCESS (307 ページを参照)	エンベロープ From アドレス、エンベロープ To アドレス、ソースおよび宛先チャネルに基づいて、受信接続をブロックする。書き換え後、エイリアス展開の前に To アドレスを調べる
MAIL_ACCESS (309 ページを参照)	SEND_ACCESS テーブルと PORT_ACCESS テーブルを組み合わせた情報に基づいて受信接続をブロックする場合に使用する。SEND_ACCESS のチャネルとアドレス、および PORT_ACCESS の IP アドレスとポート番号に関する情報が基準となる
ORIG_MAIL_ACCESS (309 ページを参照)	ORIG_SEND_ACCESS テーブルと PORT_ACCESS テーブルを組み合わせた情報に基づいて受信接続をブロックする場合に使用する。ORIG_SEND_ACCESS のチャネルとアドレス、および PORT_ACCESS の IP アドレスとポート番号に関する情報が基準となる
FROM_ACCESS (310 ページを参照)	エンベロープ From アドレスに基づいてメールをフィルタリングする。このテーブルは、To アドレスが不適切な場合に使用する
PORT_ACCESS (313 ページを参照)	IP 番号に基づいて受信接続をブロックする

もっとも一般的なのは、MAIL_ACCESS および ORIG_MAIL_ACCESS によるマッピングで、SEND_ACCESS および ORIG_SEND_ACCESS に使用できるアドレスおよびチャネル情報のほか、IP アドレスやポート番号などの PORT_ACCESS マッピングテーブルを介して得られるような情報も得ることができます。

SEND_ACCESS テーブルと ORIG_SEND_ACCESS テーブル

SEND_ACCESS マッピングテーブルと ORIG_SEND_ACCESS マッピングテーブルを使用して、だれがメールを送信または受信できるのか、あるいは送受信できるのかを制御することができます。アクセスチェックは、メッセージエンベロープの From: アドレスおよびエンベロープの To: アドレス、あるいはメッセージがどのチャネルから入ってきたか、そしてどのチャネルから出ていくのかという情報に基づいて行われます。

SEND_ACCESS または ORIG_SEND_ACCESS のマッピングテーブルが存在する場合、MTA を通過するメッセージの各受取人を調べるために、MTA は以下のフォーマットの文字列が記述されているテーブルをスキャンします (縦棒文字「|」の用法に注意してください)。

src-channel|*from-address*|*dst-channel*|*to-address*

src-channel はメッセージをキューに入れるチャネル、*from-address* はメッセージの作成者アドレス、*dst-channel* はキューに入れられたメッセージの宛先となるチャンネル、*to-address* はメッセージの宛先アドレスです。これらの 4 つのフィールド内でアスタリスクを使用すると、そのフィールドの情報 (チャネルやアドレスなど) が任意のデータと一致するようになります。

この場合のアドレスは、エンベロープの From: アドレスとエンベロープの To: アドレスを指しています。SEND_ACCESS の場合は、書き換えやエイリアス展開などの処理が行われてから、エンベロープの To: アドレスが調べられます。ORIG_SEND_ACCESS の場合には、書き換え後、エイリアス展開の前に、メッセージ作成者により指定されたエンベロープの To: アドレスが調べられます。

検索文字列のパターン (テーブルの左側にあるエントリ) が一致すると、そのマッピングの結果出力が調べられます。出力に「$Y」または「$y」フラグが含まれている場合は、その特定の To: アドレスに対しメッセージを入れることが許可されます。出力に「$N」、「$n」、「$F」、または「$f」フラグが含まれている場合は、その特定のアドレスに対しメッセージをキューに入れることが拒否されます。拒否された場合は、オプションの拒否通知テキストをマッピング出力に与えることができます。その文字列は、MTA が発行する拒否通知エラーメッセージに含まれることになります。「$N」、「$n」、「$F」、「$f」以外に文字列が出力されない場合は、デフォルトの拒否通知テキストが使用されます。その他のフラグの説明については、331 ページの「アクセス制御マッピングテーブルのフラグ」を参照してください。

次の例は、mail や Pine などの UNIX ユーザエージェントから送られてきたメール、ローカル l チャネルからの入力、および TCP/IP などのチャネルからメッセージをインターネットに出力するケースを示すものです。ポストマスター以外のローカルユーザは、インターネットからメールを受信できても送信は許可されていないと仮定します。そのような制御を行う 1 つの手段として、図 10-1 に示している SEND_ACCESS マッピングテーブルの使用があります。このマッピングテーブルの例では、ローカルのホスト名が sesta.com であると想定しています。チャネル名「tcp_*」では、ワイルドカードを使って任意の TCP/IP チャネル名 (たとえば tcp_loal) と一致するようにしています。

拒否通知メッセージでは、メッセージ内の空白文字の引用符としてドル記号が使われています。ドル記号を使用しないと、拒否通知メッセージが「Internet postings are not permitted.」とならずに「Internet」だけで終わってしまいます。この例では、ローカルのポスティングに関するほかのソース (PC ベースのメールシステムであるのか、または POP または IMAP クライアントであるのかなど) は無視されていることに注意してください。

図 10-1 SEND_ACCESS マッピングテーブル

```
SEND_ACCESS

  *|postmaster@sesta.com|*|*     $Y
  *|*|*|postmaster@sesta.com     $Y
  l|*@sesta.com|tcp_*|*          $NInternet$ postings$ are$ not$ ¥
    permitted
```

注 MTA による拒否通知エラーテキストが、メッセージの差出人であるユーザに対して実際に提示されるかどうかは、メッセージの送信を試行するクライアントにより異なります。受信 SMTP メッセージを拒否するために SEND_ACCESS を使用した場合、オプションの拒否通知テキストを含む SMTP 拒否通知コードを MTA が発行することはほとんどありません。その情報に基づいてバウンスメッセージを構築し、元の差出人に戻すかどうかは、送信 SMTP クライアントによって決まります。

MAIL_ACCESS マッピングテーブルと ORIG_MAIL_ACCESS マッピングテーブル

MAIL_ACCESS マッピングテーブルは、SEND_ACCESS マッピングテーブルと PORT_ACCESS マッピングテーブルのスーパーセットです。つまり、SEND_ACCESS のチャネルとアドレス、および PORT_ACCESS の IP アドレスとポート番号の情報を組み合わせたものです。同様に、ORIG_MAIL_ACCESS マッピングテーブルは、ORIG_SEND_ACCESS マッピングテーブルと PORT_ACCESS マッピングテーブルのスーパーセットです。MAIL_ACCESS のプローブ文字列フォーマットは以下のとおりです。

port-access-probe-info | *app-info* | *submit-type* | *send_access-probe-info*

同様に、ORIG_MAIL_ACCESS のプローブ文字列フォーマットは以下のとおりです。

port-access-probe-info | *app-info* | *submit-type* | *orig_send_access-probe-info*

上記の *port-access-probe-info* は、受信 SMTP メッセージの場合、PORT_ACCESS マッピングテーブルプローブに通常含まれているすべての情報から成ります。それ以外の場合は空白になります。*app-info* は、SMTP 経由で送信されたメッセージの場合、通常は SMTP です。それ以外の場合は空白になります。*submit-type* は MAIL、SEND、SAML、または SOML のいずれか 1 つで、メッセージが Messaging Server へ送信されてきた方法に対応します。通常、この値は、メッセージとして送信されたことを表す MAIL です。SEND、SAML、または SOML は、ブロードキャスト要求 (またはブロードキャストとメッセージを組み合わせた要求) が SMTP サーバに送信された場合の値です。MAIL_ACCESS マッピングの *send-access-probe-info* は、SEND_ACCESS マッピングテーブルプローブに通常含まれているすべての情報から成ります。同様に、ORIG_MAIL_ACCESS マッピングの *orig-access-probe-info* は、ORIG_SEND_ACCESS マッピングテーブルプローブに通常含まれているすべての情報から成ります。

受信 TCP/IP 接続情報が、チャネルおよびアドレスの情報と同じマッピングテーブルにあると、特定の IP アドレスからのメッセージにどのエンベロープの From: アドレスを表示させるのかなど、何らかの制御を課す場合に便利です。電子メールの偽造を規制したり、ユーザに対し POP および IMAP クライアントの From: アドレス設定を正しく行うように奨励する効果もあります。たとえば、IP アドレスが 1.2.3.1 および 1.2.3.2 から送信されたメッセージに対してのみエンベロープの From: アドレスに vip@siroe.com を表示し、1.2.0.0 サブネット内のシステムから送信されるメッセージにはエンベロープの From: アドレスに siroe.com を表示するようなサイトでは、図 10-2 に示す MAIL_ACCESS マッピングテーブルを使用します。

図 10-2　MAIL_ACCESS マッピングテーブル

```
MAIL_ACCESS

  ! Entries for vip's two systems
  !
  TCP|*|25|1.2.3.1|*|SMTP|MAIL|tcp_*|vip@siroe.com|*|*  $Y
  TCP|*|25|1.2.3.2|*|SMTP|MAIL|tcp_*|vip@siroe.com|*|*  $Y
  !
  ! Disallow attempts to use vip's From:address from other
  ! systems
  !
  TCP|*|25|*|*|SMTP|MAIL|tcp_*|vip@siroe.com|*|*  ¥
      $N500$ Not$ authorized$ to$ use$ this$ From:$ address
  !
  ! Allow sending from within our subnet with siroe.com From:
  ! addresses
  !
  TCP|*|25|1.2.*.*|*|SMTP|MAIL|tcp_*|*@siroe.com|*|*  $Y
  !
  ! Allow notifications through
  !
  TCP|*|25|1.2.*.*|*|SMTP|MAIL|tcp_*||*|*  $Y
  !
  ! Block sending from within our subnet with non-siroe.com
  ! addresses
  !
  TCP|*|25|1.2.*.*|*|SMTP|MAIL|tcp_*|*|*|*  ¥
      $NOnly$ siroe.com$ From:$ addresses$ authorized
```

FROM_ACCESS マッピングテーブル

FROM_ACCESS マッピングテーブルは、だれがメールを送信できるのか、まただれが From: アドレスを認証アドレスに書き換えることができるのかを制御するのに使用します。

FROM_ACCESS マッピングテーブルへの入力プローブ文字列は、MAIL_ACCESS マッピングテーブルのものと似ています。違いは、宛先チャネルとアドレスがないこと、場合によっては認証済み差出人情報があることです。したがって、FROM_ACCESS マッピングテーブルが存在する場合は、メッセージが送信されるたびに Messaging Server によって以下のフォーマットで文字列が記述されているテーブルの検索が行われます (縦棒文字「|」の用法に注意してください)。

port-access-probe-info|*app-info*|*submit-type*|*src-channel*|*from-address*|*auth-from*

上記の *port-access-probe-info* は、受信 SMTP メッセージの場合、PORT_ACCESS マッピングテーブルプローブに通常含まれているすべての情報から成ります。それ以外の場合は空白になります。*app-info* は、SMTP 経由で送信されたメッセージの場合、通常は SMTP です。それ以外の場合は空白になります。*submit-type* は MAIL、SEND、SAML、または SOML のいずれか 1 つで、メッセージが MTA に送られてきた方法に対応します。通常、この値は、メッセージとして送信されたことを表す MAIL です。SEND、SAML、または SOML は、ブロードキャスト要求 (またはブロードキャストとメッセージを組み合わせた要求) が SMTP サーバに送信された場合の値です。*src-channel* はメッセージを発する (メッセージをキューに入れる) チャネル、*from-address* はメッセージの作成者アドレスです。*auth-from* は認証済み作成者アドレスですが、その情報がない場合は空白になります。

プローブ文字列のパターン (テーブルの左側にあるエントリ) が一致した場合は、そのマッピングの結果出力が調べられます。出力に「$Y」または「$y」フラグが含まれている場合には、その特定の To: アドレスに対しメッセージを入れることが許可されます。出力に「$N」、「$n」、「$F」、または「$f」フラグが含まれている場合は、その特定のアドレスに対しメッセージをキューに入れることが拒否されます。拒否された場合は、オプションの拒否通知テキストをマッピング出力に与えることができます。この文字列は、Messaging Server が発行する拒否通知エラーメッセージに含まれることになります。「$N」、「$n」、「$F」、「$f」以外に文字列が出力されない場合は、デフォルトの拒否通知テキストが使用されます。その他のフラグの説明については、331 ページの「アクセス制御マッピングテーブルのフラグ」を参照してください。

FROM_ACCESS は、作成者の情報に基づいてメッセージの送信を許可するかどうかを決定できるだけでなく、エンベロープの From: アドレスを $J フラグで許可したり、authrewrite チャネルキーワードの効果を $K フラグで変更 (受理したメッセージに Sender: ヘッダーアドレスを追加) できます。たとえば、以下のマッピングテーブルを使用し、エンベロープの From: アドレスを最初のものから認証アドレスに置き換えることができます。

```
FROM_ACCESS

  *|SMTP|*|tcp_local|*|        $Y
  *|SMTP|*|tcp_local|*|*       $Y$J$3
```

特定のソースチャネルの authrewrite をゼロ以外の値に設定する効果を変更するために FROM_ACCESS マッピングテーブルを使用する場合、認証アドレスが文字どおりである限り FROM_ACCESS を使用する必要はありません。

たとえば、tcp_local チャネルに authrewrite 2 を設定する場合は、authrewrite だけでこの効果 (文字どおりの認証済みアドレス) を得るのに十分なため、次の FROM_ACCESS マッピングテーブルは不要です。

```
FROM_ACCESS

  *|SMTP|*|tcp_local|*|      $Y
  *|SMTP|*|tcp_local|*|*     $Y$K$3
```

ただし、FROM_ACCESS の本来の目的は、図 10-3 に示すように、より複雑で微妙な変更を行うことにあります。受信メッセージに Sender: ヘッダー行を追加 (SMTP AUTH 認証済み送信者アドレスを表示) したい場合は、authrewrite キーワードだけでも十分です。ただし、SMTP AUTH 認証済み送信者アドレスがエンベロープの From: アドレスと異なる場合にのみ、受信メッセージに Sender: ヘッダー行を強制的に追加したいとします (つまり、アドレスが一致した場合には、Sender: ヘッダー行を追加しません)。さらに、エンベロープの From: にオプションのサブアドレスが含まれているというだけで SMTP AUTH およびエンベロープの From: アドレスを異なるとみなしたいとします。

図 10-3　FROM_ACCESS マッピングテーブル

```
FROM_ACCESS

  ! If no authenticated address is available, do nothing
  *|SMTP|*|tcp_local|*|              $Y
  ! If authenticated address matches envelope From:, do nothing
  *|SMTP|*|tcp_local|*|$2*          $Y
  ! If authenticated address matches envelope From:sans
  ! subaddress, do nothing
  *|SMTP|*|tcp_local|*+*@*|$2*@$4*   $Y
  ! Fall though to...
  ! ...authenticated address present, but didn't match, so force
  ! Sender:header
  *|SMTP|*|tcp_local|*|*             $Y$K$3
```

PORT_ACCESS マッピングテーブル

ディスパッチャは、IP アドレスおよびポート番号に基づいて、受信接続を許可するかどうかを選択できます。ディスパッチャは、起動時に PORT_ACCESS という名前のマッピングテーブルを探します。このファイルが見つかると、ディスパッチャは接続情報を以下のようにフォーマットします。

TCP|*server-address*|*server-port*|*client-address*|*client-port*

ディスパッチャは、すべての PORT_ACCESS マッピングエントリを照合します。マッピングの結果に「$N」または「$F」が含まれている場合には、接続を即座に終了します。それ以外の場合は、接続を許可します。「$N」または「$F」の後ろに拒否通知メッセージが続くことがあります。メッセージがある場合には、接続を断つ前にそのメッセージが送り返されます。メッセージが送り返される前に、その文字列には CRLF ターミネータが追加されることに注意してください。

$< フラグにオプションの文字列が続いており、マッピングプローブが一致しなかった場合は、Messaging Server が文字列を syslog (UNIX) またはイベントログ (NT) に送ります。$> フラグにオプションの文字列が続いており、アクセスが拒否された場合は、Messaging Server が文字列を syslog (UNIX) またはイベントログ (NT) に送ります。LOG_CONNECTION MTA オプションのビット 1 が設定されており、かつ「$N」フラグが設定されて接続が拒否されている場合は、「$T」フラグを指定することにより「T」エントリが接続ログに書き込まれるようになります。LOG_CONNECTION MTA オプ

ションのビット 4 が設定されている場合は、サイト提供のテキストを PORT_ACCESS エントリに提供し、「C」接続ログエントリに含めることが可能です。そのようなテキストを指定するには、エントリの右側に縦棒「|」を 2 つと適切なテキストを挿入します。表 10-2 に使用可能なサービスを表示します。

表 10-2 PORT_ACCESS マッピングテーブル

フラグ	説明
$Y	アクセスを許可する
フラグと引数 (引数の読み取り順序 +)	
$< 文字列	プローブが一致する場合、文字列を syslog (UNIX) またはイベントログ (NT) に送る
$> 文字列	アクセスが拒否された場合、文字列を syslog (UNIX) またはイベントログ (NT) に送る
$N 文字列	アクセスを拒否し、オプションのエラーテキスト文字列を送る
$F 文字列	「$N 文字列」と同じ。アクセスを拒否し、オプションのエラーテキスト文字列を送る
$T テキスト	LOG_CONNECTION MTA オプションのビット 1 が設定されており、かつ「$N」フラグが設定されて接続が拒否されている場合、「$T」フラグを指定することにより、「T」エントリが接続ログに書き込まれるようになる。オプションのテキスト (2 つの縦棒「|」に続けて挿入) は、接続ログエントリに含めることができる
+ 引数を伴うフラグを複数個使用する場合は、引数を縦棒文字「|」で区切り、この表に示されている順序で配置します。	

たとえば、次のマッピングは、単一のネットワークからポート 25 (標準の SMTP ポート) への SMTP 接続だけを許可します。説明テキストは送らずに特定のホストを拒否します。

```
PORT_ACCESS

  TCP|*|25|192.123.10.70|*  $N500
  TCP|*|25|192.123.10.*|*   $Y
  TCP|*|25|*|*              $N500$ Bzzzt$ thank$ you$ for$ ¥
     playing.
```

PORT_ACCESS マッピングテーブルを変更した場合、その変更内容を適用するためにディスパッチャを再起動する必要があります。コンパイルした MTA 設定ファイルを使用している場合は、変更内容を適用するために、先に設定ファイルをリコンパイルしてください。

PORT_ACCESS マッピングテーブルは、特に IP ベースの拒否通知を処理するためのものです。電子メールアドレスレベルでの一般的な制御には、SEND_ACCESS または MAIL_ACCESS マッピングテーブルが適しています。

MTA への IP アドレス接続を制限するには

PORT_ACCESS マッピングテーブルの conn_throttle.so 共有ライブラリを使用すると、特定の IP アドレスが MTA に接続する頻度を制限することができます。特定の IP アドレスによる接続の制限は、サービス拒否による過剰な接続を防ぐ場合などに便利です。

conn_throttle.so は PORT_ACCESS マッピングテーブルで使用されるライブラリで、特定の IP アドレスからの過度の MTA 接続を制限するために使用されます。以下に示すように、設定オプションはすべて接続スロットル共有ライブラリに対するパラメータとして指定されます。

$ [*server_root*/lib/conn_throttle.so,throttle,IP-address,max-rate]

IP-address は、ピリオドで区切られた数字によるリモートシステムのアドレスです。max-rate はこの IP アドレスに対して許可される 1 分当たりの最大接続数です。

throttle の代わりに throttle_p をルーチン名として使用すると、ペナルティが適用されます。throttle_p を使用すると、過去に過度の接続があった場合、接続が拒否されます。たとえば、最大接続数が 100 で、過去 1 分間に 250 の接続が試みられた場合、リモートサイトはその 1 分間における最初の 100 個の接続のあとブロックされ

るだけでなく、次の 1 分間もブロックされます。つまり、1 分が経過するごとに、その 1 分間に試行された接続数と 1 分当たりの許容最大接続数とが比較され、試行接続数が許容最大接続数より大きいと判断された場合、そのリモートシステムはブロックされます。

指定した IP アドレスの接続が 1 分当たりの最大接続数を超えなかった場合、共有ライブラリの呼び出しに失敗します。

1 分当たりの最大接続数を超過した場合は、共有ライブラリの呼び出しに成功しますが、値が返されることはありません。これは $C/$E の組み合わせで行われます。以下に、その例を示します。

```
PORT_ACCESS

  TCP|*|25|*|* ¥
$C$[server_root/lib/conn_throttle.so,throttle,$1,10]¥
$N421$ Connection$ not$ accepted$ at$ this$ time$E
```

説明 :

`$C` により、次のテーブルエントリからマッピングプロセスが続行されます。このエントリの出力文字列が、マッピングプロセスの新しい入力文字列として使用されます。

`$[`*server_root*`/lib/conn_throttle.so,throttle,$1,10]` はライブラリの呼び出しで、`throttle` はライブラリルーチン、`$1` はサーバの IP アドレス、10 は 1 分当たりの接続数のしきい値です。

`$N421$ Connection$ not$ accepted$ at$ this$ time` により、アクセスが拒否され、421 SMTP コード (一時的な接続拒否) とともに、「現在接続は受け付けられません」という旨のメッセージが返されます。

`$E` により、マッピングプロセスが即時に終了します。このエントリからの出力文字列がマッピングプロセスの最終結果として使用されます。

アクセス制御はいつ適用されるのか

Messaging Server は、可能な限り早い段階でアクセス制御マッピングを調べます。実際にどの時点で行われるかは、使用する電子メールプロトコルによって異なります。これは、必要な情報をいつ読み取れるのかという点に依存しているためです。

SMTP プロトコルの場合、FROM_ACCESS による拒否は、送信側が受取人情報やメッセージデータを送信する前に、MAIL FROM: コマンドへの応答として行われます。SEND_ACCESS または MAIL_ACCESS による拒否は、送信側がメッセージデータを送信する前に、RCPT TO: コマンドへの応答として行われます。SMTP メッセージが拒否された場合は、Messaging Server がメッセージデータを受信せずメッセージデータを確認しないため、そのような拒否を処理するためのオーバーヘッドが最小になります。

複数のアクセス制御マッピングテーブルが存在する場合、Messaging Server はそれらをすべて調べます。したがって、FROM_ACCESS、SEND_ACCESS、ORIG_SEND_ACCESS、MAIL_ACCESS、および ORIG_MAIL_ACCESS マッピングテーブルがすべて使用されることがあります。

アクセス制御マッピングをテストするには

imsimta test -rewrite ユーティリティ (特に -from、-source_channel、および -destination_channel オプション) は、アクセス制御マッピングのテストに役立ちます。例として、図 10-4 に、サンプルの SEND_ACCESS マッピングテーブルとその結果としてのプローブを示します。

図 10-4　SEND_ACCESS マッピングテーブルとプローブの例

MAPPING TABLE:

```
SEND_ACCESS

  tcp_local|friendly@siroe.com|l|User@sesta.com     $Y
  tcp_local|unwelcome@varrius.com|l|User@sesta.com  $NGo$ away!
```

PROBE:

```
$ TEST/REWRITE/FROM="friendly@siroe.com" -
_$  /SOURCE=tcp_local/DESTINATION=l User@sesta.com
...
Submitted address list:
  l
    User (SESTA.COM) *NOTIFY FAILURES* *NOTIFY DELAYS* Submitted
notifications list:


$ TEST/REWRITE/FROM="unwelcome@varrius.com" -
_$  /SOURCE=tcp_local/DESTINATION=l User@sesta.com
...
Submitted address list:
Address list error -- 5.7.1 Go away!User@sesta.com

Submitted notifications list:
```

SMTP リレーを追加するには

iPlanet Messaging Server は、デフォルトで、試行された SMTP リレーをブロックするように設定されています。つまり、認証されていない外部ソースから外部アドレスへのメッセージの送信は拒否されます (外部システムとは、サーバがあるホスト以外のシステムのことです)。ほかのシステムはすべて外部システムとみなされることから、SMTP リレーをブロックするこのデフォルト設定はかなり厳しいものだといえます。

IMAP クライアントと POP クライアントが iPlanet Messaging Server システムの SMTP サーバを通じて外部アドレス宛てのメッセージを送信し、SMTP AUTH (SASL) を使って承認を行わない場合、メッセージの送信は拒否されます。このため、内部システムとリレーを許可するサブネットを認識するように設定を変更した方がよいでしょう。

どのシステムとサブネットを内部とみなすかは、通常 INTERNAL_IP マッピングテーブルで制御されます。このテーブルは *<InstanceRoot>*/imta/config/mappings ファイルにあります。

たとえば、IP アドレスが 123.45.67.89 の iPlanet Messaging Server システムの場合、デフォルトの INTERNAL_IP マッピングテーブルは次のようになります。

```
INTERNAL_IP

   $(123.45.67.89/32)   $Y
   127.0.0.1   $Y
   *   $N
```

ここでは、$(IP-pattern/signicant-prefix-bits) シンタックスを使った初期エントリにより、32 ビットの 123.45.67.89 すべてに一致する任意の IP アドレスが、内部として一致および認識されるように指定されています。2 番目のエントリでは、ループバック IP アドレス 127.0.0.1 が内部として認識されます。最後のエントリは、その他のすべての IP アドレスが外部として認識されることを指定しています。すべてのエントリの先頭に、少なくとも 1 つのスペースが必要なことに注意してください。

最後の $N エントリの前に別の IP アドレスを指定して、エントリを追加することもできます。これらのエントリには、必ず左側に IP アドレスまたはサブネット (サブネットの指定には $(.../...) シンタックスを使用) を指定し、右側に $Y を入力します。また、既存の $(.../...) エントリを編集して、より広範囲のサブネットを受け入れるようにすることもできます。

たとえば、このサンプルのサイトにクラス C ネットワークがあり、すべての 123.45.67.0 サブネットを所有する場合は、アドレス照合に使用されるビット数を変更することにより初期エントリを変更できます。次に示すマッピングテーブルでは、32 ビットが 24 ビットに変更されています。これにより、クラス C ネットワークのすべてのクライアントが、SMTP リレーサーバを通してメールをリレーできるようになります。

```
INTERNAL_IP

   $(123.45.67.89/24)   $Y
   127.0.0.1   $Y
   *   $N
```

また、サイトが 123.45.67.80 ～ 123.45.67.99 の範囲の IP アドレスだけを持つ場合は、次のようにします。

```
INTERNAL_IP

! Match IP addresses in the range 123.45.67.80-123.45.67.95
   $(123.45.67.80/28)   $Y
! Match IP addresses in the range 123.45.67.96-123.45.67.99
   $(123.45.670.96/30)   $Y
   127.0.0.1   $Y
   *   $N
```

IP アドレスが特定の $(.../...) テストの条件に一致するかどうかを確認するには、*<InstanceRoot>*/imsimta test -match ユーティリティが便利です。一般に、*<InstanceRoot>*/imsimta test -mapping ユーティリティは、さまざまな IP アドレス入力に対し、INTERNAL_IP マッピングテーブルが望ましい結果を返すかどうかを確認するのに利用できます。

INTERNAL_IP マッピングテーブルを編集したら、必ず *<InstanceRoot>*/imsimta restart コマンド (コンパイルされた設定で実行しない場合) または *<InstanceRoot>*/imsimta refresh コマンド (コンパイルされた設定で実行する場合) を実行して、変更が適用されるようにします。

ファイルのマッピングと一般的なマッピングテーブルの形式、および imsimta コマンドラインユーティリティについては、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

外部サイトの SMTP リレーを許可する

前の項で説明したように、内部 IP アドレスはすべて INTERNAL_IP マッピングテーブルに追加しなければなりません。お使いのシステムやサイトで SMTP リレーを許可する場合は、SMTP リレーを許可する外部アドレスを内部アドレスとともに INTERNAL_IP マッピングテーブルに指定する方法がもっとも簡単です。

ただし、これらの外部システムを実際の内部システムやサイトと区別したい場合 (たとえば、ログやほかの目的のために実際の内部システムとリレーを許可する外部システムを区別する場合) は、ほかの方法でシステムを設定します。

1 つのアプローチとして、これらの外部システムからメッセージを受信する特別のチャネルを設定する方法があります。この設定を行うには、既存の *tcp_internal* チャネルに類似した *tcp_friendly* チャネルを *tcp_friendly-daemon* という正式のホスト名を使って作成します。また、リレーを許可する外部システムの IP アドレスをリストした、INTERNAL_IP マッピングテーブルと同類の FRIENDLY_IP マッピングテーブルを作成します。そして、現在の書き換え規則のすぐあとに新しい書き換え規則を追加します。現在の書き換え規則は次のようになっています。

```
! Do mapping lookup for internal IP addresses
[]    $E$R${INTERNAL_IP,$L}$U%[$L]@tcp_intranet-daemon
```

次の新しい書き換え規則を追加します。

```
! Do mapping lookup for "friendly", non-internal IP addresses []
$E$R${FRIENDLY_IP,$L}$U%[$L]@tcp_friendly-daemon
```

もう 1 つのアプローチとして、ORIG_SEND_ACCESS マッピングテーブルの最後にある $N エントリの前に、次の形式の新しいエントリを追加する方法があります。

```
  tcp_local|*@siroe.com|tcp_local|*    $Y
```

siroe.com は外部アドレスのドメインです。また、次に示すように、ORIG_MAIL_ACCESS マッピングテーブルにエントリを追加します。

```
ORIG_MAIL_ACCESS

   TCP|*|25|$(match-siroe.com-IP-addresses)|*|SMTP|MAIL|    ¥
tcp_local|*@siroe.com|tcp_local|*     $Y
   TCP|*|*|*|*|SMTP|MAIL|tcp_local|*|tcp_local|*    $N
```

$(...) の IP アドレスには、前の項で説明したシンタックスを使用します。ORIG_SEND_ACCESS によるチェックは、アドレスが正常であれば完了します。このため、より厳密なチェック、つまり IP アドレスが siroe.com の IP アドレスに一致した場合にのみ成功する ORIG_MAIL_ACCESS によるチェックを行います。

SMTP リレーブロッキングを設定する

アクセス制御マップを使うことによって、Messaging Server システムが SMTP メールのリレーに利用されるのを防ぐことができます。たとえば、ユーザのメールシステムを利用して何百、何千ものインターネットメールボックスにジャンクメールをリレーしようとする不正操作を阻止できます。

Messaging Server のデフォルトでは、ローカルの POP ユーザおよび IMAP ユーザによるリレーを含むすべての SMTP リレー操作が防止されます。

不正なリレーをブロックする一方、正しいローカルユーザによるリレーを許可するには、2 つのクラスのユーザを識別するように Messaging Server を設定する必要があります。たとえば、POP または IMAP を使用するローカルユーザの場合、SMTP リレー操作は Messaging Server に依存しています。

SMTP リレーを阻止するには、以下のいずれかの操作を行う必要があります。

- 内部メールと外部メールを識別する
- 認証ユーザのメールを識別する
- メールのリレーを防止する

内部のホストとクライアントによる SMTP リレーを可能にするには、`INTERNAL_IP` マッピングテーブルに内部 IP アドレスまたはサブネットを追加します。

MTA による内部メールと外部メールの識別方法

メールのリレーアクティビティをブロックするためには、まず、メールが同じサイトで発信された内部メールなのか、インターネットからシステムを経由して再びインターネットに戻っていく外部メールなのかを MTA が識別できなければなりません。そして、前述のクラスを許可し、後述のクラスをブロックする必要があります。この識別は、受信用 SMTP チャネルに `switchchannel` キーワードを使うことで実現できます。通常、このチャネルは `tcp_local` であり、デフォルトで設定されています。

`switchchannel` キーワードは、SMTP サーバが受信 SMTP 接続の実際の IP アドレスを調べるようにするものです。この IP アドレスは、Messaging Server によって、ドメイン内の SMTP 接続とドメイン外の接続とを識別するために書き換え規則とともに使用されます。その後、この情報は、内部と外部のメッセージトラフィックを分離するために使用されます。

以下で説明している MTA 設定では、デフォルトで、サーバが内部と外部のメッセージトラフィックを識別できるように設定されています。

- この設定ファイルでは、ローカルチャネルの直前に `defaults` チャネルおよび `noswitchchannel` キーワードを追加します。

```
! final rewrite rules
defaults noswitchchannel
! Local store
ims-ms ...
```

- 受信 TCP/IP チャネルを変更し、switchchannel および remotehost キーワードを指定します。次に例を示します。

  ```
  tcp_local smtp single_sys mx switchchannel remotehost
  TCP-DAEMON
  ```

- 受信 TCP/IP チャネル定義のあとに、同様の新しいチャネルを別の名前で追加します。以下に例を示します。

  ```
  tcp_intranet smtp single_sys mx allowswitchchannel routelocal
  tcp_intranet-daemon
  ```

  routelocal チャネルキーワードを指定すると、アドレスをチャネルに書き換える際に、MTA はこのチャネルを介してアドレスのすべての明示的ルーティングを短絡化しようとします。これにより、明示されたソースルートアドレスを経由した内部 SMTP ホストのループによるリレー試行がブロックされます。

上記の設定により、ドメイン内で生成された SMTP メールは tcp_internal チャネルから入ってくるようになります。それ以外の SMTP メールは、tcp_local チャネルから入ってきます。したがって、メールが入ってくるチャネルに基づいて内部と外部のメールが識別されます。

この設定はどのように機能するのでしょうか。ここでもっとも重要な要素は switchchannel キーワードです。キーワードは、tcp_local チャネルに適用されます。このキーワードにより、SMTP サーバにメッセージが入ってくると、サーバが受信接続のソース IP アドレスを調べるようになります。サーバは、受信接続のリテラル IP アドレスのリバースポインティングのエンベロープ書き換えを試行し、関連するチャネルを探します。ソース IP アドレスが INTERNAL_IP マッピングテーブル内の IP アドレスまたはサブネットと一致する場合は、そのマッピングテーブルを呼び出す書き換え規則によってアドレスが tcp_intranet チャネルに書き換えられます。

tcp_internal チャネルは allowswitchchannel キーワードでマークされているため、メッセージは tcp_internal チャネルに切り替えられて、そのチャネルから入ってきます。IP アドレスが INTERNAL_IP マッピングテーブルにないシステムからメッセージが入ってくる場合、リバースポインティングのエンベロープ書き換えは、

tcp_local チャネルあるいはその他のチャネルに対して書き換えを行います。ただし、tcp_internal チャネルに対する書き換えは行われません。それ以外のチャネルはデフォルトで noswitchchannel とマークされているため、メッセージは別のチャネルに切り替えられず、tcp_local チャネルのまま処理されます。

注	「tcp_local」という文字列を使用するマッピングテーブルまたは変換ファイルのエントリは、必要に応じて「tcp_*」または「tcp_intranet」に変更する必要があるかもしれないことに注意してください。

認証ユーザのメールを識別する

サイトには、物理的にネットワークの一部ではない「ローカル」のクライアントユーザが存在することがあります。これらのユーザがメールを送信すると、メッセージの送信は外部 IP アドレス (任意のインターネットサービスプロバイダ (ISP) など) から入ってきます。ユーザが SASL 認証を処理できるメールクライアントを使用している場合には、外部接続と認証接続とを識別できます。その結果に基づいて、認証ユーザによる送信を許可し、認証されていないユーザによるリレー送信試行を拒否できます。認証されているかどうかに基づく接続の識別は、受信用 SMTP チャネル (通常、tcp_local チャネル) に saslswitchchannel キーワードを使うことで実現できます。

saslswitchchannel キーワードはチャネルの切り替え先を示す引数をとり、SMTP の差出人が認証されると、送信メッセージが指定した切り替え先チャネルから入ってくるようになります。

認証ユーザによる送信であるかどうかを識別するには、以下のようにします。

1. 設定ファイルに新しい TCP/IP チャネル定義を別の名前で追加します。以下に例を示します。

   ```
   tcp_auth smtp single_sys mx mustsaslserver noswitchchannel
   TCP-INTERNAL
   ```

   このチャネルでは、通常のチャネル切り替えは行われません。それよりも前のデフォルト行で、noswitchchannel が明示あるいは暗黙に指定されているはずです。このチャネルには mustsaslserver が必要です。

2. 次の例のように、maysaslserver と saslswitchchannel tcp_auth を追加することにより、tcp_local チャネルを変更します。

   ```
   tcp_local smtp mx single_sys maysaslserver saslswitchchannel
   tcp_auth ¥
   switchchannel
   |TCP-DAEMON
   ```

この設定では、ローカルのパスワードによって認証が可能なユーザが送信した SMTP メールは tcp_auth チャネルから入ってくるようになります。認証されていない SMTP メールが内部ホストから送信された場合、そのメールは tcp_internal から入ってきます。それ以外の SMTP メールは、すべて tcp_local から入ってきます。

メールのリレーを防止する

次の例では、無許可のユーザが送信した SMTP メールのリレーをシステムが中継しないように設定しています。まず、ローカルユーザによる SMTP メールのリレーは許可することを念頭におきます。たとえば、POP ユーザおよび IMAP ユーザは、メールの送信に Messaging Server を使います。ローカルユーザには、メッセージが内部 IP アドレスから入ってくる物理的なローカルユーザのほか、ローカルユーザとして認証され得るリモートユーザも含まれます。

サーバにおけるリレーを阻止しなければならないのは、不特定多数のインターネット利用者からのメッセージです。以下の節で説明する設定では、これらのユーザクラスを識別して特定のクラスだけをブロックできます。特に、tcp_local チャネルから入り、同一のチャネルから出るメールをブロックします。そのためには、ORIG_SEND_ACCESS マッピングテーブルを使用します。

ORIG_SEND_ACCESS マッピングテーブルは、ソースチャネルと宛先チャネルに基づいてトラフィックをブロックするために使用できます。ここでは、tcp_local チャネルから入り、同一チャネルから出るトラフィックをブロックします。これは、次の ORIG_SEND_ACCESS マッピングテーブルで実現できます。

```
ORIG_SEND_ACCESS

   tcp_local|*|tcp_local|*         $NRelaying$ not$ permitted
```

この例では、メッセージが tcp_local チャネルから入り、同一のチャネルから出ることは許可されないことを示しています。つまり、このエントリを使用すると、外部からのメールを SMTP サーバで中継してインターネットに転送する処理を禁じることができます。

SEND_ACCESS マッピングテーブルではなく ORIG_SEND_ACCESS マッピングテーブルを使用するのは、ims-ms チャネルに元々一致するアドレスにブロックを適用するのではないからです (アドレスは、エイリアスまたはメーリングリストの定義を介して展開し、外部アドレスとなることがあるためです)。SEND_ACCESS マッピングテーブルでは、外部の利用者が外部ユーザに展開するメーリングリストにメールを送信したり、外部アドレスにメッセージを転送するユーザにメールを送信できるようにするのは困難です。

SMTP リレーブロッキングの RBL チェックを含む DNS 検索を使用するには

iPlanet Messaging Server には、配信や転送のために受け入れたすべてのメールが、有効な DNS 名を持つアドレスから送信されたものであるかどうかを確認するさまざまな方法があります。もっとも簡単な方法は、tcp_local チャネルに mailfromdnsverify チャネルキーワードを割り当てることです。

また iPlanet Messaging Server には、dns_verify というプログラムが用意されています。このプログラムを使うと、配信や転送のために受け入れたすべてのメールが、次に示す ORIG_MAIL_ACCESS の規則を使った有効な DNS 名を持つアドレスから送信されたものであるかどうかを確認することができます。

```
ORIG_MAIL_ACCESS

  TCP|*|*|*|*|SMTP|MAIL|*|*@*|*|*¥
$[<server_root>/bin/msg/imta/lib/dns_verify.so,¥
dns_verify,$6|$$y|$$NInvalid$ host:$ $$6$ -$ %e]
```

上の例に示されている改行記号は、このようなマッピングエントリのシンタックスにおいて非常に重要なものです。円記号は、その行が次の行に続いていることを意味しています。

また、もう 1 つの UBE 対策として、dns_verify イメージを使用し、受信接続を RBL (Realtime Blackhole List)、MAPS (Mail Abuse Prevention System)、DUL (Dial-up User List)、ORBS (Open Relay Behavior-modification System) などのリストに対してチェックすることができます。また、新しい mailfromdnsverify キーワードの場合と同じように、dns_verify 呼び出しを行わなくてもこれらのチェックを実行できる簡単な方法があります。それは dispatcher.cnf ファイルで DNS_VERIFY_DOMAIN オプションを使用する方法です。たとえば、[SERVICE=SMTP] セクションで、オプションのインスタンスをチェック対象のリストに設定します。

```
[SERVICE=SMTP]
PORT=25
! ...rest of normal options...
DNS_VERIFY_DOMAIN=rbl.maps.vix.com
DNS_VERIFY_DOMAIN=dul.maps.vix.com
!...etc...
```

この方法の短所は、内部ユーザからのメッセージを含む、通常の SMTP 受信メッセージすべてに対してチェックが行われるということです。このため効率が下がり、インターネット接続が切断された場合に問題が発生することがあります。別の方法として、PORT_ACCESS マッピングテーブル、または ORIG_MAIL_ACCESS マッピングテーブル

から dns_verify を呼び出す方法があります。PORT_ACCESS マッピングテーブルでは、最初の 1 つまたは複数のエントリに対してローカルの内部 IP アドレスまたはメッセージ送信者のチェックを行わないようにし、あとの方のエントリでほかのすべてに対して目的のチェックを行うようにすることができます。また、ORIG_MAIL_ACCESS マッピングテーブルでは、tcp_local チャネルで受信するメッセージのみをチェックする場合、内部システムやクライアントからのメッセージに対するチェックを省略することになります。以下に、dns_verify へのエントリポイントを使用した例を示します。

```
PORT_ACCESS

! Allow internal connections in unconditionally
  *|*|*|*|*  $C$|INTERNAL_IP;$3|$Y$E
! Check other connections against RBL list
  TCP|*|25|*|*¥
$C$[server_root>/bin/msg/imta/lib/dns_verify.so,¥
dns_verify_domain_port,$1,rbl.maps.vix.com.]EXTERNAL$E


ORIG_MAIL_ACCESS

  TCP|*|25|*|*|SMTP|*|tcp_local|*@*|*|*¥
$C$[<server_root>/bin/msg/imta/lib/dns_verify.so,¥
dns_verify_domain,$1,rbl.maps.vix.com.]$E
```

DNS ベースデータベースのサポート

iPlanet Messaging Server 5.2 から、dns_verify プログラムが DNS ベースのデータベースをサポートするようになりました。このデータベースは、不特定多数宛てのメールを送る可能性のある受信 SMTP 接続を判別するために使われます。一般に利用可能な DNS データベースの一部には、通常はこの目的のために使われる TXT レコードが含まれていません。その代わりに、A レコードが含まれています。

標準の設定では、特定の IP アドレスの DNS にある TXT レコードには、メッセージを拒否するときに SMTP クライアントに返すためのエラーメッセージが含まれています。しかし、TXT レコードがなく、A レコードがある場合、iPlanet Messaging Server 5.2 より前のバージョンの dns_verify は「No error text available」というメッセージを返しました。

現在、dns_verify は、TXT レコードを利用できないイベントで使われるデフォルトのテキストを指定するオプションをサポートしています。たとえば、以下の PORT_ACCESS マッピングテーブルは、このオプションを有効にする方法を示しています。

```
PORT_ACCESS

    *|*|*|*|*  $C$|INTERNAL_IP;$3|$Y$E
    TCP|*|25|*|*
$C$[/export/home/iplanet/server51/msg/bin/imta/lib/dns_verify.so
,dns_verify_domain_port,$1,dnsblock.siroe.com,Your$ host$ ($1)$
found$ on$ dnsblock$ list]$E
    * $YEXTERNAL
```

この例では、リモートシステムがドメイン dnsblock.siroe.com 内のクエリで見つかっても、TXT レコードが利用できない場合は、「Your host a.b.c.d found on dnsblock list.」というメッセージが返されます。

多数のアクセスエントリを処理する

マッピングテーブルに非常に多くのエントリを使用するサイトでは、マッピングテーブルを組織化し、特定の参照に対して一般的なデータベースを呼び出す一般的なワイルドカードエントリを利用するとよいでしょう。特定の参照に対し、2 ～ 3 件のマッピングテーブルエントリから一般的なデータベースを呼び出すほうが、数多くのエントリを直接マッピングテーブルで処理するよりもはるかに効率的です。

その一例として、だれがインターネットの電子メールを送信または受信できるのかをユーザごとに制御するサイトがあります。そのような制御は、ORIG_SEND_ACCESS などのアクセスマッピングテーブルを使って簡単に適用できます。この場合、一般的なデータベースに特定の情報 (たとえば特定のアドレスなど) をまとめて保存し、マッピングテーブルのエントリで呼び出すように設定すれば、効率と性能がかなり向上します。

たとえば、図 10-5 のマッピングテーブルを見てください。

図 10-5　ORIG_SEND_ACCESS マッピングテーブル

```
ORIG_SEND_ACCESS

  ! Users allowed to send to Internet
  !
  *|adam@siroe.com|*|tcp_local    $Y
  *|betty@siroe.com|*|tcp_local   $Y
  !...etc...
  !
  ! Users not allowed to send to Internet
  !
  *|norman@siroe.com|*|tcp_local  $NInternet$ access$ not$
      permitted
  *|opal@siroe.com|*|tcp_local    $NInternet$ access$ not$
      permitted
  !...etc...
  !
  ! Users allowed to receive from the Internet
  !
  tcp_*|*|*|adam@siroe.com        $Y
  tcp_*|*|*|betty@siroe.com       $Y
  !...etc...
  !
  ! Users not allowed to receive from the Internet
  !
  tcp_*|*|*|norman@siroe.com      $NInternet$ e-mail$ not$
      accepted
  tcp_*|*|*|opal@siroe.com        $NInternet$ e-mail$ not$
      accepted
  !...etc...
```

このように、ユーザごとに個々のエントリを記述したマッピングテーブルを使用するのではなく、より効率的な設定 (何百、何千件ものユーザを効率的に処理できる設定) を次の 図 10-6 に示します。この図には、一般的なデータベースエントリと ORIG_SEND_ACCESS マッピングテーブルが示されています。

図 10-6　データベースエントリとマッピングテーブルの例

データベースエントリ

```
SEND|adam@domain.com    $Y
SEND|betty@domain.com   $Y
! ...etc...
SEND|norman@domain.com  $NInternet$ access$ not$ permitted
SEND|opal@domain.com    $NInternet$ access$ not$ permitted
! ...etc...
RECV|adam@domain.com    $Y
RECV|betty@domain.com   $Y
! ...etc...
RECV|norman@domain.com  $NInternet$ e-mail$ not$ accepted
RECV|opal@domain.com    $NInternet$ e-mail$ not$ accepted
```

マッピングテーブル

```
ORIG_SEND_ACCESS

  ! Check if may send to Internet
  !
  *|*|*|tcp_local       $C${SEND|$1}$E
  !
  ! Check if may receive from Internet
  !
  tcp_*|*|*|*           $C${RECV|$3}$E
```

この例では、一般的なデータベースの左側に記述した文字列「SENDl」および「RECVl」を使用 (マッピングテーブルで生成される一般的なデータベースプローブ) することにより、2 種類のプローブを区別しています。一般的なデータベースプローブを「$C」および「$E」フラグで囲むのは、マッピングテーブルから一般的なデータベース呼び出しに特有の方法です。

この例では、単純なマッピングテーブルプローブが一般的なデータベースのエントリを参照するケースを示しています。より複雑なプローブのマッピングテーブルでも一般的なデータベースの使用による効果を得ることができます。

アクセス制御マッピングテーブルのフラグ

表 10-3 に、SEND_ACCESS、ORIG_SEND_ACCESS、MAIL_ACCESS、ORIG_MAIL_ACCESS、および FROM_ACCESS マッピングテーブルに関連するアクセスマッピングフラグを示します。PORT_ACCESS マッピングテーブルでは、少し異なるフラグがサポートされています (表 10-2 を参照)。

表 10-3 アクセスマッピングフラグ

フラグ	説明
$B	ビットバケットにメッセージをリダイレクトする
$H	.HELD ファイルとしてメッセージを保留する
$Y	アクセスを許可する
フラグと引数 (引数の読み取り順序 +)	
$J*address*	元のエンベロープの From: アドレスを指定の *address* に置換する *
$K*address*	元のエンベロープの Sender: アドレスを指定の *address* に置換する * ++
$I*user* | *identifier*	特定のユーザのグループ ID を調べる
$<*string*	プローブが一致する場合、*string* を syslog (UNIX、user.notice 機能と重大度) またはイベントログ (NT) に送る +++
$>*string*	アクセスが拒否された場合、*string* を syslog (UNIX、user.notice 機能と重大度) またはイベント ログ (NT) に送る +++
$D*delay*	応答を *delay* (100 分の 1 秒) だけ遅らせる。正の値はトランザクションのコマンドごとに遅らせ、負の値は、アドレスの引き渡し時 (FROM_ACCESS テーブルの SMTP MAIL FROM: コマンド、その他のテーブルの SMTP RCPT TO: コマンド) にのみ遅らせる
$T*tag*	*tag* を前に付ける
$A*header*	メッセージに *header* 行を追加する
$X *error-code*	メッセージを拒否した場合に、指定した *error-code* を含む拡張 SMTP エラーコードを発行する
$N*string*	アクセスを拒否し、オプションのエラーテキスト *string* を送る
$F*string*	*$N string* と同じ。アクセスを拒否し、オプションのエラーテキスト *string* を送る

表 10-3　アクセスマッピングフラグ (続き)

フラグ	説明
* FROM_ACCESS テーブルでのみ使用できます。 + 引数を伴うフラグを複数個使用する場合は、引数を縦棒文字「|」で区切り、この表に示されている順序で配置します。 ++ 「$K」フラグを FROM_ACCESS マッピングテーブルで有効にするには、ソースチャネルに authrewrite キーワードが含まれていなければなりません。 +++ 問題のある差出人によるサービスアタックを防ぐには「$D」フラグを使用するとよいでしょう。特に、$> エントリまたはアクセスを拒否する $< エントリで「$D」フラグを使用します。	

第 2 部 メールボックスフィルタ

第 2 部には、以下の項目があります。

- はじめに
- ユーザ単位のフィルタを作成するには
- チャネルレベルのフィルタを作成するには
- MTA 全体のフィルタを作成するには
- ユーザフィルタをデバッグするには

はじめに

フィルタは、メールメッセージに適用される 1 つ以上の条件付きアクションで構成されています。Messaging Server フィルタはサーバに保存され、サーバによって評価されます。そのため、それらは SSR (サーバ側規則) と呼ばれることもあります。Messaging Server のフィルタは、SIEVE Internet Draft の Draft 9 である SIEVE フィルタリング言語に基づいています。

管理者は、チャネルレベルのフィルタと MTA 全体のフィルタを作成し、不正メールの配信を防止できます。また、フィルタテンプレートを作成し、Delegated Administrator for Messaging のインタフェースを介してエンドユーザが利用できるようにすることも可能です。エンドユーザは、テンプレートを利用して個人用のメールボックスフィルタを構築し、受け取りたくないメールメッセージの受信を拒否できます。

サーバは、次の優先順位に従ってフィルタを適用します。

1. ユーザ単位のフィルタ

   個人用メールボックスフィルタにメッセージの許可あるいは拒否が定義されている場合は、メッセージに対してそのフィルタ処理が行われます。しかし、受取人がメールボックスフィルタを設定していない場合、またはユーザのメールボックスフィルタが適用されないメッセージの場合、Messaging Server によってチャネルレベルのフィルタが適用されます。

2. チャネルレベルのフィルタ

   チャネルレベルのフィルタにメッセージの許可あるいは拒否が定義されている場合は、メッセージに対してそのフィルタ処理が行われます。それ以外の場合は、Messaging Server によって MTA 全体のフィルタが適用されます (該当する場合)。

3. MTA 全体のフィルタ

デフォルト設定を使用した場合、それぞれのユーザはメールボックスフィルタを所有していません。ユーザが委任管理者のインタフェースを使用して 1 つまたは複数のフィルタを作成すると、それらのフィルタがディレクトリに保存され、ディレクトリの同期処理時に MTA によって読み取られます。

ユーザ単位のフィルタを作成するには

ユーザ単位のフィルタは、特定ユーザのメールボックスに送信されるメッセージに適用されます。管理者は、フィルタテンプレートを作成し、Delegated Administrator for Messaging のインタフェースを介してそのテンプレートをエンドユーザに提供できます。エンドユーザはテンプレートを利用して個人用サーバフィルタを構築し、メールボックスへのメールメッセージ配信を制御できます。つまり、特定のメールメッセージの受信を拒否したり、メールをリダイレクトしたり、あるいはメールボックスフォルダに入れるメッセージをフィルタリングすることができます。

フィルタテンプレートは、Sieve スクリプトのハードコード要素をプロンプトや入力フィールドに置換することで、Sieve スクリプトを一般化したものです。Java サーブレットは、Sieve テンプレートを解析し、ブラウザで UI ページを生成するために使用されます。エンドユーザが入力フィールドに値を入力すると、サーブレットによってそれらの値が読み取られ、ユーザのディレクトリプロフィールエントリ内の Sieve スクリプトに保存されます。プロンプトおよび入力フィールドは、Delegated Administrator のインタフェースを介してエンドユーザに提示されます。

Delegated Administrator には、サンプルのテンプレートセットが用意されています。これらのテンプレートファイルは、次のディレクトリにあります。

`nda-path/nda/nda/default/`*`lang`*`/templates/enduser/ssr/*.txt`

フィルタテンプレートは、Sieve 言語を使って変更したり新規作成したりすることができます。新規のフィルタテンプレートを作成した場合は、それを前述の ssr ディレクトリにテキスト形式で保存しなければなりません。そのファイルがだれでも読み取り可能であることを確認し、以下の例に示すように、フィルタテンプレートに LDAP エントリを追加します。

```
dn:cn=Subject Discard,cn=ssrconf,cn=en,
              cn=domainConfiguration,ou=config,o=isp
objectclass: top
objectclass: nsValueItem
cn:Subject Discard
nsvaluetype: nsValueCIS
nsvaluecis: ../templates/enduser/ssr/subject-discard.txt
```

図 10-7 にテンプレートの例を示します。

図 10-7 Seive テンプレートの例

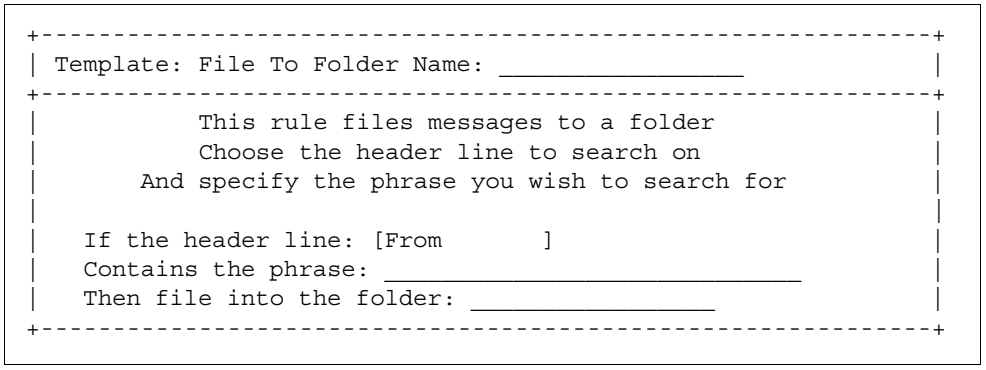
```
#RULE: $Template="File To Folder"
require "fileinto";
if header :contains # Q1
     # Q2
{
     fileinto # Q3
    ;
}

#PRE: "This rule files messages into a folder."
#PRE: "Choose the header line to search on"
#PRE: "And specify the phrase you wish to search for"
# Q1: header "If the header line"
# Q2: value "Contains the phrase"
# Q3: folder "Then file into the folder"
```

上記の例で、Q1、Q2、および Q3 は入力される値のプレースフォルダであり、UI がその値を見つけて置換します。各トークンは、その入力値の質問とデータタイプをマッピングします。

データタイプおよび関連する質問は、トークンごとのコメント行に定義されています。それらは、*token* : *data-type-variable* の形式で定義され、続いて、引用符に囲まれた文字列に実際の質問が含まれています。上記の例で、`header value`、および `folder` は、いずれも、ドロップダウンリスト、編集ボックス、あるいはその他の要素を示すデータタイプです。これらのデータタイプ変数は、UI に対し、どのタイプの情報をユーザから取得するのかを指示するものです。

テンプレートが解析されると、ダイアログが生成され、図 10-8 の例に示すようにエンドユーザに提示されます。この例では、角括弧はドロップダウンリストを示しています。

図 10-8　テンプレート出力の例



ユーザがデータを入力すると、その規則がユーザの `mailSieveRuleSource` 属性に保存されます。

テンプレートのシンタックスには、以下の規則があります。

- `#RULE` 行は、その他の行よりも前に記述され、`$Template` を定義する必要がある
- `#PRE` で始まるコメント行は、GUI ページの入力フィールドよりも前に表示される

  `#PRE` 文は、二重引用符で囲まれていなければならない
- `#POST` で始まるコメント行は、GUI ページの最後に表示される

  `#POST` 文は、二重引用符で囲まれていなければならない
- その他のコメント行は、GUI ページには表示されない
- トークンは ASCII 文字列で、大文字と小文字の区別はない。トークンに空白を挿入することはできない
- データタイプ変数は、コメント行のトークン文字列の後ろに記述する。大文字と小文字の区別はない

- 実際の質問は、データタイプ変数のすぐ後ろのコメント行に定義されており、二重引用符で囲まれている

Sieve テンプレートでは、以下のデータタイプ変数がサポートされています。

- `header` - GUI に表示される際には、リストボックスが使用され、`Subject`、`To`、`From` の各値が表示される

  Sieve 規則がユーザエントリに保存されると、`Subject` の値が Subject、Comments、Keywords に展開され、`From` の値は From、Sender、Resent-from、Resent-sender、Return-path に、さらに `To` の値は To、Cc、Bcc、Resent-to、Resent-cc、Reset-bcc に展開される

- `value` - テキストフィールドを使って値を示す
- `address` - テキストフィールドを使って値を示す。アドレスのシンタックスが RFC 822 のメールアドレス形式に準拠しているかどうかが調べられる
- `folder` - テキストフィールドを使って値を示す
- `size` - ユーザは「キロバイト」または「メガバイト」の中から選択するか、または任意の数値を指定できる
- `message` - テキストフィールドを使って値を示す

チャネルレベルのフィルタを作成するには

チャネルレベルのフィルタは、チャネルのキューに入った各メッセージに適用されます。この種のフィルタの一般的な用途は、特定のチャネルから入ってくるメッセージをブロックすることです。

チャネルレベルのフィルタを作成する手順を以下に示します。

1. SIEVE を使ってフィルタを記述します。
2. フィルタを、以下のディレクトリのファイルに保存します。

   `msg-`*instance*`/imta/config/`*file*`.filter`

   ファイルはだれでも読み取り可能で、MTA の uid によって所有されていなければなりません。

3. 以下のチャネル設定を定義します。

   `destinationfilter file:IMTA_TABLE:`*file*`.filter`

4. 設定をリコンパイルし、ディスパッチャを再起動します。

   注意 : フィルタファイルへの変更を有効にするのに、リコンパイルやディスパッチャの再起動は不要です。

destinationfilter チャネルキーワードは、対象チャネルのキューに入るメッセージのフィルタリングを有効にします。sourcefilter チャネルキーワードは、対象チャネルからキューに入るメッセージのフィルタリングを有効にします。これらのキーワードには、それぞれパラメータが 1 つ必要です。このパラメータは、そのチャネルに関連付けられたチャネルフィルタファイルへのパスを指定するものです。

destinationfilter チャネルキーワードのシンタックスは以下のとおりです。

destinationfilter *URL-pattern*

sourcefilter チャネルキーワードのシンタックスは以下のとおりです。

sourcefilter *URL-pattern*

URL-pattern は、対象チャネルのフィルタファイルへのパスを示す URL です。次の例で、*channel-name* はチャネルの名前です。

destinationfilter file:///usr/tmp/filters/*channel-name*.filter

filter チャネルキーワードは、対象チャネルにおけるメッセージのフィルタリングを有効にします。このキーワードには、パラメータが 1 つ必要です。このパラメータは、そのチャネルを介してメールを受信するエンベロープの各受取人に関連付けられたチャネルフィルタファイルへのパスを指定するものです。

filter チャネルキーワードのシンタックスは以下のとおりです。

filter *URL-pattern*

URL-pattern は、特殊な置換シーケンスを処理したあとの URL で、指定した受取人アドレスに対するフィルタファイルへのパスを示します。*URL-pattern* には、特殊な置換シーケンスを含めることができます。このシーケンスは、受取人アドレス local-part@host.domain から派生する文字列に置き換えられます。337 ページの表 10-4 に、これらの置換シーケンスを示します。

fileinto キーワードは、メールボックスフィルタの fileinto 演算子が適用されたときにアドレスをどのように変更するのかを指定するものです。次の例では、フォルダ名をサブアドレスとして元のアドレスに挿入して、元のサブアドレスを置き換えるように指定しています。

```
fileinto $U+$S@$D
```

表 10-4 置換タグ (大文字小文字を区別します)

タグ	意味
*	グループの拡張を実行する。558 ページの「グループエントリを処理する」を参照

表 10-4　置換タグ (大文字小文字を区別します) (続き)

タグ	意味
**	mailForwardingAddress 属性を拡張する。複数の値を持つ属性を設定して複数の配信先アドレスを生成できる
$$	$ 文字に置き換える
$¥	後続のテキストを小文字にする
$^	後続のテキストを大文字にする
$_	代替テキストで大文字と小文字を変換しない
$~	アドレスのローカル部分に関連付けられたホームディレクトリに対するファイルパスに置き換える
$1S	$S と同じだが、サブアドレスがない場合は何も行わない
$2S	$S と同じだが、サブアドレスがない場合は何も挿入せず前の文字を削除する
$3S	$S と同じだが、サブアドレスがない場合は何も挿入せず後続の文字を無視する
$A	アドレス (local-part@ host.domain) に置き換える
$D	ホストドメインに置き換える
$E	第 2 スペア属性の値 LDAP_SPARE_1 を挿入する
$F	配信ファイル名 (mailDeliveryFileURL 属性) を挿入する
$G	第 2 スペア属性の値 LDAP_SPARE_2 を挿入する
$H	ホストに置き換える
$I	ホストドメインを挿入する (domainUidSeparator で指定した区切り文字の右側に UID の一部を挿入)。ホストドメインがないと失敗する
$1I	$I と同じだが、ホストドメインがない場合は何も挿入しない
$2I	$I と同じだが、ホストドメインがない場合は何も挿入せず前の文字を削除する
$3I	$I と同じだが、ホストドメインがない場合は何も挿入せず後続の文字を無視する
$L	ローカル部分に置き換える
$M	UID を挿入し、ホストドメインを削除する
$P	プログラム名を挿入する (mailProgramDeliveryInfo 属性)
$S	現在のアドレスに関連づけられたサブアドレスを挿入する。サブアドレスは、元のアドレスでサブアドレス区切り (通常は +) に続くユーザ部分の該当する箇所。ただし、MTA オプションの SUBADDRESS_CHAR で指定することもできる。サブアドレスを指定しないと失敗する

表 10-4　置換タグ (大文字小文字を区別します) (続き)

タグ	意味
$U	現在のアドレスのメールボックス部分を挿入する。@ マークの左側のアドレス全体、またはその中でサブアドレス区切りの + より前の部分のいずれかが挿入される

MTA 全体のフィルタを作成するには

MTA 全体のフィルタは、MTA のキューに入るすべてのメッセージに適用されます。この種のフィルタの一般的な用途は、メッセージの宛先とは関係なく、ダイレクトメールや受信したくないメッセージをブロックすることです。MTA 全体のフィルタを作成するには次のようにします。

1. SIEVE を使ってフィルタを記述します。
2. フィルタを、次のファイルに保存します。

   msg-*instance*/imta/config/imta.filter

   このフィルタファイルは、だれでも読み取り可能でなければなりません。このファイルは自動的に使用されます。
3. 設定をリコンパイルし、ディスパッチャを再起動します。

コンパイルした設定を使用する場合、MTA 全体のフィルタファイルはコンパイルされた設定内に組み込まれています。

FILTER_DISCARD チャネルから破棄メッセージをルーティングする

デフォルトでは、メールボックスフィルタで破棄されたメッセージは、システムから即座に破棄 (削除) されます。しかし、ユーザが最初にメールボックスフィルタを設定した場合 (設定が間違っている場合)、またはデバッグを目的とする場合には、削除処理を遅らせると便利です。

メールボックスフィルタによる破棄メッセージをシステム内に一時保存し、それをあとで削除できるようにするには、次の例に示すように、まず MTA 設定に filter_discard チャネルを追加し、notices チャネルキーワードでメッセージを削除するまでの保存期間 (通常は日数) を記述します。

```
filter_discard notices 7
FILTER-DISCARD
```

次に MTA オプションファイルで FILTER_DISCARD=2 オプションを設定します。filter_discard キュー内のメッセージは、ユーザの個人用ゴミ箱フォルダの延長と考えることができます。したがって、filter_discard キュー内のメッセージに対して警告メッセージが送られたり、バウンスやリターンの要求に応じてメッセージが差出人に戻されることもありません。これらのメッセージは、final notices 値の期限となるか、imsimta return などのユーティリティを使ってバウンスを要求することによって、システムから削除されるだけです。

ユーザフィルタをデバッグするには

以下の情報は、システムのユーザフィルタに関して問題が発生した場合に役に立ちます。

dirsync プロセスは、ユーザフィルタに関する MTA の SSR データベース情報を更新します。短いフィルタは、データベース内に保存されます。長いフィルタの場合は、データベースに LDAP dn が保存されます。dirsync プロセスによってデータベースが更新されるまで、ユーザフィルタの変更内容は認識されません。

フィルタに関する問題を解決するには、以下の手順に従ってください。

- imta.cnf ファイル内で、ims-ms チャネルが次のようにマークされていることを確認します。

  ```
  filter ssrd:$a fileinto $u+$s@$d
  ```

- dirsync プロセスが configutil コマンドを使ってフィルタ情報を同期するようになっていることを確認します。

  ```
  configutil -l -o service.imta.ssrenabled -v true
  OK SET
  configutil | fgrep ssr
  service.imta.ssrenabled = true
  ```

- フィルタをテストするには、次のように imsimta test コマンドを使用します。

  ```
  imsimta test -rewrite -debug -filter user@domain
  ```

  出力で、以下の情報を探します。

  ```
  mmc_open_url called to open ssrd:user@ims-ms
     URL with quotes stripped:ssrd:user@ims-ms
  Determined to be an SSRD URL.
     Identifier:user@ims-ms-daemon
  Filter successfully obtained.
  ```

- フィルタのシンタックスに問題がある場合は、以下の情報を探します。

  ```
  Error parsing filter expression:...
  ```

このエラーからフィルタに関する問題の詳細がわかります。

- フィルタに問題がない場合は、test コマンドによって、出力の最後にフィルタが表示されます。
- フィルタに問題がある場合は、test コマンドによって、出力の最後に次の情報が表示されます。

  ```
  Address list error -- 4.7.1 Filter syntax error:user@siroe.com
  ```

  また、次に示すように、SMTP RCPT TO コマンドによって一時的なエラー応答コードが返されます。

  ```
  RCPT TO:<user@siroe.com>
  452 4.7.1 Filter syntax error
  ```

- ユーザアドレスの最終的な書き換え形式がわかっている場合には、imsimta test -url コマンドを使って MTA がそのユーザ用に使っているフィルタを確認できます。

  imsimta test -url ssrd:user@ims-ms-daemon

  imsimta test -rewrite コマンドを使用すると、ユーザアドレスの最終的な書き換え形式を見つけることができます。

第 11 章

メッセージストアを管理する

この章では、メッセージストアとメッセージストアの管理インタフェースについて説明します。この章には、以下の節があります。

概要

メッセージストアには、特定の Messaging Server インスタンス用のユーザメールボックスが格納されています。メッセージストアのサイズは、メールボックス、フォルダ、およびログファイルの数が増えるに従って増大していきます。ストアのサイズを制御するには、メールボックスのサイズ制限 (ディスク制限容量) を指定するか、許可するメッセージ総数を制限指定するか、ストア内のメッセージに関する保存期間決定ポリシーを設定します。

システムにユーザを追加していくに従い、ディスクストレージ要件も増えていきます。サーバがサポートするユーザ数によって、メッセージストアに必要な物理ディスクが 1 つであるか、複数であるかが決まります。この追加ディスク容量をシステムに統合するには、2 種類の方法が存在します。もっとも簡単な方法は、別のパーティションを追加することです。オプションで、特定のメッセージストアを担当する Messaging Server インスタンスを追加することもできます。ただし、この方法は非常に複雑です。

また、複数のホストドメインをサポートしている場合は、1 つのサーバインスタンスを単一の大規模ドメイン専用にした方がよい可能性があります。この構成を行えば、特定のドメインに対するストア管理を指定することができます。また、パーティションをさらに追加することで、メッセージストアを拡張することもできます。

iPlanet Messaging Server では、メッセージストアの管理のために、iPlanet Console インタフェースに加えてコマンドラインユーティリティのセットを提供しています。表 11-1 では、このコマンドラインユーティリティについて説明しています。これらのユーティリティの使用に関する詳細については、364 ページの「保守および回復手順を実行する」および『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

表 11-1 メッセージストアのコマンドラインユーティリティ

ユーティリティ	説明
`configutil`	ストアの設定パラメータの設定および変更を行います。
`deliver`	IMAP または POP メールクライアントがアクセスできるメッセージストアにメールを直接配信します。
`hashdir`	特定のユーザのメッセージストアを格納するディレクトリを識別します。
`iminitquota`	LDAP ディレクトリから制限容量の上限を再初期化し、使用されているディスク容量を再計算します。
`imsasm`	ユーザメールボックスの保存と回復を行います。
`imsbackup`	保存されたメッセージをバックアップします。

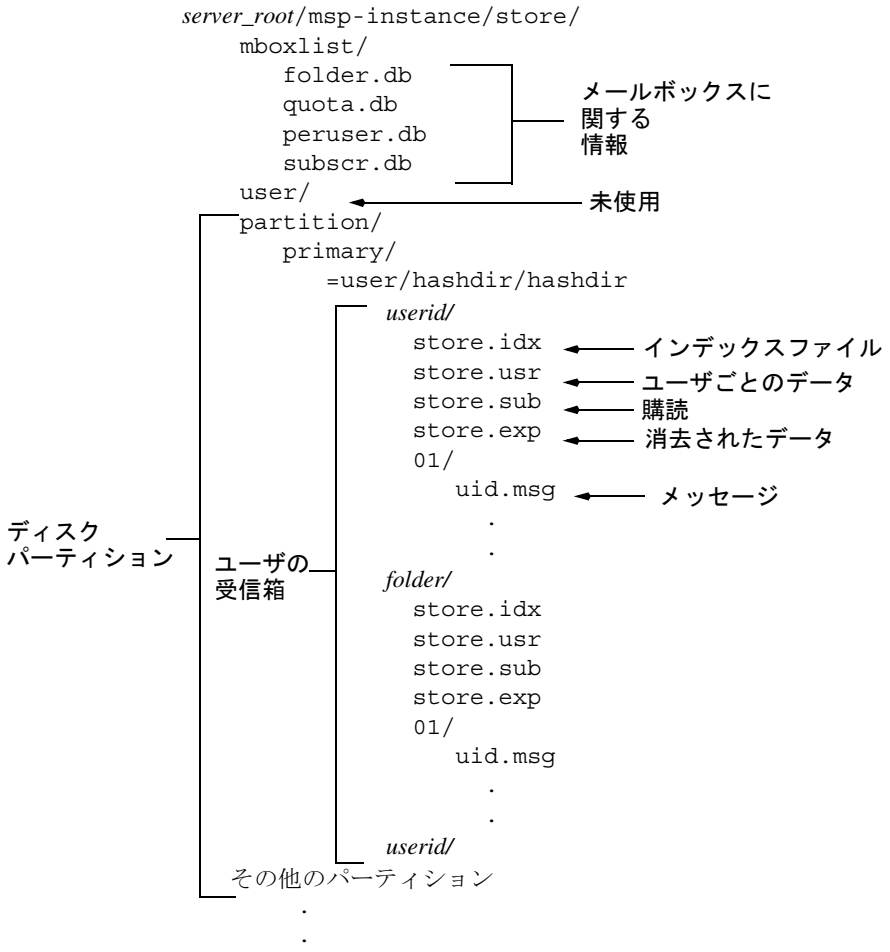
表 11-1　メッセージストアのコマンドラインユーティリティ (続き)

ユーティリティ	説明
imsexport	Certificate Management System のメールボックスを UNIX の /var/mail 形式のフォルダ内にエクスポートします。
imsrestore	バックアップされたメッセージをリストアします。
imscripter	IMAP サーバプロトコルのスクリプティングツール。1 つまたは一連のコマンドを実行します。
mboxutil	メールボックスの一覧表示、作成、削除、名前変更、移動を行い、制限容量の使用状況をレポートします。
mkbackupdir	メッセージストア内の情報を含むバックアップディレクトリを作成および同期化します。
MoveUser	ユーザのアカウントを別の Messaging Server に移動します。
quotacheck	メッセージストア内の各ユーザのメールボックスの合計を計算し、そのサイズをそれらに割り当てられている制限容量と比較します。
readership	共有の IMAP フォルダ上の読者情報を収集します。
reconstruct	破壊または破損したメールボックスを再構築します。
stored	バックグラウンドの日常タスクを実行し、ディスクに保存されたメッセージの消去や削除を行います。

メッセージストアのディレクトリレイアウト

図 11-1 は、サーバインスタンスに対するメッセージストアのディレクトリレイアウトを示しています。メッセージストアはメールボックスの内容に高速でアクセスできるように設計されています。ストアディレクトリについては、表 11-2 で説明しています。

図 11-1　メッセージストアのディレクトリレイアウト



たとえば、ディレクトリパスの例は以下のようになります。

server_root/msg-*instance*/store/partition/primary/=user/53/53/=mack1

表 11-2　メッセージストアのディレクトリの説明

場所	内容 / 説明
server_root/msg-*instance*/store/	メッセージストアのトップレベルのディレクトリ。mboxlist、user、および partition サブディレクトリを格納しています。

表 11-2　メッセージストアのディレクトリの説明 (続き)

場所	内容 / 説明
.../store/mboxlist/	サーバ上のメールボックスに関する情報やメールボックスの制限容量に関する情報が保存されたデータベース (Berkley DB) を格納しています。 ファイル folder.db には、メールボックスが保存されているパーティションの名前、ACL、および store.idx にある情報のいくつかのコピーなど、メールボックスに関する情報が格納されています。folder.db には、メールボックスごとに 1 つのエントリが存在しています。 ファイル quota.db には、制限容量および制限容量の使用状況に関する情報が格納されています。quota.db2 には、ユーザごとに 1 つのエントリが存在しています。 ファイル peruser.db には、ユーザごとのフラグに関する情報が格納されています。このフラグは、特定のユーザがメッセージを確認したかどうか、または削除したかどうかを示しています。 ファイル subscr.db には、ユーザの購読に関する情報が格納されています。
.../store/user/	使用されていません。
.../store/partition/	デフォルトの primary パーティションが格納されています。このディレクトリには、ほかのサブパーティションを定義して格納することもできます。
/*partition*/=user/	パーティションのサブディレクトリにある全ユーザのメールボックスが格納されています。メールボックスは、高速で検索できるようにハッシュ構造で保存されています。特定のユーザのメールボックスを格納するディレクトリを検索するには、hashdir ユーティリティを使用します。
/=user/*hashdir*/*hashdir*/*userid*/	*userid* という ID を持つユーザ用のトップレベルのメールフォルダ。デフォルトドメインでは、*userid* は *uid* です。ホストドメインでは、*userid* は *uid@domain* です。メッセージはこのメールフォルダに配信されます。
/*userid*/*folder*	ユーザ定義のフォルダ。

表 11-2　メッセージストアのディレクトリの説明 (続き)

場所	内容 / 説明
/*userid*/store.idx	このインデックスは、/*userid*/ ディレクトリに保存されたメールに関して、メッセージの数、このメールボックスが使用するディスクの制限容量、メールボックスが最後に追加された時間、メッセージフラグ、各メッセージの可変長情報 (ヘッダーや MIME 構造を含む)、各メッセージのサイズなどのさまざまな情報を提供します。さらにこのインデックスには、各ユーザに関する mboxlist 情報のバックアップコピーや、各ユーザに関する制限容量情報のバックアップコピーも含まれています。
/*userid*/store.usr	フォルダにアクセスしたユーザのリストが格納されています。リストされた各ユーザについて、そのユーザが最後にフォルダにアクセスした時間、ユーザが表示したメッセージのリスト、ユーザが削除したメッセージのリストといった情報が格納されています。
/*userid*/store.exp	消去されたものの、ディスクからは消し去られていないメッセージファイルのリストを格納しています。このファイルは、消去されたメッセージが存在する場合のみ表示されます。
/*userid*/store.sub	ユーザの購読に関する情報が格納されています。
/*userid*/*nn*/	*msgid*.msg の形式でメッセージが格納されているハッシュディレクトリです。*nn* には、00 ～ 99 までの数字が入ります。 たとえば、1 ～ 99 のメッセージは 00 ディレクトリに保存されており、100 ～ 199 のメッセージは 01 ディレクトリに保存されており、9990 ～ 9999 のメッセージは 99 ディレクトリに保存されており、10000 ～ 10099 のメッセージは 00 ディレクトリに保存されているという具合です。

ストアによるメッセージの消去方法

メッセージは、次の 3 段階の手順でストアから消去されます。

1. **削除** : クライアントが削除するメッセージをマークします。この時点では、クライアントは「削除済み」マークを外せばメッセージをリストアすることができます。
2. **消去** : クライアント、または指定した存続期間決定ポリシーにより、削除マークの付けられたメッセージがメールボックスから消去されます。メッセージの消去が行われると、クライアントがそれをリストアすることはできなくなります。ただし、メッセージはまだディスク上に存在しています (既存の接続を使用して同じメールボックスにアクセスできる 2 番目のクライアントは、まだメッセージを取り出すことができます)。
3. **クリーンアップ** : `stored` ユーティリティにより、1 時間以上前に消去されたメッセージをすべてディスクから消し去ります

メッセージは、**expire** オプションを設定して消去することもできます。サーバは `configutil` によって定義された存続期間決定ポリシーに基づいてメッセージを削除します。メッセージは期限が切れたら消去されますが、クリーンアップが実行されるまで物理的には削除されません (358 ページの「存続期間決定ポリシーを指定するには」を参照)。

ストアへの管理者によるアクセスを指定する

メッセージストアの管理者は、ユーザのメールボックスを表示してモニタしたり、メッセージストアに対するアクセス制御を指定することができます。ストア管理者は、すべてのサービス (POP、IMAP、HTTP、または SMTP) に対するプロキシ認証権限を持っているので、任意のユーザの権限を使用して任意のサービスを認証することができます。これらの権限により、ストア管理者は特定のユーティリティを実行してストアを管理することができます。たとえば、`MoveUser` を使用して、ストア管理者はあるシステムから別のシステムへユーザアカウントやメールボックスを移動させることができます。

この節では、Messaging Server のメッセージストアに対してストア権限を付与する方法を説明します。

注	ほかのユーザもそのストアに対する管理者権限を持っている可能性があります。たとえば、自分のサイトで Delegated Administration (DA) 製品を使用している場合、トップレベルの DA 管理者は、デフォルトではメールシステムのすべての Messaging Server に対するストア権限を持っています。デフォルトでは DA ドメイン管理者は、自分のドメインに対するストア権限を持っています。DA 管理者に関する詳細については、『iPlanet Messaging Server プロビジョニングガイド』および DA のマニュアルを参照してください。

次の項で説明するタスクを実行することができます。

- 管理者を追加するには
- 管理者エントリを変更するには
- 管理者エントリを削除するには

管理者によるストアへのアクセスは、`configutil` コマンドを使用するか、コンソールを使用して指定することができます。

コンソールを使用する場合は、以下の手順に従います。

1. 構成を行う Messaging Server をコンソールから開きます。
2. 「構成」タブをクリックして、左のペインの「メッセージストア」を選択します。
3. 右のペインの「管理者」タブをクリックします。

管理者を追加するには

コンソール : コンソールで管理者エントリを追加するには以下の手順に従います。

1. 「管理者」タブをクリックします。

   このタブでは、既存の管理者 ID が一覧表示されます。
2. 「管理者 UID」ウィンドウの横にある「追加」ボタンをクリックします。
3. 追加する管理者のユーザ ID を「管理者 UID」フィールドに入力します。

   ここで入力するユーザ ID は、iPlanet Directory Server に認識されるものでなければなりません。
4. 「OK」をクリックすると、「管理者」タブに表示されているリストに管理者 ID が追加されます。

5. 「管理者」タブで「保存」をクリックして、新たに変更した管理者リストを保存します。

コマンドライン : コマンドラインで管理者のエントリを追加する場合は、以下のようになります。

```
configutil -o store.admins -v "adminlist"
```

この *adminlist* は、スペースで区切られた管理者 ID のリストです。複数の管理者を指定する場合は、引用符でリストを囲んでください。

管理者エントリを変更するには

コンソール : コンソールでメッセージストアの管理者 UID リストにある既存のエントリを変更するには、以下の手順に従います。

1. 「管理者」タブをクリックします。
2. 「管理者 UID」ウィンドウの横にある「編集」ボタンをクリックします。
3. 「管理者 UID」フィールドに変更を入力します。
4. 「OK」をクリックして変更を送信し、「管理者の編集」ウィンドウを閉じます。
5. 「管理者」タブで「保存」をクリックして、変更した管理者リストを送信して保存します。

コマンドライン : コマンドラインでメッセージストアの管理者 UID リストにある既存のエントリを変更する場合は、以下のようになります。

```
configutil -o store.admins -v "adminlist"
```

管理者エントリを削除するには

コンソール : コンソールを使用してメッセージストアの管理者 UID リストからエントリを削除するには、以下の手順に従います。

1. 「管理者」タブをクリックします。
2. 「管理者 UID」リストで項目を選択します。
3. 「削除」をクリックして項目を削除します。
4. 「保存」をクリックして、管理者リストに変更を送信して保存します。

コマンドライン : コマンドラインでストア管理者を削除する場合は、以下のように管理者リストを編集することができます。

```
configutil -o store.admins -v "adminlist"
```

メッセージストアの制限容量について

この節では、以下の情報について説明します。

- ユーザの制限容量
- ドメインの制限容量とファミリーグループの制限容量
- Telephony Application Server に関する例外

ユーザの制限容量

ユーザのメールボックスのサイズ制限を指定することで、メッセージストアのサイズを制限することができます。以下のタイプの制限容量を指定することができます。

- ディスク制限容量は、各ユーザに割り当てられるディスク容量を制限するものです。ディスク制限容量は、ユーザのメールフォルダの数に関係なくユーザのメッセージの合計サイズに適用されるか、ユーザメッセージの合計数に適用されます。ディスク容量に限りがある場合は、ユーザのディスク制限容量を設定した方がよいでしょう。
- メッセージ制限容量は、ユーザのメールボックスに保存されるメッセージの数を制限するものです。

制限容量の情報は、LDAP 属性および設定変数として保存されます。制限容量の適用が有効になっている場合、Messaging Server は、メッセージストアにメッセージを挿入する前に制限容量キャッシュと設定ファイルをチェックして、制限容量を超えないようにします。制限容量の通知が有効になっている場合、ユーザがディスク制限容量に到達したら、エラーメッセージが送信されます。また、ユーザが制限容量に近づいたらサーバから警告メッセージを送信することも可能です。

すべてのユーザに対してデフォルトの制限容量を設定することも、個々のユーザに対して制限容量を設定することもできます。ユーザが制限容量を超えているかどうかを判別するために、Messaging Server は、まず個々のユーザに対する制限容量が設定されているかどうかを確認します。個別の制限容量が設定されていない場合、Messaging Server はすべてのユーザに対して設定されているデフォルトの制限容量を確認します。

ユーザのメッセージが制限容量を超えてしまった場合、以下のどちらかの状態になるまで、メッセージは MTA キューに残ったままとなります。

(1) ユーザのメッセージのサイズまたは数が制限容量を超えない状態になったとき。この時点で MTA によってユーザにメッセージが配信されます。(2) 未配信のメッセージが MTA キューに残留している期間が指定された猶予期間を超えてしまったとき。357 ページの「猶予期間を設定するには」を参照してください。

ディスク容量は、ユーザがメッセージを削除または消去したときや、設定された存続期間決定ポリシーに従ってサーバがメッセージを削除したときに使用可能になります。

ドメインの制限容量とファミリーグループの制限容量

特定のドメインや、ドメイン内のファミリーグループに対して制限容量を設定することもできます。これらの制限容量は強制されるものではありませんが、レポート処理を行う場合に役立ちます。

Telephony Application Server に関する例外

統一されたメッセージング要件をサポートするために、Messaging Server ではメッセージストアによって課された制限容量を無効にする機能を提供しています。これにより、特定のエージェント、つまり Telephony Application Servers (TAS) が受け取ったメッセージが確実に配信されます。TAS によって受け入れられたメッセージは特別な MTA チャネルを通るようにルーティングされ、メッセージは制限容量に関係なくストアに配信されるようになります。TAS チャネルの設定の詳細については、第 8 章「チャネル定義を設定する」を参照してください。

メッセージストアの制限容量を設定する

すべてのユーザについてのデフォルトの制限容量は、iPlanet Console または `configutil` コマンドを使用して設定することができます。また、個々のユーザ、ファミリーグループ、およびホストドメインについての制限容量も設定することができます。

このマニュアルでは、デフォルトの制限容量の設定方法について説明します。個々のユーザ、ファミリーグループ、およびホストドメインの制限容量の設定に関する詳細については、『Delegated Administrator's User Guide』を参照してください。

この節では、以下のタスクについて説明します。

- デフォルトのユーザ制限容量を指定するには
- 制限容量の適用と通知を有効にするには
- 猶予期間を設定するには

iPlanet Console を使用する場合は、以下の手順に従います。

1. iPlanet Console から構成を行う Messaging Server を開きます。
2. 「構成」タブをクリックして、左のペインの「メッセージストア」を選択します。
3. 右のペインの「制限容量」タブをクリックします。

デフォルトのユーザ制限容量を指定するには

デフォルトの制限容量は、個別の制限容量がまだ設定されていないユーザに適用されます。個別の制限容量の設定はデフォルトの制限容量よりも優先されます。

コンソール : コンソールでデフォルトの制限容量を指定するには、以下の手順に従います。

1. 「制限容量」タブをクリックします。
2. デフォルトのユーザディスク制限容量を指定するには、「デフォルトのユーザディスク制限容量」フィールドで次のオプションのどちらかを選択します。

   「無制限」: このオプションは、デフォルトのディスク制限容量を設定しない場合に選択します。

   「Size specification」: このオプションは、デフォルトのユーザディスク制限容量を特定のサイズに制限する場合に選択します。ボタンの横のフィールドに数字を入力し、ドロップダウンリストから「M バイト」または「K バイト」を選択します。
3. メッセージ数の制限を指定する場合は、「デフォルトのユーザメッセージ制限容量」ボックスに数字を入力します。
4. 「保存」をクリックします。

コマンドライン : メッセージの合計サイズについてのデフォルトのユーザディスク制限容量を指定する場合は、以下のようになります。

```
configutil -o store.defaultmailboxquota -v [ -1 | number ]
```

ここで `-1` は制限がないことを示し、*number* はバイト数を示します。

メッセージの合計数についてのデフォルトのユーザ制限を指定する場合、以下のようになります。

```
configutil -o store.defaultmessagequota -v [ -1 | number ]
```

ここで `-1` は制限がないことを示し、*number* はメッセージ数を示します。

制限容量の適用と通知を有効にするには

制限容量の適用と通知は、有効にしたり無効にしたりすることができます。サーバの動作は、表 11-3 に示すように、設定変数の設定方法によって異なります。

表 11-3 制限容量の適用と通知

	適用オン	適用オフ
通知オン	メッセージは指定された猶予期間まで据え置かれます。猶予期間が切れたら拒否されます。メッセージをメールボックスに追加することはできません。 IMAP SELECT、IMAP APPEND、SMTP メール送信機能、および配信コマンドによってエラーメッセージが表示されます。	メッセージがストアに配信されます。メッセージをメールボックスに追加することができます。 IMAP SELECT、IMAP APPEND、SMTP メール送信機能、および配信コマンドはエラーメッセージを表示しません。
通知オフ	メッセージは指定された猶予期間まで据え置かれます。猶予期間が切れたら拒否されます。メッセージをメールボックスに追加することはできません。 IMAP SELECT コマンド、配信コマンド、および SMTP メール送信機能はエラーメッセージを表示しません。 IMAP APPEND コマンドによってエラーメッセージが表示されます。	メッセージがストアに配信されます。メッセージをメールボックスに追加することができます。 IMAP SELECT、IMAP APPEND、SMTP メール送信機能、および配信コマンドはエラーメッセージを表示しません。

制限容量の適用を有効にする

コンソール : コンソールで制限容量の適用を有効にするには、以下の手順に従います。

1. 「制限容量」タブをクリックします。
2. 「容量制限実施の有効化」ボックスにチェックマークを付けます。

   このボックスでオンとオフの切り替えを行います。制限容量の適用を無効にする場合はこのボックスのチェックマークを外します。
3. 「保存」をクリックします。

コマンドライン : コマンドラインで制限容量の適用を有効にする場合は、以下のようになります。

```
configutil -o store.quotaenforcement -v [ yes | no]
```

ここで no を指定したら制限容量は適用されません。

制限容量の通知を有効にする

コンソール : コンソールで制限容量の通知を有効にするには、以下の手順に従います。

1. 「制限容量」タブをクリックします。
2. 「容量制限有効化の通知」ボックスにチェックマークを付けます。

   このボックスでオンとオフの切り替えを行います。制限容量の適用を無効にする場合はこのボックスのチェックマークを外します。
3. 制限容量の警告メッセージを定義します。

   356 ページの「制限容量の警告メッセージの定義」を参照してください。
4. 「保存」をクリックします。

コマンドライン : コマンドラインで制限容量の通知を有効にする場合は、以下のようになります。

```
configutil -o store.quotanotification -v [ yes | no ]
configutil -o store.quotaexceededmsg -v message
```

message に何も設定されなかった場合、ユーザには制限容量の警告メッセージは送信されません。

制限容量の警告メッセージの定義

ディスク制限容量を超えたユーザに送信するメッセージは、以下の手順で定義することができます。メッセージはユーザのメールボックスに送られます。

コンソール : コンソールで制限容量の警告メッセージを定義するには、以下の手順に従います。

1. 「制限容量」タブをクリックします。
2. ドロップダウンリストから使用言語を選択します。
3. ドロップダウンリストの下にあるメッセージテキストのフィールドに、送信するメッセージ内容を入力します。
4. 「保存」をクリックします。

コマンドライン : コマンドラインで制限容量の警告メッセージを定義する場合は、以下のようになります。

```
configutil -o store.quotaexceededmsg -v message
```

メッセージは RFC 822 形式でなければなりません。

警告メッセージの送信頻度を定義する場合は、以下のようになります。

```
configutil -o store.quotaexceedmsginterval -v number
```

この *number* は日数を示しています。たとえば、3 が入っていれば 3 日ごとにメッセージが送信されます。

制限容量のしきい値の指定

制限容量のしきい値を指定すれば、IMAP ユーザがディスク制限容量に到達する前に、警告メッセージを送ることができます。ユーザのディスク使用量が指定したしきい値を超えたら、サーバからユーザに警告メッセージが送信されます。

クライアントが IMAP ALERT 機能をサポートしている IMAP ユーザの場合は、ユーザがメールボックスを選択するたびに画面にメッセージが表示されます (メッセージは IMAP ログにも書き込まれます)。

コンソール : コンソールで制限容量のしきい値を指定するには、以下の手順に従います。

1. 「制限容量」タブをクリックします。
2. 「制限容量の警告のしきい値」フィールドに警告しきい値の数字を入力します。

   この数字は許可された制限容量のパーセンテージを表しています。たとえば 90% を選択した場合、ユーザは許可された制限容量の 90% を使用したところで警告を受けることになります。デフォルトは 90% です。この機能をオフにするには 100% と入力します。
3. 「保存」をクリックします。

コマンドライン : コマンドラインで制限容量のしきい値を指定する場合は、以下のようになります。

```
configutil -o store.quotawarn -v number
```

この *number* は許可された制限容量のパーセンテージを示しています。

猶予期間を設定するには

猶予期間は、メッセージを差出人にバウンスするまでメールボックスが制限容量 (ディスク容量やメッセージの数) を超えた状態でいられる期間を指定するものです。MTA がメッセージを受け取っても、メッセージは MTA キューに残り、次のいずれかの状況が発生するまでメッセージストアには配信されません。

- メールボックスが制限容量を超えない状態になったとき。この時点でメールボックスにメッセージが配信されます。
- ユーザが指定された猶予期間を過ぎても制限容量を上回ったままでいるとき。この時点でサーバが、キュー内に含まれているすべてのメッセージをバウンスします。

- メッセージがメッセージキューの最長時間より長くキューに残ったままであるとき。

たとえば、猶予期間が 2 日間に設定されているときに 1 日分の制限容量を超えた場合、新しいメッセージは引き続き受信され、キュー内に保持され、配信試行は続行します。2 日目を過ぎると、メッセージはバウンスされます。

注	猶予期間とは、メッセージがキュー内に保持される期間ではなく、キュー内に含まれているすべての受信メッセージがバウンスされるまでに、メールボックスが制限容量を超えた状態でいられる期間です。

コンソール : コンソールで、メッセージがキューに保持される猶予期間を設定するには、以下の手順に従います。

1. 「制限容量」タブをクリックします。
2. 「制限容量超過時の猶予期間」フィールドに数字を入力します。
3. ドロップダウンリストで「Day(s)」または「Hour(s)」を指定します。
4. 「保存」をクリックします。

コマンドライン : コマンドラインで制限容量の猶予期間を指定する場合は、以下のようになります。

```
configutil -o store.quotagraceperiod -v number
```

この *number* は時間数を示しています。

存続期間決定ポリシーを指定するには

存続期間決定ポリシーは、サーバによるディスク使用を制御するためのもう 1 つの手段です。1 つまたは複数のメールボックスにメッセージが保存される期間を制御することができます。ディスク容量が制限されている場合、存続期間決定ポリシーを設定してストアからメッセージを削除するとよいでしょう。ただし存続期間決定ポリシーを設定する場合には、このポリシーについてユーザを教育する必要があります。サーバがこのポリシーに基づいてストアからメッセージを削除する場合、削除の前に警告メッセージを送信しないからです。

存続期間決定の規則は、以下の条件に基づいて作成できます。

- メールボックス内のメッセージ件数
- メールボックスの合計サイズ
- メールボックス内にメッセージが残っている日数

- 指定されたサイズを超えるメッセージがメールボックスに残っている日数

1 つのメールボックスに対して複数の規則を指定する場合、有効期限に関する規則はすべて適用されますが、もっとも制約度の高い規則が優先されます。たとえば、2 つの規則が 1 つのメールボックスに適用される場合を考えてみます。一方の規則では 1000 件のメッセージが許可されており、もう一方の規則では 500 件のメッセージが許可されています。有効期限が切れた場合、サーバは 500 件のメッセージが残った状態になるまでメールボックスからメッセージを削除します。別の例では、一方の規則では 3 日間で 100,000 バイトのメッセージが許可されており、もう一方の規則では 12 日間で 1000 バイトのメッセージが許可されています。この場合、規則が結合された結果、3 日間で 100,000 バイトのメッセージサイズが許可されることになります。つまり、サーバは、メールボックスに 4 日以上存在する 100,000 バイトを超えるメッセージを削除するのです。特定のメールボックスまたはメールボックスのセットだけに特別な規則を適用させたい場合は、Exclusive パラメータを使用してください。

コンソール : コンソールを使用して新しい規則を作成するには、以下の手順に従います。

1. iPlanet Console から構成を行う Messaging Server を開きます。
2. 「構成」タブをクリックして、左のペインの「メッセージストア」を選択します。
3. 右のペインの「存続期間」タブをクリックします。
4. 「追加」をクリックして「ルールの通知」ウィンドウに進みます。
5. 新しい規則の名前を入力します。
6. この規則が適用されるターゲットフォルダを指定します。

   パス名、ファイル名、または文字列の一部分が入力できます。以下に示す IMAP ワイルドカードも使用できます。

   * - あらゆる文字列に一致します。
   % - スラッシュ (/) 以外のあらゆる文字列に一致します。

   新しい規則は、指定したパターンに一致するフォルダのみに適用されます。
7. この規則がターゲットフォルダに適用される唯一の規則である場合、「Exclusive」選択ボックスをクリックします。
8. フォルダサイズに基づいて規則を作成する場合は、以下を実行します。
   - 「メッセージ件数」フィールドには、もっとも古いメッセージが削除されるまでフォルダ内に保持されるメッセージの最大件数を指定します。
   - 「フォルダサイズ」フィールドには、フォルダサイズを数字で指定します。また、それに続くドロップダウンリストから「M バイト」または「K バイト」を選択します。

   指定したフォルダサイズを超えた場合、このサイズ内に収まるまでサーバはもっとも古いメッセージから順に削除していきます。

9. メッセージの存続期間に基づいて規則を作成する場合は、「日数」フィールドにメッセージをフォルダに残すべき日数を数字で指定します。
10. メッセージサイズに基づいて規則を作成する場合は、以下を実行します。
    - 「メッセージサイズの制限」フィールドには、フォルダ内で許可されたメッセージの最大サイズを示す数字を入力します。また、それに続くドロップダウンリストから「M バイト」または「K バイト」を選択します。
    - 「猶予期間」フィールドには、指定されたサイズを超えたメッセージをフォルダに残さなければならない日数を示す数字を入力します。

    猶予期間が過ぎたら、サーバが最大サイズを超えたメッセージを削除します。
11. 「OK」をクリックして新しい規則を「存続期間ルール」リストに追加し、「追加」ウィンドウを閉じます。
12. 「保存」をクリックして現在の「存続期間ルール」リストを送信し保存します。

コマンドライン : コマンドラインで新しい規則を作成する場合は、以下の各コマンドを使用します。ここで *name* は、この規則に付けた名前を表しています。ただし、ここではもっとも頻繁に使用する `store.expire*` オプションのみを説明しています。完全なリストについては、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

この規則が適用されるターゲットフォルダを指定するには

```
configutil -o store.expirerule.name.folderpattern -v pattern
```

たとえば、パターン `user/*` は、すべてのフォルダをターゲットにしています。パターン `user/%@siroe.com/*` は、ドメイン `siroe.com` に存在する全ユーザの全フォルダがターゲットです。さらに、パターン `user/%/Trash` では、すべてのユーザのごみ箱 (Trash) フォルダをターゲットにしています。

この規則をターゲットフォルダに適用される唯一の規則に指定するには、次のように入力します。

```
configutil -o store.expirerule.name.exclusive -v [ yes | no ]
```

もっとも古いメッセージが削除されるまでフォルダ内に保持されるメッセージの最大件数を指定するには、次のように入力します。

```
configutil -o store.expirerule.name.messagecount -v number
```

フォルダサイズを指定するには、次のように入力します。

```
configutil -o store.expirerule.name.foldersizebytes -v number
```

この *number* は、バイト数で表されたサイズです。

メッセージの存続期間を指定するには、次のように入力します。

```
configutil -o store.expirerule.name.messagedays -v number
```

この *number* は日数を示しています。

メッセージサイズを指定するには、次のように入力します。

```
configutil -o store.expirerule.name.messagesize -v number
```

この *number* は、バイト数で表されたサイズです。

指定されたサイズを超えたメッセージをフォルダに残さなければならない期間を示すには、次のように入力します。

```
configutil -o store.expirerule.name.messagesizedays -v number
```

この *number* は日数を示しています。

有効期限の日時を指定するには

有効期限の日時は次のように指定します。

```
configutil -o store.expirestart -v time （例 : 23 は 11:00PM）
configutil -o local.store.expire.workday -v day (0 ～ 6、0 は日曜 )
```

local.store.expire.workday を -1 または 6 より大きい値に設定すると、有効期限とクリーンアップが無効になります。stored は、毎日 store.expirestart で指定した時間にこの設定変数をチェックします。local.store.expire.workday が設定されていない場合、デフォルトで毎日実行されます。この変数を変更したあとで stored を再起動する必要はありません。

メッセージストアのパーティションを構成する

ユーザメールボックスはデフォルトではすべて、msg-*instance*/store/partition/ディレクトリに保存されています。partition ディレクトリは、単一または複数のサブパーティションを格納している論理的なディレクトリです。サブパーティションは、単一または複数の物理ドライブにマッピングされます。起動時には、partition ディレクトリに primary パーティションと呼ばれるサブパーティションが格納されています。

必要に応じて partition ディレクトリにパーティションが追加できます。たとえば、ユーザを体系化するために 1 つのディスクを分割する場合、以下のようになります。

```
msg-instance/store/partition/mkting/
msg-instance/store/partition/eng/
msg-instance/store/partition/sales/
```

ディスクストレージに対する要求が高まるに従い、これらのパーティションを異なる物理ディスクドライブにマッピングする必要が生じてくると考えられます。

どのディスクでもメールボックスの数を制限しなければなりません。メールボックスを複数のディスクに分散させることにより、メッセージ配信時間を短縮することができます (ただし、必ずしも SMTP の受け入れ率が変更されるわけではありません)。ディスクごとに割り当てるメールボックスの数は、ディスク容量や各ユーザに割り当てられたディスク容量によって異なります。たとえば、ユーザごとのディスク容量の割り当て量が少ない場合は、ディスクごとに割り当てるメールボックスの数を多くできます。

メッセージストアに複数のディスクを必要とする場合、RAID (Redundant Array of Inexpensive Disks) 技術を使用すれば複数ディスクの管理を容易に行うことができます。RAID 技術によってデータを一連のディスクに分散させることができます。このときディスクは単一の論理ボリュームとして表示されるので、ディスク管理が簡単になります。また、冗長性を得るために RAID 技術を使用することもできます。つまり、障害復旧用にストアを複製する目的で使用することができるわけです。

注 ディスクアクセスを向上させるには、メッセージストアとメッセージキューを別のディスクに配置しておく必要があります。

パーティションを追加するには

パーティションを追加する場合、ディスク上でパーティションが保存されている場所の絶対的な物理パスと、パーティションニックネームと呼ばれる論理名を指定します。

パーティションニックネームにより、物理パスに関係なくユーザを論理的なパーティション名にマッピングさせることができます。ユーザアカウントの設定時やユーザのメッセージストアを指定するときに、パーティションニックネームを使用できます。名前の入力に使用するのは英数字で、アルファベットは小文字を使用してください。

パーティションを作成および管理するには、サーバの実行に使用するユーザ ID が、物理パスで指定した場所への書き込み権限を持っていなければなりません。

注 パーティションを追加したら、構成情報を更新するためにサーバをいったん停止してから再起動する必要があります。

コンソール : コンソールを使用してストアにパーティションを追加するには、以下の手順に従います。

1. iPlanet Console から構成を行う Messaging Server を開きます。
2. 「構成」タブをクリックして、左のペインの「メッセージストア」を選択します。
3. 右のペインの「パーティション」タブをクリックします。

4. 「追加」ボタンをクリックします。
5. パーティションニックネームを入力します。

   これは指定したパーティションの論理名です。
6. パーティションのパスを入力します。

   これは指定したパーティションの絶対パス名です。
7. これをデフォルトのパーティションに指定するには、「デフォルトのパーティションにする」というラベルの付いた選択ボックスをクリックします。
8. 「OK」をクリックしてこのパーティション構成エントリを送信し、ウィンドウを閉じます。
9. 「保存」をクリックして現在のパーティションリストを送信し保存します。

コマンドライン : コマンドラインでストアにパーティションを追加する場合は、以下のようになります。

```
configutil -o store.partition.nickname.path -v path
```

ここで、*nickname* はパーティションの論理名、*path* はパーティションが保存されている場所の絶対パス名を示しています。

デフォルトのプライマリパーティションのパスは次のように指定します。

```
configutil -o store.partition.primary.path -v path
```

メールボックスを別のディスクパーティションに移動するには

特に設定を変更しないかぎり、メールボックスは `primary` パーティション内に作成されます。このパーティションの容量が一杯になると、メッセージを保存することができなくなります。この問題には、次のような対応策があります。

- ユーザのメールボックスのサイズを小さくする
- 容量管理ソフトウェアを使用している場合、別のディスクを追加する
- 別のパーティションを作成し (362 ページの「パーティションを追加するには」)、メールボックスを新しいパーティションに移動する

可能なかぎり、容量管理ソフトを使用して、システムにディスク容量を追加する方法をお勧めします。これは、この方法がユーザにとってもっとも透過性が高いからです。ただし、次の手順に従って、メールボックスを別のパーティションに移動することもできます。

1. 移行プロセス中は、ユーザがメールボックスに接続していない状態にしてください。このためには、ユーザに通知を出して、メールボックスの移動作業を行う前にログオフし、作業期間中にログオンしないように指示します。または、ユーザがログオフしたあと、POP、IMAP、および HTTP のサービスを使用できないように mailAllowedServiceAccess 属性を設定します『iPlanet Messaging Server プロビジョニングガイド』のユーザのプロビジョニングの章を参照してください。

注	POP、IMAP、HTTP へのアクセスを許可しないように mailAllowedServiceAccess を設定しても、ユーザがすでにメールボックスに接続している場合に、その接続が切断されることはありません。このため、メールボックスを移動する前に、すべての接続が切断されていることを確認してください。

2. ユーザのメールボックスを移動するには、次のコマンドを使用します。

   mboxutil -r user/<*userid*>/INBOX user/<*userid*>/INBOX <*partition_name*>

   例 :

   mboxutil -r user/ofanning/INBOX user/ofanning/INBOX secondary

3. 移動したユーザの LDAP エントリで mailMessageStore 属性を新しいパーティションの名前に設定します。

   例 : mailMessageStore: secondary

4. ユーザにメッセージストアへの接続が再開されたことを通知します。必要に応じて、POP、IMAP、および HTTP サービスを使用できるように mailAllowedServiceAccess 属性を変更します。

保守および回復手順を実行する

この節では、メッセージストアの保守タスクと回復タスクを実行するのに使用するユーティリティについて説明します。サーバから送信される警告のためのポストマスターメールを常に読む必要があります。また、サーバの実行状況に関する情報を記録したログファイルをモニタする必要もあります。ログファイルに関しては、第 13 章「ログ記録とログ解析」を参照してください。

この節では以下の内容について説明します。

- メールボックスを管理するには
- 制限容量をモニタするには
- ディスク容量をモニタするには
- stored ユーティリティを使用する

- メールボックスとメールボックスデータベースの修復

メールボックスを管理するには

この節では、メールボックスの管理およびモニタを行う次のユーティリティについて説明します。mboxutil、hashdir、readership。

mboxutil ユーティリティ

mboxutil コマンドを使用して、メールボックスの一般的な保守タスクを実行します。タスクには以下のものが含まれます。

- メールボックスの一覧表示
- メールボックスの作成
- メールボックスの削除
- メールボックスの名前変更
- パーティション間のメールボックスの移動

また、mboxutil コマンドを使用して制限容量に関する情報を表示することもできます。詳細は、369 ページの「制限容量をモニタするには」を参照してください。

表 11-4 は mboxutil コマンドの一覧です。シンタックスや使用要件の詳細については、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

表 11-4 mboxutil のオプション

オプション	説明
-a	すべてのユーザの制限容量に関する情報を表示します。
-c *mailbox*	指定したメールボックスを作成します。
-d *mailbox*	指定したメールボックスを削除します。
-f *file*	指定したデータファイルに一覧表示されているメールボックスを作成、削除、またはロックします。
-g *group*	指定したグループの制限容量に関する情報を表示します。
-k *mailbox cmd*	指定したメールボックスをフォルダレベルでロックし、指定したコマンドを実行し、コマンドが完了したらメールボックスのロックを解除します。
-l	サーバのすべてのメールボックスを一覧表示します。

表 11-4　mboxutil のオプション (続き)

オプション	説明
-p *pattern*	-l オプションとともに使用した場合、名前が *pattern* と一致するメールボックスのみが一覧表示されます。IMAP ワイルドカードを使用できます。
-q *domain*	指定したドメインの制限容量に関する情報を一覧表示します。
-r *oldname newname* [*partition*]	メールボックスの名前を *oldname* から *newname* に変更します。フォルダを別のパーティションに移動するには、*partition* オプションに新しいパーティションを指定します。 このオプションを使用してユーザ名を変更することができます。たとえば、mboxutil -r user/user1/INBOX user/user2/INBOX では、user1 のすべてのメールとメールボックスが user2 に移動し、新しいメッセージは新しい INBOX に表示されます (user2 がすでに存在している場合、この操作は失敗します)。
-u *user*	メールストアの現在のサイズ、制限容量 (設定されている場合)、現在使用されている制限容量の割合など、ユーザのメールストアのサイズに関する情報を一覧表示します。
-x	-l オプションとともに使用すると、メールボックスのパスとアクセス制御が表示されます。

メールボックスの命名規則

メールボックス名は、次のフォーマットで指定します。user/*userid*/*mailbox*。ここで、*userid* はメールボックスを所有するユーザ、*mailbox* はメールボックスの名前を表します。ホストドメインでは、*userid* は *uid@domain* です。

たとえば次のコマンドでは、ユーザ ID が crowe であるユーザの、INBOX という名前のメールボックスが作成されます。INBOX は、ユーザ crowe に配信されたメール用のデフォルトのメールボックスとなります。

```
mboxutil -c user/crowe/INBOX
```

重要 : INBOX という名前は、各ユーザのデフォルトのメールボックス用に確保してある名前です。INBOX は、大文字と小文字が区別されない唯一のフォルダです。ほかのフォルダ名はすべて大文字と小文字が区別されます。

例

全ユーザの全メールボックスを一覧表示するには、次のように入力します。

```
mboxutil -l
```

すべてのメールボックスを、パスと ACL の情報とともに一覧表示するには、次のように入力します。

```
mboxutil -l -x
```

ユーザ daphne に対し、INBOX というデフォルトのメールボックスを作成するには、次のように入力します。

```
mboxutil -c user/daphne/INBOX
```

ユーザ delilah に対し、projx という名前のメールフォルダを削除するには、次のように入力します。

```
mboxutil -d user/delilah/projx
```

ユーザ druscilla について、INBOX というデフォルトのメールボックスとすべてのメールフォルダを削除するには

```
mboxutil -d user/druscilla/INBOX
```

ユーザ desdemona の memos というメールフォルダの名前を、memos-april という名前に変更するには、次のように入力します。

```
mboxutil -r user/desdemona/memos user/desdemona/memos-april
```

ユーザ dulcinea の legal という名前のメールフォルダをロックするには、次のように入力します。

```
mboxutil -k user/dulcinea/legal cmd
```

この場合の *cmd* は、フォルダがロックされている間に実行するコマンドです。

ユーザ dimitria のメールアカウントを新しいパーティションに移動するには、次のように入力します。

```
mboxutil -r user/dimitria/INBOX user/dimitria/INBOX partition
```

この場合、*partition* には新しいパーティションの名前を指定します。

ユーザ dimitria のメールフォルダ personal を新しいパーティションに移動するには、次のように入力します。

```
mboxutil -r user/dimitria/personal user/dimitria/personal partition
```

hashdir ユーティリティ

メッセージストア内のメールボックスは、高速で検索できるようにハッシュ構造で保存されています。従って、特定のユーザのメールボックスを格納するディレクトリを検索するには、hashdir ユーティリティを使用します。

このユーティリティは、特定のアカウントのメッセージストアを含むディレクトリを識別します。また、メッセージストアへの相対パスをレポートします。これは d1/a7/ のようになります。このパスは、ユーザ ID に基づくディレクトリの 1 つ上のディレクトリレベルを基準にしたものです。このユーティリティによってパス情報が標準出力に送られます。

たとえば、ユーザ crowe のメールボックスへの相対パスを検索する場合は次のようになります。

```
hashdir crowe
```

readership ユーティリティ

readership ユーティリティは、メールボックスの所有者以外に、何人のユーザが共有 IMAP フォルダ内のメッセージを読んだかを報告するユーティリティです。

IMAP フォルダの所有者は、フォルダ内のメールを読む権限をほかのユーザに与えることができます。ほかのユーザにアクセス権が与えられたフォルダは、共有フォルダと呼ばれます。管理者は readership ユーティリティを使用して、所有者以外に何人のユーザが共有フォルダにアクセスしたかを表示することができます。

このユーティリティは、すべてのメールボックスをスキャンして、各共有フォルダにつき 1 行ずつ、アクセスしたユーザ数とメールボックスの名前を表示させます。ユーザ数とメールボックスの名前の間にはスペースが挿入されます。

アクセスしたユーザとは、過去の指定した日数内に共有フォルダを選択した、個別の認証を受けたユーザのことです。自分の個人用メールボックスを読んだユーザは、数には含められません。個人用メールボックスは、フォルダの所有者以外に購読者がいない場合は、レポートされません。

たとえば次のコマンドでは、最近の 15 日以内に共有の IMAP フォルダを選択したユーザをすべてカウントします。

```
readership -d 15
```

制限容量をモニタするには

mboxutil ユーティリティを使用して、制限容量の使用状況やその限界をモニタすることができます。mboxutil ユーティリティは、定義された制限容量を一覧表示し、制限容量の使用状況に関する情報を提供するレポートを生成します。制限容量と使用状況に関する数値は、キロバイト (KB) でレポートされます。

たとえば次のコマンドでは、全ユーザの制限容量に関する情報を一覧表示します。

```
mboxutil -a
```

次の例では、ユーザ crowe の制限容量に関する情報を一覧表示します。

```
mboxutil -u crowe
```

次の例では、ドメイン siroe.com の制限容量に関する情報を一覧表示します。

```
mboxutil -q siroe.com
```

ディスク容量をモニタするには

システムがディスク容量をモニタする頻度と、システムが警告を送信する環境条件を指定することができます。ディスク容量のモニタと通知について設定するには、configutil コマンドを使用してディスク容量の警告属性を設定します。表 11-5 を参照してください。

表 11-5 ディスク容量の警告属性

ディスク容量の属性	デフォルト値
alarm.diskavail.msgalarmstatinterval	3600 秒
alarm.diskavail.msgalarmthreshold	10%
alarm.diskavail.msgalarmwarninginterval	24 時間

たとえば、システムがディスク容量を 600 秒毎にモニタするようにするには、次のコマンドを指定します。

```
configutil -o alarm.diskavail.msgalarmstatinterval -v 600
```

使用可能なディスク容量が 20% を下回ったら常に警告を受け取るようにするには、次のコマンドを指定します。

```
configutil -o alarm.diskavail.msgalarmthreshold -v 20
```

警告属性の設定の詳細については、『iPlanet Messaging Server リファレンスマニュアル』および 504 ページの「ディスク容量をモニタする」を参照してください。

stored ユーティリティを使用する

stored ユーティリティは、以下の監視タスクと保守タスクをサーバに対して実行します。

- バックグラウンドと日常のメッセージ処理タスク
- デッドロックの検出とデッドロックしたデータベーストランザクションのロールバック
- 起動時の一時ファイルのクリーンアップ
- 存続期間決定ポリシーの実装
- サーバの状態、ディスク容量、サービスへの応答時間などの定期的モニタ (514 ページの「stored」を参照)
- 必要に応じて警告を生成

stored ユーティリティは毎日午後 11 時に自動的にクリーンアップと (有効期限による) 失効の操作を行います。また、これ以外の時間にもクリーンアップと失効の操作を行うように選択することもできます。

表 11-6 では stored オプションを一覧表示しています。一般的な使用例についてはこの表に従ってください。シンタックスや使用要件の詳細については、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

表 11-6 stored オプション

オプション	説明
-c	削除されたメッセージを消去するためにクリーンアップを 1 回実行します。1 回だけ実行し、終了します。-c オプションは 1 回のみの処理で、-1 オプションを指定する必要はありません。
-d	デーモンとして実行します。システムチェックを実行し、警告、デッドロック検出、およびデータベース修復をアクティブにします。
-1	1 回だけ実行し、終了します。
-n	トライアルモードでのみ実行します。メッセージを実際に期限切れにしたり、クリーンアップすることはありません。1 回だけ実行し、終了します。
-v	詳細モード出力を行います。

表 11-6　stored オプション (続き)

オプション	説明
-v -v	その他の詳細モード出力を行います。

有効期限ポリシーをテストするには、次のように入力します。

```
stored -n
```

保存期間の終了とクリーンアップを 1 回実行するには、次のように入力します。

```
stored -l -v
```

自動的なクリーンアップと失効の操作の時間を変更する場合は、以下のように configutil ユーティリティを使用します。

```
configutil -o store.expirestart -v 21
```

場合によっては、stored ユーティリティを再起動する必要があるかもしれません。たとえば、メールボックスリストのデータベースが破損した場合などです。UNIX 上で stored を再起動するには、コマンドラインで以下のコマンドを使用します。

server-root/msg-*instance*/stop-msg store
server-root/msg-*instance*/start-msg store

サーバのいずれかのデーモンがクラッシュした場合は、すべてのデーモンを停止させ、stored を含むすべてのデーモンを再起動しなくてはなりません。

メールボックスとメールボックスデータベースの修復

1 つまたは複数のメールボックスが破損した場合、reconstruct ユーティリティを使用してメールボックスまたはメールボックスデータベースを再構築し、すべての矛盾を修復することができます。

reconstruct ユーティリティは、1 つまたは複数のメールボックスまたはマスターメールボックスファイルを再構築し、すべての矛盾を修復します。このユーティリティを使うと、メールストアにおけるほとんどすべてのデータ破損を回復することができます。トランザクションの完了や、完了しなかったトランザクションのロールバックなど、低レベルのデータベースの修復には stored -d を使用します。

表 11-7 では、reconstruct オプションを一覧表示しています。シンタックスや使用要件の詳細については、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

表 11-7 reconstruct オプション

オプション	説明
-f	reconstruct を強制的に実行して、メールボックスを修復します。
-m	メールボックスデータベースを修復し、整合性チェックを行います。このオプションにより、スプール領域にあるすべてのメールボックスがチェックされ、必要に応じてメールボックスデータベースのエントリの追加または削除が行われます。このユーティリティは、データベースでエントリの追加または削除が行われると、メッセージが標準出力ファイルに出力されます。
-n	メールボックスを修復をせずに、メッセージストアだけをチェックします。メールボックス名が指定されていない場合は、-n オプションを単独で使用することはできません。メールボックス名が指定されていないときは、-n オプションを -r オプションとともに使用する必要があります。-r オプションは -p オプションと組み合わせて使用することもできます。たとえば、以下のコマンドはすべて有効です。 reconstruct -n user/dulcinea/INBOX reconstruct -n -r reconstruct -n -r -p primary reconstruct -n -r user/dulcinea/
-o	孤立したアカウントをチェックします。このオプションは、現在の Messaging Server ホスト内で、対応するエントリが LDAP にない INBOX を検索します。たとえば、-o オプションは、所有者が LDAP から削除された、または別のサーバホストに移動された INBOX を検索します。見つかった孤立アカウントのそれぞれに対し、reconstruct ユーティリティは標準出力に次のコマンドを書き込みます。 mboxutil-d user/*userid*/INBOX
-o -d *filename*	-o オプションで -d *filename* が指定されている場合、reconstruct は指定したファイルを開き、そのファイルに mboxutil -d コマンドを書き込みます。このファイルをスクリプトファイルにして、孤立したアカウントを削除することができます。
-p *partition*	パーティション名を指定します。フルパス名は使用しないでください。このオプションが指定されていない場合、reconstruct はすべてのパーティションにデフォルト設定されています。
-q	制限容量サブシステムの矛盾 (メールボックスの制限容量ルートが正しくない、または制限容量ルートで誤った容量の使用状況がレポートされるなど) を修正します。-q オプションは、ほかのサーバプロセスの実行中に実行できます。

表 11-7　reconstruct オプション (続き)

オプション	説明
-r [*mailbox*]	指定したメールボックスのパーティション領域を修復し、整合性チェックを実行します。また、-r オプションは、指定したメールボックス内のすべてのサブメールボックスも修復します。-r を指定してメールボックス引数を入力しなかった場合、ユーティリティがユーザパーティションディレクトリ内にあるすべてのメールボックスのスプール領域を修復します。

メールボックスを再構築するには

メールボックスを再構築するには -r オプションを使用します。このオプションは以下の場合に使用します。

- メールボックスにアクセスしたら次のどちらかのエラーが返された :「システム I/O エラー」または「メールボックスのフォーマットが不正です」
- メールボックスにアクセスしたらサーバがクラッシュした
- ファイルがスプールディレクトリに追加されたか、スプールディレクトリから削除された

5.0 リリースでは、reconstruct -r は、最初に整合性チェックを実行します。問題が検出されたときのみ整合性および再構築についてレポートされます。従って、このリリースでは reconstruct ユーティリティのパフォーマンスが向上しています。

reconstruct は、次の例で説明するように使用することができます。

ユーザ daphne に属するメールボックスのスプール領域を再構築するには、次のコマンドを使用します。

```
reconstruct -r user/daphne
```

メールボックスデータベースに一覧表示されたすべてのメールボックスのスプール領域を再構築するには、次のように入力します。

```
reconstruct -r
```

ただし、このオプションは注意して使用してください。メールボックスデータベースに一覧表示されたすべてのメールボックスのスプール領域を再構築する場合、メッセージストアが大規模なため非常に長い時間を要する可能性があるからです (375 ページの「reconstruct のパフォーマンス」を参照)。これよりも優れた障害復旧に対する手段は、ストア用に複数のディスクを使用することでしょう。ディスクが 1 つダウンしてもストア全体がダウンすることはないからです。ディスクが破損した場合、次のように -p オプションを使用してストアの一部分を再構築するだけですみます。

```
reconstruct -r -p subpartition
```

コマンドラインの引数にリストされたメールボックスが primary パーティションに存在する場合のみそれらを再構築するには、次のように入力します。

```
reconstruct -p primary mbox1 mbox2 mbox3
```

primary パーティションに存在するすべてのメールボックスを再構築する必要がある場合は、以下のようになります。

```
reconstruct -r -p primary
```

整合性チェックを実行せずにフォルダを再構築する場合は、-f オプションを使用します。たとえば、次のコマンドはユーザフォルダ daphne の再構築を実行します。

```
reconstruct -f -r user/daphne
```

すべてのメールボックスを修正せずにチェックする場合は、以下のように -n オプションを使用します。

```
reconstruct -r -n
```

メールボックスのチェックと修復

高レベルの整合性チェックを行い、メールボックスデータベースを修復するには次のようになります。

```
reconstruct -m
```

-m オプションは以下の場合に使用します。

- 1 つまたは複数のディレクトリがストアスプール領域から削除されたため、メールボックスデータベースのエントリも削除する必要が生じた場合。
- 1 つまたは複数のディレクトリがストアスプール領域にリストアされたため、メールボックスデータベースのエントリも追加する必要が生じた場合。
- stored -d オプションによってデータベースの整合性を保つことができない場合。

  stored -d オプションによってデータベースの整合性を保つことができない場合、以下の手順を順番に実行します。

  - すべてのサーバを停止します。
  - *server-root*/msg-*instance*/store/mboxlist 内のすべてのファイルを削除します。
  - サーバプロセスを再起動します。
  - reconstruct -m を実行して、スプール領域の内容から新しいメールボックスデータベースを構築します。

孤立したアカウントを削除するには

孤立したアカウント (孤立アカウントとは、対応するエントリが LDAP にないメールボックスのことです) を検索するには、以下のようになります。

```
reconstruct -o
```

コマンド出力が以下のように続きます。

```
reconstruct: Start checking for orphaned mailboxes
mboxutil -d user/test/annie/INBOX
mboxutil -d user/test/oliver/INBOX
reconstruct: Found 2 orphaned mailbox(es)
reconstruct: Done checking for orphaned mailboxes
```

孤立したメールボックスをリストしたファイル (このファイルはスクリプトファイルにして、孤立したアカウントを削除することができます) を作成するには、以下のようになります。ここで、このファイルは orphans.cmd という名前になります。

```
reconstruct -o -d orphans.cmd
```

コマンド出力が以下のように続きます。

```
reconstruct: Start checking for orphaned mailboxes
reconstruct: Found 2 orphaned mailbox(es)
reconstruct: Done checking for orphaned mailboxes
```

reconstruct のパフォーマンス

reconstruct によって操作を実行するのにかかる時間は、以下に挙げるようないくつかの要素によって異なってきます。

- 実行される処理と選択したオプションの種類
- ディスクパフォーマンス
- reconstruct -m 実行時のフォルダの数
- reconstruct -r 実行時のメッセージの数
- メッセージストアの全体サイズ
- システムが実行するほかの処理とシステムのビジー状態
- 実行中の POP、IMAP、HTTP、または SMTP アクティビティが存在するかどうか

reconstruct -r オプションにより、最初の整合性チェックが実行されます。このチェックでは、再構築の必要なフォルダの数に応じて reconstruct のパフォーマンスが向上します。

ユーザ数が約 2400、メッセージストアが 85G バイトで、POP、IMAP、または SMTP アクティビティが同時にサーバで実行されている場合を例として考えてみます。

- reconstruct -m に要した時間は約 1 時間でした。
- reconstruct -r -f に要した時間は約 18 時間でした。

注 reconstruct の操作にかかる時間は、サーバで POP、IMAP、HTTP、または SMTP アクティビティが実行されていない場合、大幅に減少します。

メッセージストアのバックアップとリストアを行う

バックアップとリストアは、もっとも一般的で重要な管理タスクです。メッセージストアにバックアップとリストアのポリシーを実装して、以下のような問題が発生した場合でも、データが失われないようにしておかなければなりません。

- サーバ間でメールボックスを移動
- システムのクラッシュ
- ハードウェア障害
- メッセージまたはメールボックスを誤って削除した
- システムの再インストール時またはアップグレード時の問題
- 天災 (地震、火事、台風など)

また、ユーザを移行する場合にもデータのバックアップが必要です。

Messaging Server では、メッセージストアのバックアップとリストアを行うためのコマンドラインユーティリティを提供しています。また、Messaging Server では Legato Networker® との統合ソリューションも提供しています。

Messaging Server は、単一コピーによるバックアップ手順を提供しています。特定のメッセージを格納するユーザフォルダがいくつあるかにかかわらず、バックアップ時には、メッセージファイルは最初に見つかったメッセージファイルを使用して 1 度バックアップされるだけです。2 番目のメッセージコピーは、最初のメッセージファイル名へのリンクとしてバックアップされます。さらに以下同様に続きます。backup

ユーティリティは、メッセージファイルのデバイスや inode をインデックスとして使用して、すべてのメッセージのハッシュテーブルを管理します。ただし、この方法を採用する場合はデータのリストア時に注意が必要です。詳細は、380 ページの「部分リストアに関する考察」を参照してください。

この節には、以下の項があります。

バックアップポリシーの作成

バックアップポリシーは以下のようないくつかの要素に依存しています。

- ビジネス負荷のピーク
- フルバックアップと増分バックアップ
- 同時バックアップと順次バックアップ

ビジネス負荷のピーク

システムのバックアップのスケジュールを設定する場合は、ビジネス負荷のピークを考慮に入れる必要があります。たとえば、バックアップは 2:00 a.m など早朝 (深夜) の時間帯にスケジュール設定するのが最善であると考えられます。

フルバックアップと増分バックアップ

増分バックアップとは、ストアをスキャンして変更データを見つけ、変更分だけをバックアップする方法です。フルバックアップとは、メッセージストア全体をバックアップすることです。システムが増分バックアップに対してどのくらいの頻度でフルバックアップを実行するのかを決定する必要があります。増分バックアップは、日常の保守手順の中で実行する必要があるはずです。フルバックアップは、データを移動または移行する必要のある場合に適しています。

同時バックアップと順次バックアップ

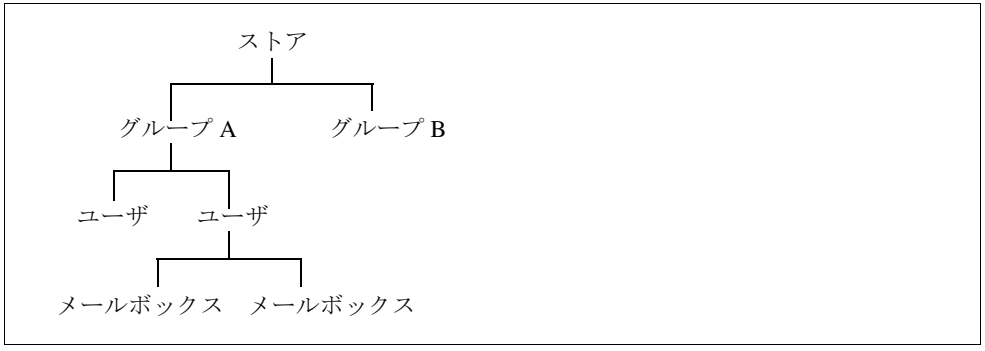
ユーザのデータが複数のディスクに保存されている場合、必要に応じて複数のユーザグループを同時にバックアップすることができます。システムリソースによっては、同時バックアップによってバックアップ手順全体の処理速度を向上させることができます。ただし、たとえばサーバのパフォーマンスに影響を与えたくないような場合、順次バックアップを実行することもあります。同時バックアップを行うか順次バックアップを行うかは、システム負荷、ハードウェア構成、使用可能なテープドライブの数など、多くの要素によって決まります。

バックアップグループを作成するには

ユーザをグループ化することで、バックアップ管理を効率的に行うことができます。たとえば、各グループに別々のバックアップセッションを指定することができます。また、複数のグループを同時にバックアップすることもできます。

ユーザメッセージがユーザの名前順に保存されている場合は、A で始まる名前のユーザが 1 つのバックアップグループとなり、B で始まる名前のユーザは別のバックアップグループになります。

メッセージストアの論理ビューは以下のようになります。



ユーザをグループにカタログ化することで、バックアップ管理を効率的に行うことができます。たとえば、各グループに別々のバックアップセッションを指定することができます。また、複数のグループを同時にバックアップすることもできます。バックアップグループの作成の詳細については、378 ページの「バックアップグループを作成するには」を参照してください。

バックアップグループを作成する場合、グループの定義を保存する設定ファイルを作成する必要があります。このファイルは `backup-groups.conf` という名前が付けられ、次のディレクトリに保存する必要があります。

server_root/msg-*instance*/config/backup-groups.conf

このファイルのフォーマットは次のとおりです。

```
groups=definitions
groups=definitions
.
.
.
```

たとえば、ユーザ ID の最初の文字でユーザをグループ化する場合、以下の定義を使用します。

```
groupA=a*
groupB=b*
groupC=c*
```

バックアップオブジェクトは、以下のようにメッセージストアの論理構造を使用して命名されます。

/server/group/user/mailbox

この *server* はメッセージストアのインスタンス名で、たとえば、以下のようになります。

```
siroe
```

Messaging Server には backup-groups 設定ファイルを作成しなくても使用することができる、事前定義のバックアップグループが含まれています。これは ALL という名前のグループで、ここにはすべてのユーザが含まれています。

Messaging Server のバックアップとリストアのユーティリティ

データのバックアップとリストアのために、Messaging Server では imsbackup および imsrestore ユーティリティが提供されています。

imsbackup および imsrestore ユーティリティは総合的なバックアップ機能を提供するものではないので注意してください。これらのユーティリティは、Legato Networker のような汎用目的ツールに見られる高度な機能は備えていません。たとえば、このユーティリティでは、テープのオートチェンジャーに対して非常に限定されたサポートが行われているだけです。複数の同時実行デバイスに単一のストアを書き込むことはできないのです。総合的なバックアップは、Legato Networker などの一般化ツールのプラグインを使用して達成することができます。Legato Networker の使用に関する詳細は、382 ページの「Legato Networker を使用するには」を参照してください。

imsbackup ユーティリティ

imsbackup ユーティリティを使用すると、選択したメッセージストアの内容を、シリアルデバイス (磁気テープ、UNIX パイプ、通常のファイルなど) に書き込むことができます。バックアップの全体または一部は、あとから imsrestore ユーティリティを使って回復できます。imsbackup の出力は、imsrestore に受け渡すことができます。

バックアップを実行するには、以下の例に示すように imsbackup コマンドを発行します。このコマンドは、user1 を backupfile にバックアップします。

```
imsbackup -f backupfile /mystore/ALL/user1
```

このコマンドはデフォルトのブロック係数である 20 を使用します。imsbackup コマンドの完全なシンタックスに関する説明は、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

imsrestore ユーティリティ

バックアップデバイスからメッセージをリストアするには、imsrestore コマンドを使用してください。たとえば、次のコマンドは backupfile から user1 のメッセージをリストアします。

```
imsrestore -f backupfile /mystore/ALL/user1
```

imsbackup コマンドの完全なシンタックスに関する説明は、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

部分リストアに関する考察

この単一コピーによるバックアップ手順では、メッセージのリストアの際に以下の点に注意する必要があります。

- **フルリストア** : フルリストアでは、リンクの付いたメッセージは、依然としてリンク先のメッセージファイルと同じ inode をポイントしています。

- **部分バックアップ / リストア** : 部分バックアップおよび部分リストアでは、メッセージストアの単一コピーの特性は保持されないことがあります。

以下のように、3 人のユーザ A、B、C に属する 3 つのメッセージが存在すると仮定してみてください。

A/INBOX/1
B/INBOX/1
C/INBOX/1

例 1 : 最初の例では、システムは部分バックアップとフルリストアを以下のように実行します。

1. ユーザ B および C をバックアップします。
2. ユーザ B および C を削除します。
3. 手順 1 のバックアップデータをリストアします。

この例では、B/INBOX/1 および C/INBOX/1 には新しい inode 番号が割り当てられ、メッセージデータはディスク上の新しい場所に書き込まれます。メッセージは 1 件だけリストアされます。2 件目のメッセージは最初のメッセージへのハードリンクです。

例 2 : この例では、システムはフルバックアップと部分リストアを以下のように実行します。

1. フルバックアップを実行します。
2. ユーザ A を削除します。
3. ユーザ A をリストアします。

A/INBOX/1 には新しい inode 番号が割り当てられます。

例 3 : この例では、複数回の部分リストアが必要となる可能性があります。

1. フルバックアップを実行します。

   B/INBOX/1 と C/INBOX/1 は A/INBOX/1 へのリンクとしてバックアップされます。
2. ユーザ A と B を削除します。
3. ユーザ B をリストアします。

   リストアユーティリティが、最初に A/INBOX をリストアするよう管理者に要求します。
4. ユーザ A と B をリストアします。

5. ユーザ A を削除します (省略可能)。

注	すべてのメッセージを部分リストアでリストアできるようにするためには、-i オプションを付けて imsbackup コマンドを実行します。-i オプションは必要に応じて各メッセージを複数回バックアップします。このオプションは POP 環境においてもっとも有効です。

Legato Networker を使用するには

Messaging Server は、Legato Networker のようなサードパーティ製のバックアップツールへのインタフェースを提供する、バックアップ API を装備しています。物理的なメッセージストア構造とデータ形式は、バックアップ API の中にカプセル化されています。バックアップ API はメッセージストアと直接対話します。さらに、バックアップサービスに対してメッセージストアの論理ビューを提示します。バックアップサービスは、メッセージストアの概念表現を使用して、バックアップオブジェクトの保存や検索を行います。

Messaging Server は Application Specific Module (ASM) を提供しています。これは、Legato Networker の save および recover コマンドによって呼び出され、メッセージストアのデータのバックアップとリストアを行います。さらに ASM は、Messaging Server の imsbackup および imsrestore ユーティリティを呼び出します。

注	この節では、Messaging Server のメッセージストアで Legato Networker を使用する方法についての情報を提供します。Legato Networker インタフェースについて理解するには、Legato のマニュアルを参照してください。

Legato Networker を使用したデータのバックアップ

Legato Networker を使用して Messaging Server メッセージストアのバックアップを行うには、Legato インタフェースを呼び出す前に以下の準備手順を実行する必要があります。

1. /usr/lib/nsr/imsasm から *server_root*/msg-*instance*/bin/imsasm へのシンボリックリンクを作成します。
2. Sun または Legato から nsrfile バイナリのコピーを取得して、それを以下のディレクトリにコピーします。

   /usr/lib/nsr/nsrfile
3. ユーザをグループ別にバックアップする必要がある場合は、以下の手順を実行します。

a. 378 ページの「バックアップグループを作成するには」の説明に従って、バックアップグループファイルを作成します。

b. 設定を確認するために、mkbackupdir.sh. を実行します。

server_root/backup 内のディレクトリ構造を確認します。このディレクトリ構造は図 11-2 に示されているようなものでなければなりません。

backup-groups.conf ファイルを指定していないと、バックアッププロセスはすべてのユーザに対して、デフォルトのバックアップグループ ALL を使用します。

4. ディレクトリ /nsr/res/ で、保存グループ用に res ファイルを作成して、バックアップの前に mkbackupdir.sh スクリプトを呼び出します。図 11-3 に示した例を参照してください。

注

Legato Networker では、保存設定の名前には最高 64 文字まで使用できます。デフォルトでは、mkbackupdir.sh は *server_root*/backup ディレクトリにストアイメージを作成します。このディレクトリ名とメールボックスの論理名を合わせたもの (たとえば siroe/groupA/fred) が 64 文字を超えた場合、mkbackupdir.sh -p を実行する必要があります。このため、mkbackupdir.sh の -p オプションの短いパス名を使用する必要があります。たとえば、次のコマンドでは /backup ディレクトリの下にバックアップイメージが作成されます。

mkbackupdir.sh -p /backup

重要 : バックアップディレクトリは、メッセージストアの所有者による書き込みが可能でなければなりません (例 : mailsrv)。

図 11-2 には、バックアップグループのディレクトリ構造のサンプルが示されています。

図 11-2 バックアップグループのディレクトリ構造

```
siroe-groupA-a1
             -a2
     -groupB-b1
             -b2
     -groupC-c1
             -c2
```

図 11-3 に、res ファイルのサンプルとして、/nsr/res ディレクトリにある IMS.res という名前のファイルを示します。

図 11-3　サンプルの res ファイル

```
type: savepnpc
precmd: "echo mkbackupdir started",
   "/usr/siroe/server5/msg-siroe/bin/mkbackupdir.sh -p /backup"
pstcmd: "echo imsbackup Completed";
timeout: "12:00 pm";
```

ここまでの準備が完了したら、以下の手順に従って Legato Networker インタフェースを実行します。

1. 必要に応じて Messaging Server 保存グループを作成します。
   a. `nwadmin` を実行します。
   b. Customize | Group | Create の順に選択します。
2. バックアップコマンドとして `savepnpc` を使用して、バックアップクライアントを作成します。
   a. `mkbackupdir` によって作成されるディレクトリに対して保存設定を行います。

      単一セッションのバックアップには、*server_root*`/backup` を使用します。

      同時バックアップには、*server_root*`/backup/`*server*`/`*group* を使用します。

      378 ページの「バックアップグループを作成するには」の定義に従って *group* があらかじめ作成されていることを確認します。

      また、同時実行するバックアップセッションの数も設定する必要があります。

      385 ページの「例 : Networker でバックアップクライアントを作成する」を参照してください。
3. Group Control | Start の順に選択して、バックアップ設定のテストを行います。

例 : Networker でバックアップクライアントを作成する : Networker でバックアップクライアントを作成するには、nwadmin から、Client | Client Setup | Create の順に選択します。

```
Name: siroe
Group: IMS
Savesets:/backup/siroe/groupA
   /backup/siroe/groupB
   /backup/gotmail/groupC
          .
          .
Backup Command: savepnpc
Parallelism: 4
```

Legato Networker を使用したデータのリストア

データの回復は、Legato Networker の nwrecover インタフェースまたは recover コマンドラインユーティリティを使用して実行できます。以下の例では、ユーザ a1 の INBOX を回復しています。

```
recover -a -f -s siroe /backup/siroe/groupA/a1/INBOX
```

次の例では、メッセージストア全体を回復しています。

```
recover -a -f -s siroe /backup/siroe
```

サードパーティのバックアップソフトウェア (Legato 以外) を使用するには

iPlanet Messaging Server では、コマンドライン imsbackup と Solstice Backup (Legato Networker) の 2 つのメッセージストアバックアップソリューションを提供しています。メッセージストア全体をバックアップするために imbackup を単体で実行する大規模なメッセージストアの場合、非常に長い時間がかかってしまう可能性があります。Legato ソリューションでは、複数のバックアップデバイスでのバックアップセッションの同時実行をサポートしています。バックアップを同時実行することにより、バックアップ時間を大幅に短縮できます (毎時 25G バイトのデータバックアップが達成できます)。

その他のサードパーティのバックアップソフトウェア (Netbackup など) を使用する場合は、以下の方法によってバックアップソフトウェアを iPlanet Messaging Server に統合します。

1. ユーザをグループに分割し (378 ページの「バックアップグループを作成するには」を参照)、*server_root*/msg-<instance>/config/ ディレクトリの下に backup-groups.conf ファイルを作成します。

   たとえば、ユーザを UID によってグループ化するには、/usr/iplanet/server5/msg-siroe/config/backup-groups.conf で次の定義を使用します。

   ```
   groupA=a*
   groupB=b*
   groupC=c*
   . . .
   ```

   注 このバックアップソリューションは追加のディスク容量を必要とします。すべてのグループを同時にバックアップするには、メッセージストアの 2 倍のサイズのディスク容量が必要になります。ディスク容量に余裕のない場合は、ユーザを小規模なグループに分け、グループセット単位でバックアップしていきます たとえば、group1 ～ group5、group6 ～ group10 というようになります。バックアップ後、グループデータファイルを削除します。

2. imsbackup を実行して、準備領域にあるファイルに各グループをバックアップします。

   このためのコマンドは、imsbackup -f <device> /<instance>/<group> です。

   複数の imsbackup プロセスを同時に実行することができます。たとえば、以下のようになります。

   ```
   # imsbackup -f- /siroe/groupA > /bkdata/groupA &
   # imsbackup -f- /siroe/groupB > /bkdata/groupB &
   ...
   ```

   imsbackup は大きなサイズのファイルをサポートしていないため、バックアップデータが 2G バイトを超える場合は -f- オプションを使用して、データを stdout に書き込み、ファイルへ出力を受け渡します。

3. サードパーティ製のバックアップソフトウェアを使用して、準備領域 (上の例では /bkdata) のグループデータファイルをバックアップします。

4. ユーザをリストアするには、ユーザのグループファイル名を確認し、そのファイルをテープからリストアし、imsrestore を使用してデータファイルからユーザをリストアします。

   imsrestore は大きなサイズのファイルをサポートしていません。データファイルが 2G バイトを超える場合は、次のコマンドを使用します。

   ```
   # cat /bkdata/groupA | imsrestore -f- /siroe/groupA/andy
   ```

メッセージストアをトラブルシューティングする

この節では、障害に備えてメッセージストアを保守する際のガイドラインについて説明します。また、メッセージストアが壊れたり、予期せずシャットダウンされた場合に使用する、その他のメッセージストアの回復手順についても説明します。メッセージストア回復の追加手順に関する節は、371 ページの「メールボックスとメールボックスデータベースの修復」の続きになります。

この節を読む前に、この章と、『iPlanet Messaging Server リファレンスマニュアル』のコマンドラインユーティリティおよび `configutil` の章を再度お読みになることを強くお勧めします。この節では、以下の項目について説明します。

- 387 ページの「標準的なメッセージストアのモニタ手順」
- 389 ページの「一般的な問題と解決策」
- 392 ページの「メッセージストアの回復手順」

標準的なメッセージストアのモニタ手順

ここでは、メッセージストアのモニタの標準的な手順の概要を説明します。ここで説明する手順は、全般的な状況のチェック、テスト、および標準的な保守を行う場合に役立つものです。

その他の情報については、512 ページの「メッセージストアをモニタする」を参照してください。

ハードウェアの容量のチェック

メッセージストアには、十分な追加のディスク容量とハードウェアリソースが必要です。メッセージストアがディスク容量とハードウェア容量の上限に近づくと、メッセージストアに問題が発生することがあります。

ディスク容量の不足は、メールサーバの問題と障害を発生させる、もっとも一般的な原因です。メッセージストアに書き込む領域がないと、メールサーバでエラーが発生します。さらに、利用可能なディスク容量が一定のしきい値より少なくなると、メッセージ配信やログ記録などに関連する多数の問題が発生します。`stored` プロセスのクリーンアップ機能が失敗し、削除されたメッセージがメッセージストアから消去されていないと、ディスク容量が急激に不足することがあります。

ディスク容量のモニタの詳細は、369 ページの「ディスク容量をモニタするには」および 512 ページの「メッセージストアをモニタする」を参照してください。

ログファイルのチェック

ログファイルをチェックして、メッセージストアプロセスが設定どおりに実行されていることを確認します。Messaging Server は、サポートしている主なプロトコルまたはサービス (SMTP、IMAP、POP、および HTTP) ごとに一連のログファイルを作成します。 ログファイルは、Console を使って確認するか、*server-root*/msg-*instance*/log/ ディレクトリ内で確認します。ログファイルは定期的にモニタする必要があります。

ログ記録はサーバパフォーマンスに影響することがあります。より詳細なログ記録を指定するほど、一定期間にログファイルが多くのディスク容量を占有することになります。効果的に定義する必要がありますが、現実的なログローテーション、有効期間、サーバのバックアップポリシーなどを考慮する必要があります。サーバのログポリシーの定義の詳細は、第 13 章「ログ記録とログ解析」を参照してください。

stored プロセスのチェック

stored 機能は、存続期間決定ポリシーを実行したり、ディスクに保存されているメッセージを消去して、メッセージデータベースのデッドロック操作やトランザクション操作などの、さまざまな重要なタスクを実行します。stored が実行を停止すると、最終的には Messaging Server に問題が発生します。start-msg が実行されているときに stored が起動していないと、ほかのプロセスも起動しません。

- stored プロセスが実行中かどうかをチェックします。pid ファイルは、stored (*server-root*/msg-*instance*/config/store.pid) によって作成および更新されます。
- stored プロセスによって以下の機能のいずれかが試行されたとき、常に以下のファイル (*server-root*/msg-*instance*/config/ ディレクトリ内) のタイムスタンプが更新されることを確認します。

表 11-8　stored 操作

stored 操作	機能
stored.ckp	データベースのチェックポイントが開始されたときに押される。約 1 分ごとにスタンプが付けられる
stored.lcu	データベースログのクリーンアップごとに押される。約 5 分ごとにタイムスタンプが付けられる
stored.per	ユーザ単位のデータベース書き込み時に押される。タイムスタンプは 1 時間ごとに付けられる

- *server-root*/msg-*instance*/store/mailboxlist 内に作成されたログファイルをチェックします。
- デフォルトのログファイル *server-root*/msg-*instance*/log/default/default 内の stored メッセージをチェックします。

stored プロセスの詳細は、370 ページの「stored ユーティリティを使用する」および、『iPlanet Messaging Server リファレンスマニュアル』の Messaging Server コマンドラインユーティリティの章の stored ユーティリティを参照してください。

stored 機能のモニタの詳細は、512 ページの「メッセージストアをモニタする」を参照してください。

データベースログファイルをチェックする

データベースログファイルは、sleepycat トランザクションのチェックポイントログファイル (*server-root*/msg-*instance*/store/mboxlist ディレクトリ内) を指します。ログファイルが蓄積されると、データベースのチェックポイント設定は行われません。通常は、単一の期間内に、2 つまたは 3 つのデータベースログファイルがあります。ログファイルがそれ以上ある場合は、問題がある可能性があります。

ユーザフォルダのチェック

ユーザフォルダをチェックする場合は、以下のコマンドを実行します。reconstruct -r -n (recursive nofix)。これにより、ユーザフォルダおよびレポートのエラーを確認します。reconstruct コマンドの詳細は、371 ページの「メールボックスとメールボックスデータベースの修復」を参照してください。

コアファイルのチェック

コアファイルは、プロセスが予期せず終了したときにのみ存在します。コアファイルを確認することは、メッセージストアに問題がありそうなときは特に重要です。

一般的な問題と解決策

この節では、以下のようなメッセージストアの一般的な問題と解決策の一覧を示します。

- 389 ページの「ユーザメールボックスディレクトリに関する問題」
- 391 ページの「ストアの全体に関する問題」

ユーザメールボックスディレクトリに関する問題

ユーザメールボックスに関する問題が発生するのは、メッセージストアの損傷が少数のユーザに限られていて、システム全体に対する損傷がないときです。ユーザメールボックスのディレクトリに関する問題を識別、分析、および解決する際は、以下のガイドラインを参考にしてください。

1. ログファイル、エラーメッセージ、またはユーザが見た異常な動作を確認します。

2. デバッグ情報と履歴を保存しておくには、*server-root*/msg-*instance*/store/mboxlist/ ユーザディレクトリ全体を、メッセージストアの外側の別の場所にコピーします。
3. 問題が発生している可能性のあるユーザフォルダを見つけるには、reconstruct -r -n コマンドを実行する必要があります。reconstruct を使用しても問題のあるフォルダが見つからない場合は、該当のフォルダが folder.db 内にない可能性があります。

   reconstruct -r -n コマンドを使用してもフォルダが見つからない場合は、hashdir コマンドを使用して場所を確認します。hashdir の詳細は、367 ページの「hashdir ユーティリティ」および、『iPlanet Messaging Server リファレンスマニュアル』の Messaging Server コマンドラインユーティリティの章の hashdir ユーティリティを参照してください。
4. ファイルが見つかったら、ファイルを調べ、権限をチェックし、適切なファイルのサイズを確認します。
5. reconstruct -r (-n オプションは付けない) を使用して、メールボックスを再構築します。
6. reconstruct で問題が検出されない場合は、reconstruct -r -f コマンドを使用して、メールフォルダを強制的に再構築することができます。
7. フォルダが mboxlist ディレクトリ (*server-root*/msg-*instance*/store/mboxlist) 内にはなく、partition ディレクトリ (*server-root*/msg-*instance*/store/partition) にある場合は、全体的な矛盾がある可能性があります。この場合は、reconstruct -m コマンドを実行する必要があります。
8. 上記の手順が機能しない場合は、store.idx ファイルを削除してから、再度 reconstruct コマンドを実行してください。

警告 問題のあるファイルが reconstruct では見つからないファイルであることがわかっている場合は、store.idx ファイルだけを削除してください。

9. 原因が問題を起こすメッセージに限られている場合は、メッセージファイルをメッセージストアの外側の別の場所にコピーしてから、mailbox/ ディレクトリ上で reconstruct -r コマンドを実行する必要があります。
10. フォルダがディスク (*server-root*/msg-*instance*/store/mboxlist/partition/ ディレクトリ) 上にあっても、明らかにデータベース (*server-root*/msg-*instance*/store/mboxlist/ ディレクトリ) 内にはないことがわかった場合は、reconstruct -m コマンドを実行してメッセージストアの整合性をチェックします。

reconstruct コマンドの詳細は、371 ページの「メールボックスとメールボックスデータベースの修復」を参照してください。

ストアの全体に関する問題

メッセージストアの障害がすべてのユーザに影響する問題やシステムの全体的な損傷の結果によるものであるとわかった場合は、以下のガイドラインに従ってシステムを回復することができます。

1. メッセージストアプロセスを停止します。

   a. メッセージストアプロセスが停止したことを確認したら、メッセージストアプロセスを再起動します。

   b. stored プロセスを実行してデータベースを回復します。

   たいていの場合、データベースの障害は自動的に回復されます。この処理は、stored が起動すると、それによって、キャッシュファイルとデータベースファイルに対してデータベースのログファイルを解析するデータベース回復が開始されるために発生します。データベース回復は、データベースを整合性のある状態に置こうとします。

2. stored プロセスコマンドでメッセージストアを起動しようとしているときに、msg-start コマンドが突然終了した場合は、stored が失敗したか、ストアを回復しようとしているかのいずれかです。

   stored がメッセージストアを起動しようとしているときにこのプロセスが異常終了した場合は、データベースをリストアするために stored プロセスが大きなログファイルを確認している可能性があります。

   a. *server-root*/msg-*instance*/log/default/ ディレクトリをチェックして、stored が解析していた情報を確認します。

   b. また、設定ファイルと pidfile.store ファイルを確認することもできます。

      pidfile.store ファイルは、pid とともに stored プロセスの状態を示します。pidfile は、回復中は init 状態を示し、stored プロセスがデータベースの修復を終えたときは ready 状態を示します。

3. pidfile が ready 状態を示すと、データベースはすでに回復済みで、メッセージストアの残りの部分を再起動することができます。

   a. store プロセスを起動し、reconstruct -m コマンドを実行します。reconstruct の詳細は、371 ページの「メールボックスとメールボックスデータベースの修復」を参照してください。

   b. テストアカウントをモニタし、ログファイルを確認して、ユーザメールボックスディレクトリが有効かどうかを確認します。

      個々のユーザメールボックスが損傷している場合は、reconstruct -r コマンドを実行します。

   c. メッセージストアの損傷の度合いが大きい場合は、停止しているメッセージストアプロセスを使って修復する必要があるかもしれません。392 ページの「メッセージストアの回復手順」を参照してください。

4. pidfile が ready 状態に変わらない場合、stored プロセスは、mboxlist ログファイルを確認中であるか、データベースを回復できないかのいずれかです。

   a. *server-root*/msg-*instance*/store/mboxlist ディレクトリ内に多数のデータベースログファイルがある場合は、stored プロセスが init 状態を終了できない可能性があります。さらに、データベースの回復に時間がかかっている可能性があります。ログファイルが 20 ～ 30 個あると、大半のマシンでは処理に非常に時間がかかります。このような場合は、stored プロセスを停止し、*server-root*/msg-*instance*/store/mboxlist ディレクトリ内のファイルを削除してから、スナップショットまたは高速回復プロセスを開始する必要があります。

   b. stored プロセスでメッセージストアを回復できない場合は、ほぼ間違いなくデータベースが壊れています。その場合は、データベースのスナップショットコピーをリストアするか、高速回復を実行する必要があります。詳細は、392 ページの「メッセージストアの回復手順」を参照してください。

警告 プロセスがデータベースにアクセスしているときは、絶対にプロセスを終了しないでください。init 状態のときに stored プロセスを終了すると、データベースを既存の mboxlist データから回復することができなくなります。その結果、データは削除されてしまいます。データベースにアクセスしている別のプロセスを終了すると、データベースは矛盾のある状態のままになり、メッセージストア全体をシャットダウンしてから再起動することが必要になります。

メッセージストアの回復手順

この節では、メッセージストアを再構築または修復するための回復手順について説明します。

- **高速回復を実行するには** : 標準的な修復ができないほどデータベースが壊れているときは、高速回復を使用します (標準的なメールボックスの修復の詳細は、371 ページの「メールボックスとメールボックスデータベースの修復」を参照)。さらに、高速回復では、メッセージストアを即座に表示することができます。標準的なメッセージストア回復手順 (371 ページの「メールボックスとメールボックスデータベースの修復」を参照) と同様、高速回復でも reconstruct コマンドを使用する必要があります。
- **データベーススナップショットのバックアップを作成するには と データベーススナップショットを使用してメッセージストアを回復するには** : データベースが損傷した場合は、前のバージョンのデータベースを使用できるため、ユーザフォルダの大部分は即座にリストアすることができます。リストアを実行したら、reconstruct コマンドとともに高速回復手順を使用して、データベースの置換と再構築を行うことができます。

高速回復を実行するには

データベースに矛盾があるときは、標準の回復時に reconstruct ユーティリティを使用します (371 ページの「メールボックスとメールボックスデータベースの修復」を参照)。

データベースが標準的な方法では修復できないほど壊れている場合は、以下の手順に従って、高速回復のために reconstruct ユーティリティを使用することができます。

1. メッセージストアプロセスを停止します。
2. すべてのストアプロセスが停止していることを確認します。
3. *server-root*/msg-*instance*/store/mboxlist/* ファイルを、あとで確認するために安全な場所にコピーします。
4. *server-root*/msg-*instance*/store/mboxlist/ ディレクトリ内のすべてのファイルを削除します。
5. stored、imapd、popd、mshttpd のようなメッセージストアプロセスを起動します。
6. reconstruct -m ユーティリティを実行して、folder.db を再構築します。

データベーススナップショットのバックアップを作成するには

メールボックスデータベースとログファイルのバックアップ (スナップショットともいう) を作成することによって、メッセージストアの破損に備えることができます。データベースが破損してしまった場合は、スナップショットを使用してデータベースを置き換えれば、データベースを再構築しなくて済みます。スナップショット機能は、データベースの整合性のあるコピーを作成し、それによってデータベースを回復することができます。これらのバックアップを保存しておくための十分なディスク容量があることを確認してください。

注　特に指定されていないかぎり、iPlanet Messaging Server 5.2 で使われるデータベーススナップショットパラメータは 表 11-9 に示されているものだけです。

表 11-9 に、データベーススナップショットの作成に使われる 3 つの configutil パラメータを示します。これらのデータベーススナップショットは、回復時に stored プロセスによって呼び出されます。

表 11-9　configutil のデータベーススナップショットパラメータ

データベーススナップショットパラメータ	説明
local.store.snapshotpath	mboxlist ディレクトリのコピー先のパスを指定します。メッセージストア所有者に権限が設定されます。スナップショットはサブディレクトリに置かれます。

表 11-9　configutil のデータベーススナップショットパラメータ (続き)

データベーススナップショットパラメータ	説明
local.store.snapshotinterval	スナップショットの時間の間隔。時間は分単位。この手順は少なくとも 1 日 1 回実行することをお勧めします。
local.store.snapshotdirs	ディスク上に保存する個々のスナップショットの数。最低は 2 個で、デフォルトは 3。現在のもので修復できないことが判明する前に、データベースの良好なバックアップを作成しておくことをお勧めします。

データベースのバックアップを作成するには、configutil コマンドを使用して次のパラメータの値を指定します。

```
configutil -o local.store.snapshotinterval -v number
```

ここで、*number* には stored がデータベースをバックアップする頻度を指定します。この *number* は、分間隔を示しています。

```
configutil -o local.store.snapshotpath -v path
```

この *path* は、バックアップコピーの置かれる場所を示しています。

警告　初期のリリースの Messaging Server のデータベーススナップショットユーティリティは、現在のユーティリティと同じ方法では機能しません。このため、旧バージョンの Messaging Server のスナップショットユーティリティを Messaging Server 5.2 で使用することはお勧めできません。

データベーススナップショットを使用してメッセージストアを回復するには

データベーススナップショットを使用してデータベースを回復するためには、メッセージストアのレイアウトを熟知していることが必須です。詳細は、345 ページの「メッセージストアのディレクトリレイアウト」を参照してください。

データベーススナップショットが作成されると (393 ページの「データベーススナップショットのバックアップを作成するには」を参照)、それらは src サブディレクトリに保存されます。これらのファイルは、最終的には、回復済みのデータベースが保存されている dst *server-root*/msg-*instance*/store/mboxlist/ ディレクトリに移されます。

スナップショットファイルに加え、スナップショットの作成中に作成される制御ファイルがあります。表 11-10 に、データベーススナップショットの制御ファイルを示します。これらのファイルはメッセージストア所有者が所有することに注意してください。

表 11-10 データベーススナップショット制御ファイル

制御ファイル	説明
dst/.nosnap	設定データが再読み込みされない場合でも、データベーススナップショットプロセスを無効にします。
dst/.snaprst	以前のすべてのスナップショットを無効としてマークします。このファイルは最初の新しいスナップショットのあとで削除されます。
dst/.catrecov	stored プロセスをトリガして重大な回復処理を開始し、スナップショットを使用可能な形式にリストアします。
src/.snaptime	有効なスナップショットがディレクトリ内にあることを示します。このファイルのタイムスタンプは、スナップショットが完了した時間を示します。

以下の手順で、データベーススナップショット、制御ファイル、src/、および dst/ ディレクトリを使用して手動回復を実行する方法を説明します。

1. 回復を実行する前に、自分がメッセージストアの所有者であることを確認します。
2. メッセージストアプロセスを停止し、すべてのプロセスが停止していることを確認します。
3. *server-root*/msg-*instance*/store/mboxlist/ ディレクトリ内のファイルを、あとで確認するために安全な場所にコピーします。
4. スナップショットがある場合は、メッセージストアと置き換えることができるかどうかを確認します。詳細は、393 ページの「データベーススナップショットのバックアップを作成するには」を参照してください。
   a. *.snaptime ファイルを使用して、バックアップの有効性と時間を確認します。スナップショットにある該当のログファイルが多すぎる場合は、別のスナップショットを確認します。
   b. データベースに関する問題が取り込まれていない最新の有効なスナップショットを選びます。

      利用できるスナップショットがない場合は、高速回復手順に従ってください。詳細は、393 ページの「高速回復を実行するには」を参照してください。
5. *server-root*/msg-*instance*/store/mboxlist/ ディレクトリ内のファイルは壊れているため、これらのファイルをすべて削除します。

6. 該当のスナップショットファイルを、選択したスナップショットから *server-root*/msg-*instance*/store/mboxlist/ ディレクトリにコピーします。ただし、*.snaptime ファイルはコピーしないでください。
7. touch コマンドを使用して、*server-root*/msg-*instance*/store/mboxlist/ ディレクトリに .catrecov ファイルを作成します。

   .catrecov ファイルは、重大な回復処理を実行する必要があるメッセージストアを示すものです。
8. メッセージストアプロセスを起動します。
9. stored プロセスをモニタします。stored プロセスが回復を行います。
10. stored プロセスが回復を行ったあとで *server-root*/msg-*instance*/store/mboxlist/.catrecov ファイルが削除されていることを確認します。削除されていないと、メッセージストアは、起動したとき常に重大な回復処理が必要であるとみなしてしまいます。
11. reconstruct -m を実行して、snaptime ファイルと破損したデータベースの間の相違点を修復します。

第 12 章

セキュリティとアクセス制御を設定する

iPlanet Messaging Server は、広範囲にわたる柔軟なセキュリティ機能をサポートします。これらの機能を使用して、メッセージが横取りされないようにしたり、侵入者がユーザや管理者になりすますことを防いだり、メッセージングシステム内の特定部分へのアクセスを特定のユーザだけに許可したりできます。

Messaging Server のセキュリティアーキテクチャは、iPlanet サーバのセキュリティアーキテクチャ全体の一部です。このアーキテクチャは、最大の相互運用性と一貫性を実現するために業界標準と公開プロトコルに基づいて構築されています。そのため、Messaging Server のセキュリティポリシーを実装するには、この章だけでなく他のドキュメントも参照する必要があります。特に、『iPlanet Managing Servers with Netscape Console』には、Messaging Server のセキュリティを設定するために必要な情報が記載されています。

この章には、以下の節があります。

サーバのセキュリティについて

サーバのセキュリティには広範囲にわたる説明が含まれます。ほとんどの企業では、承認されたユーザだけがサーバにアクセスできること、パスワードや識別情報が安全なこと、他のユーザになりすました通信ができないこと、必要に応じて通信の機密性を確保できることなどがすべてメッセージングシステムの重要な要件になります。

サーバのセキュリティはさまざまな方法で攻撃される可能性があるため、さまざまな方法でセキュリティを強化します。この章では、暗号化、認証、およびアクセス制御の設定を中心に、次の Messaging Server のセキュリティに関する項目について説明します。

- **ユーザ ID とパスワードログイン**: ユーザは、IMAP、POP、HTTP、または SMTP にログインするためにユーザ ID とパスワードを入力する必要があります。また、メッセージの受取人に差出人の認証情報を送信するには、SMTP パスワードログインを使用する必要があります。
- **暗号化と認証**: TLS プロトコルおよび SSL プロトコルを使用して通信を暗号化し、クライアントを認証するようにサーバを設定します。
- **管理者のアクセス制御**: Netscape Console のアクセス制御機能を使って、Messaging Server へのアクセス権や個別のタスクを委任します。
- **TCP クライアントアクセス制御**: フィルタリング技術を使用して、サーバの POP、IMAP、HTTP、および 認証済み SMTP サービスに接続できるクライアントを制御します。

この章では、Messaging Server に関連するすべてのセキュリティとアクセスの問題について説明するわけではありません。この章で説明していないセキュリティ関連の項目として、以下のものがあります。

- **物理的なセキュリティ**: サーバマシンを物理的に保護しないと、ソフトウェアのセキュリティは意味を持たない場合があります。
- **メッセージの暗号化 (S/MIME)**: S/MIME (Secure Multipurpose Internet Mail Extensions) を使用すると、差出人はメッセージを暗号化して送信し、受取人は暗号化されたメッセージを受け取って保存し、メッセージを読む場合にのみ解読することができるようになります。S/MIME は、Messaging Server に関する特別な設定や作業をしなくても利用できます。S/MIME は、クライアントでだけ使用できます。S/MIME の設定方法については、使用するクライアントのマニュアルを参照してください。Messenger Express クライアントインタフェースは、電子メールメッセージの暗号化をサポートしません。
- **メッセージストアへのアクセス**: Messaging Server に対して、複数のメッセージストア管理者を定義できます。これらの管理者は、メールボックスの表示と監視を行ったり、メールボックスへのアクセスを制御したりできます。詳細は、第 11 章「メッセージストアを管理する」を参照してください。

- **エンドユーザアカウントの設定**: エンドユーザアカウント情報は、主に Delegated Administrator 製品を使って管理します。詳細は、Delegated Administrator のマニュアルを参照してください。また、コンソールのインタフェースを使ってエンドユーザアカウントを管理することもできます。詳細は、付録 D「メールユーザとメーリングリストを管理する」を参照してください。
- **不特定多数宛てメール (UBE) のフィルタリング**: 第 10 章「メールのフィルタリングとアクセス制御」を参照してください。

iPlanet では、セキュリティに関するさまざまな説明を含んだ数多くのマニュアルを作成しています。この章に記載した内容の背景情報や、その他のセキュリティ関連情報については、iPlanet の Web サイト (http://docs.iplanet.com) を参照してください。

HTTP のセキュリティについて

Messaging Server は、ユーザ ID とパスワードによる認証とクライアント証明書による認証の両方をサポートしています。ただし、クライアントとサーバ間におけるネットワーク接続の処理方法は、この 2 つのプロトコルでいくつか異なります。

POP、IMAP、または SMTP クライアントが Messaging Server にログインすると、接続が確立され、セッションが開始されます。この接続は、セッションの間中、すなわちログインからログアウトまで維持されます。新しい接続を確立する場合は、クライアントが再びサーバで認証される必要があります。

HTTP クライアントが Messaging Server にログインする場合は、サーバからクライアントに固有のセッション ID が与えられます。クライアントは、このセッション ID を使って、セッション中に複数の接続を確立できます。HTTP クライアントは接続するたびに再認証を行う必要はありません。ただし、セッションが切断された場合やクライアントが新しいセッションを確率する必要がある場合だけは、クライアントが、再び認証を行う必要があります。指定した時間にわたり HTTP セッションのアイドル状態が続くと、サーバは自動的に HTTP セッションを切断し、クライアントがログアウトされます。デフォルトの時間は 2 時間です。

HTTP セッションのセキュリティを向上させるには、以下の方法を使用します。

- セッション ID は、特定の IP アドレスにバインドされる
- 各セッション ID には、タイムアウト値が関連付けられているので、指定時間にわたりセッション ID が使用されないと、そのセッション ID は無効になる
- 使用中のすべてのセッション ID のデータベースをサーバが管理するため、クライアントは ID を偽造できない
- セッション ID は、cookie ファイルではなく URL 内に保管される

設定パラメータを指定して接続のパフォーマンスを向上させる方法については、第 3 章「POP、IMAP、および HTTP サービスの設定」を参照してください。

認証メカニズムを構成する

認証メカニズムは、クライアントが識別情報をサーバに提示する方法の 1 つです。Messaging Server は SASL (Simple Authentication and Security Layer) プロトコルで定義されている認証方法をサポートし、さらに、証明書に基づく認証もサポートします。この節では、SASL による認証メカニズムについて説明します。証明書に基づく認証の詳細は、404 ページの「暗号化と証明書に基づく認証を構成する」を参照してください。

Messaging Server は、パスワードに基づく認証の場合、以下の SASL 認証方法をサポートします。

- PLAIN - このメカニズムは、ユーザのプレーンテキストパスワードをネットワーク経由で渡すので、パスワードが盗まれる可能性があります。

  この問題は、SSL を使用することによって軽減できます。詳細は、404 ページの「暗号化と証明書に基づく認証を構成する」を参照してください。

- DIGEST-MD5 - RFC 2831 で定義されているチャレンジ / レスポンス型の認証メカニズム。Messaging Multiplexor では、DIGEST-MD5 はサポートされていません。
- CRAM-MD5 - APOP に似たチャレンジ / レスポンス型の認証メカニズムですが、他のプロトコルでの使用にも適しています。RFC 2195 に定義されています。
- APOP - POP3 プロトコルでのみ使用できるチャレンジ / レスポンス型の認証メカニズム。RFC 1939 に定義されています。

チャレンジ / レスポンス型の認証メカニズムでは、サーバからクライアントにチャレンジ文字列が送られます。クライアントは、そのチャレンジのハッシュとユーザのパスワードを使用して応答します。クライアントの応答がサーバ自体のハッシュと一致すると、ユーザが認証されます。ハッシュは元のデータに戻すことができないため、ネットワークを介して送信してもユーザのパスワードが危険にさらされることはありません。

注 POP、IMAP、および SMTP サービスは、すべての SASL メカニズムをサポートします。HTTP サービスは、プレーンテキストパスワードによるメカニズムだけをサポートします。

プレーンテキストパスワードへのアクセスを構成するには

CRAM-MD5、DIGEST-MD5、または APOP SASL の認証メソッドでは、ユーザのプレーンテキストパスワードにアクセスする必要があります。次の手順を実行する必要があります。

1. パスワードがクリアテキストで保存されるように Directory Server を構成します。
2. Directory Server がクリアテキストのパスワードを使用していることを認識できるように、Messaging Server を構成します。

パスワードが保存されるように Directory Server を構成するには

CRAM-MD5、DIGEST-MD5、または APOP メカニズムを有効にするには、次のようにパスワードがクリアテキストで保存されるように Directory Server を構成する必要があります。

1. コンソールで、構成する Directory Server を開きます。
2. 「環境設定」タブをクリックします。
3. 左のペインで「データベース」を開きます。
4. 右のペインで「パスワード」をクリックします。
5. 「パスワードの暗号化」ドロップダウンリストで「クリアテキスト」を選択します。

注	この変更は、将来作成するユーザにのみ影響を与えます。既存のユーザは、この変更を加えたあとで移行するか、パスワードを再設定する必要があります。

Messaging Server を構成するには

次に、Directory Server がクリアテキストのパスワードを使用していることを認識できるように Messaging Server を構成することができます。これにより、Messaging Server で APOP、CRAM-MD5、および DIGEST-MD5 を安全に使用できるようになります。

```
configutil -o sasl.default.ldap.has_plain_passwords -v 1
```

値を 0 または null (" ") に設定すると、これらのチャレンジ / レスポンス型の SASL メカニズムを無効にすることができます。

注	既存のユーザは、パスワードを再設定または移行するまで APOP、CRAM-MD5、または DIGESt-MD5 を使用できません (次の「ユーザの移行」を参照)。

ユーザを移行するには

configutil を使用して、移行するユーザに関する情報を指定できます。たとえば、ユーザパスワードを変更する場合や、適切なユーザエントリがないメカニズムを使ってクライアントが認証を試みている場合に、この情報を指定します。

```
configutil -o sasl.default.transition_criteria -v value
```

value には、次のいずれかを指定できます。

- CHANGE - ユーザパスワードを変更する場合は、サーバがプレーンテキストを受け入れるように移行します。デフォルトでは、このキーワードが使用されます。
- CLIENT - クライアントが適切なユーザエントリのないメカニズムを使用しようとしている場合、サーバはプレーンテキストのパスワードを使用して認証するようにクライアントに要求します。その後、サーバは、同じパスワード値を使用してそのメカニズム用の適切なエントリを作成します。
- PLAIN - クライアントがプレーンテキストパスワードを使用する場合に、サーバがプレーンテキストを受け入れるように移行します。

ユーザを正常に移行するには、Messaging Server がユーザパスワード属性に書き込みアクセスできるように、Directory Server の ACI を設定する必要があります。そのためには、次の手順を実行します。

1. コンソールで、構成する Directory Server を開きます。
2. 「ディレクトリ」タブをクリックします。
3. ユーザ / グループツリーのベース接尾辞を選択します。
4. 「オブジェクト」メニューから「アクセス権」を選択します。
5. 「Messaging Server エンドユーザ管理者書き込みアクセス権 (Messaging Server End User Administrator Write Access Rights)」に対する ACI を選択 (ダブルクリック) します。
6. 「ACI 属性」をクリックします。
7. 既存の属性のリストに userpassword 属性を追加します。
8. 「OK」をクリックします。

ユーザパスワードログイン

Messaging Server にログインしてメールの送受信を行うには、ユーザがパスワードを入力する必要があります。これは承認されていないアクセスを防ぐための最初の防御手段です。Messaging Server では、IMAP、POP、HTTP、および SMTP の各サービスに対して、パスワードに基づくログインがサポートされています。

IMAP、POP、HTTP のパスワードログイン

デフォルトでは、内部ユーザは、Messaging Server からメッセージを取得するためにパスワードを送信する必要があります。POP、IMAP、HTTP のサービスごとにパスワードログインを有効または無効にできます。POP、IMAP、HTTP サービスのパスワードログインの詳細は、57 ページの「パスワードに基づくログイン」を参照してください。

ユーザパスワードは、クリアテキストまたは暗号文 (POP を除く) の形式で、ユーザのクライアントソフトウェアからサーバに転送できます。クライアントとサーバの両方が、SSL を使用できるように構成され、かつ必要な強度の暗号化 (410 ページの「SSL を有効化し符号化方式を選択するには」を参照) をサポートする場合に、暗号化が実行されます。

ユーザ ID とパスワードは、LDAP ユーザディレクトリに保存されます。最小長などのパスワードに関するセキュリティ条件は、ディレクトリポリシーの要件によって決まり、Messaging Server では管理されません。

パスワードに基づくログインの代わりに証明書に基づくログインを使用できます。証明書に基づくログインについては、SSL の説明とともにこの章で後述します。413 ページの「証明書に基づくログインを設定するには」を参照してください。

プレーンテキストパスワードによるログインの代わりに、チャレンジ / レスポンス型の SASL メカニズムを使用できます。

SMTP パスワードログイン

デフォルトでは、Messaging Server の SMTP サービスに接続してメッセージを送信する場合に、ユーザはパスワードを入力する必要がありません。しかし、認証 SMTP を使用可能にするために、SMTP サービスへのパスワードログインを有効にすることができます。

認証 SMTP は、クライアントがサーバに対して認証を行うことを可能にする、SMTP プロトコルの拡張機能です。認証は、メッセージの送受信時に実行されます。認証 SMTP の主要な用途は、他のユーザが悪用できるオープンリレーの発生を防ぎながら、ローカルユーザが移動先から (または自宅用の ISP を使用して) メールを送信 (リレー) できるようにすることです。クライアントは、「AUTH」コマンドを使用してサーバに対する認証を行います。

SMTP パスワードログイン (すなわち認証 SMTP) を有効にする方法については、233 ページの「SMTP 認証、SASL、TLS」を参照してください。

認証 SMTP は、SSL 暗号化とともに使用することも、SSL 暗号化を使わずに使用することもできます。

暗号化と証明書に基づく認証を構成する

この節には、以下の項があります。

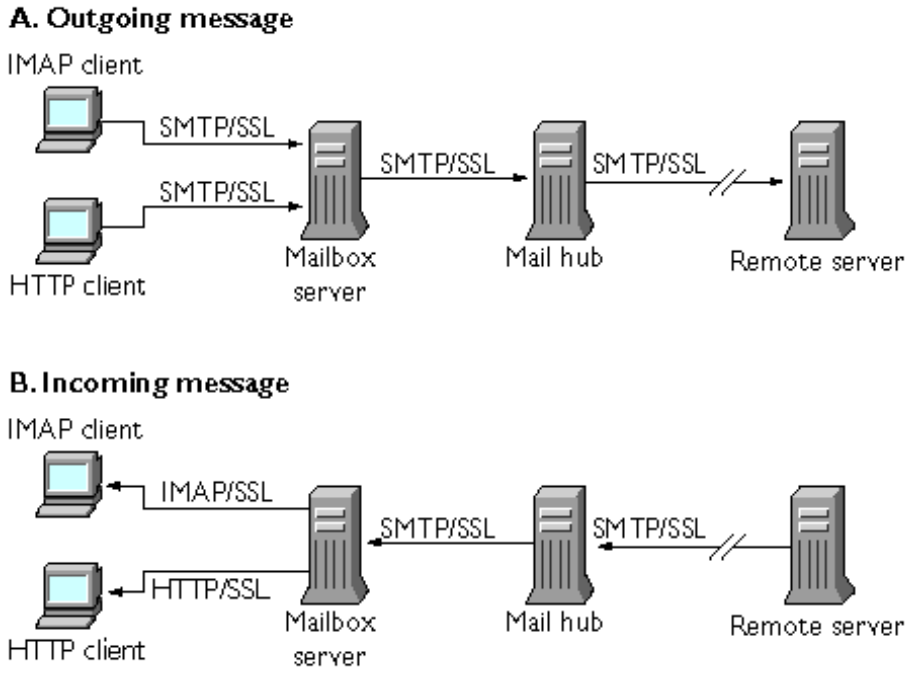
iPlanet Messaging Server では、クライアントとサーバ間で、暗号化された通信および証明書に基づく認証を行うために TLS (Transport Layer Security) プロトコルを使用します。TLS プロトコルは、SSL (Secure Sockets Layer) プロトコルとも呼ばれます。iPlanet Messaging Server は、SSL バージョン 3.0 および 3.1 をサポートしています。TLS には、SSL との完全な互換性があり、必要な SSL 機能がすべて含まれています。

SSL に関する背景情報については、『Managing Servers with Netscape Console』の付録の「Introduction to SSL」を参照してください。SSL は、公開鍵暗号化の概念に基づいています。この概念については、『Managing Servers with Netscape Console』の付録の「Introduction to Public-Key Cryptography」を参照してください。

Messaging Server とそのクライアント間、および Messaging Server と他のサーバ間におけるメッセージの転送が暗号化される場合は、通信が盗聴される危険性はほとんどありません。また、接続しているクライアントが認証済みの場合は、侵入者がそれらのクライアントになりすます (スプーフィングする) 危険性もほとんどありません。

SSL は、IMAP4、HTTP、および SMTP のアプリケーションレイヤの下のプロトコルレイヤとして機能します。SMTP と SMTP/SSL は同じポートを使用しますが、HTTP と HTTP/SSL は異なるポートを必要とします。IMAP と IMAP/SSL は、同じポートを使用することも異なるポートを使用することもできます。図 12-1 に示すように、SSL は、送信メッセージと受信メッセージの両方で、メッセージ通信の特定の段階で作用します。

図 12-1 Messaging Server での暗号化された通信



SSL は、ホップ間の暗号化を提供しますが、中間にある各サーバ上ではメッセージは暗号化されません。終端間の暗号化を実現するには、クライアントが S/MIME をサポートしている必要があります。

注 送信メッセージの暗号化を有効にするには、チャネル定義を変更して、`maytls` や `musttls` などの `tls` チャネルキーワードを追加する必要があります。詳細は、『iPlanet Messaging Server リファンレンスマニュアル』を参照してください。

SSL 接続を設定する際のオーバーヘッドによって、サーバのパフォーマンスが低下する可能性があります。メッセージングシステムの設計とパフォーマンスの分析を行う際には、セキュリティ要件とサーバのパフォーマンスのバランスをとる必要があります。

注	SSL はすべての iPlanet サーバでサポートされており、SSL の有効化と設定を行うために使用する Console インタフェースは多くのサーバでほとんど同じです。そのため、この章で説明するタスクのいくつかについては、『iPlanet Managing Servers with Netscape Console』に詳しい説明が記載されています。それらのタスクについては、この章では要約だけを説明します。

証明書の入手

SSL の用途が暗号化か認証かにかかわらず、Messaging Server 用のサーバ証明書を入手する必要があります。この証明書は、使用するサーバの識別情報をクライアントや他のサーバに提供します。

内部モジュールと外部モジュールを管理するには

サーバ証明書によって、キーのペアの所有権と有効性が確立されます。キーのペアは、データの暗号化と解読に使用される数値です。サーバの証明書とキーのペアは、そのサーバの識別情報を示します。これらは、サーバ内部または取り外し可能な外部のハードウェアカード (スマートカード) の証明書データベース内に保存されます。

iPlanet サーバは、PKCS (Public-Key Cryptography System) #11 API に準拠するモジュールを使用して、キーと証明書のデータベースにアクセスします。通常、特定のハードウェアデバイスの PKCS #11 モジュールは、そのデバイスの供給元から入手できます。Messaging Server でそのデバイスを使用する前に、このモジュールを Messaging Server にインストールする必要があります。Messaging Server にプリインストールされている「Netscape Internal PKCS # 11 Module」は、サーバ内部の証明書データベースを使用する単一の内部ソフトウェアトークンをサポートします。

証明書を使用できるようにサーバを設定する場合は、証明書とそのキーを格納するためのデータベースを作成し、PKCS #11 モジュールをインストールする必要があります。外部のハードウェアトークンを使用しない場合は、サーバ上に内部データベースを作成し、Messaging Server に含まれるデフォルトの内部モジュールを使用します。外部トークンを使用する場合は、スマートカードリーダーハードウェアを接続し、そのハードウェアの PKCS #11 モジュールをインストールします。

外部モジュールか内部モジュールかにかかわらず、PKCS #11 モジュールは、コンソールを使用して管理できます。PKCS #11 モジュールをインストールするには、次の手順を実行します。

1. カードリーダーハードウェアを Messaging Server ホストマシンに接続し、ドライバをインストールします。
2. コンソールの「PKCS #11 Management」インタフェースを使用して、インストールしたドライバ用の PKCS #11 モジュールをインストールします。

詳細な手順については、『iPlanet Managing Servers with Netscape Console』の SSL に関する章を参照してください。

ハードウェア暗号化アクセラレータのインストール: 暗号化用に SSL を使用する場合は、ハードウェア暗号化アクセラレータをインストールすることによって、メッセージの暗号化と解読のパフォーマンスを向上させることができます。一般的に、暗号化アクセラレータは、サーバマシンに常設されたハードウェアボードとソフトウェアドライバで構成されます。iPlanet Messaging Server は、PKCS #11 API に準拠したアクセラレータモジュールをサポートしています。これらは、基本的に独自のキーを格納しないハードウェアトークンで、キーの格納には内部データベースが使用されます。まず、製造元の指示に従ってハードウェアとドライバをインストールすることにより、アクセラレータをインストールします。その後、PKCS #11 モジュールをインストールすることにより、ハードウェア証明書トークンをインストールします。

サーバ証明書を要求するには

サーバ証明書を要求するには、iPlanet Console でサーバを開き、「証明書セットアップウィザード」を実行します。このウィザードには、「コンソール」メニューまたは Messaging Server の「暗号化」タブからアクセスできます。このウィザードを使用して、次のタスクを実行します。

1. 証明書要求を作成します。
2. 電子メールを使用して、証明書を発行する認証局 (CA) に要求を送信します。

認証局 (CA) から電子メールによる応答を受け取ったら、メールをテキストファイルとして保存し、証明書セットアップウィザードを使用してそのファイルをインストールします。

詳細な手順については、『iPlanet Managing Servers with Netscape Console』の SSL に関する章を参照してください。

証明書をインストールするには

インストールは、要求とは別の手順で実行します。認証局 (CA) から証明書要求に対する応答の電子メールを受け取ったら、電子メールをテキストファイルとして保存し、もう一度証明書セットアップウィザードを実行して、次のように証明書としてファイルをインストールします。

1. 入手済みの証明書をインストールすることを指定します。
2. 指示に従って、証明書のテキストをフィールド内に貼り付けます。

詳細な手順については、『iPlanet Managing Servers with Netscape Console』の SSL に関する章を参照してください。

注	CA の証明書 (以下に説明) をインストールする場合にも、この手順を実行する必要があります。サーバはこの証明書を使用して、クライアントによって提示された証明書の信頼性を判断します。

信頼できる CA の証明書をインストールするには

認証局 (CA) の証明書をインストールする場合も、証明書セットアップウィザードを使用します。CA 証明書は、認証局自体の身元を証明します。サーバは、クライアントや他のサーバを認証するプロセスで、これらの CA 証明書を使用します。

たとえば、パスワードに基づく認証 (157 ページの「証明書に基づくログインの設定」を参照) に加え、証明書に基づく認証にも対応するように会社の環境を設定した場合は、クライアントが提示する可能性のある証明書の発行元として信頼できる CA の証明書をすべてインストールする必要があります。これらの CA は、社内組織の場合もあれば、商業機関、政府機関、他の企業などの外部組織の場合もあります。認証用 CA 証明書の使用方法については、『iPlanet Managing Servers with Netscape Console』の「Introduction to Cryptography」を参照してください。

Messaging Server をインストールすると、いくつかの商用認証局の CA 証明書もインストールされます。他の商用認証局の CA 証明書を追加する場合や、社内使用のために (iPlanet Certificate Server を使用して) 独自の認証局を開発する場合は、追加の CA 証明書を入手して、インストールする必要があります。

注	Messaging Server により自動的に提供される CA 証明書は、インストール時にはクライアント証明書用の信頼できる証明書としてマークされていません。これらの CA から発行されるクライアント証明書を信頼できるものにする必要がある場合は、信頼設定を編集する必要があります。この手順については、409 ページの「証明書と信頼できる CA の管理」を参照してください。

新しい CA 証明書を要求してインストールするには、次の手順を実行します。

1. Web ページからまたは電子メールを利用して認証局に連絡し、その CA 証明書をダウンロードします。
2. 受け取った証明書のテキストをテキストファイルとして保存します。

3. 証明書セットアップウィザードを使用し、前の節で説明した手順に従って証明書をインストールします。

詳細な手順については、『iPlanet Managing Servers with Netscape Console』の SSL に関する章を参照してください。

証明書と信頼できる CA の管理

サーバには、クライアントの認証に使用する、信頼できる CA の証明書を必要な数だけインストールできます。

コンソールでサーバを開き、「コンソール」メニューの「証明書の管理」コマンドを選択すると、Messaging Server にインストールされている証明書の信頼設定の表示や編集、または任意の証明書の削除を行うことができます。この手順については、『iPlanet Managing Servers with Netscape Console』の SSL に関する章を参照してください。

パスワードファイルの作成

任意の iPlanet サーバ上で、証明書セットアップウィザードを使用して証明書を要求すると、ウィザードによってキーのペアが作成されます。このキーのペアは、あとで内部モジュールのデータベースまたはスマートカード内にある外部データベースに格納します。次に、このプライベートキーを暗号化するために使われるパスワードの入力を要求されます。あとでこのキーを解読するには、この同じパスワードを使用する必要があります。ウィザードでは、パスワードはどこにも記録されません。

SSL を有効にしている iPlanet サーバでは、ほとんどの場合、起動時に管理者がキーのペアの解読に必要なパスワードを入力します。ただし、Messaging Server では、パスワードを何度も入力する手間を省き (少なくとも 3 つのサーバプロセスで入力が必要)、さらに無人でサーバを再起動できるように、パスワードファイルからパスワードが読み取られます。

パスワードファイルは、`sslpassword.conf` という名前が付けられ、ディレクトリ *server-instance*`/config/` に格納されます。ファイル内の各エントリは、次のフォーマットで 1 行ずつ記述されます。

moduleName:password

moduleName は使用される (内部または外部) PKCS #11 モジュールの名前です。*password* はそのモジュールのキーのペアを暗号化するためのパスワードです。パスワードは、クリアテキスト (暗号化されていないテキスト) として保存されます。

Messaging Server には、デフォルトのパスワードファイルが用意されています。このファイルには、次のような内部モジュールとデフォルトのパスワードのエントリが 1 つだけ含まれています。

```
Internal (Software) Token:netscape!
```

内部証明書をインストールするときにデフォルト以外のパスワードを指定する場合は、指定するパスワードに合わせてパスワードファイル内の上記の行を編集する必要があります。外部モジュールをインストールする場合は、ファイルに新しい行を追加し、モジュール名とモジュール用に指定するパスワードを記述する必要があります。

警告 管理者はサーバ起動時にモジュールパスワードの入力を要求されません。そのため、管理者のアクセスが適切に制御されていること、およびサーバホストマシンとそのバックアップの物理的なセキュリティが確保されていることの確認が重要になります。

SSL を有効化し符号化方式を選択するには

コンソールを使用すると、SSL を有効にし、Messaging Server がクライアントとの暗号通信で使用できる符号化方式を選択できます。

符号化方式について

符号化方式とは、暗号化プロセスでデータの暗号化と解読に使用されるアルゴリズムのことです。各符号化方式によって強度が異なります。つまり、強度の高い符号化方式で暗号化したメッセージほど、承認されていないユーザによる解読が困難になります。

符号化方式では、キー (長い数値) をデータに適用することによってデータを操作します。一般的に、符号化方式で使用するキーが長いほど、適切な解読キーを使わずにデータを解読することが難しくなります。

クライアントは、Messaging Server と SSL 接続を開始するときに、サーバに対して、希望する暗号化用の符号化方式とキー長を伝えます。暗号化された通信では、両方の通信者が同じ符号化方式を使用する必要があります。一般的に使用される符号化方式とキーの組み合わせは数多くあります。そのため、サーバは柔軟な暗号化サポートを提供する必要があります。iPlanet Messaging Server では、最大 6 つの符号化方式とキー長の組み合わせをサポートできます。

表 6.1 に、Messaging Server が SSL 3.0 を使用する場合にサポートする符号化方式の一覧を示します。この表には、概要を記載しています。詳細は、『iPlanet Managing Servers with Netscape Console』の「Introduction to SSL」の節を参照してください。

表 12-1　Messaging Server の SSL 符号化方式

符号化方式	説明
128 ビットの暗号化と MD5 メッセージ認証を使用した RC4	RSA が提供する符号化方式で、もっとも高速で、もっとも強度の高い符号化方式と暗号化キーの組み合わせを提供する
168 ビットの暗号化と SHA メッセージ認証を使用した DES	米国政府の標準となっている符号化方式で、低速で、強度の高い符号化方式と暗号化キーの組み合わせを提供する
56 ビットの暗号化と SHA メッセージ認証を使用した DES	米国政府の標準となっている符号化方式で、低速で、中程度の強度の符号化方式と暗号化キーの組み合わせを提供する
40 ビットの暗号化と MD5 メッセージ認証を使用した RC4	RSA が提供する符号化方式で、もっとも高速で、強度の低い符号化方式と暗号化キーの組み合わせを提供する
40 ビットの暗号化と MD5 メッセージ認証を使用した RC2	RSA が提供する符号化方式で、低速で、強度の低い符号化方式と暗号化キーの組み合わせを提供する
暗号化なし、MD5 メッセージ認証のみ	暗号化を使用せず、認証用のメッセージダイジェストのみ使用する

特定の符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、すべての符号化方式をサポートする必要があります。ただし、特定の暗号化方式の使用が法律で制限されている国もあります。また、米国の輸出規制法規が緩和される前に開発されたクライアントソフトウェアの中には、強度の高い暗号化を使用できないものもあります。40 ビットの符号化方式では、偶発的な漏洩は防ぐことができますが、セキュリティが確保されないため、意図的な攻撃を防ぐことはできません。

コンソール： コンソールを使用して SSL を有効にし、符号化方式を選択するには、次の手順を実行します。

1. コンソールで、符号化方式の設定を変更する Messaging Server を開きます。
2. 左のペインの「環境設定」タブをクリックし、「サービス」フォルダを選択します。
3. 右のペインの「暗号化」タブをクリックします。
4. 「SSL の利用」ボックスにチェックマークを付け、サーバの SSL を有効にします。
5. RSA 符号化方式を使用可能にする場合は、「RSA」ボックスにチェックマークを付けます。

6. 「トークン」ドロップダウンリストから、使用するトークンを選択します。
7. 「証明書」ドロップダウンリストから、使用する証明書を選択します。
8. 「符号化方式のプリファレンス」をクリックして、使用可能な符号化方式のリストを表示します。
9. ボックスをクリックしてサーバでサポートする 1 つまたは複数の符号化方式を選択します。

SSL を完全に無効にするには、「SSL の利用」ボックスのチェックマークを外します。

注	送信メッセージの暗号化を有効にするには、チャネル定義を変更して、`maytls` や `musttls` などの `tls` チャネルキーワードを追加する必要があります。詳細は、『iPlanet Messaging Server リファンレンスマニュアル』を参照してください。

コマンドライン : 次のように、コマンドラインを使用して SSL を有効にし、符号化方式を選択することもできます。

SSL を有効化・無効化するには、次のように入力します。

```
configutil -o nsserversecurity -v [ on | off ]
```

RSA 符号化方式を有効化・無効化するには、次のように入力します。

```
configutil -o encryption.rsa.nssslactivation -v [ on | off ]
```

トークンを指定するには、次のように入力します。

```
configutil -o encryption.rsa.nsssltoken -v tokenname
```

証明書を指定するには、次のように入力します。

```
configutil -o encryption.rsa.nssslpersonalityssl -v certname
```

RSA 符号化方式を有効にする場合は、トークンと証明書も指定する必要があります。

優先する符号化方式を選択するには、次のように入力します。

```
configutil -o encryption.nsssl3ciphers -v cipherlist
```

cipherlist は、カンマで区切られた符号化方式のリストです。

証明書に基づくログインを設定するには

iPlanet サーバでは、パスワードに基づくログインに加えて、デジタル証明書の確認によるユーザ認証もサポートしています。証明書に基づく認証では、クライアントはサーバとの SSL セッションを確立し、ユーザの証明書をサーバに提出します。その後、サーバが、提出された証明書の信頼性を評価します。証明書の信頼性が確認されると、そのユーザは認証済みであるとみなされます。

証明書に基づくログインを実行できるように Messaging Server を設定するには、次の手順を実行します。

1. 使用しているサーバ用の証明書を入手します (詳細は、406 ページの「証明書の入手」を参照)。
2. 証明書セットアップウィザードを実行して、サーバが認証するユーザに証明書を発行する、信頼できる認証局の証明書をインストールします (詳細は、408 ページの「信頼できる CA の証明書をインストールするには」を参照)。

   サーバのデータベース内に信頼できる CA の証明書が 1 つでもあるかぎり、サーバは接続するクライアントに対してクライアント証明書を要求します。
3. SSL を有効にします (詳細は、410 ページの「SSL を有効化し符号化方式を選択するには」を参照)。
4. サーバが提出された証明書の情報に基づいて LDAP ユーザディレクトリを適切に検索するように、サーバの `certmap.conf` ファイルを編集します (省略可)。

   ユーザの証明書内の電子メールアドレスと、ユーザのディレクトリエントリ内の電子メールアドレスが一致する場合は、`certmap.conf` ファイルを編集する必要はありません。また、検索を最適化したり、提出された証明書をユーザエントリ内の証明書と照合したりする必要もありません。

   `certmap.conf` のフォーマットと変更可能な部分の詳細は、『iPlanet Managing Servers with Netscape Console』の SSL に関する章を参照してください。

上記の手順を実行したあとに、ユーザが、IMAP または HTTP にログインできるようにクライアントで SSL セッションを確立すると、Messaging Server からクライアントに対してユーザの証明書が要求されます。サーバによって信頼されている CA から発行された証明書をクライアントが提出し、かつ証明書の識別情報がユーザディレクトリ内のエントリと一致する場合、そのユーザは、認証され、ユーザに適用されるアクセス制御規則に応じたアクセス権が与えられます。

証明書に基づくログインを有効にするためにパスワードに基づくログインを無効する必要はありません。パスワードに基づくログインが許可されている場合 (デフォルトの状態) に、この節で説明した作業を実行すると、パスワードに基づくログインと証明書に基づくログインの両方がサポートされます。その場合は、クライアントが SSL セッションを確立し、証明書を提出すると、証明書に基づくログインが使用されます。クライアントが SSL を使用しない場合、または証明書を提出しない場合は、パスワードを要求されます。

証明書に基づく認証を使用するための iPlanet サーバ全体とクライアントの設定方法については、『Single Sign-On Deployment Guide』を参照してください。

SMTP プロキシを使用した SSL パフォーマンスの最適化方法

SMTP プロキシを使用すると、SMTP プロトコルの待ち時間が増加するため、ほとんどのサイトでは SMTP プロキシを使用しません。ただし、SMTP 接続を保護するために SSL を頻繁に使用する大規模サイトでは、SSL とプロキシ専用の 1 台のサーバ上で、すべてのプロトコルのすべての SSL 操作を実行することで、SSL アクセラレータハードウェアに対する投資効果を最大化する必要があります。SMTP プロキシを使用すると、フロントエンドのプロキシサーバで SSL を処理し、メールキューを別の MTA マシン上に置くことができます。この方法により、各タスクに最適なハードウェアを個別に購入して構成することができます。

SMTP プロキシのインストール方法については、429 ページの「SMTP プロキシをインストールするには」を参照してください。

Messaging Server への管理者アクセスを構成する

この節には、以下の項があります。

この節では、サーバ管理者による Messaging Server へのアクセスを制御する方法について説明します。特定の Messaging Server および Messaging Server タスクへの管理アクセスは、委任サーバ管理を行うときに発生します。

委任サーバ管理は、ほとんどの iPlanet サーバが持っている機能で、管理者が、他の管理者に対して、個々のサーバやサーバ機能へのアクセス権を選択して提供できる機能を意味します。この章では、委任されたサーバのタスクについて簡単に説明します。詳細は、『iPlanet Managing Servers with Netscape Console』のサーバ管理の委任に関する章を参照してください。さらに、『iPlanet Messaging Server プロビジョニングガイド』

の「Provisioning Messaging Server 管理者のプロビジョニング」も参照してください。『プロビジョニングガイド』では、サーバ管理者 (Messaging Server を構成できる管理者)、および iDA 管理者 (システム内のユーザやグループを追加、変更、削除できる管理者) について説明しています。

委任管理の階層

ネットワーク上に最初の iPlanet サーバをインストールすると、インストールプログラムによって、LDAP ユーザディレクトリに構成管理者グループと呼ばれるグループが自動的に作成されます。デフォルトでは、構成管理者グループのメンバーには、ネットワーク上のすべてのホストおよびサーバに対する無制限のアクセス権が与えられます。

構成管理者グループは、次のようなアクセス階層の最上位に位置します。このようなアクセス階層を構築して、Messaging Server の委任管理を実装することができます。

1. **構成管理者** : iPlanet サーバのネットワークの「スーパーユーザ」。すべてのリソースに対する完全なアクセス権を持ちます。
2. **サーバ管理者** : ドメイン管理者は、各タイプのサーバを管理するためのグループを作成することがあります。たとえば、管理ドメイン内またはネットワーク全体にあるすべての Messaging Server を管理するためにメッセージング管理者グループを作成する場合があります。このグループのメンバーは、その管理ドメイン内のすべての Messaging Server にアクセスできます (他のサーバにはアクセスできません)。
3. **タスク管理者** : 上記のすべての管理者は、単一または複数の Messaging Server に対する制限付きアクセス権を持つグループを作成したり、そのようなアクセス権を持つ個別のユーザを指定したりできます。指定されたタスク管理者は、特定の制限されたサーバタスク (サーバの起動または停止、特定のサービスのログへのアクセス) だけを実行できます。

管理者は、コンソールが提供する便利なインタフェースを使用して、次のタスクを実行できます。

- グループまたは個人に特定の Messaging Server に対するアクセス権を与える。次の節の「サーバ全体に対するアクセス権の提供」を参照
- そのアクセス権を特定の Messaging Server 上での特定のタスクに制限する。416 ページの「特定タスクへのアクセスを限定するには」を参照

サーバ全体に対するアクセス権を与えるには

ユーザまたはグループに Messaging Server の特定のインスタンスに対するアクセス権を与えるには、次の手順を実行します。

1. アクセス権を与える対象の Messaging Server へのアクセス権を持っている管理者として、コンソールにログインします。
2. 「コンソール」ウィンドウでそのサーバを選択します。

   「コンソール」メニューから「オブジェクト」を選択し、「アクセス権の設定」を選択します。
3. そのサーバへのアクセス権を持つユーザおよびグループのリストに対する追加や編集を行います。

詳細な手順については、『iPlanet Managing Servers with Netscape Console』のサーバ管理の委任に関する章を参照してください。

特定の Messaging Server へのアクセス権を持つユーザおよびグループのリストの設定が済んだら、次に説明する ACI を使用して、特定のサーバタスクをリスト内の特定のユーザまたはグループに委任することができます。

特定タスクへのアクセスを限定するには

一般的に、管理者はサーバに接続して 1 つ以上の管理タスクを実行します。コンソールの「Messaging Server タスク」フォームには、頻繁に実行される管理タスクが一覧表示されます。

デフォルトでは、特定の Messaging Server にアクセスできると、そのサーバのすべてのタスクにアクセスできます。ただし、タスクフォーム内の各タスクには、一連のアクセス制御インストラクション (ACI) を関連付けることができます。サーバは、接続しているユーザ (サーバ全体に対するアクセス権をすでに持っているユーザ) にタスクへのアクセス権を与える前に、これらの ACI を参照します。実際、タスクフォームには、そのユーザがアクセス権を持っているタスクだけが表示されます。

Messaging Server へのアクセス権がある場合は、アクセスできる任意のタスクに関する ACI を作成または編集して、他のユーザやグループがそのタスクに対して持つことができるアクセス権を制限できます。

接続しているユーザまたはグループが持つことができるタスクアクセス権を制限するには、次の手順を実行します。

1. 制限付きアクセス権を与える対象の Messaging Server へのアクセス権を持っている管理者として、コンソールにログインします。

2. サーバを開き、そのサーバのタスクフォームで、タスクのテキストをクリックして、タスクを選択します。
3. 「編集」メニューの「アクセス権の設定」を選択し、アクセス規則のリストに対する追加や編集を行い、ユーザまたはグループに必要なアクセス権を与えます。
4. 必要に応じて、他のタスクについて同じ手順を繰り返します。

   詳細な手順については、『iPlanet Managing Servers with Netscape Console』のサーバ管理の委任に関する章を参照してください。

ACI とその作成方法の詳細は、『iPlanet Managing Servers with Netscape Console』のサーバ管理の委任に関する章を参照してください。

POP、IMAP、および HTTP サービスへのクライアントアクセスを構成する

この節には、以下の項があります。

Messaging Server には、IMAP、POP、HTTP の各サービスを個別に制御できる精巧なアクセス制御機能があります。これにより、クライアントによるサーバへのアクセスを広範囲に細かく制御できます。

大企業やインターネットサービスプロバイダのメッセージングサービスを管理する場合、これらの機能を使用して、スパム (大量メール送信) や DNS スプーフィングを行うユーザをシステムから除外したり、ネットワークの全般的なセキュリティを強化したりできます。不特定多数宛てメールを制御するための具体的な方法については、第 10 章「メールのフィルタリングとアクセス制御」を参照してください。

注 IP アドレスによるアクセス制御が重要な問題ではない場合は、この節で説明しているフィルタを作成する必要はありません。最小限のアクセス制御だけが必要な場合は、その設定手順について、424 ページの「大半のアクセスを許可」を参照してください。

クライアントアクセスフィルタのしくみ

Messaging Server のアクセス制御機能は、プログラムであり、TCP デーモンと同じポートで応答を待機します。このプログラムは、アクセスフィルタを使用してクライアントの識別情報を確認し、クライアントがフィルタリングプロセスを通過した場合に、そのクライアントに対してデーモンへのアクセス権を与えます。

Messaging Server の TCP クライアントアクセス制御システムは、必要な場合、その処理の一部として、次のようなソケットの終端アドレスの分析を行います。

- 両方の終端の逆引き DNS 検索 (名前に基づくアクセス制御を行うため)
- 両方の終端の正引き DNS 検索 (DNS スプーフィングを検出するため)
- `Identd` コールバック (クライアントエンドのユーザがクライアントホストに認識されていることを調べるため)

システムは、この情報をフィルタと呼ばれるアクセス制御文と比較して、アクセスの許可または拒否を決定します。サービスごとに、個別の許可フィルタと拒否フィルタのセットを使用して、アクセスを制御します。許可フィルタは明示的にアクセスを許可し、拒否フィルタは明示的にアクセスを禁止します。

クライアントがサービスへのアクセスを要求すると、アクセス制御システムは、そのクライアントのアドレスまたは名前情報を、以下の条件を使用して順番に対象のサービスのフィルタと比較します。

- 検索は、最初の一致項目が見つかった時点で終了する。許可フィルタは、拒否フィルタより先に処理されるため、許可フィルタが優先される
- クライアント情報が対象のサービスの許可フィルタに一致した場合は、アクセスが許可される
- クライアント情報がそのサービスの拒否フィルタに一致した場合は、アクセスが拒否される
- 許可フィルタと拒否フィルタのどちらにも一致しなかった場合は、アクセスが許可される。ただし、許可フィルタだけがあり、拒否フィルタがない場合は、許可フィルタに一致しないと、アクセスが拒否される

ここで説明するフィルタの構文は柔軟性に富んでいるため、わかりやすい簡単な方法で、さまざまなアクセス制御ポリシーを実装できます。許可フィルタと拒否フィルタは自由に組み合わせて使用できますが、大半のアクセスを許可するフィルタまたは大半のアクセスを拒否するフィルタを使用すると、ほとんどのポリシーを実装できます。

以下の節では、フィルタの構文について詳しく説明し、さらに使用例を紹介します。アクセスフィルタの作成手順については、425 ページの「各サービス用のアクセスフィルタを作成するには」を参照してください。

フィルタの構文

フィルタ文は、サービス情報とクライアント情報の両方を含んでいます。サービス情報には、サービス名、ホスト名、ホストアドレスを含めることができます。クライアント情報には、ホスト名、ホストアドレス、ユーザ名を含めることができます。サービス情報とクライアント情報の両方で、ワイルドカード名やパターンを使用できます。

以下に、非常に単純な形式のフィルタを示します。

service: hostSpec

service には、サービス名 (smtp、pop、imap、http など) を指定し、*hostSpec* には、ホスト名、IP アドレス、またはアクセス要求元のクライアントを表すワイルドカード名やパターンを指定します。フィルタが処理されるときに、アクセス要求元のクライアントが *client* に一致すると、*service* で指定されているサービスへのアクセスが (フィルタのタイプに応じて) 許可または拒否されます。次に例を示します。

```
imap: roberts.newyork.siroe.com
pop: ALL
http: ALL
```

これらが許可フィルタの場合は、最初の行によって roberts.newyork.siroe.com というホストに対して、IMAP サービスへのアクセスが許可されます。さらに 2 行目と 3 行目によって、それぞれ POP サービスと HTTP サービスへのアクセスがすべてのクライアントに許可されます。これらが拒否フィルタの場合は、それらのクライアントによる指定したサービスへのアクセスが拒否されます。ALL などのワイルドカード名の詳細は、420 ページの「ワイルドカード名」を参照してください。

フィルタ内のサーバ (サービス) 情報やクライアント情報は、これよりも少々複雑になることがあります。次に、その場合の一般的な形式を示します。

serviceSpec: *clientSpec*

serviceSpec は、*service* または *service@hostSpec* のどちらかを示し、*clientSpec* は、*hostSpec* または *user@hostSpec* のどちらかを示します。*user* はアクセス要求元のクライアントホストに関連付けられたユーザ名 (またはワイルドカード名) です。次にフィルタの例を 2 つ示します。

```
pop@mailServer1.siroe.com: ALL
imap: srashad@xyz.europe.siroe.com
```

これらが拒否フィルタの場合、最初のフィルタは、すべてのクライアントに対して、ホスト mailServer1.siroe.com 上の SMTP サービスへのアクセスを拒否します。2 番目のフィルタは、ホスト xyz.europe.siroe.com のユーザ srashad に対して、IMAP サービスへのアクセスを拒否します。これらの詳細なサーバおよびクライアントに対する指定を使用する状況については、422 ページの「サーバホストの指定」および 423 ページの「クライアントのユーザ名の指定」を参照してください。

もっとも一般的なフィルタの形式は次のようになります。

serviceList: *clientList*

serviceList は、1 つ以上の *serviceSpec* エントリで構成され、*clientList* は、1 つ以上の *clientSpec* エントリで構成されます。*serviceList* と *clientList* 内の各エントリは、空白またはカンマで区切ります。

この場合、フィルタが処理されるときに、アクセス要求元のクライアントが、*clientList* 内の *clientSpec* エントリのいずれかと一致すると、*serviceList* で指定されているすべてのサービスへのアクセスが (フィルタのタイプに応じて) 許可または拒否されます。次に例を示します。

pop、imap、http: .europe.siroe.com .newyork.siroe.com

これが許可フィルタの場合、europe.siroe.com ドメインおよび newyork.siroe.com ドメイン内のすべてのクライアントに対して、POP、IMAP、HTTP サービスへのアクセスが許可されます。ドメインやサブネットを指定する場合の先頭に付けるドットや他のパターンの使用方法については、421 ページの「ワイルドカードのパターン」を参照してください。

ワイルドカード名

以下のワイルドカード名を使用して、サービス名、ホストの名前やアドレス、またはユーザ名を表すことができます。

表 12-2 ワイルドカード名

ワイルドカード名	説明
ALL	汎用のワイルドカード。すべての名前に一致する
LOCAL	すべてのローカルホスト (ドット文字を含まない名前を持つホスト) に一致する。ただし、正規名のみを使用しているシステムの場合は、ローカルホスト名もドットを含むため、このワイルドカードに一致しない

表 12-2　ワイルドカード名 (続き)

ワイルドカード名	説明
UNKNOWN	名前が不明なすべてのユーザ、あるいは名前またはアドレスが不明なすべてのホストに一致する このワイルドカード名は、次のことに注意して使用する必要がある 一時的な DNS サーバの問題により、ホスト名が使用できなくなる場合がある。このような場合、UNKNOWN を使用しているすべてのフィルタはすべてのクライアントホストに一致する ソフトウェアが通信相手のネットワークのタイプを識別できない場合は、ネットワークアドレスを使用できない。そのような場合、UNKNOWN を使用しているすべてのフィルタは、そのネットワーク上にあるすべてのクライアントホストに一致する
KNOWN	名前が認識されているすべてのユーザ、または名前とアドレスが認識されているすべてのホストに一致する このワイルドカード名は、次のことに注意して使用する必要がある 一時的な DNS サーバの問題により、ホスト名が使用できなくなる場合がある。このような場合、KNOWN を使用しているすべてのフィルタはどのクライアントホストにも一致しない ソフトウェアが通信相手のネットワークのタイプを識別できない場合は、ネットワークアドレスを使用できない。そのような場合、KNOWN を使用しているすべてのフィルタは、そのネットワーク上にあるどのクライアントホストにも一致しない
DNSSPOOFER	IP アドレスと DNS 名が一致しないすべてのホストに一致する

ワイルドカードのパターン

サービスまたはクライアントアドレスを指定するときは、次のパターンを使用できます。

- ドット文字 (.) から始まる文字列。ホスト名の最後の部分が指定したパターンに一致する場合、そのホスト名は一致します。たとえば、ワイルドカードパターン `.siroe.com` は、ドメイン `siroe.com` 内のすべてのホストに一致します。
- ドット文字 (.) で終わる文字列。ホストアドレスの最初の数値フィールドが指定したパターンに一致する場合、そのホストアドレスは一致します。たとえば、ワイルドカードパターン `123.45.` は、サブネット `123.45.0.0` 内のすべてのホストのアドレスに一致します。

- n.n.n.n/m.m.m.m 形式の文字列。このワイルドカードパターンは、*net/mask* のペアと解釈されます。ホストアドレスの *net* が、アドレスと *mask* のビット単位の論理積と等しい場合、そのホストアドレスは一致します。たとえば、123.45.67.0/255.255.255.128 というパターンは、123.45.67.0 ～ 123.45.67.127 の範囲内のすべてのアドレスに一致します。

EXCEPT 演算子

アクセス制御システムでは、1 つの演算子がサポートされています。この EXCEPT 演算子を使うと、*serviceList* または *clientList* 内に複数のエントリがある場合に、名前やパターンの一致に関する例外を指定することができます たとえば、次のような式を使用します。

list1 EXCEPT *list2*

この式では、*list1* に一致するもので、*list2* に一致しないものが、すべて一致します。

次に例を示します。

```
ALL: ALL EXCEPT isserver.siroe.com
```

これが拒否フィルタの場合、ホストマシン isserver.siroe.com 上のクライアントを除くすべてのクライアントに対して、すべてのサービスへのアクセスが拒否されます。

EXCEPT 句は入れ子にすることができます。次に入れ子の式の例を示します。

list1 EXCEPT *list2* EXCEPT *list3*

これは次の式と同様に評価されます。

list1 EXCEPT (*list2* EXCEPT *list3*)

サーバホストの指定

serviceSpec エントリにサーバホストの名前またはアドレス情報を含めることで、要求される特定のサービスをフィルタ内で識別することができます。この場合、次の形式でエントリを指定します。

service@hostSpec

この機能は、Messaging Server ホストマシンが、異なるインターネットホスト名を持つ複数のインターネットアドレス用に設定されている場合に有効です。サービスプロバイダの場合、この機能を使用することで、異なるアクセス制御規則を持つ複数のドメインを 1 つのサーバインスタンス上でホストできます。

クライアントのユーザ名の指定

RFC 1413 に記載された identd サービスをサポートするクライアントホストマシンの場合は、フィルタの *clientSpec* エントリ内にクライアントのユーザ名を含めることにより、サービスを要求している特定のクライアントを識別することができます。この場合、次の形式でエントリを指定します。

user@hostSpec

user は、クライアントの identd サービスによって返されるユーザ名 (またはワイルドカード名) です。

フィルタ内でクライアントユーザ名を指定すると便利ですが、次のことに注意する必要があります。

- identd サービスは認証機能ではないため、クライアントシステムが安全性に欠ける場合は、クライアントから返されるクライアントユーザ名を信頼することができません。一般的に、特定のユーザ名を使用せずに、ALL、KNOWN、UNKNOWN などのワイルドカード名だけを使用します。
- identd は最新のクライアントマシンではサポートされていないため、最近の導入ではあまり付加価値がありません。将来のバージョンでは identd のサポートを廃止することが検討されているため、今後もこの機能を使う必要がある場合は iPlanet にお知らせください。
- ユーザ名の検索は時間がかかるので、すべてのユーザについて検索を実行すると、identd をサポートしていないクライアントのアクセスが遅くなる場合があります。ユーザ名の検索を選択的に実行すると、この問題を緩和することができます。たとえば次のように指定します。

  serviceList: @xyzcorp.com ALL@ALL

  この場合、xyzcorp.com ドメイン内のユーザは、ユーザ名の検索を実行せずに一致します。ただし、他のすべてのシステムについては、ユーザ名の検索が実行されます。

ユーザ名検索の機能は、クライアントホスト上の承認されていないユーザからの攻撃を防ぐために役立つ場合があります。たとえば、一部の TCP/IP の実装環境では、侵入者が rsh (リモートシェルサービス) を使用して信頼されているクライアントホストになりすます場合があります。クライアントホストが ident サービスをサポートしている場合は、ユーザ名の検索を使用してそのような攻撃を検出できます。

フィルタの例

この節では、さまざまなアクセス制御方法の例を紹介します。これらの例を参照する際には、許可フィルタが拒否フィルタよりも先に処理されること、一致するものが見つかった時点で検索が終了すること、および一致するものがまったく見つからないとアクセスが許可されることに注意してください。

ここに記載した例では、IP アドレスではなく、ホスト名とドメイン名を使用します。フィルタにアドレス情報やネットマスク情報を含めておくと、ネームサービスに障害が発生した場合の信頼性を向上させることができます。

大半のアクセスを拒否

この例では、デフォルトでアクセスを拒否します。明示的に許可したホストだけにアクセスを許可します。

デフォルトのポリシー (アクセスなし) は、次のような 1 つの単純な拒否フィルタを使用して実装します。

```
ALL: ALL
```

このフィルタは、許可フィルタによって明示的にアクセスを許可されていないすべてのクライアントに対して、すべてのサービスへのアクセスを拒否します。この場合の許可フィルタは、たとえば次のようになります。

```
ALL: LOCAL @netgroup1

ALL: .siroe.com EXCEPT externalserver.siroe.com
```

最初の規則は、ローカルドメイン内のすべてのホスト (ドットを含まないホスト名を持つすべてのホスト) からのアクセス、および `netgroup1` というグループのメンバーからのアクセスを許可します。2 番目の規則では、先頭にドットが付いたワイルドカードパターンを使用することで、`siroe.com` ドメイン内のすべてのホストからのアクセスを許可しますが、ホスト `externalserver.siroe.com` は除外されます。

大半のアクセスを許可

この例では、デフォルトでアクセスを許可します。明示的に拒否したホストだけにアクセスを拒否します。

デフォルトのポリシー (アクセス許可) により、許可フィルタは不要になります。次のように、アクセスを拒否するクライアントのリストを拒否フィルタ内に明示的に指定します。

```
ALL: externalserver.siroe1.com, .siroe.asia.com

ALL EXCEPT pop: contractor.siroe1.com, .siroe.com
```

最初のフィルタは、特定のホストおよびドメインに対して、すべてのサービスを拒否します。2 番目のフィルタは、特定のホストおよびドメインからの POP アクセスだけを許可します。

スプーフィングされたドメインのアクセスを拒否

フィルタ内で、DNSSPOOFER を使用すると、ホスト名のスプーフィングを検出できます。DNSSPOOFER を指定すると、アクセス制御システムによって正引きまたは逆引きの DNS 検索が実行され、クライアントが提示したホスト名とホストの実際の IP アドレスが一致するかどうかが調べられます。以下に拒否フィルタの例を示します。

```
ALL: DNSSPOOFER
```

このフィルタは、IP アドレスとその DNS ホスト名が一致しないすべてのリモートホストに対して、すべてのサービスを拒否します。

仮想ドメインへのアクセス制御

メッセージングシステムで仮想ドメインを使用し、1 つのサーバインスタンスが複数の IP アドレスおよびドメイン名に関連付けられている場合は、許可フィルタと拒否フィルタを組み合わせて各仮想ドメインのアクセスを制御できます。たとえば、次のような許可フィルタを使用できます。

```
ALL@msgServer.siroe1.com: @.siroe1.com

ALL@msgServer.siroe2.com: @.siroe2.com

...
```

この場合、次のような拒否フィルタと組み合わせることができます。

```
ALL: ALL
```

各許可フィルタは、domain*N* 内のホストだけに、msgServer.siroe*N*.com に対応する IP アドレスを持つサービスへの接続を許可します。他の接続はすべて拒否されます。

各サービス用のアクセスフィルタを作成するには

IMAP、POP、HTTP の各サービス用の許可フィルタと拒否フィルタを作成できます。SMTP サービス用に作成することもできますが、認証済みの SMTP セッションにしか適用されないため、あまり価値はありません。認証されていない SMTP セッションへのアクセスを制御する方法については、第 10 章「メールのフィルタリングとアクセス制御」を参照してください。

コンソール : コンソールを使用してフィルタを作成するには、次の手順を実行します。

1. コンソールで、アクセスフィルタを作成する Messaging Server を開きます。

2. 「環境設定」タブをクリックします。
3. 左のペインで「サービス」フォルダを開き、そのフォルダの下にある「IMAP」、「POP」、または「HTTP」を選択します。
4. 右のペインの「アクセス」タブをクリックします。

   このタブの「許可」フィールドと「拒否」フィールドに、そのサービスの既存の許可フィルタと拒否フィルタが表示されます。フィールド内の各行がそれぞれ 1 つのフィルタを表します。どちらのフィールドに対しても、以下の操作を実行できます。

   - 新しいフィルタを追加するには、「追加」をクリックします。「Allow フィルタ」ウィンドウまたは「Deny フィルタ」ウィンドウが表示されます。ウィンドウに新しいフィルタのテキストを入力し、「OK」をクリックします。
   - フィルタを編集する場合は、フィルタを選択して「編集」をクリックします。「Allow フィルタ」ウィンドウまたは「Deny フィルタ」ウィンドウが表示されます。ウィンドウに表示されたフィルタのテキストを編集し、「OK」をクリックします。
   - フィルタを削除する場合は、フィルタを選択して「削除」をクリックします。

許可フィルタまたは拒否フィルタの順序を変更する必要がある場合は、フィルタが適切な順序になるまで、削除と追加の操作を繰り返します。

フィルタの構文の指定方法とさまざまな例については、419 ページの「フィルタの構文」を参照してください。その他の例については、424 ページの「フィルタの例」を参照してください。

コマンドライン : 次のように、コマンドラインを使用して許可フィルタや拒否フィルタを指定することもできます。

各サービス用のアクセスフィルタを作成または編集するには、次のように入力します。

```
configutil -o service.service.domainallowed -v filter
```

service には `pop`、`imap`、`http` のいずれかを指定し、*filter* は、419 ページの「フィルタの構文」で説明した構文規則に従って指定します。

各サービス用の拒否フィルタを作成または編集するには、次のように入力します。

```
configutil -o service.service.domainnotallowed -v filter
```

service には `pop`、`imap`、`http` のいずれかを指定し、*filter* は、419 ページの「フィルタの構文」で説明した構文規則に従って指定します。

HTTP プロキシ認証用のアクセスフィルタを作成するには

すべてのストア管理者は、任意のサービスに対してプロキシ認証を行うことができます (ストア管理者の詳細は、349 ページの「ストアへの管理者によるアクセスを指定する」を参照)。HTTP サービスの場合にだけ、すべてのエンドユーザがサービスに対してプロキシ認証を行うことができます。ただし、ユーザが使用するクライアントホストが、プロキシ認証アクセスフィルタを介してアクセスを許可されている必要があります。

プロキシ認証を使用すると、ポータルサイトなどの他のサービスが、ユーザを認証して、HTTP ログインサービスに認証資格情報を渡すことができます。たとえば、1 つのポータルサイトが複数のサービスを提供し、そのうちの 1 つが Messenger Express の Web ベースの電子メールだとします。HTTP プロキシ認証機能を使用すると、エンドユーザはポータルサービスに対する認証を一度行うだけで済み、電子メールにアクセスするために再び認証を行う必要はありません。ただし、ポータルサイトでは、クライアントとサービス間のインタフェースとして機能するログインサーバを構成する必要があります。Messenger Express の認証用にログインサーバを設定する場合は、iPlanet が提供する Messenger Express 認証 SDK を利用できます。

この節では、許可フィルタを使用し、IP アドレスを基準として、HTTP プロキシ認証を許可する方法について説明します。ログインサーバの設定方法や Messenger Express 認証 SDK の使用方法については説明しません。Messenger Express 用のログインサーバの設定方法や、認証 SDK の使用方法については、iPlanet の担当者に問い合わせてください。

コンソール : HTTP サービスに対するプロキシ認証用のアクセスフィルタを作成するには、次の手順を実行します。

1. コンソールで、アクセスフィルタを作成する Messaging Server を開きます。
2. 「環境設定」タブをクリックします。
3. 左のペインで「サービス」フォルダを開き、そのフォルダの下にある「HTTP」を選択します。
4. 右のペインの「プロキシ」タブをクリックします。

このタブの「許可」フィールドに、既存のプロキシ認証用の許可フィルタが表示されます。

5. 新しいフィルタを作成する場合は、「追加」をクリックします。

   「Allow フィルタ」ウィンドウが表示されます。ウィンドウに新しいフィルタのテキストを入力し、「OK」をクリックします。
6. 既存のフィルタを編集する場合は、フィルタを選択して、「編集」をクリックします。

「Allow フィルタ」ウィンドウが表示されます。ウィンドウに表示されたフィルタのテキストを編集し、「OK」をクリックします。

7. 既存のフィルタを削除する場合は、「許可」フィールドからフィルタを選択し、「削除」をクリックします。
8. 「プロキシ」タブでの変更作業が終了したら、「保存」をクリックします。

許可フィルタの構文については、419 ページの「フィルタの構文」を参照してください。

コマンドライン : 次のように、コマンドラインを使用して、HTTP サービスに対するプロキシ認証用のアクセスフィルタを指定することもできます。

```
configutil -o service.service.proxydomainallowed -v filter
```

filter は、419 ページの「フィルタの構文」で説明した構文規則に従って指定します。

POP before SMTP を有効にする

SMTP リレーサーバのセキュリティを提供する方法としては、SMTP 認証または *SMTP Auth* (RFC 2554) をお勧めします。SMTP Auth は、認証済みのユーザだけに MTA を介したメール送信を許可します。ただし、一部のレガシークライアントは、*POP before SMTP* だけをサポートします。この場合には、後述のように、POP before SMTP を有効にすることができます。ただし、可能な場合は、POP before SMTP を使用するのではなく、POP クライアントをアップグレードするようにユーザに指示します。POP before SMTP をサイトに導入すると、ユーザがクライアントに依存するようになり、インターネットのセキュリティ標準を守れなくなります。これにより、エンドユーザがハッキングの危険にさらされ、さらにパフォーマンスが低下して、サイトの処理が遅くなります。これは、最後の正常な POP セッションの IP アドレスを追跡して同期する必要があるためです。

iPlanet Messaging Server での POP before SMTP の実装は、SIMS や Netscape Messaging Server での実装とはまったく異なっています。POP before SMTP をサポートするには、POP と SMTP プロキシの両方を使用するように Messaging Multiplexor (MMP) を構成します。SMTP クライアントが SMTP プロキシに接続すると、プロキシは、メモリ内キャッシュで最新の POP 認証をチェックします。同じクライアント IP アドレスからの POP 認証が見つかった場合、SMTP プロキシは、ローカルとローカル以外の両方の受取人宛てのメッセージを許可する必要があることを SMTP サーバに通知します。

SMTP プロキシをインストールするには

1. 『iPlanet Messaging Server インストールガイド』の説明に従って、iPlanet Messaging Multiplexor (MMP) をインストールします。
2. MMP 上で SMTP プロキシを有効にします。

   以下の文字列を

   server_root/bin/msg/mmp/lib/SmtpProxyAService@25|587

   server_root/mmp-*hostname*/AService.cfg ファイルの ServiceList オプションに追加します。このオプションは、1 行に記述し、改行を入れないようにします。

注 MMP をアップグレードすると、MMP 用の既存の 4 つの設定ファイルに対応する次の 4 つの新しいファイルが作成されます。

AService-def.cfg、ImapProxyAService-def.cfg、PopProxyAService-def.cfg、SmtpProxyAService-def.cfg

これらのファイルは、インストーラによって作成されます。docs 内に記述された 4 つの設定ファイルは、インストールプロセスによって作成されず、また影響も受けません。MMP は、起動時に、通常の設定ファイルを検索します。通常の設定ファイルが見つからない場合、MMP は、それぞれの *AService-def.cfg ファイルをコピーして、対応する *AService.cfg という名前を付けます。

3. 各 SMTP リレーサーバ上で、SMTP チャネルオプションファイル tcp_local_option の PROXY_PASSWORD オプションを設定します。

   SMTP プロキシは、SMTP サーバに接続する際に、実際の IP アドレスとその他の接続情報を SMTP サーバに通知する必要があります。この情報により、SMTP サーバは、リレーブロッキングやその他のセキュリティポリシー (POP before SMTP を含む) を適切に適用できるようになります。この操作はセキュリティ上重要な操作であり認証される必要があります。MMP SMTP プロキシと SMTP サーバの両方で構成されたプロキシパスワードにより、第三者によるこの機能の悪用が確実に防止されます。

   例： PROXY_PASSWORD *A_Password*

4. POP before SMTP をサポートするように SMTP プロキシを構成します。
   a. *server_root*/mmp-*instance*/SmtpProxyAService.cfg 設定ファイルを編集します。

以下の SMTP プロキシオプションは、IMAP プロキシおよび POP プロキシの同名のオプションとまったく同じように機能します。『iPlanet Messaging Server インストールガイド』の「Installing the Messaging Multiplexor」を参照してください。またこれらのオプションについては、『iPlanet Messaging Server リファレンスマニュアル』の「暗号化 (SSL) オプション」の節を参照してください。

`LdapURL`、`LogDir`、`LogLevel`、`BindDN`、`BindPass`、`Timeout`、`Banner`、`SSLEnable`、`SSLSecmodFile`、`SSLCertFile`、`SSLKeyFile`、`SSLKeyPasswdFile`、`SSLCipherSpecs`、`SSLCertNicknames`、`SSLCacheDir`、`SSLPorts`、`CertMapFile`、`CertmapDN`、`ConnLimits`、`TCPAccess`

上記のリストにないその他の MMP オプション (`BacksidePort` オプションを含む) は、現在のところ SMTP プロキシには適用されません。

次の 5 つのオプションを追加します。

`SmtpRelays`。このオプションは、スペースで区切られた SMTP リレーサーバホスト名 (およびオプションのポート) のリストで、ラウンドロビンリレー用に使用されます。これらのリレーサーバは、`XPROXYEHLO` 拡張キーワードをサポートしている必要があります。このオプションは必須で、デフォルト値はありません。**例** : `default:SmtpRelays  manatee:485 gonzo mothra`

`SmtpProxyPassword`。SMTP リレーサーバ上でソースチャネルの変更を認証するために使用されるパスワードです。このオプションは必須で、デフォルト値はありません。また、SMTP サーバ上の `PROXY_PASSWORD` オプションと一致している必要があります。
例 : `default:SmtpProxyPassword  A_Password`

`EhloKeywords`。このオプションは、プロキシがクライアントを通過させるために使用する、EHLO 拡張キーワードのリストを提供します。また、デフォルト値のセットも提供します。MMP は、SMTP リレーから返される EHLO のリストから、認識できない EHLO キーワードをすべて削除します。`EhloKeywords` は、リストから削除されないようにする必要のある、追加の EHLO キーワードを指定します。デフォルト値は空白ですが、SMTP プロキシは以下のキーワードをサポートするので、これらのキーワードをこのオプションで指定する必要はありません。`8BITMIME`、`PIPELINING`、`DSN`、`ENHANCEDSTATUSCODES`、`EXPN`、`HELP`、`XLOOP`、`ETRN`、`SIZE`、`STARTTLS`、`AUTH`

以下に、使用頻度の少ない「TURN」拡張キーワードを使用するサイトで使用できる指定例を示します。
例 : `default:EhloKeywords  TURN`

PopBeforeSmtpKludgeChannel オプションは、POP before SMTP で認証される接続で使用する MTA チャネルの名前に設定されます。デフォルト値は空白です。POP before SMTP を有効にする必要があるユーザは、一般に tcp_intranet の設定を使用します。SSL のパフォーマンスを最適化するためにこのオプションを指定する必要はありません (414 ページの「SMTP プロキシを使用した SSL パフォーマンスの最適化方法」を参照)。

例 : `default:PopBeforeSmtpKludgeChannel  tcp_intranet`

ClientLookup。このオプションはデフォルトで no に設定されます。yes に設定すると、クライアントの IP アドレスに関する DNS 逆引き検索が無条件に実行されるため、SMTP リレーサーバで検索を行う必要がなくなります。このオプションは、ホストドメインごとに設定できます。

例 : `default:ClientLookup  yes`

b. PopProxyAService.cfg 設定ファイルに PreAuth オプションと AuthServiceTTL オプションを設定します。SSL のパフォーマンスを最適化するためにこのオプションを指定する必要はありません (414 ページの「SMTP プロキシを使用した SSL パフォーマンスの最適化方法」を参照)。

注 POP before SMTP を機能させるために、IMAP または SMTP のプロキシ設定ファイル内で、AuthServiceTTL を設定する必要は**ありません**。

これらのオプションは、POP 認証後にユーザがメールの送信を許可される時間を秒単位で指定します。一般的な設定は、900 ～ 1800 (15 ～ 30 分) です。

例 :

```
default:PreAuth     yes
default:AuthServiceTTL     900
```

c. オプションで、MMP が、SMTP リレーからの応答を待つ時間を秒単位で指定することができます。この時間が経過すると MMP はリスト内の次の SMTP リレーを試行します。

デフォルトは 10 (秒) です。SMTP リレーへの接続が失敗すると、MMP は、このフェイルオーバータイムアウトと同じ時間 (分単位) が経過するまで、そのリレーへの接続を試行しません。つまり、フェイルオーバータイムアウトが 10 秒のときに、あるリレーへの接続が失敗したとすると、MMP は、10 分間経過するまでそのリレーを再試行しません。

例 : `default:FailoverTimeout  10`

SMTP サービスへのクライアントアクセスを構成する

SMTP サービスへのクライアントアクセスの構成方法については、第 10 章「メールのフィルタリングとアクセス制御」を参照してください。

第 13 章

ログ記録とログ解析

iPlanet Messaging Server では、ログファイルを作成して、管理に関連するサーバのイベント、サーバでサポートされているプロトコル (SMTP、POP、IMAP、HTTP) を使用した通信関連のイベント、およびサーバで処理されるその他のプロセスに関するイベントを記録できます。このログファイルを調べれば、サーバのアクションをさまざまな観点からモニタすることができます。

MTA は他のサービスとは異なるログ機構を使用しているため、iPlanet Console を使ってログサービスを設定したりログを表示することはできません。その代わり、設定ファイルに情報を指定することで、MTA のログ機能を設定します。この章は、以下のように 3 部構成になっています。第 1 部では概要について、第 2 部ではメッセージストアおよび管理サービスのログ、第 3 部では MTA サービスのログについて説明します。

433 ページの「第 1 部 : 概要」

435 ページの「第 2 部 : サービスログ (メッセージストア、Administration Server、MTA)」

446 ページの「第 3 部 : サービスログ (MTA)」

第 1 部 : 概要

Messaging Server ログファイルの作成と管理のためにポリシーをカスタマイズすることができます。この章では、ログファイルの種類と構造、およびログファイルの管理と表示方法について説明します。この章には、以下の節があります。

ログ記録されるサービス

Messaging Server は、サポートしている主なプロトコル (サービス) ごとに一連のログファイルを作成します。各種類のログファイルは、個別にカスタマイズしたり表示することができます。表 13-1 に、ログ記録が可能なサービスのリストとそれぞれのログファイルに関する説明を示します。

表 13-1　ログ記録されるサービス

サービス	ログファイルの説明
Admin	Administration Server を介した iPlanet Console と Messaging Server 間の通信 (大半は複数の CGI プロセスを経る) に関連するログイベントが記録されます。
SMTP	サーバの SMTP アクティビティに関連するログイベントが記録されます。
IMAP	サーバの IMAP4 アクティビティに関連するログイベントが記録されます。
POP	サーバの POP3 アクティビティに関連するログイベントが記録されます。
HTTP	サーバの HTTP アクティビティに関連するログイベントが記録されます。
デフォルト	サーバのその他のアクティビティ (コマンドラインユーティリティやその他のプロセスなど) に関連するログイベントが記録されます。

サードパーティ製のツールを使ってログを解析する

iPlanet Messaging Server ではサポートされていないログ解析やレポート生成を行うには、別のツールを使用する必要があります。ログファイルは、テキストエディタや標準のシステムツールで操作できます。

正規表現による構文解析をサポートするスクリプト可能なテキストエディタを使用すると、この章で説明しているような特定の条件に基づくログエントリの検索や抽出を行い、その結果を並べ替えたり、集計や統計を行うこともできます。

UNIX 環境では、UNIX の `syslog` ファイルを操作するために開発された既存のレポート生成ツールを変更して使用することもできます。パブリックドメインの `syslog` 操作ツールを使用する場合は、そのツールにおいて、日付 / 時刻形式と、Messaging Server のログエントリにはあって `syslog` エントリにはない 2 つの特殊コンポーネント (*facility* と *logLevel*) の変更が必要になる場合があります。

第 2 部 : サービスログ (メッセージストア、Administration Server、MTA)

ここでは、POP、IMAP、HTTP、MTA、Admin、および Default (表 13-1 を参照) の各サービスのログについて説明します。

これらのサービスの場合、iPlanet Console を使用してログの設定と表示を行うことができます。設定内容は、どのイベントを何件まで記録するかに影響します。これらの設定とその他の特徴を使用して、ログファイル解析時のログイベントの検索条件を微調整することができます。MTA のサービスログの詳細は、**446 ページの「第 3 部 : サービスログ (MTA)」**を参照してください。

第 2 部には以下の節があります。

ログの特徴

ここでは、メッセージストアと管理サービスに関するログの特徴 (ログレベル、ログイベントのカテゴリ、ログファイル名の命名規則、ログファイルのディレクトリ) について説明します。

ログレベル

ログのレベル (優先順位) は、ログのアクティビティの詳細度を定義します。優先順位レベルが高いほど、詳細度は低くなります。優先順位 (重要度) の高いイベントだけがログに記録されるためです。レベルを下げると、ログは詳細なものとなり、より多くのイベントがログファイルに記録されます。

ログレベルは、`logfile.`*service*`.loglevel` 設定パラメータを設定することによって、POP、IMAP、HTTP、Admin、および Default の各サービスごとに個別に設定できます (440 ページの「ログオプションを定義、設定する」を参照)。また、ログレベルを使用して、ログイベントを検索するときにフィルタリングすることもできます。表 13-2 に、利用可能なレベルを示します。これらのログレベルは、UNIX の `syslog` 機構で定義されるログレベルのサブセットです。

表 13-2 メッセージストアと管理サービスのログレベル

レベル	説明
Critical	もっとも詳細度の低いログ。メールボックスや実行に必要なライブラリにサーバがアクセスできない場合など、サーバに重大な問題や致命的な状態が発生したときに、イベントがログに記録されます。
Error	クライアントまたは他のサーバへの接続試行に失敗した場合など、エラー状態が発生したときに、イベントがログに記録されます。
Warning	サーバがクライアントから送られた通信を解釈できない場合など、警告状態が発生したときに、イベントがログに記録されます。
Notice	ユーザがログインに失敗したり、セッションが終了した場合など、通知 (通常の状態だが重要な状況) が発生したときに、イベントがログに記録されます。
Information	ユーザがログオンやログオフを行ったり、メールボックスを作成したり名前を変更した場合など、重要なアクションが行われたときに、イベントがログに記録されます。
Debug	もっとも詳細度の高いログ。デバッグを行う場合のみ役立ちます。各プロセスまたはタスク内の個々のステップごとにイベントがログに記録されるため、問題の箇所を正確に突き止めることができます。

特定のログレベルを選択すると、そのレベルのイベントとそれ以上のレベル (詳細度の低い) のイベントがログに記録されます。デフォルトのログレベルは、`Notice` です。

注 より詳細なログを指定するほど、ログファイルがより多くのディスク容量を占有することになります。ガイドラインについては、440 ページの「ログオプションを定義、設定する」を参照してください。

ログイベントのカテゴリ

サポートされているサービスまたはプロトコル内で、Messaging Server は、どの機能領域で発生したかに基づいて、ログイベントをより細かくカテゴリに分類します。各ログイベントには、それを生成した機能領域の名前が含まれています。これらのカテゴリは、イベントを検索する際のフィルタリングに使用できます。表 13-3 に、Messaging Server がログのために認識するカテゴリのリストを示します。

表 13-3 ログイベントの発生場所のカテゴリ

機能領域	説明
General	プロトコルまたはサービスに関連するアクション全般
LDAP	LDAP ディレクトリデータベースにアクセスする Messaging Server に関連するアクション
Network	ネットワークの接続に関連するアクション (ソケットエラーはこのカテゴリに分類される)
Account	ユーザアカウントに関連するアクション (ユーザログインはこのカテゴリに分類される)
Protocol	プロトコル固有のコマンドに関連するプロトコルレベルのアクション (POP、IMAP、または HTTP 機能によって返されるエラーはこのカテゴリに分類される)
Stats	サーバの統計収集に関連するアクション
Store	メッセージストアへのアクセスに関連する低レベルのアクション (読み取り / 書き込みエラーはこのカテゴリに分類される)

ログ検索でカテゴリをフィルタとして使用する場合の例については、444 ページの「ログを検索、表示する」を参照してください。

メッセージストアと管理サービスのログファイル名の命名規則

POP、IMAP、HTTP、Admin、および Default サービスのログファイルには、同一の命名規則が適用されます。各ログファイル名の形式は、以下のとおりです。

service.sequenceNum.timeStamp

表 13-4 に、メッセージストアのログファイル名の命名規則を示します。

表 13-4　メッセージストアと管理サービスのログファイル名の命名規則

コンポーネント	定義
service	ログ対象のサービス : POP、IMAP、HTTP、Admin、Default。
sequenceNum	ログファイルディレクトリ内に作成されたログファイルの順番を表す整数。新しいログファイルほど、値が大きくなります。シーケンス番号はロールオーバーすることはなく、サーバのインストール時に始まり、そのサーバを使用している限り常に増え続けます。
timeStamp	ファイルが作成された日付と時刻を示す整数。この値は UNIX 標準の時刻形式で表されます。つまり、1970 年 1 月 1 日午前 0 時から経過した秒数です。

たとえば、`imap.63.915107696` という名前のログファイルは、IMAP ログファイルのディレクトリで 63 番目に作成されたログファイルであり、1998 年 12 月 31 日午後 12 時 34 分 56 秒に作成されたログファイルです。

無制限のシーケンス番号をタイムスタンプと組み合わせることによって、解析するファイルのローテーション、有効期間、および選択がより柔軟になります。詳細については、440 ページの「ログオプションを定義、設定する」を参照してください。

ログファイルのディレクトリ

ログ記録される各サービスごとに、1 つのディレクトリが割り当てられ、ログファイルはそこに保存されます。IMAP ログファイルや POP ログファイルなどの各サービスのログファイルは、それぞれのディレクトリ内に一緒に保存されます。各ディレクトリの場所、そのディレクトリ内に保存できるログファイルの数、およびファイルのサイズを設定することができます。

すべてのログファイルを保存するのに十分な容量があることを確認してください。ログレベルが低い (詳細度が高い) ほど、ログファイルのサイズは大きくなります。

ログレベル、ログローテーション、ログの有効期間、およびサーバのバックアップポリシーを正しく定義することが重要です。ログファイルディレクトリのすべてがバックアップされ、また、過負荷にならないようにするためです。これらを正しく定義しないと、情報を失ってしまうことがあります。440 ページの「ログオプションを定義、設定する」を参照してください。

ログファイルの形式

Messaging Server によって作成されたメッセージストアおよび管理サービスのログファイルのコンテンツの形式は、すべて同じです。ログファイルは複数行のテキストファイルであり、各行に 1 つのログイベントが記述されています。サポートされている各サービスに対するすべてのイベントは、通常は以下のような形式で記述されています。

dateTime hostName processName[pid]: category logLevel: eventMessage

表 13-5 に、ログファイルのコンポーネントを示します。このイベント記述形式は、日付 / 時刻形式が異なることと追加コンポーネント (*category* と *logLevel*) があることを除けば、UNIX の `syslog` 機構で定義されているものと同じです。

表 13-5 メッセージストアと管理サービスのログファイルのコンポーネント

コンポーネント	定義
dateTime	イベントがログ記録された日付と時刻。*dd/mm/yyyy hh:mm:ss* の形式で表記されます。時間帯フィールドは GMT を基準とした *+/-hhmm* で表記されます。たとえば、以下のようになります。 `02/Jan/1999:13:08:21 -0700`
hostName	サーバが動作しているホストマシンの名前。たとえば、`showshoe`。 **注 :** ホスト上に複数の Messaging Server インスタンスがある場合は、プロセス ID (pid) を使用して、ログイベントのインスタンスを区別することができます。
processName	イベントを生成したプロセスの名前。たとえば、`cgi_store`。
pid	イベントを生成したプロセスのプロセス ID。たとえば、`18753`。
category	イベントが属するカテゴリ。たとえば、`General` (437 ページの表 13-3 を参照)。
logLevel	イベントのログレベル。たとえば、`Notice` (436 ページの表 13-2 を参照)
eventMessage	イベント固有の説明メッセージで、長さは任意。たとえば、`Log created (894305624)`

以下に、iPlanet Console を使って表示したログイベントの例を示します。

```
02/May/1998:17:37:32 -0700 showshoe cgi_store[18753]:
General Notice:
  Log created (894155852)

04/May/1998:11:07:44 -0400 xyzmail cgi_service[343]:General Error:
  function=getserverhello|port=2500|error=failed to connect

03/Dec/1998:06:54:32 +0200 SiroePost imapd[232]:Account Notice:
  close [127.0.0.1] [unauthenticated] 1998/12/3 6:54:32
  0:00:00 0 115 0
```

IMAP および POP のイベントエントリの末尾は、3 つの数になることがあります。上の例では 0 115 0 です。最初の数字はクライアントによって送信されたバイト数、2 番目の数字はサーバによって送信されたバイト数、3 番目の数字は選択されたメールボックス (POP の場合は常に 1) です。

ログファイルを「ログビューア」ウィンドウに表示するときは、特定のログレベルやカテゴリ、または特定のプロセス ID などのイベント内の特定のコンポーネントを検索することによって、表示するイベントを制限することができます。詳細は、444 ページの「ログを検索、表示する」を参照してください。

各ログエントリのイベントメッセージの形式は、ログに記録されるイベントの種類に固有のものです。つまり、各サービスごとに、イベントメッセージに含まれるコンテンツが定義されます。多くのイベントメッセージは単純で明白なものですが、複雑なものもあります。

ログオプションを定義、設定する

メッセージストアおよび管理サービスのログ設定は、管理者のニーズに合わせて定義することができます。ここでは、最適な設定とポリシーを決定するために役立つ情報と、それらの適用方法を説明します。

柔軟なログ構造

ログファイルの名前の形式 (*service.sequenceNum.timeStamp*) により、柔軟なログローテーションとバックアップポリシーを設計することができます。イベントはサービスごとに別のファイルに記録されるため、問題をすばやく簡単に隔離することができます。また、ファイル名の中のシーケンス番号は常に増え続け、タイムスタンプは常に一意であるため、指定したシーケンス番号の限界に達しても、新しいログファイルが古いログファイルを単純に上書きしてしまうことはありません。古いログファイルの上書きや削除が行われるのは、ログファイルの保存期間や最大数、合計ログ容量など、より柔軟性のある制限がその限界に達したときだけです。

Messaging Server では、管理やバックアップを簡素化できるように、ログファイルの自動ローテーションがサポートされています。後続のログイベントを記録するために、手動で現在のログファイルを回収して新しいログファイルを作成する必要はありません。現在のログファイル以外、ディレクトリ内にあるものはすべて、サーバを停止したり、新しいログファイルの作成をサーバに手動で指定しなくても、いつでもバックアップすることができます。

ログポリシーを設定する際に、合計ログ容量、ログファイルの最大数、個々のファイルサイズ、ファイルの最長保存期間、およびログファイルローテーションの頻度といったオプションを、サービスごとに設定することができます。

適切なオプションを決定する

複数の制限を設定する必要があることと、それらの中にはログファイルのローテーションや削除を引き起こすものがあることを理解しておいてください。最初に限界に達する制限が、制御の中心となります。たとえば、ログファイルの最大サイズを 3.5M バイトに設定し、毎日新しいログを作成するように設定したとします。しかし、24 時間以内に 3.5M バイト以上のデータが記録される場合は、1 日に複数のログファイルが作成されることになります。このため、ログファイルの最大数が 10 個、最長保存期間が 8 日に設定されている場合でも、ログのローテーションが早いため、8 日間経過する前に 10 個のファイルが作成され、最長保存期間まで達することはありません。

以下は Messaging Server の管理ログに備えられているデフォルト値であり、適切なオプションを決定する際に役立ちます。

ディレクトリ内のログファイルの最大数 : 10
ログファイルの最大サイズ : 2M バイト
全ログファイルの合計最大サイズ : 20M バイト
最小空きディスク容量 : 5M バイト
ログロールオーバー時間 : 1 日
最長有効期間 : 7 日
ログのレベル : Notice

この設定の場合、サーバ管理ログのデータは 1 日当たり約 2M バイト蓄積され、バックアップは週 1 回作成され、管理ログの保存に割り当てられている合計容量は最低 25M バイトです (ログレベルがより詳細な場合、これらの設定では不十分なことがあります)。

POP、IMAP、または HTTP のログの場合も、同様の設定から始めるとよいでしょう。すべてのサービスのログ容量要件が上記のデフォルト値とほとんど同じである場合、最初は約 150M バイトの合計ログ容量を設定することをお勧めします (ここに示した設定はあくまでも一般例であり、実際の条件はこれとはかなり異なる場合があります)。

ログオプションを設定するには

メッセージストアのログ設定を制御するオプションは、iPlanet Console またはコマンドラインを使用して設定することができます。

これらのオプションの最適な設定は、ログデータの累積される頻度によって異なります。1M バイトの保存領域には、約 4,000 ～ 10,000 件のログエントリを記録できます。適度にビジー状態のサーバでは、ログレベルが低い場合 (Notice など)、週に何百メガバイトものログデータが記録されることもあります。以下の設定を参考にしてください。

- 使用可能な保存領域の上限に合わせてログレベルを設定します。つまり、使用可能な保存領域の上限に基づき、ログデータの累積頻度を考慮してログレベルを判断します。
- 検索処理に影響が出ないように、ログファイルのサイズを設定します。ローテーションのスケジュールと合計保存容量の上限を考慮して調整します。ログエントリの累積頻度に基づいて、最大値を設定してもかまいません。この最大値は、ローテーションが自動的に発生するまでに蓄積されるサイズよりも少し大きめのサイズに設定します。最大ファイルサイズとファイルの最大数を掛けて得られる値が、合計保存領域の上限とほぼ等しくなります。

  たとえば、IMAP ログローテーションが毎日、1 日当たりに累積される IMAP ログデータが 3M バイト、IMAP ログの合計保存領域の上限が 25M バイトの場合、IMAP ログファイルの最大サイズは 3.5M バイトに設定します (この例では、すべてのログファイルが最大サイズと最大ファイル数に達してしまうほど急速にログデータが累積された場合は、いくつかのログデータが失われる可能性があります)。
- サーバのバックアップを週 1 回行い、IMP ログファイルを毎日ローテーションする場合、IMAP ログファイルの最大数を約 10 個 (個々のログサイズの上限を超える場合のローテーション頻度を考慮) と指定し、最長保存期間を 7 日または 8 日に指定します。
- ハードウェアの容量とサーバに対して計画したバックアップスケジュールに基づいて、合計保存領域の上限を設定します。ログデータの累積頻度を予測し、サーバのバックアップ周期を超えないように合計保存容量の上限を少し大きめに設定します。

  たとえば、IMAP ログファイルデータの累積が 1 日平均 3M バイト、サーバのバックアップが週 1 回の場合、ディスクの保存領域が十分にあることを前提として、IMAP ログの記憶領域の上限は 25 ～ 30M バイトに設定します。
- 安全性を確保するため、ログファイルを保存するボリュームに、最小空きディスク容量を設定します。ログファイルサイズ以外の要因によってボリュームがいっぱいになった場合は、いっぱいになったディスクにログデータを書き込もうとして障害が発生する前に、古いログファイルが削除されます。

ログ情報は、サーバが提供するログファイルではなく、syslog 機構に送るように選択することもできます。ログ情報を syslog に送るには、`syslogfacility` オプションを以下のように設定します。

```
configutil -o logfile.service.syslogfacility -v value
```

ここで、*service* は `admin`、`pop`、`imap`、`imta`、または `http` で、*value* は `user`、`mail`、`daemon`、`local0` から `local7`、または none です。

値が設定されると、設定値に対応する syslog 機構のログにメッセージが記録され、その他のすべてのログファイルサービスオプションが無視されます。オプションが設定されていない場合、または値が none の場合、Messaging Server ログファイルが使用されます。

コンソール : iPlanet Console を使用してログオプションを設定するには、以下の手順に従います。

1. ログファイルオプションを設定する Messaging Server を開きます。
2. 「環境設定」タブをクリックし、左側のパネルで「ログファイル」フォルダを開き、サービス (IMAP、HTTP、Admin など) のログファイルを選択します。
3. 「詳細レベル」ドロップダウンリストからログレベルを選択します。
4. 「ログファイルのディレクトリパス」フィールドに、ログファイルの保存先となるディレクトリの名前を入力します。
5. 「各ログのファイルサイズ」フィールドに、ログファイルの最大サイズを入力します。
6. 「新規アクセスログ作成」フィールドに、ログローテーションのスケジュールの値を入力します。
7. 「ディレクトリ当たりのログ数」および「ログが次の日付よりも古い場合」フィールドに、バックアップスケジュールを考慮に入れて、最大ログファイル数と期限を示す値を入力します。
8. 「ログサイズの合計が次の値を超えたとき」フィールドに、合計保存領域の上限を入力します。
9. 「残りディスク容量が次の値以下になった場合」フィールドに、確保しておく空きディスク容量の最小値を入力します。

コマンドライン : コマンドラインでログオプションを設定するには、以下の例のように `configutil` コマンドを使用します。

ログレベルを設定するには、以下のように指定します。

```
configutil -o logfile.service.loglevel -v level
```

ここで、*service* は `admin`、`pop`、`imap`、`imta`、または `http`、*loglevel* は `Nolog`、`Critical`、`Error`、`Warning`、`Notice`、`Information`、または `Debug` です。

ログファイルのディレクトリパスは、以下のように指定します。

```
configutil -o logfile.service.logdir -v dirpath
```

各ログの最大ファイルサイズは、以下のように指定します。

```
configutil -o logfile.service.maxlogfilesize -v size
```

size にはバイト数を指定します。

ログローテーションのスケジュールは、以下のように指定します。

```
configutil -o logfile.service.rollovertime -v number
```

number には秒数を指定します。

ディレクトリ内の最大ログファイル数は、以下のように指定します。

```
configutil -o logfile.service.maxlogfiles -v number
```

保存容量の上限は、以下のように指定します。

```
configutil -o logfile.service.maxlogsize -v number
```

number にはバイト数を指定します。

確保しておく空きディスク容量の最小値は、以下のように指定します。

```
configutil -o logfile.service.minfreediskspace -v number
```

number にはバイト数を指定します。

ログの保存期間は、以下のように指定します。

```
configutil -o logfile.service.expirytime -v number
```

number には秒数を指定します。

ログを検索、表示する

iPlanet Console には、メッセージストアおよび管理サービスに関するログデータを表示するための基本的なインタフェースがあります。個々のログファイルを選択したり、それらのファイル内で柔軟なフィルタリングによる検索を行うことができます。

ログファイルはサービスごとに分かれており、それぞれ作成順に一覧表示されます。検索するログファイルを選択したら、検索パラメータを指定して検索対象を個々のイベントに限定することができます。

検索パラメータ

以下に、表示するログデータを指定するための検索パラメータを示します。

- **期間** : イベントを検索する期間の開始と終了を指定するか、検索する日数 (現時点からさかのぼる日数) を指定します。サーバのクラッシュやその他の問題の原因となったログイベントを調べるために、通常は期間の範囲を指定します。また、現在のログファイルの中で今日のイベントだけを見る場合は、期間を 1 日に指定することもできます。
- **ログのレベル** : ログレベルを指定できます (435 ページの「ログレベル」を参照)。特定の問題を検出するために該当するレベルを選択します。たとえば、サーバがダウンした原因を調べる場合は Critical、失敗したプロトコルコールを検出する場合は Error を選択します。
- **機能領域** : 機能領域を指定できます (437 ページの「ログイベントのカテゴリ」を参照)。問題が含まれている機能領域がわかっている場合は、その機能領域を選択することができます。たとえば、サーバのクラッシュにディスクエラーが関連していると思われる場合は Store、問題が IMAP プロトコルコマンドエラーにあると思われる場合は Protocol を選択します。
- **テキスト検索パターン** : テキスト検索パターンを指定して検索対象を絞ることができます。検索するイベントについてすでにわかっている、イベント時刻、プロセス名、プロセス ID、およびイベントメッセージの一部 (リモートホスト名、関数名、エラー番号など) などのイベントコンポーネント (439 ページの「ログファイルの形式」を参照) を、ワイルドカードを使用して検索することができます。

  検索パターンには、以下の特殊文字およびワイルドカード文字を使用することができます。

  `*` 任意の文字セット (例 : `*.com`)
  `?` 任意の 1 文字 (例 : `199?`)
  [*nnn*] *nnn* の中の任意の文字 (例 : `[aeiou]`)
  [`^`*nnn*] *nnn* 以外の任意の文字 (例 : `[^aeiou]`)
  [*n-m*] *n-m* の範囲内の任意の文字 (例 : `[A-Z]`)
  [`^`*n-m*] *n-m* の範囲外の任意の文字 (例 : `[^0-9]`)
  ¥ エスケープ文字 : `*`、`?`、`[`、または `]` の前に配置してそれらを文字として使用

注 : 検索では大文字と小文字が区別されます。

以下に、ログレベルと機能領域を組み合わせた、表示するログの検索例を示します。

- 失敗したログインを表示するには、Account 機能領域 (および Notice レベル) を指定します。これは、潜在的なセキュリティ違反を調べるときに役立ちます。
- 接続に関する問題を調べるには、Network 機能 (およびすべてのログレベル) を指定します。
- サーバの機能に関する基本的な問題を調べるには、すべての機能 (および Critical ログレベル) を指定します。

検索対象を指定し、結果を表示するには

指定したサービスに属する固有の特徴を持つログイベントを検索するには、以下の手順に従います。

1. iPlanet Console で、調べるログファイルがある Messaging Server を開きます。
2. 以下のいずれかの方法で、指定したサービスログの「ログファイルの内容」タブを表示します。
   - 「タスク」タブをクリックしてから、「サービスログの表示」をクリックします。サービスは、ログに記録されているサービスの名前 (「IMAP サービス」や「管理」など) です。
   - 「環境設定」タブをクリックし、左側のパネルで「ログファイル」フォルダを開き、サービス (IMAP や Admin など) のログファイルを選択します。次に、右側のパネルの「コンテンツ」タブを選択します。
3. ログに記録されたサービスの「コンテンツ」タブが表示されます。
4. 「ログファイル名」フィールドで、調べたいログファイルを選択します。
5. 「選択したログの表示」ボタンをクリックして「ログビューア」ウィンドウを開きます。
6. 「ログビューア」ウィンドウで、検索パラメータを指定します (前述の「検索パラメータ」を参照)。
7. 「更新」をクリックして検索を実行し、「ログエントリ」フィールドに結果を表示します。

第 3 部 : サービスログ (MTA)

MTA は、メッセージがキューに出し入れされるたびにログを作成することができます。また、ディスパッチャエラーとデバッグ出力も生成できます。第 3 部には以下の節があります。

チャネルごとにログを制御したり、すべてのチャネル上のメッセージアクティビティのログを記録するよう指定することができます。初期設定では、すべてのチャネルでのログ記録が無効になっています。

ログを有効にすると、メッセージが MTA チャネルを通過するたびに `mail.log*` ファイルにエントリが書き込まれます。これらのログエントリは、MTA (または特定のチャネル) を通過するメッセージの数の統計を取ったり、メッセージが送信または配信されたかどうか、いつ送信または配信されたかを調べるときに役立ちます。

特定の MTA チャネルを通過するメッセージの数の統計をとるだけであれば、その該当する MTA チャネルだけでログチャネルキーワードを有効にしてもかまいません。ほとんどのサイトでは、すべての MTA チャネルでのログを有効にしています。特に、問題を突き止める場合、問題を診断する最初のステップは、メッセージが意図していたチャネルに送られているかどうかに注目することです。すべてのチャネルに対してログを有効にしておくと、このような問題を調べる際に役立ちます。

警告 ログが有効になっている場合は、`mail.log` が大きくなり続けるため、そのままにしておくと利用可能なディスク容量がなくなってしまいます。このファイルのサイズをモニタし、定期的に不要なコンテンツを削除してください。ファイル全体を削除することもできます。この場合、必要に応じて新しいファイルが作成されます。

MTA のログを有効にするには

特定のチャネルのログを有効にするには、以下のように MTA 設定ファイルのチャネル定義に `logging` キーワードを追加します。

channel-name keyword1 keyword2 `logging`

また、ログファイルやログレベルなどのディレクトリパスのような設定パラメータの数も、設定することができます。435 ページの「第 2 部 : サービスログ (メッセージストア、Administration Server、MTA)」を参照してください。

すべてのチャネルのメッセージアクティビティをログファイルに記録する場合は、MTA 設定ファイルのチャネルブロックセクションの先頭に、defaults チャネルブロックを追加します。たとえば、以下のようになります。

```
defaults logging

l defragment charset7 us-ascii charset8 iso-8859-01
siroe.com
```

`defaults` チャネルは、MTA 設定ファイルの最初の空白行のすぐ後ろにあります。`defaults logging` 行の前後に空白行を指定することが重要です。

メッセージがキューに入ったりキューから取り出されるたびに、メッセージがログに記録されます。ログエントリはすべて、MTA ログディレクトリ (msg-*instance*/log/imta/mail.log_current) にある mail.log_current ファイルに記録されます。

毎晩午前 0 時頃に実行されるメッセージ返送ジョブは、累積されたログファイル mail.log に既存の mail.log_yesterday を追加し、現在の mail.log_current ファイルの名前を mail.log_yesterday に変更してから、新しい mail.log_current ファイルを開始します。connection.log* ファイルに対しても同様の処理が行われます。

LOG_MESSAGES_SYSLOG オプションを 1 に設定して、MTA ログメッセージを syslog (UNIX) またはイベントログ (Windows NT) に送ることができます。0 はデフォルトで、syslog (イベントログ) ログを実行しないことを示します。

その他の MTA ログオプションを指定するには

ログが有効になっているときに与えられる基本的な情報のほかにも、MTA オプションファイルにさまざまな LOG_* MTA オプションを設定することにより、オプションの情報フィールドを含めることができます。オプションファイルの詳細については、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

- LOG_MESSAGE_ID : エントリとメッセージの相関関係を示すことができます。
- LOG_FILENAME : 特定のメッセージファイルの配信試行回数を、即座に確認しやすくなります。また、MTA が複数の受取人宛てのメッセージを、どのような場合に別々のメッセージファイルに分割してディスク上に保存するのかを知る際にも役立ちます。
- LOG_CONNECTION : TCP/IP 接続とメッセージトラフィックのログが記録されます。接続のログエントリは、デフォルトでは mail.log* ファイルに書き込まれますが、connection.log* ファイルに書き込むこともできます。SEPARATE_CONNECTION_LOG オプションを参照してください。
- SEPARATE_CONNECTION_LOG : 接続のログエントリを connection.log ファイルに書き込むように指定する際に使用します。
- LOG_PROCESS : LOG_CONNECTION とともに使用すると、接続エントリとそれに対応するメッセージエントリの相関関係をプロセス ID によって示すことができます。
- LOG_USERNAME : メールをキューに入れるプロセスに関連付けられたユーザ名を mail.log ファイルに保存するかどうかを制御します。SASL (SMTP AUTH) を使用している SMTP 送信の場合は、ユーザ名フィールドが認証ユーザ名 (接頭辞としてアスタリスクが付いたもの) になります。

MTA ログエントリの形式

MTA ログファイルは、ASCII テキストとして記述されます。デフォルトでは、図 13-1 に示すように、各ログファイルエントリに 8 個または 9 個のフィールドがあります。

図 13-1 MTA ログエントリの形式

```
19-Jan-1998 19:16:57.64 l tcp_local E 1 adam@sesta.com
 rfc822;marlowe@siroe.com marlowe@siroe.com
```

ログエントリには以下の情報が含まれています。

1. エントリが記録された日付と時刻。
2. ソースチャネルのチャネル名 (上の例では `l`)。
3. 宛先チャネルのチャネル名 (上の例では `tcp_local`)。SMTP チャネルの場合、`LOG_CONNECTION` が有効になっているときは、プラス記号「+」が SMTP サーバの受信を示し、マイナス記号「-」が SMTP クライアント経由の送信を示します。
4. エントリのタイプ (E)。表 13-6 を参照。
5. メッセージのサイズ (1)。デフォルトではキロバイト単位で表されますが、MTA オプションファイルで `BLOCK_SIZE` キーワードを使用して単位を変更することもできます。
6. エンベロープ From: アドレス (`adam@sesta.com`)。通知メッセージのようにエンベロープ From: アドレスが空のメッセージの場合、このフィールドは空白になります。
7. エンベロープ To: アドレスの元の形式 (`marlowe@siroe.com`)。
8. エンベロープ To: アドレスのアクティブな (現在の) 形式 (`marlowe@siroe.com`)。
9. 配信ステータス (SMTP チャネルのみ)。

表 13-6 に、ログエントリのコードを示します。

表 13-6 ログエントリのコード

エントリ	説明
D	キューからの取り出しに成功
DA	SASL (認証) でのキューからの取り出しに成功
DS	TLS (セキュリティ) でのキューからの取り出しに成功

表 13-6　ログエントリのコード (続き)

エントリ	説明
DSA	TLS および SASL (セキュリティと認証) でのキューからの取り出しに成功
E	エンキュー
EA	SASL (認証) でキューに入れることに成功
ES	TLS (セキュリティ) でキューに入れることに成功
ESA	TLS および SASL (セキュリティと認証) でキューに入れることに成功
J	キューに入れる試行の拒否 (スレーブチャネルプログラムによる拒否)
Q	キューからの取り出しで一時的な失敗
R	キューからの取り出し試行で受取人アドレスの拒否 (マスターチャネルプログラムによる拒否)、または失敗 / 差し戻しメッセージの生成
W	未配信メッセージに関する警告メッセージの生成
Z	数人の受取人に対しては成功したが、この受取人に対しては一時的に失敗。すべての受取人の元のメッセージファイルはキューから取り出され、それに代わって新しいメッセージファイルが入れられ、その他の失敗した受取人がすぐにキューに入れられます。
SMTP チャネルの LOG_CONNECTION + または − エントリ	
C	接続終了
O	接続開始
X	接続拒否
Y	接続が確立される前に試行に失敗
I	ETRN コマンド受信

LOG_CONNECTION、LOG_FILENAME、LOG_MESSAGE_ID、LOG_NOTARY、LOG_PROCESS、および LOG_USERNAME がすべて有効になっている場合、形式は図 13-2 に示されているようになります。この例のログエントリ行は改行されて表示されていますが、実際のログエントリは 1 行で記述されます。

図 13-2　その他のフィールドを含むログ形式

```
19-Jan-1998 13:13:27.10 HOSTA   2e2d.2.1 tcp_local   l
 E 1 service@siroe.com rfc822;adam@sesta.com
 adam 276 /imta/queue/l/ZZ01IWFY9ELGWM00094D.00
 <01IWFVYLGTS499EC9Y@siroe.com> inetmail
 siroe.com (siroe.com [192.160.253.66])
```

前述の説明に含まれていない追加のフィールドは、以下のとおりです。

1. チャネルプロセスを実行しているノードの名前 (例では HOSTA)。
2. プロセス ID (16 進数) と、その後ろに続くピリオド (ドット) 文字とカウント。マルチスレッドのチャネルエントリ (tcp_* チャネルエントリなど) の場合は、プロセス ID とカウントの間にスレッド ID も挿入されています。上の例では、プロセス ID は 2e2d.2.1 です。
3. メッセージの NOTARY (配達証明書要求) フラグ。整数値で表記 (例では 276)。
4. MTA キュー領域内のファイル名 (例では /imta/queue/l/ZZ01IWFY9ELGWM00094D.00)。
5. メッセージ ID (例では <01IWFVYLGTS499EC9Y@siroe.com>)。
6. 実行プロセスの名前 (例では inetmail)。UNIX での SMTP サーバなどのディスパッチャプロセスの場合、通常は inetmail (SASL を使用しなかった場合)。
7. 接続情報 (例では siroe.com (siroe.com [192.160.253.66]))。接続情報は、送信システムが HELO/EHLO 行に示す名前 (受信 SMTP メッセージの場合) や、チャネルの正規のホスト名 (他の種類のチャネルの場合) など、送信システムまたはチャネル名で構成されています。TCP/IP チャネルの場合、送信システムの「実際の」名前、つまり、DNS リバース検索によってレポートされるシンボリック名や IP アドレスは、ident* チャネルキーワードを使用して括弧内にレポートすることもできます。229 ページの「IDENT 検索」を参照してください。この例では、DNS によって見つかった名前と IP アドレスの両方を表示するように指定するキーワードの 1 つ (たとえば、デフォルトの identnone キーワード) が使用されていると仮定しています。

MTA ログファイルを管理する

毎晩午前 0 時頃に実行されるメッセージ返送ジョブは、累積されたログファイル mail.log に既存の mail.log_yesterday を追加し、現在の mail.log_current ファイルの名前を mail.log_yesterday に変更してから、新しい mail.log_current ファイルを開始します。connection.log* ファイルに対しても同様の処理が行われます。

MTA は自動的にロールオーバーを実行して現在のファイルを維持しますが、エントリが累積される mail.log ファイルは、ファイルのバックアップ、切り捨て、削除などのタスクのポリシーを決めて管理する必要があります。

ログファイルの管理方法を検討するときは、MTA の定期的な返送ジョブが、サイトが提供する *server-instance*/imta/bin/daily_cleanup プロシージャ (存在する場合) を実行することに注意してください。このため、サイトによっては独自のクリーンアップ方法を提供していることもあります。たとえば、古い mail.log ファイルの名前を週に 1 回 (または月に 1 回) 変更するなどです。

MTA メッセージログの例

MTA メッセージファイルにログ記録されるフィールドの形式とフィールドのリストは、設定したログオプションによって異なります。ここでは、いくつかの典型的なログエントリの解釈の例を示します。その他のオプションのフィールドについては、448 ページの「その他の MTA ログオプションを指定するには」を参照してください。

注	ここではログファイルエントリが複数行にわたって表示されていますが、実際のログファイルエントリは 1 行で記述されます。

ログファイルを確認するときは、通常のシステムでは一度に多くのメッセージが処理されていることに留意してください。通常、特定のメッセージに関連するエントリは、同時に処理されているその他のメッセージに関連するエントリの間に散らばっています。基本的なログ情報は、MTA を通過するメッセージの数が全部でいくつあるかを把握するのに役立ちます。

同じ受取人への同じメッセージに関連する特定のエントリを関連付ける場合は、LOG_MESSAGE_ID を有効にします。特定のメッセージを MTA キュー領域にある特定のファイルと関連付けたり、エントリを見てキューからの取り出しに成功していない特定のメッセージの配信を何回試行したかを確認する場合は、LOG_FILENAME を有効にします。SMTP メッセージ (TCP/IP チャネル経由で処理されるメッセージ) の場合、リモートシステムとの TCP 接続を送信メッセージと関連付けるには、LOG_PROCESS と何らかのレベルの LOG_CONNECTION を有効にします。

図 13-3 に、ローカルユーザが送信 TCP/IP チャネルからインターネットなどにメッセージを送信する場合に見られる、基本的なログエントリの例を示します。この例では、LOG_CONNECTION が有効になっています。(1) と (2) の行は 1 つのエントリで、実際のログファイルでは 1 行で記述されます。同様に、(3) 〜 (7) の行も 1 つのエントリで、実際のログファイルでは 1 行で記述されます。

図 13-3 ログ : ローカルユーザが送信メッセージを送った場合

```
19-Jan-1998 19:16:57.64 l             tcp_local    E 1 (1)
 adam@sesta.com rfc822;marlowe@siroe.com marlowe@siroe.com (2)

19-Jan-1998 19:17:01.16 tcp_local                  D 1 (3)
 adam@sesta.com rfc822;marlowe@siroe.com marlowe@siroe.com (4)
 dns;thor.siroe.com
 (TCP|206.184.139.12|2788|192.160.253.66|25) (5)
 (THOR.SIROE.COM -- Server ESMTP [iMS V5.0 #8694]) (6)
 smtp;250 2.1.5 marlowe@siroe.com and options OK. (7)
```

1. この行は、ブロックメッセージ (1) をチャネル 1 からチャネル tcp_local のキューに入れたときの日付と時刻 (E) を示します。
2. この部分は、実際にはログファイルでは (1) と同じ行に表示されます。ここでは印刷上の理由から改行されています。エンベロープ From: アドレス (この例では adam@sesta.com) と、エンベロープ To: アドレスの元のバージョンと現在のバージョン (この例では marlowe@siroe.com) を示しています。
3. ブロックメッセージ (1) を tcp_local チャネルのキューから取り出したときの日付と時刻 (D) を示しています。つまり、tcp_local チャネルがリモートの SMTP サーバへの送信に成功したことを示しています。
4. エンベロープ From: アドレス、元のエンベロープ To: アドレス、および現在の形式のエンベロープ To: アドレスを示しています。
5. 接続先の実際のシステムの名前が DNS で thor.siroe.com であること、ローカルの送信システムの IP アドレスが 206.184.139.12 で、ポート 2788 から送信されていること、リモートの宛先システムの IP アドレスが 192.160.253.66 で、接続ポートが 25 であることを示しています。
6. リモートの SMTP サーバの SMTP 見出し行を示しています。
7. このアドレスに返された SMTP ステータスコードを示しています。250 は基本的な SMTP 成功コードであり、このリモート SMTP サーバは拡張 SMTP ステータスコードと追加テキストで応答しています。

図 13-4 は図 13-3 に示されているログエントリと似ていますが、LOG_FILENAME=1 および LOG_MESSAGE_ID=1 を設定することによって、ファイル名とメッセージ ID を含む追加の情報もログ記録されています。(1) と (2) を参照してください。特に、メッセージ ID は、エントリとそれに関連するメッセージの相関関係を示すために使われます。

図 13-4 ログ : オプションのログフィールドを含む場合

```
19-Jan-1998 19:16:57.64 l            tcp_local    E 1
 adam@sesta.com rfc822;marlowe@siroe.com marlowe@siroe.com
 /imta/queue/tcp_local/ZZ01ISKLSKLZLI90N15M.00
   <01ISKLSKC2QC90N15M@sesta.com> (1)

19-Jan-1998 19:17:01.16 tcp_local                 D 1
 adam@sesta.com rfc822;marlowe@siroe.com marlowe@siroe.com
 /imta/queue/tcp_local/Z01ISKLSKLZLI90N15M.00
   <01ISKLSKC2QC90N15M@sesta.com> (2)
 dns;thor.siroe.com (TCP|206.184.139.12|2788|192.160.253.66|25)
 (THOR.SIROE.COM -- Server ESMTP [iMS V5.0 #8694])
 smtp;250 2.1.5 marlowe@siroe.com and options OK.
```

図 13-5 は、LOG_FILENAME=1、LOG_MESSAGE_ID=1、および LOG_CONNECTION=1 を有効にして、複数の受取人に送信する例を示しています。ここでは、ユーザ adam@sesta.com が MTA メーリングリスト test-list@sesta.com に送信し、それが bob@sesta.com、carol@varrius.com、および david@varrius.com に展開されています。それぞれの受取人の元のエンベロープ To: アドレスはすべて test-list@sesta.com ですが、現在のエンベロープ To: アドレスはそれぞれの受取人ごとに異なるアドレスであることに注意してください。2 つのファイル (チャネル l と送信チャネル tcp_local 用) がありますが、メッセージ ID は同じです。

図 13-5 ログ : リストに送信する場合

```
19-Jan-1998 20:01:44.10 l           l           E 1
 adam@sesta.com rfc822;test-list@sesta.com bob
 imta/queue/l/ZZ01ISKND3DE1K90N15M.00
 <01ISKND2H8MS90N15M@sesta.com>

19-Jan-1998 20:01:44.81 l           tcp_local   E 1
 adam@sesta.com rfc822;test-list@sesta.com carol@varrius.com
 imta/queue/tcp_local/ZZ01ISKND2WS1I90N15M.00
 <01ISKND2H8MS90N15M@sesta.com>

19-Jan-1998 20:01:44.81 l           tcp_local   E 1
 adam@sesta.com rfc822;test-list@sesta.com david@varrius.com
 imta/queue/tcp_local/ZZ01ISKND2WS1I90N15M.00
 <01ISKND2H8MS90N15M@sesta.com>

19-Jan-1998 20:01:50.69 l                       D 1
 adam@sesta.com rfc822;test-list@sesta.com bob
 imta/queue/l/ZZ01ISKND3DE1K90N15M.00
 <01ISKND2H8MS90N15M@sesta.com>

19-Jan-1998 20:01:57.36 tcp_local               D 1
 adam@sesta.com rfc822;test-list@sesta.com carol@varrius.com
 imta/queue/tcp_local/ZZ01ISKND2WS1I90N15M.00
 <01ISKND2H8MS90N15M@sesta.com>
 dns;gw.varrius.com (TCP|206.184.139.12|2788|192.160.253.66|25)
 (gw.varrius.com -- SMTP Sendmail)
 smtp;250 OK.

19-Jan-1998 20:02:06.14 tcp_local               D 1
 adam@sesta.com rfc822;test-list@sesta.com david@varrius.com
 imta/queue/tcp_local/ZZ01ISKND2WS1I90N15M.00
 <01ISKND2H8MS90N15M@sesta.com>
 dns;gw.varrius.com (TCP|206.184.139.12|2788|192.160.253.66|25)
 (gw.varrius.com -- SMTP Sendmail)
 smtp;250 OK.
```

図 13-6 は、存在しないドメイン (ここでは very.bogus.com) に送信しようとしたことを示しています。つまり、存在しないことが MTA の書き換え規則によって通知されないドメイン名であり、また、送信 TCP/IP チャネルに一致するドメイン名に送信しようとしました。この例では、LOG_FILENAME=1 と LOG_MESSAGE_ID=1 という MTA オプションが設定されていると仮定しています。

TCP/IP チャネルが作動していて、DNS のドメイン名をチェックしているとき、DNS はそのような名前は存在しないというエラーを返します。(5) の「拒否」エントリ (R) のように DNS はエラーを返し、(6) のようにドメイン名が不正であることを示します。

メッセージが発行されたあとでアドレスが拒否されたため、MTA は元の差出人への返送メッセージを生成します。MTA は新しい拒否メッセージを元の差出人のキューに入れ (1)、元の送信メッセージを削除する ((5) の R エントリ) 前にポストマスターにコピーを送信します (4)。

(2) と (8) に示すように、返送メッセージなどの通知メッセージには空のエンベロープ From: アドレスがあります。エンベロープ From: フィールドは空白で示されています。MTA が生成した返送メッセージが最初にキューに入れられることにより、新しい通知メッセージのメッセージ ID の後ろに元のメッセージのメッセージ ID が表示されます (3)。(この情報は MTA で常に利用できるわけではありませんが、利用できる場合は、失敗した送信メッセージに対応するログエントリを、通知メッセージに対応するログエントリに関連付けることができます。) この通知メッセージは、プロセスチャネルのキューに入れられたあと、該当する宛先チャネルのキューに入れられます (7)。

図 13-6 ログ : 存在しないドメインに送信する場合

```
19-JAN-1998 20:49:04 l            tcp_local    E 1
 adam@sesta.com rfc822;user@very.bogus.com user@very.bogus.com
 imta/queue/tcp_local/ZZ01ISKP0S0LVQ94DU0K.00
 <01ISKP0RYMAS94DU0K@SESTA.COM>

19-JAN-1998 20:49:33 tcp_local    process      E 1 (1)
 rfc822;adam@sesta.com adam@sesta.com  (2)
 imta/queue/process/ZZ01ISKP0S0LVQ94DTZB.00
 <01ISKP22MW8894DTAS@SESTA.COM>,<01ISKP0RYMAS94DU0K@SESTA.COM>
(3)

19-JAN-1998 20:49:33 tcp_local    process      E 1 (4)
 rfc822;postmaster@sesta.com postmaster@sesta.com
 imta/queue/process/ZZ01ISKP0S0LVQ94DTZB.00
 <01ISKP22MW8894DTAS@SESTA.COM>,<01ISKP0RYMAS94DU0K@SESTA.COM>

19-JAN-1998 20:50:07 tcp_local                 R 1 (5)
 adam@sesta.com rfc822;user@very.bogus.com user@very.bogus.com
 imta/queue/tcp_local/ZZ01ISKP0S0LVQ94DU0K.00
 <01ISKP0RYMAS94DU0K@SESTA.COM>
 Illegal host/domain name found (6)

19-JAN-1998 20:50:08 process      l            E 3 (7)
 rfc822;adam@sesta.com adam (8)
 imta/queue/l/ZZ01ISKP23BUQS94DTYL.00
 <01ISKP22MW8894DTAS@SESTA.COM>

19-JAN-1998 20:50:08 process      l            E 3
 rfc822;postmaster@sesta.com postmaster
 imta/queue/l/ZZ01ISKP23BUQS94DTYL.00
 <01ISKP22MW8894DTAS@SESTA.COM>

19-JAN-1998 20:50:12 l                         D 3
 rfc822;adam@sesta.com adam
 imta/queue/l/ZZ01ISKP23BUQS94DTYL.00
 <01ISKP22MW8894DTAS@SESTA.COM>

19-JAN-1998 20:50:12 l                         D 3
 rfc822;postmaster@sesta.com postmaster
 imta/queue/l/ZZ01ISKP23BUQS94DTYL.00
 <01ISKP22MW8894DTAS@SIROE.COM>
```

図 13-7 は、リモートシステムの不正アドレスに送信しようとした場合の例を示しています。この例では、LOG_FILENAME=1 および LOG_MESSAGE_ID=1 という MTA オプションと、LOG_BANNER=1 および LOG_TRANSPORTINFO=1 というチャネルオプションが設定されていると仮定しています。(1) の拒否エントリ (R) に注意してください。図 13-6 の拒否エントリとは異なり、この例の拒否エントリではリモートシステムに接続されたことが示されており、また、(2)、(3) にリモート SMTP サーバが発行した SMTP エラーコードが示されています。(2) に示されている情報は、LOG_BANNER=1 および LOG_TRANSPORTINFO=1 というチャネルオプションが設定されていることを前提としています。

図 13-7 ログ : 存在しないリモートユーザに送信する場合

```
20-JAN-1998 13:11:05 l            tcp_local    E 1
 adam@sesta.com rfc822;nonesuch@siroe.com nonesuch@siroe.com
 imta/queue/tcp_local/ZZ01ISLNBB1JOE94DUWH.00
 <01ISLNBAWV3094DUWH@sesta.com>

20-JAN-1998 13:11:08 tcp_local    process      E 1
 rfc822;adam@sesta.com adam@sesta.com
 imta/queue/process/ZZ01ISLNBB1JOE94DSGB.00
 <01ISLNBFKIDS94DUJ8@sesta.com>,<01ISLNBAWV3094DUWH@sesta.com>

20-JAN-1998 13:11:08 tcp_local    process      E 1
 rfc822;postmaster@sesta.com postmaster@sesta.com
 imta/queue/process/ZZ01ISLNBB1JOE94DSGB.00
 <01ISLNBFKIDS94DUJ8@sesta.com>,<01ISLNBAWV3094DUWH@sesta.com>

20-JAN-1998 13:11:11 tcp_local                 R 1     (1)
 adam@sesta.com rfc822;nonesuch@siroe.com nonesuch@siroe.com
 imta/queue/tcp_local/ZZ01ISLNBB1JOE94DUWH.00
 <01ISLNBAWV3094DUWH@sesta.com>
 dns;thor.siroe.com
 (TCP|206.184.139.12|2788|192.160.253.66|25)          (2)
 (THOR.SIROE.COM -- Server ESMTP [iMS V5.0 #8694])
 smtp; 553 unknown or illegal user:nonesuch@siroe.com  (3)

20-JAN-1998 13:11:12 process      l            E 3
 rfc822;adam@sesta.com adam
 imta/queue/l/ZZ01ISLNBGND1094DQDP.00
 <01ISLNBFKIDS94DUJ8@sesta.com>

20-JAN-1998 13:11:12 process      l            E 3
 rfc822;postmaster@sesta.com postmaster
 imta/queue/l/ZZ01ISLNBGND1094DQDP.00
 <01ISLNBFKIDS94DUJ8@sesta.com>

20-JAN-1998 13:11:13 l                         D 3
 rfc822;adam@sesta.com adam@sesta.com
 imta/queue/l/ZZ01ISLNBGND1094DQDP.00
 <01ISLNBFKIDS94DUJ8@sesta.com>

20-JAN-1998 13:11:13 l                         D 3
 rfc822;postmaster@sesta.com postmaster@sesta.com
 imta/queue/l/ZZ01ISLNBGND1094DQDP.00
 <01ISLNBFKIDS94DUJ8@sesta.com>
```

図 13-8 は、MTA がリモート側のメッセージ送信の試行を拒否した場合のログエントリを示しています。(この例では、有効になっている LOG_* オプションがないと仮定されているため、基本的なフィールドだけがエントリにログ記録されています。LOG_CONNECTION オプションを有効にすると、J エントリなどにその他の情報フィールドが追加されます。) この例は、ORIG_SEND_ACCESS マッピングを使って SMTP リレーブロッキング (322 ページの「SMTP リレーブロッキングを設定する」を参照) が設定されている MTA の場合の例です。

```
ORIG_SEND_ACCESS

! ...numerous entries omitted...
!
   tcp_local|*|tcp_local|*   $NRelaying$ not$ permitted
```

alan@very.bogus.com は内部アドレスではありません。したがって、リモートユーザ harold@varrius.com が MTA システムを介してリモートユーザ alan@very.bogus.com にリレーしようとしても、拒否されます。

図 13-8 ログ : リモート側のメッセージ送信試行が拒否される場合

```
28-May-1998 12:02:23 tcp_local                J 0      (1)
 harold@varrius.com rfc822; alan@very.bogus.com        (2)
 550 5.7.1 Relaying not permitted: alan@very.bogus.com (3)
```

1. このログは、MTA がリモート側のメッセージ送信の試行を拒否した日付と時刻を示しています。拒否は J レコードで示されています。(図 13-6 と図 13-7 で示されているように、MTA チャネルがメッセージを送信しようとして拒否され、それが R レコードで示されています。)
2. 試行されたエンベロープ From: アドレスと To: アドレスが示されています。この場合、利用できる元のエンベロープ To: 情報がなかったため、フィールドは空です。
3. このエントリには、MTA がリモート (試行した差出人) 側に発行した SMTP エラーメッセージが含まれています。

図 13-9 に、メッセージを最初の試行で配信できなかったために、MTA が何度もメッセージを送信しようとする場合のログエントリの例を示します。この例では、LOG_FILENAME=1 と LOG_MESSAGE_ID=1 というオプションが設定されていると仮定しています。

図 13-9 ログ : 配信試行が複数回行われた場合

```
15-Jan-1998 10:31:05.18 tcp_internal   tcp_local  E 3 (1)
 adam@hosta.sesta.com rfc822;user@some.org user@some.org
 imta/queue/tcp_local/ZZ01IS3D2ZP7FQ9UN54R.00
 <01IRUD7SVA3Q9UN2D4@sesta.com>

15-Jan-1998 10:31:10.37 tcp_local                 Q 3 (2)
 adam@hosta.sesta.com rfc822;user@some.org user@some.org
 imta/queue/tcp_local/ZZ01IS3D2ZP7FQ9UN54R.00         (3)
 <01IRUD7SVA3Q9UN2D4@sesta.com>
 TCP active open:Failed connect()    Error: no route to host (4)

 ...several hours worth of entries...

15-Jan-1998 12:45:39.48 tcp_local                 Q 3 (5)
 adam@hosta.sesta.com rfc822;user@some.org user@some.org
 imta/queue/tcp_local/ZY01IS3D2ZP7FQ9UN54R.00         (6)
 <01IRUD7SVA3Q9UN2D4@sesta.com>
 TCP active open:Failed connect()    Error:no route to host

 ...several hours worth of entries...

15-Jan-1998 16:45:24.72 tcp_local                 Q 3
 adam@hosta.sesta.com rfc822;user@some.org user@some.org
 imta/queue/tcp_local/ZX01IS67NY4RRK9UN7GP.00         (7)
 <01IRUD7SVA3Q9UN2D4@sesta.com>
 TCP active open:Failed connect()    Error: connection refused (8)

 ...several hours worth of entries...

15-Jan-1998 20:45:51.55 tcp_local                 D 3 (9)
 adam@hosta.sesta.com rfc822;user@some.org user@some.org
 imta/queue/tcp_local/ZX01IS67NY4RRK9UN7GP.00
 <01IRUD7SVA3Q9UN2D4@sesta.com>
 dns;host.some.org (TCP|206.184.139.12|2788|192.1.1.1|25)
 (All set, fire away)
 smtp; 250 Ok
```

1. メッセージは tcp_internal チャネルに入ります。これは、おそらく POP または IMAP クライアント、または SMTP リレーとして MTA を使用している組織内の別のホストから来たものです。MTA はこれを、送信 tcp_local チャネルのキューに入れます。

2. 最初の配信試行に失敗しています。これは Q エントリで示されています。
3. これが最初の配信試行であることは、ZZ* ファイル名からわかります。
4. この配信試行は、TCP/IP パッケージがリモート側への経路を見つけられなかったために失敗しました。図 13-6 とは異なり、DNS は宛先ドメイン名 some.org を否定しません。「no route to host」というエラーは、送信側と受信側の間にネットワーク上の問題があることを示しています。
5. MTA の定期的なジョブの次の実行時に、配信が再試行され、再び失敗しています。
6. ここでファイル名が ZY* になり、2 回目の試行であることを示しています。
7. ファイル名が ZX* になり、3 回目の失敗した試行であることを示しています。
8. 定期的なジョブが配信を再試行し、再び失敗しています。ただし、ここでは TCP/IP パッケージがリモートの SMTP サーバに接続できなかったことが示されているのではなく、リモートの SMTP サーバが接続を受け入れないことを示しています。(リモート側のネットワーク上の問題は解決されても、SMTP サーバをまだ起動していない、またはその SMTP サーバのメッセージ処理が追いつかないなどの理由で、MTA が接続しようとした時点で接続が受け入れられなかったことが考えられます。)
9. メッセージがキューから取り出されています。

図 13-10 に、メッセージが変換チャネルを通過する場合の例を示します。このサイトには、以下のような CONVERSIONS マッピングテーブルがあると仮定しています。

```
CONVERSIONS

  IN-CHAN=tcp_local;OUT-CHAN=l;CONVERT   Yes
```

この例では、LOG_FILENAME=1 と LOG_MESSAGE_ID=1 というオプションが設定されていると仮定しています。

図 13-10 ログ : 変換チャネルを通過する受信 SMTP メッセージ

```
04-Feb-1998 00:06:26.72 tcp_local    conversion   E 9 (1)
 amy@siroe.edu rfc822;bert@sesta.com bert@sesta.com
 imta/queue/conversion/ZZ01IT5UAMZ4QW98518O.00
 <01IT5UALL14498518O@siroe.edu>

04-Feb-1998 00:06:29.06 conversion   l            E 9 (2)
 amy@siroe.edu rfc822;bert@sesta.com bert
 imta/queue/l/ZZ01IT5UAOXLDW98509E.00
 <01IT5STUMUFO984Z8L@siroe.edu>

04-Feb-1998 00:06:29.31 conversion                D 9 (3)
 amy@siroe.edu rfc822;bert@sesta.com bert
 imta/queue/conversion/ZZ01IT5UAMZ4QW98518O.00
 <01IT5UALL14498518O@siroe.edu>

04-Feb-1998 00:06:32.62 l                         D 9 (4)
 amy@siroe.edu rfc822;bert@siroe.com bert
 imta/queue/l/ZZ01IT5UAOXLDW98509E.00
 <01IT5STUMUFO984Z8L@siroe.edu>
```

1. 外部ユーザ amy@siroe.edu からのメッセージがチャネル l の受取人 bert@sesta.com に届きました。しかし、CONVERSIONS マッピングエントリにより、このメッセージは直接チャネル l には送られず、最初に変換チャネルのキューに入れられます。
2. 変換チャネルが実行され、メッセージがチャネル l のキューに入れられます。
3. 変換チャネルはメッセージをキューから取り出す (古いメッセージファイルを削除する) ことができます。
4. 最後に、チャネル l のキューからメッセージが取り出され (配信され) ています。

図 13-11 に、LOG_CONNECTION=3 によって接続ログが有効になっているときの送信メッセージのログ出力を示します。この例では、LOG_PROCESS=1、LOG_MESSAGE_ID=1、および LOG_FILENAME=1 も設定されていると仮定されています。この例は、ユーザ adam@sesta.com が 3 人の受取人 (bobby@hosta.sesta.com、carl@hosta.sesta.com、および dave@hostb.sesta.com) に同じメッセージ (各メッセージコピーのメッセージ ID は同じ) を送信している場合を示しています。この例では、メッセージが single_sys チャネルキーワードで示された tcp_local チャネル (普段使用しているチャネル) から送信されていると仮定しています。したがって、(1)、(2)、(3) で示されているように、それぞれの受取人に対して、別々のメッ

セージファイルが別々のホスト名のディスク上に作成されます。bobby@hosta.sesta.com と carl@hosta.sesta.com の受取人は同じメッセージファイルに保存されますが、dave@hostb.sesta.com の受取人は別のメッセージファイルに保存されます。

図 13-11 ログ : 送信接続ログ

```
19-Feb-1998 10:52:05.41 1e488.0 l            tcp_local    E 1
 adam@sesta.com rfc822;bobby@hosta.sesta.com bobby@hosta.sesta.com
 imta/queue/tcp_local/ZZ01ITRF7BO388000FCN.00  (1)
  <01ITRF7BDHS6000FCN@SESTA.COM>

19-Feb-1998 10:52:05.41 1e488.0 l            tcp_local    E 1
 adam@sesta.com rfc822;carl@hosta.sesta.com carl@hosta.sesta.com
 imta/queue/tcp_local/ZZ01ITRF7BO388000FCN.00  (2)
  <01ITRF7BDHS6000FCN@SESTA.COM>

19-Feb-1998 10:52:05.74 1e488.1 l            tcp_local    E 1
 adam@sesta.com rfc822;dave@hostb.sesta.com dave@hostb.sesta.com
 imta/queue/tcp_local/ZZ01ITRF7C11FU000FCN.00  (3)
  <01ITRF7BDHS6000FCN@SESTA.COM>

19-Feb-1998 10:52:10.79 1f625.2.0 tcp_local    -            O (4)
 TCP|206.184.139.12|5900|206.184.139.66|25
 SMTP/hostb.sesta.com/mailhub.sesta.com        (5)

19-Feb-1998 10:52:10.87 1f625.3.0 tcp_local    -            O (6)
 TCP|206.184.139.12|5901|206.184.139.70|25
 SMTP/hosta.sesta.com/hosta.sesta.com          (7)

19-Feb-1998 10:52:12.28 1f625.3.1 tcp_local                 D 1
 adam@sesta.com rfc822;bobby@hosta.sesta.com bobby@hosta.sesta.com
 imta/queue/tcp_local/ZZ01ITRF7BO388000FCN.00
  <01ITRF7BDHS6000FCN@SESTA.COM>
 hosta.sesta.com dns;hosta.sesta.com           (8)
 (TCP|206.184.139.12|5901|206.184.139.70|25)
 (hosta.sesta.com -- Server ESMTP [iMS V5.0 #8790])
 (TCP|206.184.139.12|5901|206.184.139.70|25)
 smtp;250 2.1.5 bobby@hosta.sesta.com and options OK.

19-Feb-1998 10:52:12.28 1f625.3.1 tcp_local                  D 1
 adam@sesta.com rfc822;carl@hosta.sesta.com carl@hosta.sesta.com
 imta/queue/tcp_local/ZZ01ITRF7BO388000FCN.00
  <01ITRF7BDHS6000FCN@SESTA.COM>
 hosta.sesta.com dns;hosta.sesta.com
 (TCP|206.184.139.12|5901|206.184.139.70|25)
 (hosta.sesta.com -- Server ESMTP [iMS V5.0 #8790])
 (TCP|206.184.139.12|5901|206.184.139.70|25)
 smtp;250 2.1.5 carl@hosta.sesta.com and options OK.

19-Feb-1998 10:52:12.40 1f625.3.2 tcp_local     -            C (9)
 TCP|206.184.139.12|5901|206.184.139.70|25
 SMTP/hosta.sesta.com/hosta.sesta.com
```

```
19-Feb-1998 10:52:13.01 1f625.2.1 tcp_local                    D 1
 adam@sesta.com rfc822;dave@hostb.sesta.com dave@hostb.sesta.com
 imta/queue/tcp_local/ZZ01ITRF7C11FU000FCN.00
  <01ITRF7BDHS6000FCN@SESTA.COM>
 mailhub.sesta.com dns;mailhub.sesta.com
 (TCP|206.184.139.12|5900|206.184.139.66|25)
 (MAILHUB.SESTA.COM -- Server ESMTP [iMS V5.0 #8694])
 (TCP|206.184.139.12|5900|206.184.139.66|25)
 smtp;250 2.1.5 dave@hostb.sesta.com and options OK.

19-Feb-1998 10:52:13.05 1f625.2.2 tcp_local     -              C (10)
 TCP|206.184.139.12|5900|206.184.139.66|25
 SMTP/hostb.sesta.com/mailhub.sesta.com
```

1. 1 人目の受取人へのメッセージがキューに入れられます。
2. 次に、2 人目の受取人へのメッセージがキューに入れられます。
3. 最後に、3 人目の受取人へのメッセージがキューに入れられます。
4. LOG_CONNECTION=3 が設定されているため、MTA がこのエントリを書き込みます。マイナス記号「-」は、このエントリが送信接続であることを示しています。「O」は、このエントリが接続開始に対応することを意味しています。この接続開始はスレッド 2 とスレッド 3 によって実行されていますが、これらの接続開始に対するマルチスレッド TCP/IP チャネルに同じプロセスが使用されているため、プロセス ID は同じ (1f625) であることに注意してください。
5. 2 つの異なるリモートシステムに接続するため、別々のスレッドにあるマルチスレッド SMTP クライアントがそれぞれの接続を開いています。最初の接続はこのエントリで、2 番目の接続は 7 に示されています。エントリのこの部分には、送信側と受信側の IP 番号とポート番号、および最初のホスト名と DNS 検索で見つかったホスト名の両方が示されています。SMTP/*initial-host*/*dns-host* には、最初のホスト名と、DNS MX レコード検索を実行したあとで使用されるホスト名が表示されています。mailhub.sesta.com は、hostb.sesta.com の MX サーバであることがわかります。
6. マルチスレッド SMTP クライアントが、別のスレッドで 2 番目のシステムとの接続を開いています (プロセスは同じ)。
7. 2 つの異なるリモートシステムに接続するため、別々のスレッドにあるマルチスレッド SMTP クライアントがそれぞれの接続を開いています。2 番目の接続はこのエントリで、最初の接続は上記の 5 に示されています。エントリのこの部分には、送信側と受信側の IP 番号とポート番号、および最初のホスト名と DNS 検索で見つかったホスト名の両方が示されています。この例では、hosta.sesta.com というシステムがメールを直接受信することがわかります。
8. この例に示されているように、特定の接続エントリのほか、LOG_CONNECTION=3 によって接続に関連する情報が標準のメッセージエントリに組み込まれます。

9. LOG_CONNECTION=3 が設定されているため、MTA がこのエントリを書き込みます。メッセージ (この例では bobby と carl のメッセージ) がキューから取り出されたあと、接続が終了します。このエントリでは C で示されています。
10. LOG_CONNECTION=3 が設定されているため、MTA がこのエントリを書き込みます。メッセージ (この例では dave のメッセージ) がキューから取り出されたあと、接続が終了します。このエントリでは C で示されています。

図 13-12 に、LOG_CONNECTION=3 によって接続ログが有効になっているときの受信 SMTP メッセージのログ出力を示します。

図 13-12 ログ : 受信接続ログ

```
19-Feb-1998 17:02:08.70 tcp_local    +             O  (1)
 TCP|206.184.139.12|25|192.160.253.66|1244 SMTP       (2)

19-Feb-1998 17:02:26.65 tcp_local    l             E 1
 service@siroe.com rfc822;adam@sesta.com adam
 THOR.SIROE.COM (THOR.SIROE.COM [192.160.253.66])     (3)

19-Feb-1998 17:02:27.05 tcp_local    +             C  (4)
 TCP|206.184.139.12|25|192.160.253.66|1244 SMTP

19-Feb-1998 17:02:31.73 l                          D 1
 service@siroe.com rfc822;adam@sesta.com adam
```

1. リモートシステムが接続を開きます。「O」は、このエントリが接続開始に対応したものであることを示しています。「+」は、このエントリが受信接続であることを示しています。
2. 接続の IP 番号とポートが示されています。このエントリでは、受信システム (ログファイルエントリを記録しているシステム) の IP アドレスは 206.184.139.12、ポート番号は 25、送信システムの IP アドレスは 192.160.253.66、ポートは 1244 です。
3. このエントリは、受信 TCP/IP チャネル (tcp_local) からチャネル 1 の受取人に送られるメッセージがキューに入っていることを示しています。LOG_CONNECTION=3 が有効になっているため、デフォルト以外の情報も含まれています。特に、送信システムがその HELO または EHLO 行に示す名前、接続 IP 番号の DNS リバース検索で見つかった送信システムの名前、および送信システムの IP アドレスが、すべてログに記録されます。この動作に影響するチャネルキーワードについては、第 8 章「チャネル定義を設定する」を参照してください。

4. 受信接続が閉じています。「O」は、このエントリが接続終了に対応したものであることを示しています。「+」は、このエントリが受信接続であることを示しています。

ディスパッチャのデバッグとログファイル

ディスパッチャエラーとデバッグ出力 (有効になっている場合) は、MTA ログディレクトリ内の dispatcher.log ファイルに書き込まれます。

デバッグ出力は、ディスパッチャ設定ファイルの DEBUG オプションを使って有効にするか、または IMTA_DISPATCHER_DEBUG 環境変数 (UNIX) を使ってプロセスレベルで有効にすることができます。

DEBUG オプションまたは IMTA_DISPATCHER_DEBUG 環境変数 (UNIX) は、16 進数で 32 ビットのデバッグマスクを定義するものです。すべてのデバッグ機能を有効にするには、オプションを 1 に設定するか、またはシステム全体で論理 / 環境変数を FFFFFFFF に定義します。表 13-7 に、各ビットの説明を示します。

表 13-7 ディスパッチャデバッグビット

ビット	16 進値	10 進値	使用目的
0	x 00001	1	サービスディスパッチャのメインモジュールの基本的なデバッグ。
1	x 00002	2	サービスディスパッチャのメインモジュールの特別なデバッグ。
2	x 00004	4	サービスディスパッチャ設定ファイルのログ処理。
3	x 00008	8	サービスディスパッチャに関するその他の基本的なデバッグ。
4	x 00010	16	サービスの基本的なデバッグ。
5	x 00020	32	サービスの特別なデバッグ。
6	x 00040	64	プロセスに関連するサービスのデバッグ。
7	x 00080	128	使用されていません。
8	x 00100	256	サービスディスパッチャとプロセス通信の基本的なデバッグ。
9	x 00200	512	サービスディスパッチャとプロセス通信の特別なデバッグ。
10	x 00400	1024	パケットレベル通信のデバッグ。
11	x 00800	2048	使用されていません。
12	x 01000	4096	ワーカープロセスの基本的なデバッグ。
13	x 02000	8192	ワーカープロセスの特別なデバッグ。
14	x 04000	16384	その他のワーカープロセスのデバッグ (特に接続ハンドオフ)。

表 13-7　ディスパッチャデバッグビット (続き)

ビット	16 進値	10 進値	使用目的
15	x 08000	32768	使用されていません。
16	x 10000	65536	サービスディスパッチャ I/O に対するワーカープロセスの基本的なデバッグ。
17	x 20000	131072	サービスディスパッチャ I/O に対するワーカープロセスの特別なデバッグ。
20	x 100000	1048576	統計の基本的なデバッグ。
21	x 200000	2097152	統計の特別なデバッグ。
24	x 1000000	16777216	PORT_ACCESS 拒否を dispatcher.log ファイルにログ。

Solaris のシステムパラメータ

システムのヒープサイズ (datasize) は、ディスパッチャによるスレッドスタックの使用を考慮して十分なサイズに設定する必要があります。各ディスパッチャサービスに対して、STACKSIZE*MAX_CONNS を計算し、それらの計算値を合計します。システムのヒープサイズは、この合計値の 2 倍以上でなければなりません。

ディスパッチャ設定ファイルで提供されるディスパッチャサービスは、さまざまなシステムパラメータの必要要件に影響を与えます。

ヒープサイズ (すなわち、デフォルトの datasize) を表示するには、以下の csh コマンドを使用します。

```
# limit
```

または、以下の ksh コマンドを使用します。

```
# ulimit -a
```

または、以下のユーティリティを使用します。

```
# sysdef
```

第 14 章

MTA のトラブルシューティング

この章では、MTA (Message Transfer Agent) のトラブルシューティングのための一般的なツール、方法、手順について説明します。この章には、以下の節があります。

モニタ手順に関連する項目は、第 15 章「iPlanet Messaging Server をモニタする」で参照できます。

注 この章を読む前に、このマニュアルの第 6 章から第 10 章と、『iPlanet Messaging Server リファレンスマニュアル』の MTA 設定およびコマンドラインユーティリティに関する章をもう一度確認してください。

トラブルシューティングの概要

MTA トラブルシューティングの最初の段階の 1 つは、診断を始める場所を決めることです。該当する問題によって、ログファイルにあるエラーメッセージを検索することもできます。また、標準 MTA プロセスのすべてをチェックしたり、MTA 設定を見直したり、個々のチャネルを起動して停止することもできます。

どの方法を使用する場合も、MTA のトラブルシューティングを行う際は次の点を考慮してください。

- メッセージの受け入れが設定や環境に関する問題 (たとえば、ディスク容量や制限容量の問題) によって妨げられていないか ?
- メッセージがキューに入れられたときに、ディスパッチャやジョブコントローラなどの MTA サービスが実行されていたか ?
- ネットワーク接続やルーティングの問題が、リモートシステム上でメッセージの未着や配信ミスの原因になっていないか ?
- 問題が発生したのは、メッセージをキューに入れる前後か ?

この章の以下の節で、これらの問題に対する処置を説明しています。

MTA のトラブルシューティングの標準的な手順

この節では、MTA のトラブルシューティングの標準的な手順の概要を説明します。問題が発生してもエラーメッセージが生成されない場合、エラーメッセージに十分な診断情報がない場合、あるいは MTA の全般的な状況のチェック、テスト、および標準的な保守を行う場合は、以下の手順に従ってください。

- 471 ページの「MTA 設定をチェックする」
- 471 ページの「メッセージキューディレクトリをチェックする」
- 472 ページの「危険なファイルの所有権をチェックする」
- 473 ページの「ジョブコントローラとディスパッチャが実行中であることをチェックする」
- 474 ページの「ログファイルをチェックする」
- 475 ページの「チャネルプログラムを手動で実行する」
- 476 ページの「個々のチャネルを起動および停止する」
- 477 ページの「MTA のトラブルシューティングの例」

MTA 設定をチェックする

imsimta test -rewrite ユーティリティを使って、アドレス設定をテストしてください。このユーティリティを使うと、実際にメッセージを送信することなく、MTA のアドレス書き換えとチャネルマッピングをテストすることができます。詳細については、『iPlanet Messaging Server リファレンスマニュアル』の MTA コマンドラインユーティリティの章を参照してください。

通常このユーティリティは、メッセージをキューに入れるチャネルとともに、適用されるアドレス書き換えを表示します。ただし、このユーティリティは、MTA 設定のシンタックスエラーが発生すると、エラーメッセージを発行します。出力が希望するものでない場合は、設定を修正することもできます。

メッセージキューディレクトリをチェックする

メッセージが MTA メッセージキューディレクトリ (通常は */server-root*/msg-*instance*/imta/queue/) にあるかどうかをチェックしてください。希望するメッセージが MTA メッセージキューディレクトリにあるかどうかをチェックするには、imsimta qm のようなコマンドラインユーティリティを使用します。imsimta qm の詳細については、『iPlanet Messaging Server リファレンスマニュアル』と 523 ページの「imsimta qm counters」とを参照してください。

imsimta test -rewrite の出力が正しいようであれば、メッセージが実際に MTA メッセージキューサブディレクトリに置かれているかどうかをチェックします。これを行うには、メッセージのログを有効にします (MTA ログの詳細は、446 ページの「第 3 部 : サービスログ (MTA)」を参照)。次に、ディレクトリ */server-root*/msg-*instance*/log/imta/ にある mail.log_current ファイルを調べます。特定のメッセージをそのメッセージ ID で追跡して、メッセージが MTA メッセージキューサブルーチンに置かれていることを確認できます。メッセージが見つからない場合は、ファイルのディスク容量やディレクトリアクセス権に関する問題がある可能性があります。

危険なファイルの所有権をチェックする

iPlanet Messaging Server をインストールしたときに、メールサーバのユーザアカウント (デフォルトでは nobody) を選択したはずです。以下のディレクトリ、サブディレクトリ、およびファイルは、このアカウントが所有している必要があります。

/*server-root*/msg-*instance*/imta/queue/
/*server-root*/msg-*instance*/log/imta/
/*service-root*/msg-*instance*/imta/tmp

以下の UNIX システムのコマンド例にあるようなコマンドを使用して、これらのディレクトリの保護と所有権をチェックできます。

```
ls -l -p -d /usr/iplanet/server5/msg-budgie/imta/queue
drwx------ 6 nobody bin 512 Feb 7 09:32
/usr/iplanet/server5/msg-budgie/imta/queue

ls -l -p -d /usr/iplanet/server5/msg-budgie/log/imta
drwx------ 2 nobody bin 1536 Mar 10 09:00
/usr/iplanet/server5/msg-budgie/log/imta

ls -l -p -d /usr/iplanet/server5/msg-budgie/imta/tmp
drwx------ 2 nobody bin 512 Feb 7 10:00
/usr/iplanet/server5/msg-budgie/imta/tmp
```

以下の UNIX システムのコマンド例のようなコマンドを使用して、/*server-root*/msg-*instance*/imta/queue にあるファイルが MTA アカウントによって所有されていることをチェックします。

```
ls -l -p -R /usr/iplanet/server5/msg-budgie/imta/queue
```

ジョブコントローラとディスパッチャが実行中であることをチェックする

MTA ジョブコントローラは、大半の送信 (マスター) チャネルジョブなどの、MTA が処理するジョブの実行を行います。

MTA チャネルの中には、MTA のマルチスレッド SMTP チャネルのように、受信メッセージを処理する常駐サーバプロセスを含むものもあります。このようなサーバは、チャネルのスレーブ (受信) 方向を扱います。MTA ディスパッチャは、そのような MTA サーバの作成を行います。ディスパッチャの設定オプションは、サーバの可用性、作成されたサーバの数、各サーバが処理できる接続の数を制御します。

ジョブコントローラとディスパッチャがあるかどうかをチェックし、MTA サーバと処理するジョブが実行中かどうかを確認するには、`imsimta process` コマンドを使用します。このコマンドは、アイドル状態では `job_controller` および `dispatcher` プロセスになります。たとえば、以下のようになります。

```
imsimta process

USER       PID S   VSZ    RSS    STIME     TIME   COMMAND
mailsrv  9567  S  18416   9368   02:00:02  0:00 /opt/iplanet/
  server5/bin/msg/imta/bin/tcp_smtp_server

mailsrv  6573  S  18112   5720   Jul_13    0:00 /opt/iplanet/
  server5/bin/msg/imta/bin/job_controller

mailsrv  9568 S  18416   9432   02:00:02   0:00 /opt/iplanet/
  server5/bin/msg/imta/bin/tcp_smtp_server

mailsrv  6574 S  17848   5328   Jul_13     0:00 /opt/iplanet/
  server5/bin/msg/imta/bin/dispatcher
```

ジョブコントローラがない場合、*/server-root*/*msg*-instance/imta/queue ディレクトリにあるファイルはバックアップされ、メッセージは配信されません。ディスパッチャがなければ、SMTP 接続を受信することはできません。

`imsimta process` の詳細は、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

ジョブコントローラもディスパッチャもない場合は、*/server-root*/msg-*instance*/log/imta/ ディレクトリにある `dispatcher.log-*` または `job_controller.log-*` ファイルを確認します。

ログファイルが存在しないか、エラーが示されていない場合は、imsimta start コマンドを使ってプロセスを開始してください。詳細は、『iPlanet Messaging Server リファレンスマニュアル』の MTA コマンドラインユーティリティの章を参照してください。

注 imsimta process を実行するときは、ディスパッチャまたはジョブコントローラの複数のインスタンスが実行されていないようにしてください。

ログファイルをチェックする

MTA が処理するジョブが正常に実行されていても、メッセージがメッセージキューディレクトリに残っている場合は、ログファイルを調べて何が起きているかを見ることができます。すべての MTA ログファイルはディレクトリ */server-root*/msg-*instance*/log/imta 内に作成されています。表 14-1 に、MTA が処理するさまざまなジョブのログファイル名の形式を示します。

表 14-1 MTA ログファイル

ファイル名	ログファイルの内容
*channel*_master.log-*uniqueid*	*channel* のマスタープログラム (通常はクライアント) の出力
*channel*_slave.log-*uniqueid*	*channel* のスレーブプログラム (通常はサーバ) の出力
dispatcher.log-*uniqueid*	ディスパッチャのデバッグ。このログは、ディスパッチャの DEBUG オプションが設定されているかどうかにかかわらず作成されます。ただし、デバッグの詳細情報を入手するには、DEBUG オプションをゼロ以外の値に設定する必要があります。
imta	配信に関する問題が発生した場合の ims-ms チャネルのエラーメッセージ。
job_controller.log-*uniqueid*	ジョブコントローラのログ。このログは、ジョブコントローラの DEBUG オプションが設定されているかどうかにかかわらず作成されます。ただし、デバッグの詳細情報を入手するには、DEBUG オプションをゼロ以外の値に設定する必要があります。
tcp_smtp_server.log-*uniqueid*	tcp_smtp_server. のデバッグ。このログ内の情報はサーバ固有の情報であり、メッセージに対するものではありません。
return.log-*uniqueid*	定期的な MTA メッセージバウンサージョブのデバッグ出力。option.dat 内で return_debug オプションを使用している場合は、このログファイルが作成されます。

注 それぞれのログファイルは、以前に同じチャネルによって作成されたログを上書きしないように、固有 ID (*uniqueid*) で作成されます。特定のログファイルを見つける際は、`imsimta view` ユーティリティを使用できます。`imsimta purge` コマンドを使用して、古いログファイルをパージすることもできます。詳細は、『iPlanet Messaging Server リファレンスマニュアル』の MTA コマンドラインユーティリティの章を参照してください。

channel`_master.log-`*uniqueid* および *channel*`_slave.log-`*uniqueid* のログファイルは、次のような状況で作成されます。

- 現在の設定にエラーがある場合。
- `master_debug` または `slave_debug` キーワードが `imta.cnf` ファイル内のチャネルに設定されている場合。
- `mm_debug` が `option.dat` ファイル (*/server-root/msg-*`instance/imta/config/` ディレクトリ内) でゼロ以外の値 (`mm_debug > 0`) に設定されている場合。

チャネルのマスターおよびスレーブプログラムのデバッグについては、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

チャネルプログラムを手動で実行する

MTA の配信問題を診断するときは、特に、1 つ以上のチャネルに対するデバッグを有効にしたあとで、MTA 配信ジョブを手動で実行することをお勧めします。

`imsimta submit` コマンドは、MTA ジョブコントローラにチャネルの実行を通知します。問題のチャネルに対してデバッグが有効になっている場合は、表 14-1 で示すように、`imsimta submit` でディレクトリ */server-root*`/msg-`*instance*`/log/imta` 内にログファイルが作成されます。

`imsimta run` コマンドは、現在アクティブなプロセスのもとでチャネルに対する送信を実行し、また、端末に出力を送信します。ジョブの送信自体に問題があると思われる場合は特に、ジョブを送信するよりもこの方法をお勧めします。

注 チャネルを手動で実行するには、ジョブコントローラが実行されている必要があります。

`imsimta submit` コマンドと `imsimta run` コマンドのシンタックス、オプション、パラメータ、例の詳細は、『iPlanet Messaging Server リファレンスマニュアル』の MTA コマンドラインユーティリティの章を参照してください。

個々のチャネルを起動および停止する

場合によっては、個々のチャネルを停止して再起動することで、メッセージキューの問題の診断とデバッグが行いやすくなることもあります。メッセージキューを停止して、キューに入れられたメッセージを検査し、ループまたはスパム攻撃があるかどうかを確認することができます。

特定のチャネルへの送信処理 (キューからの取り出し) を停止するには

1. `imsimta qm stop` コマンドを使用して、特定のチャネルを停止します。これにより、ジョブコントローラを停止する必要がなくなり、設定を再コンパイルしなくてすみます。以下の例では、`conversion` チャネルを停止しています。

   `imsimta qm stop conversion`

2. 処理を再開するには、`imsimta qm start` コマンドを使用してチャネルを再起動します。以下の例では、`conversion` チャネルを再起動しています。

   `imsimta qm start conversion`

`imsimta qm start` コマンドと `imsimta qm stop` コマンドの詳細は、『iPlanet Messaging Server リファレンスマニュアル』の MTA コマンドラインユーティリティの章を参照してください。

特定のドメインまたは IP アドレスからの受信処理 (チャネルのキューに入れる) を停止するには

クライアントホストに一時的な SMTP エラーを返している間に、特定のドメインまたは IP アドレスからの受信メッセージ処理を停止したい場合は、以下の操作のいずれかを実行することができます。これを実行すると、メッセージはシステム上に保持されることはありません。305 ページの「第 1 部 マッピングテーブル」を参照してください。

- 特定のホストまたはドメイン名からの受信処理を停止するには、MTA マッピングファイル (通常は */server-root*/msg-*instance*/imta/config/mappings) にある ORIG_SEND_ACCESS マッピングテーブルに以下のアクセス規則を追加します。

```
ORIG_SEND_ACCESS

    *|*@sesta.com|*|*              $X4.2.1|$NHost$ blocked
```

このようにすると、差出人のリモート MTA はメッセージをシステム上に保持し、受信処理を再開するまで定期的にそのメッセージを再送信し続けるようになります。

- 特定の IP アドレスからの受信処理を停止するには、MTA マッピングファイル (通常は */server-root*/msg-*instance*/imta/config/mappings) にある PORT_ACCESS マッピングテーブルに以下のアクセス規則を追加します。

```
PORT_ACCESS

    TCP|*|25|IP_address_to_block|*                    $N500$ unable$ to$ ¥
    connect$ at$ this$ time
```

ドメインまたは IP アドレスからの受信処理を再開するときは、必ず上記の規則をマッピングテーブルから削除し、設定を再コンパイルしてください。さらに、各マッピングテーブルごとに固有のエラーメッセージを作成することもできます。これを行うことで、使用中のマッピングテーブルを確認することができます。

MTA のトラブルシューティングの例

この節では、特定の MTA の問題のトラブルシューティング方法をステップバイステップで説明します。この例では、メールの受取人は電子メールメッセージの添付ファイルを受信しませんでした。注 : MIME プロトコルの用語に沿って、この節では「添付ファイル」のことを「メッセージ部分」と呼びます。前述のトラブルシューティング方法を使用して、メッセージ部分が見えなくなった場所と原因を確認します (470 ページの「MTA のトラブルシューティングの標準的な手順」を参照)。以下のステップで、メッセージが MTA を通じてとるパスを確認することができます。さらに、メッセージ部分が見えなくなったのがキューに入れられる前後かどうかを確認することができます。これを行うには、関連ファイルを取り込みながら、チャネルを手動で停止してから起動する必要があります。

注 メッセージをチャネルを通じて手動で起動するときは、ジョブコントローラが実行されている必要があります。

メッセージパスにあるチャネルを識別する

メッセージパスにあるチャネルを識別することによって、該当するチャネルに `master_debug` および `slave_debug` キーワードを適用することができます。これらのキーワードはチャネルのマスターおよびスレーブログファイルにデバッグ出力を生成します。そのマスターおよびスレーブデバッグ情報により、メッセージ部分が見えなくなった場所が識別しやすくなります。

1. ディレクトリ */server-root*/msg-*instance*/imta/config にある `option.dat` ファイルに `log_message_id=1` を追加します。このパラメータにより、`mail.log_current` ファイルにあるメッセージ ID: ヘッダー行が表示されます。

2. `imsimta cnbuild` を実行して設定を再コンパイルします。
3. `imsimta restart dispatcher` を実行して、SMTP サーバを再起動します。
4. エンドユーザにメッセージ部分を含むメッセージを再送信してもらいます。
5. メッセージが通過するチャネルを確認します。

   チャネルを識別する方法にはいろいろありますが、以下の方法をお勧めします。

   a. UNIX プラットフォームの場合は、`grep` コマンドを使用して、*/server-root*`/msg-`*instance*`/log/imta/` ディレクトリにある `mail.log_current` ファイルでメッセージ ID: ヘッダー行を検索します。Windows NT プラットフォームの場合は、`find` コマンドを使用します。

   b. メッセージ ID: ヘッダー行が見つかったら、E (キューに入れる) および D (キューから取り出す) レコードを検索して、メッセージのパスを確認します。ログエントリコードの詳細は、449 ページの「MTA ログエントリの形式」を参照してください。この例の場合は、以下の E および D レコードを見てください。

```
29-Aug-2001 10:39:46.44 tcp_local conversion     E 2 ...

29-Aug-2001 10:39:46.44 conversion tcp_intranet  E 2 ...

29-Aug-2001 10:39:46.44 tcp_intranet             D 2 ...
```

左側のチャネルはソースチャネルで、右側のチャネルは宛先チャネルです。この例では、E レコードと D レコードは、メッセージのパスが `tcp_local` チャネルから `conversion` チャネルに移り、最後に `tcp_intranet` チャネルに移っていることを示しています。

データを収集するためにチャネルを手動で起動および停止する

この節では、チャネルを手動で起動したり停止する方法を説明します。詳細については、476 ページの「個々のチャネルを起動および停止する」を参照してください。メッセージのパスにあるチャネルを手動で起動したり停止することによって、メッセージとログファイルを MTA プロセスのさまざまな段階で保存することができます。これらのファイルは、後述の 480 ページの「メッセージに問題が発生した場所を確認する」の節で使用できます。

1. 十分なデバッグ情報を提供するためには、ディレクトリ */server-root*`/msg-`*instance*`/imta/config` にある `option.dat` ファイルに `mm_debug=5` を設定します。

2. ディレクトリ */server-root*/msg-*instance*/imta/config 内の imta.cnf ファイルにある該当するチャネルに、slave_debug キーワードと master_debug キーワードを追加します。

   a. リモートシステムから送信されるメッセージ部分を含むメッセージの受信チャネル (または最初のダイアログの間にメッセージが切り替えられるチャネル) で、slave_debug キーワードを使用します。この例では、slave_debug キーワードが tcp_local チャネルに追加されています。

   b. メッセージが通過し、477 ページの「メッセージパスにあるチャネルを識別する」で識別されたほかのチャネルに、master_debug キーワードを追加します。この例では、master_debug キーワードは conversion チャネルと tcp_intranet チャネルに追加されます。

   c. imsimta restart dispatcher コマンドを実行して SMTP サーバを再起動します。

3. imsimta qm stop コマンドと imsimta qm start コマンドを使用して、特定のチャネルを起動して停止します。これらのキーワードの使用の詳細は、476 ページの「個々のチャネルを起動および停止する」を参照してください。

4. メッセージファイルの取り込み処理を開始するには、エンドユーザにメッセージ部分を含むメッセージを再送信してもらいます。

5. メッセージがチャネルに入るときに、メッセージが imsimta qm stop コマンドによって停止されていると、メッセージはチャネル内で停止します。詳細は、ステップ 手順 3 を参照してください。

   a. メッセージのパスにある次のチャネルを手動で起動する前に、メッセージファイルをコピーして名前を変更します。以下の UNIX プラットフォームの例を見てください。

      ```
      # cp ZZ01K7LXW76T7O9TD0TB.00 ZZ01K7LXW76T7O9TD0TB.KEEP1
      ```

      通常、メッセージファイルは、*/server-root*/msg-*instance*/imta/queue/*destination_channel*/001 のようなディレクトリにあります。*destination_channel* は、メッセージが通過する次のチャネル (tcp_intranet など) です。*destination_channel*l ディレクトリにサブディレクトリ (001、002 など) を作成する場合は、チャネルに subdirs キーワードを追加します。

   b. メッセージが処理される順番を識別するために、メッセージをトラップしてコピーするたびに、メッセージの拡張子に番号を付けることをお勧めします。

6. チャネルでメッセージの処理を再開し、メッセージのパスにある次の宛先チャネルのキューに入れます。これを行うには、imsimta qm start コマンドを使用します。

7. /*server-root*/msg-*instance*/log/imta/ ディレクトリにある対応するチャネルログファイル (たとえば、tcp_intranet_master.log-*) をコピーして、保存します。追跡しているメッセージのデータを含む該当するログファイルを選択します。必ず、コピーするファイルが、チャネルで受信するメッセージのタイムスタンプおよび Subject ヘッダーと一致するようにします。tcp_intranet_master.log-* の例では、ファイルが削除されないように、ファイルを tcp_intranet_master.keep という名前で保存しています。
8. 最終的な宛先に達するまで、ステップ 5 ～ 7 を繰り返します。

   ステップ 手順 7 でコピーしたログファイルは、ステップ 手順 5 でコピーしたメッセージファイルと相互に関連させる必要があります。たとえば、メッセージ部分がないためにすべてのチャネルを停止した場合は、conversion_master.log-* ファイルと tcp_intranet_master.log-* ファイルを保存します。ソースチャネルのログファイル tcp_local_slave.log-* も保存します。さらに、それぞれの宛先チャネルの対応するメッセージファイルのコピーを保存します。つまり、conversion チャネルの ZZ01K7LXW76T7O9TD0TB.KEEP1、tcp_intranet チャネルの ZZ01K7LXW76T7O9TD0TB.KEEP2 を保存します。
9. メッセージとログファイルがコピーされたら、デバッグオプションを削除します。
   a. ディレクトリ /*server-root*/msg-*instance*/imta/config の imta.cnf ファイルにある該当するチャネルから、slave_debug キーワードと master_debug キーワードを削除します。
   b. mm_debug=0 をリセットし、ディレクトリ /*server-root*/msg-*instance*/imta/config の option.dat ファイルにある log_message_id=1 を削除します。
   c. imsimta cnbuild を使用して設定を再コンパイルします。
   d. imsimta restart dispatcher コマンドを実行して SMTP サーバを再起動します。

メッセージに問題が発生した場所を確認する

1. チャネルプログラムの起動と停止が終わるまでには、トラブルシューティングのために使用できる以下のファイルがあるはずです。
   a. 各チャネルプログラムのメッセージファイルのすべてのコピー (たとえば、ZZ01K7LXW76T7O9TD0TB.KEEP1)
   b. tcp_local_slave.log-* ファイル
   c. 各宛先チャネルの *channel*_master.log-* ファイルのセット

d. メッセージのパスを示す mail.log_current レコードのセット

どのファイルにも、mail.log_current レコードにあるメッセージ ID: ヘッダー行に一致するタイムスタンプとメッセージ ID がある必要があります。メッセージが受取人にバウンスされた場合は例外です。バウンスされたメッセージには元のメッセージとは異なるメッセージ ID 値が付いています。

2. tcp_local_slave.log-* ファイルを調べて、メッセージがキューに入れられたときにメッセージにメッセージ部分があったかどうかを確認します。

SMTP ダイアログとデータを見て、クライアントマシンから何が送信されたかを確認します。

メッセージ部分が tcp_local_slave.log-* ファイルになかった場合、問題が発生したのはメッセージが MTA に入る前です。結果として、メッセージはメッセージ部分なしでキューに入れられています。このような場合、問題は、差出人のリモート SMTP サーバまたは差出人のクライアントマシンで発生した可能性があります。

3. メッセージファイルを詳しく調べて、メッセージ部分が変更されたり欠落した場所を確認します。

メッセージファイルにメッセージ部分が変更されたり欠落したことが示されていた場合は、前のチャネルのログファイルを調べます。たとえば、tcp_intranet チャネルに入っているメッセージのメッセージ部分が変更されたり欠落した場合は、conversion_master.log-* ファイルを確認する必要があります。

4. メッセージの最終的な宛先を確認します。

tcp_local_slave.log、メッセージファイル (例 : ZZ01K7LXW76T7O9TD0TB.KEEP1)、および *channel*_master.log-* でメッセージ部分が変更されていないようであれば、MTA がメッセージを変更したのではなく、メッセージ部分は最終的な宛先へのパスの次のステップで消えています。

最終的な宛先が ims-ms チャネル (メッセージストア) である場合は、メッセージ部分がこの転送の間または転送のあとに欠落したかどうかを確認するために、メッセージをサーバからクライアントマシンにダウンロードすることもできます。宛先チャネルが tcp_* チャネルの場合は、メッセージのパスにある MTA に移動する必要があります。iPlanet Messaging Server の MTA の場合は、トラブルシューティング処理すべてを繰り返す必要があります (477 ページの「メッセージパスにあるチャネルを識別する」、478 ページの「データを収集するためにチャネルを手動で起動および停止する」、およびこの節を参照)。その他の MTA が自分の管理下にない場合は、問題を報告したユーザが特定のサイトに問い合わせる必要があります。

一般的な MTA の問題と解決策

この節では、MTA の設定と操作で一般的に起こりやすい問題と解決策を示します。

設定ファイルまたは MTA データベースに対する変更が有効にならない

設定、マッピング、変換、セキュリティ、オプション、またはエイリアスファイルに対する変更が有効になっていない場合は、以下のステップを実行したかどうかをチェックします。

1. 設定を再コンパイルします (`imsimta cnbuild` を実行)。
2. 該当するプロセス (`imsimta restart dispatcher` など) を再起動します。
3. クライアント接続を再度確立します。

MTA が、メールを送信するが受信しない

ほとんどの MTA チャネルは、スレーブまたはチャネルプログラムに依存して、受信メッセージを受信します。MTA がサポートしているいくつかの転送プロトコル (TCP/IP や UUCP など) の場合、転送プロトコルが標準サーバではなく MTA スレーブプログラムをアクティブにしていることを確認する必要があります。ネイティブの `sendmail` SMTP サーバから MTA の SMTP サーバへの置換は、iPlanet Messaging Server のインストールの際に実行されます。詳細は、『iPlanet Messaging Server UNIX 用インストールガイド』を参照してください。

マルチスレッド SMTP サーバの場合、SMTP サーバの起動はディスパッチャによって制御されます。ディスパッチャが SMTP サービスより大きいか等しい MIN_PROCS 値を使用して構成されている場合は、少なくとも 1 つの SMTP サーバプロセスが常に実行している必要があります (SMTP サービスの MAX_PROCS 値によっては複数の場合もあります)。imsimta process コマンドを使用して、SMTP サーバプロセスがあるかどうかをチェックすることもできます。詳細については、『iPlanet Messaging Server リファレンスマニュアル』の MTA コマンドラインユーティリティの章を参照してください。

受信 SMTP 接続時のタイムアウト

受信 SMTP 接続時のタイムアウトは、システムリソースやその割り当てに関連していることがよくあります。以下の方法を使用して、受信 SMTP 接続時のタイムアウトの原因を識別することができます。

1. 同時に許可する受信 SMTP 接続の数をチェックします。これは SMTP サービスのディスパッチャ設定である MAX_PROCS および MAX_CONNS によって制御され、許可できる同時接続数は MAX_PROCS*MAX_CONNS です。接続数が少なすぎる場合、システムリソースに余裕があれば、この数を増やすことを考慮してください。
2. 使用できるもう 1 つの方法は、TELNET セッションを開くことです。以下の例では、ユーザは 127.0.0.1 ポート 25 に接続しています。接続すると、220 個の見出しが返されます。たとえば、以下のようになります。

```
telnet 127.0.0.1 25
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
220 budgie.sesta.com -- Server ESMTP (iPlanet Messaging Server
5.1 (built May  7 2001))
```

接続して 220 個の見出しを受信しても、その他のコマンド (ehlo や mail from など) が応答を許可していない場合は、imsimta test -rewrite を実行して設定が正しいことを確認する必要があります。imsimta dirsync コマンドを使用している場合は、最近このコマンドを実行したことを確認します。dirsync が失敗した場合は、SMTP サーバ内のコマンドが応答を受信しないこともあります。このような場合、imsimta dirsync -F を実行すると問題を解決できます。ただし、dirsync ロックファイルが最初に削除されている必要があります。

3. 220 個の見出しの応答が遅い場合や、SMTP サーバで pstack コマンドを実行すると以下の iii_res* 関数 (名前解決検索が実行されていることを示す) が表示される場合があります。

```
febe2c04 iii_res_send (fb7f4564, 28, fb7f4de0, 400, fb7f458c,
fb7f4564) + 142c
febdfdcc iii_res_query (0, fb7f4564, c, fb7f4de0, 400, 7f) + 254
```

このような場合は、localhost/127.0.0.1 のような共通ペアでも、ホストが名前解決のリバース検索を行う必要があることも考えられます。このようなパフォーマンスの低下を回避するには、/etc/nsswitch.conf ファイルでのホストの検索順を並べ替える必要があります。このためには、/etc/nsswitch.conf ファイルの以下の行を変更します。

```
hosts: dns nis [NOTFOUND=return] files
```

次のように変更します。

```
hosts: dns nis [NOTFOUND=return] files
```

/etc/nsswitch.conf ファイルでこのように変更すれば、パフォーマンスを向上させることができます。複数の SMTP サーバで必要のない検索を実行するのではなく、少数の SMTP サーバでメッセージを処理する必要があります。

4. 受信 SMTP over TCP/IP メールを処理しているチャネル (通常は、tcp_local と tcp_intranet) 上に slave_debug キーワードを設定することもできます。これを実行したあと、最新の tcp_local_slave.log-*uniqueid* ファイルを見直して、タイムアウトになったメッセージの特性を識別します。たとえば、受取人の数が多すぎる受信メッセージがタイムアウトになった場合は、チャネル上で expandlimit キーワードを使用することを考慮してください。

システムが過負荷になっていて拡張されすぎている場合は、タイムアウトを完全に回避するのは難しくなります。

メッセージがキューから取り出されない

TCP/IP 配信中に発生したエラーは、一時的なことがよくあります。通常、MTA は、問題が発生したときにメッセージを残し、それを定期的に再試行します。大規模なネットワークでは通常、あるホスト上で定期的な機能停止が起こっても、ほかのホスト接続は適切に作動しています。問題を検証するには、配信試行に関連するエラーのログファイルを調べます。「smtp_open の致命的なエラー」のようなエラーメッセージが表示されることもあります。このようなエラーは特別なものではなく、通常はネットワークに関する一時的な問題と関連しています。TCP/IP ネットワークに関する問題をデバッグするには、PING、TRACEROUTE、NSLOOKUP のようなユーティリティを使用します。

以下の例は、メッセージが `xtel.co.uk` への配信待ちでキューに入ったままになっている理由を確認するためのステップを示しています。メッセージがキューから取り出されない理由を確認するには、MTA が TCP/IP 上で SMTP メールを配信するために使用するステップを再作成することができます。

```
% nslookup -query=mx xtel.co.uk ( 手順 1)

Server: LOCALHOST
Address: 127.0.0.1

Non-authoritative answer:
XTEL.CO.UK  preference = 10, mail exchanger = nsfnet-relay.ac.uk
( 手順 2)

% telnet nsfnet-relay.ac.uk 25 ( 手順 3)
Trying... [128.86.8.6]
telnet: Unable to connect to remote host: Connection refused
```

1. NSLOOKUP ユーティリティを使用して、MX レコードがこのホストに存在していることを確認します。MX レコードが存在していない場合、直接ホストへの接続を試みる必要があります。MX レコードが存在している場合、指定された MX リレーに接続する必要があります。MTA は MX 情報を優先して処理します (優先して処理しないように設定されている場合は除く)。230 ページの「TCP/IP MX レコードのサポート」も参照してください。
2. この例では、DNS (ドメインネームサービス) は `xtel.co.uk` の指定された MX リレーの名前を返しています。これは MTA の実際の接続先になるホストです。複数の MX リレーがリストにある場合、MTA は各 MX レコードを、優先度がもっとも低いものから順に、連続して試行します。

3. リモートホストへの接続がある場合は、SMTP サーバのポート 25 への TELNET を使用して、受信 SMTP 接続を受け入れているかどうかチェックする必要があります。

注 ポートを指定しないで TELNET を使用すると、リモートホストが通常の TELNET 接続を受け入れることがわかります。これは、SMTP 接続を受け入れることを示すわけではありません。多くのシステムは、正規の TELNET 接続は受け入れても SMTP 接続は拒否します。または、その逆になります。そのため、常に SMTP ポートのテストを行う必要があります。

前述の例では、リモートホストは SMTP ポートへの接続を拒否しています。これが、MTA がメッセージの配信に失敗した理由です。この接続は、リモートホストの設定ミスやリモートホスト上での何らかのリソース不足のために拒否されることがあります。このような場合は、ローカルで問題解決を行うことはできません。通常は、MTA にメッセージの再試行を続けさせることになります。

DNS を使用しない TCP/IP ネットワーク上で iPlanet Messaging Server が稼働している場合は、ステップ (手順 1) と (手順 2) をスキップすることができます。代わりに、TELNET を使用して、問題となっているホストに直接アクセスすることができます。MTA が使用するホスト名と同じホスト名を使用する際は、注意してください。ホスト名を確認するには、MTA の最後の試行に関連するログファイルを確認します。ホストファイルを使用している場合は、ホスト名情報が正しいことを確認する必要があります。ホスト名ではなく DNS を使用することを、強くお勧めします。

TCP/IP ホストへの接続をテストする場合、インタラクティブテストを使用して問題が発生しないのであれば、ほぼ確実に、問題は MTA が最後にメッセージを配信しようとしたあとに解決されています。該当するチャネルで `imsimta submit tcp_channel` を再度実行して、メッセージがキューから取り出されているかどうかを確認することができます。

MTA メッセージが配信されない

メッセージ転送に関する問題のほかに、2 つの一般的な問題があります。この問題はメッセージキューにある未処理のメッセージに起因することがあります。

1. キューキャッシュはキューディレクトリにあるメッセージと同期しません。MTA キューサブディレクトリにある配信待ちのメッセージファイルは、インメモリキューキャッシュに入れられます。起動時にチャネルプログラムは、このキューキャッシュを調べて、キューにあるどのメッセージを配信するかを確認します。メッセージファイルがキューの中にあっても、対応するキューキャッシュエントリがない場合もあります。

a. 特定のファイルがキューキャッシュにあるかどうかをチェックするには、`imsimta cache -view` ユーティリティを使用します。ファイルがキューキャッシュにない場合は、キューキャッシュを同期させる必要があります。

   キューキャッシュは、通常は 4 時間ごとに同期されます。必要に応じて、`imsimta cache -sync` を使用してキャッシュを手動で再同期することができます。同期が終わると、チャネルプログラムは、新しいメッセージが処理されたあとで、元の未処理メッセージを処理します。デフォルト (4 時間) を変更する場合は、`sync_time=`*timeperiod* を追加することで、ディレクトリ */server-root*/msg-*instance*/imta/config にある `job_controller.cnf` ファイルを変更する必要があります。ここで、*timeperiod* は、キューキャッシュを同期させる頻度です。*timeperiod* は 30 分より長くする必要があります。以下の例では、`job_controller.cnf` のデフォルトのグローバルセクションに `sync_time=02:00` を追加することで、キューキャッシュの同期間隔が 2 時間に変更されます。

```
! VERSION=5.0
!IMTA job controller configuration file
!
!Global defaults
tcp_port=27442
secret=N1Y9[HzQKW
slave_command=NULL
sync_time=02:00
```

   `imsimta submit` *channel* を実行して、`imsimta cache -sync` を実行したあとにメッセージのバックログを空にすることができます。メッセージのバックログが大きい (1000 以上) 場合、チャネルを空にするのに時間がかかることがあるので、注意してください。

   キューキャッシュの情報の概要については、`imsimta qm -maint dir -database -total` を実行してください。

b. キューキャッシュを同期させてもメッセージがまだ配信されない場合は、ジョブコントローラを再起動する必要があります。これを行うには、`imsimta restart job_controller` コマンドを使用します。

   ジョブコントローラを再起動すると、メッセージのデータ構造がディスク上のメッセージキューから再構築されます。

警告 ジョブコントローラの再起動は最後の手段です。ほかの手段をすべて使用し尽くすまでは実行しないでください。

ジョブコントローラの詳細は、108 ページの「ジョブコントローラ」を参照してください。

2. 処理するログファイルを作成できないために、チャネル処理プログラムの実行は失敗します。アクセス権、ディスク容量、および制限容量をチェックします。

メッセージがループしている

メッセージがループしていることを MTA が検出すると、そのメッセージは HELD ファイルとして保持されます。489 ページの「.HELD メッセージを診断して整理する」を参照してください。場合によっては、MTA がメッセージループを検出できないときもあります。

最初のステップは、メッセージがループしている理由を確認することです。問題のメッセージのコピー (MTA キュー領域にあるとき)、問題のメッセージに関連する MTA メールログエントリ (該当チャネルの MTA 設定ファイルで `logging` チャネルキーワードが有効になっている場合)、および該当チャネルの MTA チャネルのデバッグログファイルを確認します。問題のメッセージの From: および To: アドレス、Received: ヘッダー行、およびメッセージ構造 (メッセージ内容のカプセル化の種類) を確認して、発生したメッセージループの種類を特定することができます。

一般的によくある原因として、以下のものがあります。

1. ポストマスターアドレスが壊れている。

   MTA では、電子メールを受信するために、ポストマスターアドレスが正しく機能しなければなりません。ポストマスターへのメッセージがループしている場合は、メッセージを受信できるアカウントをポイントする適切なポストマスターアドレスが設定されているかどうかチェックします。

2. Received: ヘッダー行を削除すると、MTA はメッセージのループを検出できなくなります。

   通常のメッセージループの検出は、Received: ヘッダー行に基づいています。Received: ヘッダー行が MTA システム自体で明示的に、あるいはファイアウォールのような別のシステム上で削除されている場合、メッセージループを適切に検出できなくなることがあります。このような場合は、Received: ヘッダー行が知らないうちに削除されていないかどうかチェックします。また、メッセージがループしている根本的な原因もチェックします。考えられる原因は、システム名の割り当ての問題 (システムが自分の名前の変形を認識しないように設定されている場合)、DNS の問題、該当するシステムに承認可能なアドレス情報がないこと、あるいはユーザアドレス転送エラーなどです。

3. ほかのメッセージングシステムによる通知メッセージの処理が正しくなく、通知メッセージに応答して再度カプセル化されたメッセージが生成されている。

   インターネット規格では、通知メッセージ (メッセージ配信やメッセージ差し戻しのレポート) にメッセージループを防ぐための空のエンベロープ From: アドレスがあることを必要としています。ただし、メッセージングシステムによってはこのような通知メッセージを正しく処理しない場合もあります。このようなメッセージングシステムは、通知メッセージを転送または返送するときに、新しいエンベロープ From: アドレスを挿入することがあります。これがメッセージループの原因になることもあります。解決策は、通知メッセージを正しく処理していないメッセージングシステムを修復することです。

.HELD メッセージを診断して整理する

メッセージがサーバまたはチャネル間で返送されていることを MTA が検出すると、配信は停止され、メッセージは */server-root*/msg-*instance*/imta/queue/*channel* にある、接尾辞が .HELD のファイルに格納されます。通常、メッセージのループが発生するのは、各サーバまたはチャネルがメッセージの配信をほかのサーバやチャネルが担当するとみなしたときです。

たとえば、エンドユーザが、2 つの別々のメールホスト上のメッセージを互いのホストに転送するオプションを設定しているとします。sesta.com アカウントに対して、varrius.com アカウントへのメール転送を有効にしています。また、この設定が有効であることを忘れて、varrius.com アカウントに対して sesta.com アカウントへのメール転送を有効にしています。

ループは、MTA の設定に誤りがあるために発生することもあります。たとえば、MTA ホスト X は、mail.sesta.com のメッセージがホスト Y に送信されるとみなします。しかし、ホスト Y はホスト X が mail.sesta.com のメッセージを処理すべきとみなし、結果としてホスト Y はホスト X にメールを返信することになります。

このような場合、MTA はメッセージを無視し、それ以上配信は試行されません。このような問題が発生したときは、メッセージ内のヘッダー行を見て、サーバまたはチャネルがメッセージをバウンスしているかどうか確認します。必要であればエントリを修正してください。

以下の手順に従って .HELD メッセージを再試行することもできます。

1. 拡張子 .HELD を 00 以外の任意の 2 桁の数字 (たとえば、06) に変更します。

注	.HELD ファイルの名前を変更する前に、メッセージのループが停止していることを確認してください。

2. imsimta cache -sync を実行します。このコマンドを実行すると、キャッシュが更新されます。
3. imsimta submit *channel* または imsimta run *channel* を実行します。

これらのステップは何回か実行することが必要かもしれません。これは、Received: ヘッダー行が蓄積され、それによってメッセージに再度 .HELD とマークが付けられている可能性があるためです。

受信したメッセージがエンコードされている

MTA が送信したメッセージは、エンコードされた形式で受信されます。たとえば、以下のようになります。

```
Date: Wed, 04 Jul 2001 11:59:56 -0700 (PDT)
From: "Desdemona Vilalobos" <Desdemona@sesta.com>
To: santosh@varrius.com
Subject: test message with 8bit data
MIME-Version: 1.0
Content-type: TEXT/PLAIN; CHARSET=ISO-8859-1
Content-transfer-encoding: QUOTED-PRINTABLE


2=00So are the Bo=F6tes Void and the Coal Sack the same?=
```

これらのメッセージは、MTA デコーダコマンド `imsimta decode` を使用すれば、デコードされて表示されます。詳細は、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

RFC 821 で定義されているように、ASCII 文字 (7 ビット文字セット) を送信できるのは SMTP プロトコルのみです。実際には、ネゴシエーションが行われていない 8 ビット文字の転送は SMTP 経由では無効であり、いくつかの SMTP サーバでさまざまな問題の原因になることがあります。たとえば、SMTP サーバが計算量の多いループに陥ってしまうことがあります。メッセージは何度も繰り返し送信されます。8 ビット文字は SMTP サーバをクラッシュさせることがあります。最終的に、8 ビット文字セットは、8 ビットデータを扱えないブラウザやメールボックスに大きな損害をもたらす可能性があります。

以前に使用されていた SMTP クライアントには、8 ビットデータを含むメッセージを処理するときのオプションが 3 つしかありませんでした。メッセージを配信不能として差出人に返送するオプション、メッセージをエンコードするオプション、RFC 821 の直接違反でメッセージを送信するオプションです。しかし、MIME および SMTP 拡張の出現により、現在では、ASCII 文字セットを使用することによって 8 ビットデータをエンコードする標準のエンコーディングがあります。

前述の例で受取人は、MIME コンテンツタイプが TEXT/PLAIN のエンコードされたメッセージを受信しています。リモート SMTP サーバ (MTA SMTP クライアントからのメッセージの転送先) は、8 ビットデータの転送をサポートしていません。元のメッセージに 8 ビット文字が含まれていたため、MTA はメッセージをエンコードする必要があります。

SSR (Server-Side Rules) が作動していない

フィルタは、メールメッセージに適用される 1 つ以上の条件付きアクションで構成されています。フィルタはサーバ上に保存されて評価されるため、SSR (Server-Side Rules) と言われることがよくあります。

ディレクトリ同期コマンド (`imsimta dirsync`) は、ユーザのフィルタに関する情報を含む MTA の SSR データベースを更新します。SSR データベースは短いフィルタ (1016 バイト未満) を保存し、LDAP DN は長いフィルタ用に使われます。MTA は、`imsimta dirsync` コマンドでディレクトリサーバが更新されるまではユーザのフィルタに対する変更を認識しません。SSR の詳細は、332 ページの「第 2 部 メールボックスフィルタ」を参照してください。

この節では、SSR に関する以下の情報について説明します。

- 491 ページの「SSR 規則をテストする」
- 492 ページの「トラブルシューティングの手順」
- 492 ページの「一般的なシンタックスの問題」

SSR 規則をテストする

- 以下のコマンドを使用して、MTA のユーザフィルタをチェックします。

```
# imsimta test -rewrite -debug -filter user@domain
```

出力では以下の情報を探します。

```
mmc_open_url called to open ssrf: user@ims-ms
   URL with quotes stripped: ssrd: user@ims-ms
Determined to be a SSRD URL.
   Identifier: user@ims-ms-daemon
Filter successfully obtained.
```

- さらに、slave_debug キーワードを tcp_local チャネルに追加して、フィルタが適用される状態を確認することができます。この結果は tcp_local_slave.log ファイルに表示されます。十分なデバッグ情報を得るためには、ディレクトリ /server-root/msg-*instance*/imta/config にある option.dat ファイルに mm_debug=5 を追加します。

トラブルシューティングの手順

SSR の問題を診断する前は、必ず以下の手順を確認しておいてください。

- imsimta dirsync コマンドを使用している場合は、ims-ms チャネルに、以下のフィルタ
  ssrd:$a
  および
  fileinto $u+$s@$d が、ディレクトリ /*server-root*/msg-*instance*/imta/config の imta.cnf ファイル内でマークされていることを確認します。
- imsimta dirsync コマンドを使用している場合は、imsimta dirsync コマンドがフィルタ情報を適切に同期することを確認します。これを行うには、ディレクトリ /*server-root*/msg-*instance*/ から以下のコマンドを実行します。このコマンドは必ずメッセージングサーバのユーザとして実行してください。

```
# configutil -l -o service.imta.ssrenabled -v true
OK SET
# configutil | fgrep ssr
local.imta.ssrenabled = yes
service.imta.ssrenabled = true
```

一般的なシンタックスの問題

- フィルタにシンタックスの問題がある場合は、tcp_local_slave.log-* ファイルで以下のメッセージを探します。
  Error parsing filter expression:...
  - フィルタが適正であれば、出力の最後にフィルタ情報があります。
  - フィルタが不正であれば、出力の最後に以下のエラーがあります。
    Address list error -- 4.7.1 Filter syntax error:
    desdaemona@sesta.com

    また、フィルタが不正であれば、SMTP RCPT TO コマンドによって一時的なエラー応答コードが返されます。

```
RCPT TO: user@domain
452 4.7.1 Filter syntax error
```

一般的なエラーメッセージ

MTA が起動に失敗すると、コマンドラインに一般的なエラーメッセージが表示されます。この節では、共通の一般的なエラーメッセージの説明と診断を示します。

注 MTA 設定を診断するには、`imsimta test -rewrite -debug` ユーティリティを使用して MTA のアドレス書き換えとチャネルマッピング処理を調べます。このユーティリティを使用すれば、メッセージを実際に送信しなくても設定をチェックすることができます。471 ページの「MTA 設定をチェックする」を参照してください。

MTA サブコンポーネントは、この章では説明していないほかのエラーメッセージを発行することもあります。各サブコンポーネントの詳細は、『iPlanet Messaging Server リファレンスマニュアル』の MTA コマンドラインユーティリティの章と、第 6 章から第 10 章を参照してください。ここでは、以下のタイプのエラーについて説明します。

- 493 ページの「mm_init でのエラー」
- 497 ページの「コンパイル済み設定のバージョンが一致していない」
- 497 ページの「スワップ空間のエラー」
- 498 ページの「ファイルのオープンまたは作成エラー」
- 499 ページの「不正なホスト / ドメインエラー」
- 500 ページの「SMTP チャネルでのエラー : os_smtp_* エラー」

mm_init でのエラー

mm_init でのエラーは、通常は MTA の設定の問題を示します。`imsimta test -rewrite` ユーティリティを実行する場合は、これらのエラーが表示されます。`imsimta cnbuild` などのその他のユーティリティ、チャネル、サーバ、またはサーバがこのようなエラーを返すこともあります。

よく発生する mm_init エラーには以下のものがあります。

- 494 ページの「エイリアスが同じではありません」
- 494 ページの「エイリアスインクルードファイルを開くことができません」
- 494 ページの「重複するエイリアスが見つかりました」
- 494 ページの「チャネルテーブル内でホストが重複しています」
- 495 ページの「重複するマッピング名が見つかりました」
- 495 ページの「マッピング名が長すぎます」

- 495 ページの「ch_ 機能の初期化中のエラー : コンパイルした文字セットのバージョンが一致しません」
- 495 ページの「ch_ 機能の初期化中のエラー : 空き容量がありません」
- 496 ページの「システムのローカルホストエイリアスまたは固有名詞が長すぎます」
- 496 ページの「同じエイリアスアドレスがありません」
- 496 ページの「チャネルの正規のホスト名がありません」
- 496 ページの「正規のホスト名が長すぎます」

エイリアスが同じではありません

エイリアスファイルのエントリの右側が適切にフォーマットされていません。

エイリアスインクルードファイルを開くことができません

エイリアスファイルに含まれているファイルを開くことができません。

重複するエイリアスが見つかりました

エイリアスファイルの 2 つのエントリが両方とも左側にあります。重複するものを見つけて削除する必要があります。`error line #XXX` というエラーメッセージを探します。`XXX` は行番号です。この行にある重複のエイリアスを修正することができます。

チャネルテーブル内でホストが重複しています

このエラーメッセージは、MTA の設定に 2 つのチャネル定義があり、両方に同じ正規ホスト名があることを示しています。

MTA 設定ファイル (`imta.cnf`) の書き換え規則 (上部) に関係のない空白行があると、MTA は設定ファイルの残りの部分をチャネル定義と解釈します。ファイルの最初の行が空白でないことを確認してください。同じパターンを持つ書き換え規則 (左側) が複数あると、MTA はそれらの書き換え規則を、一意でない正規のホスト名を含むチャネル定義と解釈します。正規のホスト名が重複しているチャネル定義がないかどうか、また、ファイル上部 (書き換え規則の部分) に不適切な空白行がないかどうか、MTA の設定をチェックしてください。

重複するマッピング名が見つかりました

このメッセージは、2 つのマッピングテーブルに同じ名前が付いていて、これらの重複するマッピングテーブルのいずれかを削除する必要があることを示します。ただし、マッピングファイル内のフォーマットエラーによって、MTA が何かを間違ってマッピングテーブル名と解釈することもあります。たとえば、マッピングテーブルエントリが適切にインデントされていないと、MTA はエントリの左側が実際にマッピングテーブル名であるとみなします。マッピングファイルが一般の形式であることと、マッピングテーブル名をチェックしてください。

注 空白行はマッピングテーブル名を含む行の前と後ろに付ける必要があります。ただし、空白行をマッピングテーブルのエントリ間に入れないでください。

マッピング名が長すぎます

このエラーは、マッピングテーブル名が長すぎるので、短くする必要があることを示しています。マッピングファイル内のフォーマットエラーによって、MTA が何かを間違ってマッピングテーブル名と解釈することもあります。たとえば、マッピングテーブルエントリが適切にインデントされていないと、MTA はエントリの左側が実際にマッピングテーブル名であるとみなします。マッピングファイルとマッピングファイル名をチェックしてください。

ch_ 機能の初期化中のエラー : コンパイルした文字セットのバージョンが一致しません

このメッセージが表示された場合は、`imsimta chbuild` コマンドを使用して、コンパイル済みの文字セットテーブルを再コンパイルして再インストールする必要があります。詳細は、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

ch_ 機能の初期化中のエラー : 空き容量がありません

通常、このエラーメッセージは、MTA 文字セットの内部テーブルのサイズを変更し、以下のコマンドでコンパイル済み文字セットテーブルを再構築する必要があることを意味しています。

```
imsimta chbuild -noimage -maximum -option
imsimta chbuild
```

この変更を加える前に、ほかには何も再コンパイルまたは再起動する必要がないことを確認してください。`imsimta chbuild` の詳細は、『iPlanet Messaging Server リファレンスマニュアル』の MTA コマンドラインユーティリティの章を参照してください。

システムのローカルホストエイリアスまたは固有名詞が長すぎます

このエラーは、ローカルホストエイリアスまたは固有名詞が長すぎることを示します (オプションで、チャネルブロックの 2 番目以降の名前の右側にある)。ただし、MTA 設定ファイル内でこのエラーより前にシンタックスエラー (書き換え規則に関係のない空白行がある場合など) がある場合は、MTA が何かを間違ってチャネル定義と解釈することもあります。設定ファイルの指定されている行をチェックするだけでなく、その行より上にほかのシンタックスエラーがないかどうかもチェックしてください。特に、このエラーが発生した行が書き換え規則を意図する行である場合は、その行より上に関係のない空白行がないかどうかを必ずチェックしてください。

同じエイリアスアドレスがありません

エイリアスファイル内のエントリの右側 (変換値) がありません。

チャネルの正規のホスト名がありません

このエラーは、チャネル定義ブロックに必須の 2 番目の行 (正規のホスト名の行) がないことを示しています。チャネル定義ブロックの詳細は、『iPlanet Messaging Server リファレンスマニュアル』の MTA の設定およびコマンドラインユーティリティの章と、第 8 章「チャネル定義を設定する」を参照してください。それぞれのチャネル定義ブロックの前と後ろには空白行が必要ですが、空白行をチャネル定義のチャネル名と正規のホスト名の行の間に入れることはできません。また、空白行は MTA 設定ファイルの書き換え規則部分には入れることはできません。

正規のホスト名が長すぎます

チャネルの正規のホスト名 (チャネル定義ブロックの 2 行目) は、長さが 40 オクテットに制限されています。チャネル上で長めの正規ホスト名を使用しようとしている場合は、それをプレースホルダ名まで短くしてから、書き換え規則を使用してその長めの名前がその短い正規ホスト名に一致するようにします。このような状況は、l (ローカル) チャネルホスト名を使用しているときに起こることがあります。たとえば、以下のようになります。

Original l Channel:

```
!delivery channel to local /var/mail store
l subdirs 20 viaaliasrequired maxjobs 7 pool LOCAL_POOL
newt.salamander.lizard.gecko.komododragon.com
```

Create Place Holder:

```
!delivery channel to local /var/mail store
l subdirs 20 viaaliasrequired maxjobs 7 pool LOCAL_POOL
newt
```

Create Rewrite Rule:

```
newt.salamander.lizard.gecko.komododragon.com    $U%$D@newt
```

l (ローカル) チャネルを使用しているときは、REVERSE マッピングテーブルを使用する必要があります。使用法とシンタックスの詳細は、『iPlanet Messaging Server リファレンスマニュアル』の MTA の設定の章を参照してください。

MTA 設定ファイル内でこのエラーより前にシンタックスエラー (書き換え規則に関係のない空白行があった場合など) がある場合は、MTA が何かを間違ってチャネル定義と解釈することもあります。このため、書き換え規則を意図していたとしても、正規のホスト名と解釈されてしまうことがあります。設定ファイルの指定されている行をチェックするだけでなく、その行より上にほかのシンタックスエラーがないかどうかもチェックしてください。特に、このエラーが発生した行が書き換え規則を意図する行である場合は、その行より上に関係のない空白行がないかどうかを必ずチェックしてください。

コンパイル済み設定のバージョンが一致していない

`imsimta cnbuild` ユーティリティの機能の 1 つとして、MTA の設定情報を、すばやく読み込むことができるイメージにコンパイルする機能があります。コンパイル済みフォーマットは厳密に定義されており、多くの場合、異なるバージョンの MTA 間では実質的に異なっています。小さな変更はパッチリリースとして発生することもあります。

このような変更が発生すると、互換性のないフォーマットを検出するために、内部バージョンフィールドも変更されます。互換性のないフォーマットを検出すると、MTA コンポーネントは上記のエラーで停止します。この問題の解決策は、`imsimta cnbuild` コマンドを使って新しいコンパイル済み設定を生成することです。

また、`imsimta restart` コマンドを使用して常駐 MTA サーバプロセスを再起動することも良い方法です。これによって、常駐 MTA サーバプロセスは更新された設定情報を取得することができます。

スワップ空間のエラー

適切な動作を保証するために、メッセージングシステム上に十分なスワップ空間を設定することが重要です。必要なスワップ空間の量は設定によって異なります。調整の際に一般的に推奨されるのは、スワップ空間の量を主記憶容量の少なくとも 3 倍にすることです。

以下のようなエラーメッセージは、スワップ空間が不足していることを示しています。

```
jbc_channels: chan_execute [1]: fork failed: Not enough space
```

このエラーはジョブコントローラのログファイルで見られることがあります。その他のスワップ空間のエラーは設定によって異なります。

以下のコマンドを使用して、スワップ空間の空き容量と使用容量を確認します。

- Solaris システム : `swap -s` (MTA プロセスがビジー状態のとき)、`ps -elf`、または `tail /var/adm/messages`
- HP-UX システム : `swapinfo` または `tail /var/adm/syslog/syslog.log`
- Windows NT システム : より多くの空間が必要な場合や、ほかの場所により高速のドライブがある場合は、デフォルトのハードドライブ (`C:¥` など) 以外のドライブ上にページングファイルサイズを設定することができます。利用可能な容量をチェックしたり、新しいページングファイルサイズを設定するには、以下の手順に従います。
  - コントロールパネルの [システムのプロパティ] (または [システム]) をクリックします。
  - [パフォーマンス] タブをクリックします。
  - [仮想メモリ] の [変更] をクリックします。
  - [仮想メモリ] ウィンドウに各ドライブのページングファイルサイズが表示されます。

ファイルのオープンまたは作成エラー

メッセージを送信するために、MTA は設定ファイルを読み取って、MTA メッセージキューディレクトリにメッセージファイルを作成します。設定ファイルは、MTA または MTA の SDK に対して書かれたプログラムが読み取ることのできるものでなければなりません。適切な権限はこれらのファイルのインストール中に割り当てられます。設定ファイルを作成する MTA ユーティリティとプロシージャも、権限を割り当てます。ファイルがシステムマネージャ、特権を持つほかのユーザ、またはサイト固有のプロシージャによって保護されている場合、MTA は設定情報を読み取ることができない場合があります。その結果、「ファイルオープン」エラーや予測不能な動作が発生します。設定ファイルの読み取りに関する問題が発生したときは、`imsimta test -rewrite` ユーティリティが追加情報をレポートします。『iPlanet Messaging Server リファレンスマニュアル』の MTA の章にある `imsimta test -rewrite` の説明を参照してください。

MTA が、権限を持つアカウントから機能していて、権限のないアカウントからは機能していないように見える場合は、MTA テーブルディレクトリのファイルアクセス権が問題の原因と思われます。設定ファイルとそのディレクトリのアクセス権をチェックしてください。472 ページの「危険なファイルの所有権をチェックする」を参照してください。

「ファイル作成」エラーは、通常、MTA メッセージキューディレクトリにメッセージファイルを作成する際に問題が発生したことを示しています。ファイル作成に関する問題の診断については、471 ページの「メッセージキューディレクトリをチェックする」を参照してください。

不正なホスト / ドメインエラー

このエラーは、ブラウザで MTA にアドレスを指定したときに見られることがあります。また、このエラーは、据え置かれて、エラー返送メールメッセージの一部として返送されることがあります。どちらの場合もこのエラーメッセージは、MTA が指定したホストにメールを配信できないことを示しています。メールが指定したホストに送信されていない原因を確認するには、以下のトラブルシューティング手順に従います。

- 該当するアドレスにスペルミスがないかどうか、コピーミスがないかどうか、存在していないホストまたはドメインの名前を使用していないかどうかを確認します。
- `imsimta test -rewrite` ユーティリティを使って該当するアドレスを実行します。このユーティリティを使用してもアドレスで「不正なホスト / ドメイン」エラーが返される場合は、MTA の `imta.cnf` ファイルと関連ファイルにアドレスを処理する規則がありません。MTA が正しく設定されているかどうか、設定の際のすべての質問に適切に回答したかどうか、設定情報が最新のものになっているかどうかを確認してください。
- `imsimta test -rewrite` によってアドレスでエラーが発生しない場合、MTA はアドレスの処理方法を決定できますが、ネットワーク転送はそれを受け入れません。追加の詳細については、配信試行の際に作成された該当するログファイルを調べることができます。一時的なネットワークのルーティングエラーまたはネームサービスエラーが発生したことにより、エラーメッセージが返されることはありません。ただし、ドメインネームサーバの設定が大幅に間違っていると、このようなエラーが発生する可能性があります。
- インターネット上の場合は、MX レコード検索をサポートするように TCP/IP チャネルが正しく設定されているかどうかチェックします。多くのドメインアドレスはインターネットに直接アクセスすることはできず、メールシステムが正しく MX エントリを解決する必要があります。インターネット上の場合、および TCP/IP が MX レコードをサポートするように設定されている場合は、MX サポートを有効にするように MTA を設定する必要があります。詳細は、225 ページの「TCP/IP 接続と DNS 検索のサポート」を参照してください。TCP/IP パッケージが MX レコード検索をサポートするように設定されていない場合は、MX 専用ドメインにアクセスすることはできません。

SMTP チャネルでのエラー : os_smtp_* エラー

os_smtp_open、os_smtp_read、os_smtp_write エラーなどの os_smtp_* エラーは、必ずしも MTA エラーではありません。これらのエラーは、MTA がネットワーク層で発生した問題をレポートするときに生成されます。たとえば、os_smtp_open エラーは、リモート側へのネットワーク接続を開くことができなかったことを意味します。MTA は、アドレスエラーやチャネル設定エラーのために無効なシステムに接続するよう設定されていることがあります。一般的に os_smtp_* エラーは、DNS またはネットワーク接続の問題が原因です (特に、直前に処理していたのがチャネルまたはアドレスの場合)。os_smtp_read または os_smtp_write エラーは、一般的に、接続がリモート側で強制終了されたか、ネットワーク上の問題によるものであることを示しています。

多くの場合、ネットワークおよび DNS の問題は実際には一時的です。ときどき発生する os_smtp_* エラーは、通常は気にしなくても大丈夫です。ただし、これらのエラーが頻繁に表示される場合は、根本的なネットワーク上の問題がある可能性があります。

特定の os_smtp_* エラーに関する詳細情報を入手するには、該当するチャネル上でデバッグを有効にします。試行された SMTP ダイアログの詳細を示す、デバッグチャネルのログファイルを調べます。特に、ネットワークの問題が SMTP ダイアログのどこのタイミングで発生したかを確認します。このタイミングは、ネットワークまたはリモート側の問題の種類を示していることがあります。場合によっては、ネットワークレベルのデバッグ (たとえば、TCP/IP パケットトレース) を実行して、何を送信または受信したかを確認することもできます。

第 15 章

iPlanet Messaging Server をモニタする

多くの場合、適切に計画され設定されたサーバには、管理者による過度の介入は必要ありません。ただし、管理者は、サーバに問題の兆候がないかモニタする必要があります。この章では、iPlanet Messaging Server のモニタ機能について説明します。この章には、以下の節があります。

トラブルシューティングの手順については、第 14 章「MTA のトラブルシューティング」を参照してください。

毎日のモニタ作業

毎日行う必要のあるもっとも重要な作業は、ポストマスターメールのチェック、ログファイルのモニタ、および stored ユーティリティの設定です。これらの作業について、以降で説明します。

ポストマスターメールをチェックする

Messaging Server には、ポストマスター電子メール用に設定されている定義済み管理メーリングリストがあります。このメーリングリストに含まれているユーザは、ポストマスター宛てに送信されたメールを自動的に受信します。

ポストマスターメールの規則は RFC822 に定義されています。RFC822 では、すべての電子メールサイトでポストマスターという名前のユーザまたはメーリングリスト宛てに送信されたメールを受け取り、このアドレスに送信されたメールを実際のユーザに配信することを要求しています。postmaster@host.domain に送られるすべてのメッセージは、ポストマスターアカウントまたはメーリングリストに送られます。

通常、ユーザは、ポストマスターアドレス宛てに自分のメールサービスに関する電子メールを送信します。ポストマスターは、たとえば、ローカルユーザからはサーバ応答時間に関するメールを受信し、ほかのサーバ管理者からはサーバへのメール送信時に発生した問題に関するメールを受信します。ポストマスターメールは毎日チェックする必要があります。

また、ポストマスターアドレスに特定のエラーメッセージを送信するようにサーバを設定することもできます。たとえば、MTA がメッセージをルーティングまたは配信できないときは、ポストマスターアドレスに送信される電子メールによってそのことを知ることができます。また、ポストマスターに例外状態の警告 (ディスク容量の低下やサーバ応答の不良) を送ることもできます。

ログファイルをモニタおよび管理する

iPlanet Messaging Server は、サポートされている主なプロトコル (SMTP、IMAP、POP、HTTP) またはサービスごとに一連のログファイルを作成します。ログファイルは定期的にモニタする必要があり、サーバに問題がある場合は特に必要です。

ログ記録はサーバのパフォーマンスに影響することがあります。より詳細なログ記録を指定するほど、一定期間にログファイルが多くのディスク容量を占有することになります。効果的に定義する必要がありますが、現実的なログローテーション、有効期間、サーバのバックアップポリシーなどを考慮する必要があります。サーバのログポリシーの定義の詳細は、第 13 章「ログ記録とログ解析」を参照してください。

stored ユーティリティを設定する

stored ユーティリティは、以下のような、サーバの自動モニタおよび管理を実行します。

- バックグラウンドおよび日常のメッセージ処理タスク
- デッドロックの検出とデッドロックしたデータベーストランザクションのロールバック
- 起動時の一時ファイルのクリーンアップ
- 存続期間決定ポリシーの実装
- サーバの状態、ディスク容量、サービスへの応答時間などの定期的モニタ
- 必要に応じて警告を生成

stored ユーティリティは、毎日深夜 0 時に、クリーンアップと有効期限による失効の操作を自動的に実行します。詳細は、514 ページの「stored」を参照してください。

システムのパフォーマンスをモニタする

ここでは、特に iPlanet Messaging Server のモニタ機能について説明しますが、サーバがあるシステムもモニタする必要があります。適切に設定されたサーバでも、適切に調整されていないシステム上では正しく作動しないことがあります。また、サーバエラーの兆候は、ハードウェアに電子メールを処理するための十分な機能がないことを示している場合があります。この章では、システムパフォーマンスのモニタの詳細についてすべて説明しているわけではありません。これらの手順の多くはプラットフォーム固有のものであり、プラットフォーム固有のシステムのマニュアルを参照することが必要になる場合もあります。パフォーマンスをモニタする手順を以下に示します。

- 503 ページの「終端間メッセージ配信時間をモニタする」
- 504 ページの「ディスク容量をモニタする」
- 506 ページの「CPU 使用状況をモニタする」

終端間メッセージ配信時間をモニタする

電子メールは時間どおりに配信する必要があります。これがサービス契約の要件になっていることもあります。また、メールをできるだけ速く配信することは良いポリシーでもあります。終端間の時間が遅いことは、多くの事柄を示している可能性があります。たとえば、サーバが正しく作動していない、1 日の特定の時間にメッセージが処理不能になる、既存のハードウェアリソースの容量を超えている、などです。

終端間メッセージ配信時間の不良の兆候

メールの配信に、通常よりも長い時間がかかります。

終端間メッセージ配信時間をモニタするには

メッセージを送信および受信する機能を使用します。サーバのホップ間のヘッダー時間、および始点と取り出しの時間を比較します。

ディスク容量をモニタする

ディスク容量の不足は、メールサーバの問題と障害を発生させる、もっとも一般的な原因です。MTA キューまたはメッセージストアに書き込む領域がないと、メールサーバでエラーが発生します。さらに、ログファイルをモニタおよびクリーンアップしないと、ログファイルが制御できないほど大きくなり、ディスク容量を使い果たすことがあります。

`stored` のクリーンアップ機能が失敗し、削除されたメッセージがメッセージストアから消去されていないと、ディスク容量が急激に不足することがあります。MTA メッセージキューが大きくなりすぎたり、メッセージストアが利用可能なディスク容量より大きくなったり、モニタしていないログファイルが制御できないほど大きくなったりすることも、ディスク容量の低下を招きます (ログファイルには LDAP、MTA、および Message Access など、多数のものがあり、それらの各ログファイルは別のディスクに保存することができます)。

ディスク容量に関する問題の兆候

容量の低下によって発生する兆候は、ディスクやパーティションによって異なります。MTA キューがオーバーフローして SMTP 接続を拒否したり、メッセージが ims_master キューに残されたままでメッセージストアに配信されなくなったり、ログファイルがオーバーフローすることがあります。

ディスク容量をモニタするには

システムの構成に従って、さまざまなディスクやパーティションをモニタする必要があります。たとえば、MTA キューが 1 つのディスクやパーティション上にあり、メッセージストアが別の場所にあり、ログファイルがさらに別の場所にあるとします。この場合、それらの容量のそれぞれをモニタする必要があり、その容量をモニタする方法は異なることがあります。

メッセージストアをモニタする

メッセージストアのディスク容量は、75% を超えないようにすることをお勧めします。メッセージストアのディスク使用量をモニタするには、`configutil` ユーティリティを使用して以下の警告属性を設定します。

- alarm.diskavail.msgalarmstatinterval
- alarm.diskavail.msgalarmthreshold
- alarm.diskavail.msgalarmwarninginterval

これらのパラメータを設定することによって、システムがディスク容量をモニタする頻度と、どのような状況で警告を送信するかを指定することができます。たとえば、システムに 600 秒ごとにディスク容量をモニタさせる場合は、以下のコマンドを指定します。

```
configutil -o alarm.diskavail.msgalarmstatinterval -v 600
```

利用可能なディスク容量が 20% より低くなったとき常に警告を受け取る場合は、以下のコマンドを指定します。

```
configutil -o alarm.diskavail.msgalarmthreshold -v 20
```

これらのパラメータの詳細については、515 ページの表 15-1 を参照してください。

MTA キューとログ領域をモニタする

MTA キューのディスクおよびログ領域のディスク使用量をモニタする必要があります。

CPU 使用状況をモニタする

CPU 使用状況が高い場合は、使用状況のレベルに対して CPU 容量が不足しているか、または適切なサイクルより多くの CPU サイクルを使用しているプロセスがあることを示しています。

CPU 使用状況に関する問題の兆候

システムの応答が悪く、ユーザのログインに時間がかかり、配信速度が遅くなります。

CPU 使用状況をモニタするには

CPU 使用状況のモニタは、プラットフォーム固有のタスクです。関連するプラットフォームのマニュアルを参照してください。

MTA をモニタする

この節には、以下の項目があります。

メッセージキューのサイズをモニタする

メッセージキューが過度に大きくなる場合は、メッセージが配信されていない、配信が遅延されている、あるいは入るのが速すぎてシステムがメッセージを配信できないことを示していることがあります。これは、膨大なメッセージがシステムに送られるサービス拒否攻撃に遭っている、ジョブコントローラが実行されていないなど、さまざまな原因によって発生します。

メッセージキューの詳細は、105 ページの「チャネルメッセージキュー」, 485 ページの「メッセージがキューから取り出されない」および 486 ページの「MTA メッセージが配信されない」を参照してください。

メッセージキューに関する問題の兆候

- ディスク容量使用状況が高くなる

- ユーザが適切な時間内にメッセージを受信できない
- メッセージキューのサイズが異常に大きい

メッセージキューのサイズをモニタするには

メッセージキューをモニタする最良の方法は、`imsimta qm` を使用することです。523 ページの「imsimta qm counters」を参照してください。

キューディレクトリ (*/ServeRoot*/msg-*instance*/imta/queue/) 内のファイルの数もモニタすることができます。ファイルの数はサイト固有であり、「多すぎる」ものを見つけるための基準を作る必要があります。これは、キューファイルのサイズを 2 週間以上記録して、おおよその平均をとることによって行います。

配信エラーの頻度をモニタする

配信エラーは、外部サイトへのメッセージの配信試行のエラーです。配信エラーの頻度の大幅な増加は、DNS サーバの故障や、接続への応答時のリモートサーバのタイムアウトなど、ネットワークに関する何らかの問題の兆候です。

配信エラーの頻度に関する問題の兆候

表面的な問題の兆候はありません。多数の `Q` レコードは、`mail.log_current` に現れます。

配信エラーの頻度をモニタするには

配信エラーは、ログエントリコード `Q` とともに MTA ログに記録されます。`msg-`*instance*`/log/imta/mail.log_current` ファイル内の Q レコードを確認します。

受信 SMTP 接続をモニタするには

指定した IP アドレスからの受信用 SMTP 接続の数が異常に増加した場合は、以下の状況を示しています。

- 外部ユーザがメールをリレーしようとしている
- 外部ユーザがサービス拒否攻撃を行おうとしている

認証されていない SMTP 接続の兆候

- **外部ユーザによるメールのリレー** - 表面的には問題発生の兆候はない

- **サービス拒否攻撃** - 外部のメッセージ要求により SMTP サーバを過負荷にする試行

受信用 SMTP 接続をモニタするには

- **外部ユーザによるメールのリレー** - ログエントリレコード J (拒否されたリレー) を含むレコードの msg-*instance*/log/imta/mail.log_current を確認します。リモート IP アドレスのログを有効にするには、option.dat ファイルに以下の行を追加します。

  ```
  log_connection=1
  ```

  この機能を有効にすると、わずかながらパフォーマンスが低下します。

- **サービス拒否攻撃** - SMTP サーバに接続しているユーザとその人数を調べるには、netstat コマンドを実行し、SMTP ポート (デフォルトは 25) の接続を確認します。以下に例を示します。

  ```
  Local address        Remote
  address                      State

  192.18.79.44.25    192.18.78.44.56035    32768    0 32768      0
  CLOSE_WAIT
  192.18.79.44.25    192.18.136.54.57390    8760    0 24820      0
  ESTABLISHED
  192.18.79.44.25    192.18.26.165.48508   33580    0 24820      0
  TIME_WAIT
  ```

  最初に、システムで特定の読み取りが異常かどうかを判断するために、SMTP 接続の適切な数とその状態 (ESTABLISHED、CLOSE_WAIT など) を決定する必要があります。

  多数の接続が SYN_RECEIVED 状態にある場合は、ネットワークがうまく稼働していなかったり、サービス拒否攻撃が行われていることもあります。さらに、SMTP サーバプロセスの有効期間は制限されています。これは、dispatcher.cnf ファイルの MTA 設定変数 MAX_LIFE_TIME によって制御されます。デフォルトは 86,400 秒 (1 日) です。同様に、MAX_LIFE_CONNS は、サーバプロセスがその有効期間中に処理できる接続の最大数を指定します。特定の SMTP サーバが長時間稼働している場合は、調査することもできます。

ディスパッチャおよびジョブコントローラのプロセスをモニタする

MTA が機能するためには、ディスパッチャおよびジョブコントローラプロセスが動作している必要があります。種類ごとに 1 つのプロセスが必要です。

ディスパッチャおよびジョブコントローラのプロセスダウンの兆候

ディスパッチャがダウンしていたり十分なリソースがない場合、SMTP 接続は拒否されます。

ジョブコントローラがダウンしている場合、キューのサイズが大きくなります。

ディスパッチャおよびジョブコントローラのプロセスをモニタするには

`dispatcher` および `job_controller` というプロセスが存在しているかどうかチェックします。473 ページの「ジョブコントローラとディスパッチャが実行中であることをチェックする」を参照してください。

メッセージアクセスをモニタする

この節には、以下の項目があります。

imapd、popd、および httpd をモニタする

これらのプロセスによって、IMAP、POP、および Web メールサービスにアクセスします。これらのいずれかが実行されていないか応答がない場合、サービスは正しく機能しません。サービスが実行されていても過負荷の場合は、モニタすることでそれを検出し、より適切に設定し直すことができます。

imapd、popd、および httpd に関する問題の兆候

接続が拒否されたり、システムが遅すぎて接続できません。たとえば、IMAP が実行されていないときに IMAP に直接接続しようとすると、以下のようなメッセージが表示されます。

```
telnet 0 143
Trying 0.0.0.0...
telnet: Unable to connect to remote host: Connection refused
```

クライアントと接続しようとすると、以下のようなメッセージが表示されます。

```
netscape is unable to connect to the server at the location you have
specified. The server may be down or busy.
```

imapd、popd、および httpd をモニタするには

- SNMP によってモニタすることができます。

  SNMP を設定している場合は、これらのプロセスをモニタすることをお勧めします。付録 A「SNMP サポート」を参照してください。サーバ情報は、Network Services Monitoring MIB にあります。

- ログファイルをチェックします。

  msg-instance/log/*service* ディレクトリを確認します。このディレクトリで *service* を http、IMAP、または POP にすることができます。このディレクトリで、ログファイルの数を確認します。1 つは *service* (imap、pop、http) というファイル名です。ほかのファイル名には、サービスの名前、シーケンス番号、およびサービス名に連結された日付が使われます。たとえば、以下のようになります。

  ```
  imap  imap.29.1010221593 imap.31.1010394412  imap.33.1010567224
  ```

  サービス名だけを含むファイルは、最新のログです。それ以外のファイルは、シーケンス番号 (ここでは 29、31、33) 順に並べられ、シーケンス番号が一番大きいファイルが次に新しいファイルです (第 13 章「ログ記録とログ解析」を参照)。

  サーバが停止した場合は、以下のように表示されることがあります。

  ```
  [05/Jan/2002:08:36:38 -0800] gotmail-a imapd[10275]: General
  Warning: iPlanet Messaging Server IMAP4 5.2 (built Dec  9 2001)
  shutting down
  ```

- `counterutil` を使ってチェックできます。516 ページの「counterutil」および『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

- プラットフォーム固有のコマンドを実行して、imapd、popd、および httpd プロセスが実行中かどうかを確認します。たとえば、Solaris では、`ps` コマンドを使用し、`imapd`、`popd`、および `mshttpd` を検索することができます。Windows NT では、タスクマネージャ ウィンドウまたはコマンドラインを使用することができます。

- 515 ページの「推奨される stored パラメータ」に記載されているサーバ応答設定パラメータを設定することによって、指定したサーバのパフォーマンスしきい値に対する警告を設定することができます。

stored をモニタする

stored は、存続期間決定ポリシーを実行したり、ディスクに保存されているメッセージを消去して、メッセージデータベースのデッドロック操作やトランザクション操作などの、さまざまな重要なタスクを実行します。stored が実行を停止すると、最終的には Messaging Server に問題が発生します。start-msg が実行されているときに stored が起動していないと、ほかのプロセスも起動しません。stored の詳細は、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

stored に関する問題の兆候

表面的な問題の兆候はありません。

stored をモニタするには

- stored プロセスが実行中かどうかをチェックします。stored は、pidfile.store という、msg-*instance*/config 内の pid ファイルを作成し、更新します。この pid ファイルは、復元中の init 状態と準備中の ready 状態を示します。たとえば、以下のようになります。

  ```
  231: cat pidfile.store
  28250
  ready
  ```

  1 行目の数字は stored のプロセス ID です。

  ```
  232: ps -eaf | grep stored
  mailsrv 28250     1  0   Jan 05 ?        8:44
  /usr/iplanet/server5/bin/msg/admin/bin/stored -d
  ```

- msg-*instance*/store/mailboxlist 内に作成されたログファイルをチェックします。すべてのログファイルが直接 stored の問題によって作成されるわけではありません。ログファイルは、imapd が壊れている場合やデータベースに問題がある場合にも作成されることがあります。

- msg-*instance*/config にある以下のファイルのタイムスタンプをチェックします。

  stored.ckp - チェックポイントで試行が行われたときに押される。1 分ごとにタイムスタンプが付けられる
  stored.lcu - データベースログのクリーンアップごとに押される。5 分ごとにタイムスタンプが付けられる
  stored.per - ユーザ単位のデータベース書き込み時に押される。60 分ごとにタイムスタンプが付けられる

- デフォルトのログファイル msg-*instance*/log/default/default 内の stored メッセージをチェックします。

LDAP Directory Server をモニタする

この節には、以下の項目があります。

slapd をモニタする

LDAP ディレクトリサーバ (slapd) は、メッセージングシステムのディレクトリ情報を提供します。slapd がダウンしていると、システムは正しく作動しません。slapd 応答時間が遅すぎると、ログイン速度、および LDAP 検索を必要とするほかのトランザクションに影響を及ぼします。

slapd に関する問題の兆候

- クライアント POP、IMAP、または Web メール認証が失敗するか、予定よりも時間がかかる
- MTA が正しく動作しない

slapd をモニタするには

- ns-slapd プロセスが実行中かどうかをチェックします。
- slapd-*instance*/logs/ にある slapd ログファイルの access および errors をチェックします。
- ユーザ検索時の ns-slapd 応答時間をチェックします。
- Admin Console を表示して slapd をモニタします。

メッセージストアをモニタする

メッセージはデータベースに保存されています。ディスク上のユーザの分散、メールボックスのサイズ、ディスクの要件は、ストアのパフォーマンスに影響します。この節には、以下の項目があります。

メッセージストアデータベースのロック状態をモニタする

データベースロックの状態は、さまざまなサーバプロセスで保持されます。これらのデータベースロックは、メッセージストアのパフォーマンスに影響することがあります。デッドロックの場合、メッセージが適切な速度でストアに挿入されないため、結果として *ims-ms* チャネルキューが大きくなります。キューをバックアップするのにはいくつかの正当な理由があります。従って、キューの長さの履歴をとっておくと、問題を診断するのに便利です。

メッセージストアのデータベースロックに関する問題の兆候

多数のトランザクションが蓄積され、解決されません。

メッセージストアのデータベースロックをモニタするには

`counterutil -o db_lock` コマンドを使用します。

mboxutil ディレクトリ内のデータベースログファイルの数をモニタする

データベースログファイルは、sleepycat トランザクションのチェックポイントログファイル (`msg-`*instance*`/store/mboxlist`) を指します。作成されるログファイルは、データベースのチェックポイントが発生しないという問題の兆候です。また、`stored` の問題による場合もあります。

データベースログファイルの問題の兆候

通常は、2 つまたは 3 つのログファイルがあります。ログファイルがそれ以上ある場合は、潜在的に重大な問題があることを示しています。メッセージストアはメッセージと制限容量のためにいくつかのデータベースを使用します。それらに問題があるとすべてのメールサーバに問題が発生することがあります。

データベースログファイルをモニタするには

`msg-`*instance*`/store/mboxlist` ディレクトリを見て、ファイルが 2 つか 3 つしかないことを確認します。

モニタ用のユーティリティとツール

モニタには、以下のツールを利用できます。

stored

`stored` ユーティリティはサーバ上で保守タスクを実行しますが、モニタも実行できます。指定されている場合は、サーバの状態、ディスク容量、サービスへの応答時間を定期的にチェックでき、ポストマスターへの電子メールメッセージの形式で警告を発することができます (510 ページを参照)。

警告は、電子メールメッセージの形式で、`stored` からポストマスターに送られ、指定された状態を警告します。一定のしきい値を超えたときに `stored` が送信する電子メール警告のサンプルを以下に示します。

Subject: ALARM: server response time in seconds of "ldap_siroe.com_389" is 10
Date: Tue, 17 Jul 2001 16:37:08 -0700 (PDT)
From: postmaster@siroe.com
To: postmaster@siroe.com

Server instance: /usr/iplanet/server5/msg-europa
Alarmid: serverresponse
Instance: ldap_siroe_europa.com_389
Description: server response time in seconds
Current measured value (17/Jul/2001:16:37:08 -0700): 10
Lowest recorded value: 0
Highest recorded value: 10
Monitoring interval: 600 seconds
Alarm condition is when over threshold of 10
Number of times over threshold: 1

stored でディスクおよびサーバのパフォーマンスをモニタする頻度と、どのような状況下で警告を送るかを指定することができます。これは、configutil コマンドを使用して警告パラメータを設定することによって行います。有用な stored パラメータとそのデフォルト設定を表 15-1 に示します。

表 15-1 推奨される stored パラメータ

パラメータ	説明 (デフォルト)
alarm.msgalarmnoticehost	(localhost) 警告メッセージの送信先のマシン
alarm.msgalarmnoticeport	(25) 警告メッセージの送信時に接続する SMTP ポート
alarm.msgalarmnoticercpt	(Postmaster@localhost) 警告通知の送信先
alarm.msgalarmnoticesender	(Postmaster@localhost) 警告の差出人のアドレス
alarm.diskavail.msgalarmdescription	ディスク利用度警告の説明
alarm.diskavail.msgalarmstatinterval	(3600) ディスク利用度のチェック間隔 (秒)。ディスク使用状況をチェックしない場合は、0 に設定する
alarm.diskavail.msgalarmthreshold	(10) 利用可能なディスク容量の割合。この値を下回ると警告が送信される
alarm.diskavail.msgalarmthresholddirection	(-1) 利用可能なディスク容量がしきい値 (-1) より低いか、しきい値 (1) より高いときに警告を発行するかどうかを指定する
alarm.diskavail.msgalarmwarninginterval	(24). ディスク利用度警告が繰り返される間隔 (時)
alarm.serverresponse.msgalarmdescription	サーバ応答警告の説明
alarm.serverresponse.msgalarmstatinterval	(600) サーバ応答のチェックの間隔 (秒)。サーバの応答をチェックしない場合は、0 に設定する
alarm.serverresponse.msgalarmthreshold	(10) サーバ応答時間 (秒) がこの値を超えると、警告が発行される
alarm.serverresponse.msgalarmthresholddirection	(1) サーバ応答時間がしきい値より大きい (1) か、しきい値より小さい (-1) ときに、警告を発行するかどうかを指定する
alarm.serverresponse.msgalarmwarninginterval	(24) サーバ応答警告が繰り返される間隔 (時)。

counterutil

このユーティリティは、さまざまなシステムカウンタから取得した統計情報を提供します。以下は、現在利用できるカウンタオブジェクトのリストです。

```
counterutil -l
entry = alarm
entry = diskusage
entry = serverresponse
entry = db_lock
entry = db_log
entry = db_mpool
entry = db_txn
entry = popstat
entry = imapstat
entry = httpstat
entry = cgimsg
```

それぞれのエントリはカウンタオブジェクトを表し、このオブジェクトに使用できるさまざまなカウントを提供します。この節では、alarm、diskusage、serverresponse、db_lock、popstat、imapstat、および httpstat カウンタオブジェクトについてのみ説明します。counterutil コマンドの使用法の詳細は、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

counterutil の出力

`counterutil` にはさまざまなフラグがあります。このユーティリティのコマンドの形式は次のとおりです。

```
counterutil -o CounterObject -i 5 -n 10
```

ここで、

`-o` *CounterObject* は、カウンタオブジェクト `alarm`、`diskusage`、`serverresponse`、`db_lock`、`popstat`、`imapstat`、および `httpstat` を表します。

`-i 5` は、5 秒の間隔を指定します。

`-n 10` は、反復回数 (デフォルト : 無限) を表します。

`counterutil` の使用例を以下に示します。

```
counterutil -o imapstat -i 5 -n 10
Monitor counteroobject (imapstat)
registry /gotmail/iplanet/server5/msg-gotmail/counter/counter opened
counterobject imapstat opened

count = 1 at 972082466 rh = 0xc0990 oh = 0xc0968
```

```
global.currentStartTime [4 bytes]: 17/Oct/2000:12:44:23 -0700
global.lastConnectionTime [4 bytes]: 20/Oct/2000:15:53:37 -0700
global.maxConnections [4 bytes]: 69
global.numConnections [4 bytes]: 12480
global.numCurrentConnections [4 bytes]: 48
global.numFailedConnections [4 bytes]: 0
global.numFailedLogins [4 bytes]: 15
global.numGoodLogins [4 bytes]: 10446
  ...
```

counterutil を使用した警告統計

これらの警告統計は、stored が送信する警告を指します。警告カウンタは以下の統計を提供します。

表 15-2　counterutil alarm 統計

接尾辞	説明
alarm.countoverthreshold	しきい値を超えた回数
alarm.countwarningsent	送信された警告の数
alarm.current	現在のモニタ値
alarm.high	これまでに記録された最高値
alarm.low	これまでに記録された最低値
alarm.timelastset	最後に現在の値が設定された時間
alarm.timelastwarning	最後に警告が送信された時間
alarm.timereset	最後にリセットが行われた時間
alarm.timestatechanged	最後に警告状態が変わった時間
alarm.warningstate	警告状態 (yes(1) または no(0))

counterutil を使用した IMAP、POP、および HTTP 接続の統計

現在の IMAP、POP、および HTTP 接続の数、ログインに失敗した回数、開始時間からの接続合計などの情報を得るために、コマンド `counterutil -o` *CounterObject* `-i 5 -n 10` を使用することができます。ここで、*CounterObject* は、カウンタオブジェクト `popstat`、`imapstat`、または `httpstat` を表します。`imapstat` 接尾辞の意味を 表 15-3 に示します。`popstat` および `httpstat` オブジェクトは、同じ情報を同じ形式と構造で提供します。

表 15-3　`counterutil imapstat` 統計

接尾辞	説明
`currentStartTime`	現在の IMAP サーバプロセスの開始時間
`lastConnectionTime`	最後に新しいクライアントが受け入れられた時間
`maxConnections`	IMAP サーバが処理した同時接続の最大数
`numConnections`	現在の IMAP サーバが処理した接続の総数
`numCurrentConnections`	アクティブな接続の現在の数
`numFailedConnections`	現在の IMAP サーバが処理した失敗した接続の数
`numFailedLogins`	現在の IMAP サーバが処理した失敗したログインの数
`numGoodLogins`	現在の IMAP サーバが処理した成功したログインの数

counterutil を使用したディスク使用状況の統計

コマンド `counterutil -o diskusage` は以下の情報を生成します。

表 15-4　`counterutil diskstat` 統計

接尾辞	説明
`diskusage.availSpace`	ディスクパーティションで利用できる合計容量
`diskusage.lastStatTime`	最後に統計がとられた時間
`diskusage.mailPartitionPath`	メールパーティションのパス
`diskusage.percentAvail`	利用できるディスクパーティション容量の割合
`diskusage.totalSpace`	ディスクパーティションの合計容量

サーバ応答の統計

コマンド counterutil -o serverresponse は、以下の情報を生成します。この情報は、サーバが稼働中かどうかと、サーバの応答速度をチェックする際に便利です。

表 15-5 counterutil serverresponse 統計

接尾辞	説明
http.laststattime	最後に http サーバ応答がチェックされた時間
http.responsetime	http の応答時間
imap.laststattime	最後に imap サーバ応答がチェックされた時間
imap.responsetime	imap の応答時間
pop.laststattime	最後に pop サーバ応答がチェックされた時間
pop.responsetime	pop の応答時間
ldap_host1_389.laststattime	最後に ldap_host1_389 サーバ応答がチェックされた時間
ldap_host1_389.responsetime	ldap_host1_389 の応答時間
ugldap_host2_389.laststattime	最後に ugldap_host2_389 サーバ応答がチェックされた時間
ugldap_host2_389.responsetime	ugldap_host2_389 の応答時間

ログファイル

Messaging Server は、SMTP、IMAP、POP、および HTTP のイベント記録をログに保存します。Messaging Server ログファイルの作成と管理用のポリシーはカスタマイズ可能です。

ログ記録はサーバのパフォーマンスに影響を与えることがあるため、サーバに負担がかからないよう、非常に慎重に検討する必要があります。詳細については、第 13 章「ログ記録とログ解析」を参照してください。

imsimta counters

MTA は、アクティブなチャネルのそれぞれに対して、Mail Monitoring MIB (RFC 1566) に基づいてメッセージトラフィックのカウンタを累積します。チャネルカウンタは、使用している電子メールシステムの傾向や調子を示すためのものです。チャネルカウンタは、メッセージトラフィックを正確に計算するためのものではありません。正確な計算については、第 13 章「ログ記録とログ解析」に記載されている MTA ログを参照してください。

MTA チャネルカウンタは、利用可能な最軽量メカニズムを使用して実装されるため、実際の操作での影響はわずかです。チャネルカウンタはさらに処理を行おうとはしません。つまり、セクションのマッピングの試行が失敗した場合やセクション内のロックの 1 つをほぼ即座に取得できない場合は、情報が記録されず、システムが停止している場合は、メモリ内セクションに含まれている情報は永久に失われます。

imsimta counters -show コマンドによって MTA チャネルメッセージの統計が得られます (以下を参照)。最小値が何も示されないときは、これらのカウンタを調べる必要があります。チャネルによっては、実際の最小値は負の値です。負の値は、カウンタがゼロになった (たとえば、カウンタのクラスタレベルのデータベースが作成された) 時点でチャネルのキューに入れられたメッセージがあることを示します。これらのメッセージがキューから取り出されるとき、関連するチャネルのカウンタは減少し、それによって最小値が負になります。このようなカウンタの場合、正確な「絶対」値は、初期化以降にカウンタが保持していた最小値を差し引いた現在の値です。

```
Channel           Messages    Recipients    Blocks
-------           --------    ----------    -------
tcp_local
   Received        29379        79714       982252                          (1)
   Stored             61          113        -2004                          (2)
   Delivered       29369        79723       983903 (29369 first time)       (3)
   Submitted       13698        13699        18261                          (4)
   Attempted           0            0            0                          (5)
   Rejected            1           10            0                          (6)
   Failed            104          104         4681                          (7)

   Queue time/count         16425/29440 = 0.56                               (8)
   Queue first time/count   16425/29440 = 0.56                               (9)

   Total In Assocs            297637
   Total Out Assocs            28306
```

1) Received は、tcp_local という名前のチャネルのキューに入れられたメッセージの数です。つまり、ほかのチャネルによって directory チャネルのキューに入れられたメッセージ (mail.log* ファイル内の E レコード) です。

2) Stored は、チャネルキューに保存された配信されるメッセージの数です。

3) Delivered は、チャネル tcp_local によって処理された (キューから取り出された) メッセージの数です (つまり、mail.log* ファイル内の D レコード)。キューからの取り出しとは、正常な配信 (別のチャネルのキューに入れること) か、またはメッセージが差出人に戻ってきたためにキューから取り出すことのいずれかを指します。通常これは、Received の数から Stored の数を引いた数に相当します。

MTA は、最初の試行でキューから取り出されたメッセージ数も記録します。この数は括弧で囲んで示されます。

4) Submitted は、チャネル tcp_local によって別のチャネルのキューに入れられたメッセージ (mail.log ファイル内の E レコード) の数です。

5) Attempted は、キューから取り出す際に一時的な問題が発生したメッセージ (mail.log* ファイル内の Q または Z レコード) の数です。

6) Rejected は、拒否されたキューからの取り出し試行 (mail.log* ファイル内の J レコード) の数です。

7) Failed は、失敗したキューからの取り出し試行 (mail.log* ファイル内の R レコード) の数です。

8) Queue time/count は、配信されるメッセージがキューに入っていた時間の平均時間です。これは、最初の試行で配信されたメッセージ (9) を参照) と、追加の配信試行が必要になった (通常はそのためにキューの中で長い間待機している) メッセージの両方が対象になっています。

9) Queue first time/count は、最初の試行で配信されたメッセージがキューに入っていた時間の平均時間です。

送信されたメッセージの数が配信されたメッセージの数より大きくなっていることに注意してください。この原因のほとんどは、メッセージがチャネルのキューから取り出される (配信される) たびに少なくとも 1 つ (場合によっては複数) の新しいメッセージがキューに入れられる (送信される) ためです。たとえば、メッセージが異なるチャネル経由で 2 人の受取人に届けられる場合は、2 つのメッセージがキューに入れられる必要があります。すなわち、メッセージがバウンスされる場合は、差出人にコピーが返送され、もう 1 つのコピーがポストマスターに送信されることがあります。通常は 2 件の送信になります (両方とも同じチャネル経由で届けられる場合を除く)。

通常は、Submitted と Delivered の間の接続はチャネルのタイプによって異なります。たとえば、変換チャネルでは、メッセージはほかの任意のチャネルのキューに入れられ、そのあと変換チャネルがそのメッセージを処理し、それを 3 番目のチャネルのキューに入れ、元のチャネルのキューから取り出されたものとしてメッセージをマークします。個々のメッセージのパスは以下のとおりです。

```
ほかの場所 -> 変換チャネル    E レコード   Received
変換チャネル -> ほかの場所    E レコード   Submitted
変換チャネル                  D レコード   Delivered
```

ただし、tcp_local のように「通過」しなくても 2 つの部分 (スレーブとマスター) があるチャネルの場合は、Submitted と Delivered の間の接続はありません。Submitted カウンタが tcp_local チャネルの SMTP サーバ部分を処理する必要があるのに対し、Delivered は tcp_local チャネルの SMTP クライアント部分を処理する必要があります。これらは 2 つのまったく別のプログラムであり、それぞれから送られるメッセージはまったく別のものになることがあります。

SMTP サーバ に送信されるメッセージ

```
tcp_local -> ほかの場所  E レコード    Submitted
```

SMTP クライアント経由でほかの SMTP ホストに送信されるメッセージ

```
ほかの場所 -> tcp_local  E レコード    Received
tcp_local               D レコード    Delivered
```

チャネルのキューからの取り出し (配信) により、少なくとも 1 つの新しいメッセージがキューに入れられ (送信され) ます。複数になることもあります。たとえば、メッセージが異なるチャネル経由で 2 人の受取人に届けられる場合は、2 つのメッセージがキューに入れられる必要があります。すなわち、メッセージがバウンスされる場合は、差出人にコピーが返送され、もう 1 つのコピーがポストマスターに送信されることがあります。通常は同じチャネル経由で届けられます。

UNIX および NT での実装

パフォーマンス上の理由から、MTA はメモリ内にチャネルカウンタのキャッシュを保持します。これには、UNIX では共有メモリセクションを使用し、NT では共有ファイルマッピングオブジェクトを使用します。ノード上のプロセスがメッセージをキューに入れたりキューから取り出すときに、このプロセスがそのメモリ内キャッシュ内のカウンタを更新します。チャネルが作動しているときにメモリ内セクションが存在しない場合、メモリ内セクションは自動的に作成されます (imta start コマンドも、存在しない場合はメモリ内キャッシュを作成します)。

imta counters -clear コマンドまたは imta qm コマンドの counters clear は、カウンタをゼロにリセットするために使用することもあります。

imsimta qm counters

imsimta qm counters ユーティリティは、MTA チャネルのキューメッセージカウンタを表示します。　このユーティリティは、root または mailsrv として実行する必要があります。出力されるフィールドは 519 ページの「imsimta counters」に記載されているものと同じです。使用法の詳細は、『iPlanet Messaging Server リファレンスマニュアル』を参照してください。

例 1 :

```
imsimta qm counters show
```

```
Channel                 Messages     Recipients    Blocks
----------------------  ----------  ----------    ----------
autoreply
   Received              13077        13859         264616
   Stored                   92           91           -362
   Delivered             12985        13768         264978
   Submitted              2594         2594           3641
...
```

例 2 :

```
imsimta qm counters today
```

```
4370 messages processed so far today
Your license permits an unlimited number of messages per day.
```

SNMP を使用した MTA のモニタ

iPlanet Messaging Server では、SNMP (Simple Network Management Protocol) を利用したシステムモニタ機能がサポートされています。Sun Net Manager や HP OpenView などの SNMP クライアント (ネットワークマネージャとも呼ばれる) を使って、iPlanet Messaging Server の特定の部分をモニタすることができます。ただし、SNMP クライアントは本製品に付属していません。詳細については付録 A「SNMP サポート」を参照してください。

メールボックスの制限容量チェックのための mboxutil

mboxutil ユーティリティを使用して、メールボックスの制限容量の使用状況と制限をモニタすることができます。mboxutil ユーティリティは、定義されている制限容量と制限をリストし、制限容量に関する情報を出力する、レポートを生成します。制限容量と使用状況はキロバイトでレポートされます。

たとえば、以下のコマンドはすべてのユーザの制限容量情報を一覧表示します。

```
% mboxutil -a
------------------------------------------------------------------------
Domain red.siroe.com (diskquota = not set msgquota = not set) quota usage
------------------------------------------------------------------------
diskquota        size(K)    %use    msgquota      msgs    %use    user
# of domains = 1
# of users = 705

no quota         50418              no quota      4392            ajonkish
no quota         5                  no quota      2               andrewt
no quota         355518             no quota      2500            aniksri
...
```

以下の例では、ユーザ sorook の制限容量の使用状況を示します。

```
% mboxutil -u sorook
------------------------------------------------------------------------
quota usage for user sorook
------------------------------------------------------------------------
diskquota      size(K)    %use    msgquota      msgs    %use    user

no quota       1487                no quota      305               sorook
```

付録 A

SNMP サポート

iPlanet Messaging Server では、SNMP (Simple Network Management Protocol) を利用したシステムモニタ機能がサポートされています。Sun Net Manager や HP OpenView などの SNMP クライアント (ネットワークマネージャとも呼ばれる) を使って、iPlanet Messaging Server の特定の部分をモニタすることができます。ただし、SNMP クライアントは本製品に付属していません。iPlanet Messaging Server のモニタの詳細は、第 15 章「iPlanet Messaging Server をモニタする」を参照してください。

この章では、Messaging Server の SNMP サポートを使用する方法について説明します。また、SNMP から得られる情報の種類についても簡単に説明します。ただし、この章では、それらの情報を表示する方法については取り上げていません。SNMP クライアントを使って SNMP ベースの情報を表示する方法については、SNMP クライアントのマニュアルを参照してください。このマニュアルには、Messaging Server の SNMP 実装で使用できるデータの一部も紹介されています。MIB の詳細は、RFC 2788 および RFC 2789 を参照してください。

この章には、以下の節があります。

SNMP の実装

iPlanet Messaging Server には、Network Services Monitoring MIB (RFC 2788) と Mail Monitoring MIB (RFC 2789) という 2 つの標準化された MIB が実装されています。Network Services Monitoring MIB は POP、IMAP、HTTP、SMTP などのサーバのネットワークサービスをモニタするためのもので、Mail Monitoring MIB は MTA をモニタするためのものです。Mail Monitoring MIB では、各 MTA チャネルのアクティブ状態と、その履歴をモニタすることができます。アクティブ状態のモニタでは、現在キュー内にあるメッセージとオープンなネットワーク接続に焦点があてられます。たとえば、キュー内にあるメッセージの数や、オープンなネットワーク接続のソース IP アドレスなどです。一方、履歴のモニタからは、累積による統計が提供されます。たとえば、処理したメッセージの合計数や、受信接続の合計数などです。

注	Messaging Server SNMP モニタ機能の詳細は、RFC 2788 および RFC 2789 を参照してください。

SNMP がサポートされているのは Solaris 8 プラットフォームだけです。このほかのプラットフォームについては、今後のリリースで SNMP をサポートする予定です。Solaris での SNMP サポートは、Solaris の原初 SNMP テクノロジーである Solstice Enterprise Agents (SEA) を利用しています。Solaris 8 システムに SEA をインストールする必要はありません。必要なランタイムライブラリはすでにインストールされています。

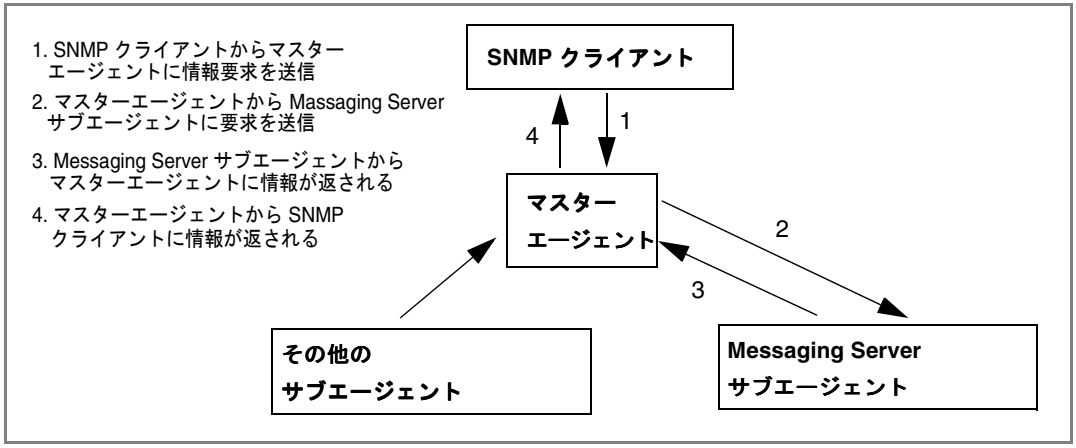
Messaging Server SNMP サポートには、次のような制限があります。

- SNMP を通じてモニタできる Messaging Server のインスタンスは、ホストコンピュータ当たり 1 つのみである。
- SNMP サポートは、モニタ用のみである。SNMP 管理はサポートされていない。
- SNMP トラップは実装されない (RFC 2788 に、トラップを使用しない同様の機能が記述されています)。

Messaging Server での SNMP の動作

Solaris プラットフォームでは、Messaging Server SNMP プロセスは SNMP サブエージェントであり、起動時にプラットフォームの原初 SNMP マスターエージェントに自動的に登録されます。クライアントからの SNMP 要求は、マスターエージェントに送られます。次に Messaging Server 宛ての要求は、マスターエージェントから Messaging Server サブエージェントプロセスに送られます。最後に Messaging Server サブエージェントプロセスによって要求が処理され、その応答がマスターエージェントを通じてクライアントに送られます。図 A-1 に、このプロセスを示します。

図 A-1　　SNMP の情報フロー



Solaris 8 で iPlanet Messaging Server 用の SNMP サポートを設定する

SNMP モニタ機能によって生じるオーバーヘッドは非常に小さなものですが、Messaging Server は SNMP サポートを無効にした状態で出荷されています。SNMP サポートを有効にするには、次のコマンドを実行します。

```
# su user-id-for-ims
# configutil -o local.snmp.enable -v 1
# start-msg snmp
```

SNMP を有効にすると、パラメータを指定せずに `start-msg` コマンドを実行するだけで、SNMP サブエージェントプロセスがその他の Messaging Server プロセスとともに自動的に起動するようになります。

Messaging Server SNMP サブエージェントが動作するためには、Solaris の原初 SNMP マスターエージェントが実行されていなければなりません。Solaris の原初 SNMP マスターエージェントは `snmpdx` デーモンであり、通常これは Solaris の起動プロセスの一部として起動します。

要求を受信する UDP ポートは、SNMP サブエージェントによって自動的に選択されます。必要であれば、次のコマンドを使ってサブエージェントに固定の UDP ポートを割り当てることもできます。

```
# configutil -o local.snmp.port -v port-number
```

この設定は、あとでポート番号にゼロを指定することによって取り消すことができます。ゼロ (デフォルト) に指定すると、Messaging Server により、サブエージェントが使用可能な任意の UDP ポートを自動的に選択することが許可されます。

/etc/snmp/conf ディレクトリには、2 つの SNMP サブエージェント設定ファイルがあります。1 つは SNMP アクセス制御情報を含む ims.acl で、もう 1 つは SNMP MIB OID 登録情報を含む ims.reg です。

通常、これらのファイルを編集する必要はありません。Messaging Server によって提供される MIB は読み取り専用で、ims.reg ファイルでポート番号を指定する必要はありません。ポート番号を指定した場合は、configutil ユーティリティでもポート番号を設定した場合を除き、ここで指定した値が使用されます。configutil でポート番号を設定した場合は、そのポート番号がサブエージェントで使用されます。これらのファイルを編集した場合は、変更を反映させるために SNMP サブエージェントをいったん停止してから再起動する必要があります。

```
# stop-msg snmp
# start-msg snmp
```

Windows プラットフォーム用の SNMP を設定する

SNMP モニタ機能によって生じるオーバーヘッドは非常に小さなものですが、Messaging Server は SNMP サポートを無効にした状態で出荷されています。SNMP サポートを有効にするには、DOS プロンプトから以下のコマンドを発行します。

```
X:¥> server_root¥msg-instance¥configutil /o local.snmp.enable /v 1
X:¥> %SYSTEMROOT%¥SYSTEM32¥regsvr32.exe server_root¥bin¥msg¥imta¥bin¥madmand.dll
```

次に、Windows サービスユーティリティを使って SNMP サービスを再起動します。このサービスユーティリティは、Microsoft 管理コンソール と呼ばれることもあります。

Messaging Server SNMP サポートが動作するためには、Windows SNMP サービスが実行されていなければなりません。デフォルトでは、Windows SNMP サービスは Windows NT とともにインストールされません。Windows SNMP サービスは手作業でインストールする必要があります。

Windows NT で、以下のように SNMP サービスをインストールします。

1. **コントロールパネル**で、「**ネットワーク**」アイコンをマウスの右ボタンでクリックします。
2. 「**ネットワーク**」ウィンドウで、「**サービス**」タブを選択します。

3. 「**サービス**」ダイアログで、「**追加**」ボタンをクリックします。
4. 「**ネットワーク サービスの選択**」ポップアップウィンドウの「**ネットワークサービス**」リストボックスで、「**SNMP サービス**」を選択します。次に、「**OK**」ボタンをクリックします。
5. これで、Windows で SNMP サービスがインストールされます。インストールを完了するために Windows NT CD-ROM が必要になることもあります。

SNMP サービスのインストールの詳細は、Microsoft の Windows のマニュアルを参照してください。

Messaging Server の SNMP サポートを無効にするには、以下のコマンドを発行します。

```
X:¥> server_root¥msg-instance¥configutil /o local.snmp.enable /v 0
X:¥> %SYSTEMROOT%¥SYSTEM32¥regsvr32.exe /u server_root¥bin¥msg¥imta¥bin¥madmand.dll
```

次に、Windows サービスユーティリティから SNMP サービスを再起動します。

Windows プラットフォームでは、`start-msg snmp` コマンドと `stop-msg snmp` コマンドには効力はありません。Messaging Server の SNMP サポートは Windows SNMP サービス内で実行されます。SNMP サポートを開始または停止するには、Windows SNMP サービスを開始または停止する必要があります。

SNMP クライアントからモニタする

RFC 2788 および RFC 2789 のベース OID は次のとおりです。

mib-2.27 = 1.3.6.1.2.10.27

mib-2.28 = 1.3.6.1.2.1.28

SNMP クライアントをこれら 2 つの OID にポイントし、SNMP コミュニティに「パブリック」としてアクセスします。

お使いの SNMP クライアントに MIB のコピーを読み込みたい場合は、*<server_root>*`/plugins/snmp` ディレクトリにある ASCII 版の MIB を利用できます。ファイル名は `rfc2788.mib` と `rfc2789.mib` です。これらの MIB を SNMP クライアントソフトウェアに読み込む方法については、SNMP クライアントソフトウェアのマニュアルを参照してください。これらの MIB で使用される `SnmpAdminString` データタイプは、古いバージョンの SNMP クライアントで認識されないことがあります。その場合には、同じディレクトリにある `rfc2248.mib` と `rfc2249.mib` を使用してください。

Unix プラットフォームにおける他の iPlanet 製品との共存

SNMP サポートが提供されている他の Netscape 製品または iPlanet 製品では、プラットフォームの原初マスターエージェントを置き換えて SNMP サポートを有効にします。これらの iPlanet 製品を Messaging Server と同じホストで実行し、両者を SNMP でモニタする場合は、『iPlanet Managing Servers with Netscape Console』の第 7 章 (http://docs.iplanet.com/docs/manuals/console/42/html/7_snmp.htm#1024620) の説明に従って iPlanet Proxy SNMP Agent を設定します。これにより、Messaging Server SNMP サブエージェント (原初 SNMP エージェント) が他の iPlanet 製品の非原初 iPlanet SNMP サブエージェントと共存できるようになります。

Messaging Server の SNMP の情報

この節では、SNMP を通じて提供される Messaging Server 情報について簡単に説明します。詳細は、RFC 2788 および RFC 2789 で個々の MIB テーブルを参照してください。RFC/MIB の用語では、メッセージングサービス (MTA、HTTP など) がアプリケーション (`appl`)、Messaging Server ネットワーク接続がアソシエーション (`assoc`)、および MTA チャネルが MTA グループ (`mtaGroups`) と呼ばれていることに注意してください。

Messaging Server の複数のインスタンスを同時にモニタできるプラットフォームでは、`applTable` に複数の MTA とサーバのセット、また他のテーブルに複数の MTA が存在する場合があります。

注	MIB でレポートされる累積値 (配信済みメッセージの合計数や、IMAP 接続の合計数など) は、再起動時、ゼロにリセットされます。

各サイトには、モニタに関してそれぞれ異なるしきい値と重要な値があります。うまく機能している SNMP クライアントでは、傾向の分析を行い、過去の傾向から急にそれた場合に警告を送信することができます。

applTable

applTable には、サーバ情報があります。これは 1 次元のテーブルであり、MTA の行が 1 つと、WebMail HTTP、IMAP、POP、SMTP、および SMTP 送信サーバが有効の場合は、これらに対応する行がそれぞれ 1 つずつ含まれています。このテーブルには、バージョン情報、作動時間、現在の動作のステータス (up、down、congested)、現在の接続数、接続の累積合計数、およびその他の関連するデータがあります。

以下に、applTable (mib-2,27.1.1) のデータ例を示します。

```
applTable:

    applName.1[1] = mailsrv-1[2] MTA on mailsrv-1.west.sesta.com
    applVersion.1 = 5.1
    applUptime.1 = 7322[3]
    applOperStatus.1 = up[4]
    applLastChange.1 = 7422[3]
    applInboundAssociations.1 = 5
    applOutboundAssociations.1 = 2
    applAccumulatedInboundAssociations.1 = 873
    applAccumulatedOutboundAssociations.1 = 234
    applLastInboundActivity.1 = 1054822[3]
    applLastOutboundActivity.1 = 1054222[3]
    applRejectedInboundAssociations.1 = 0[5]
    applFailedOutboundAssociations.1 = 17
    applDescription.1 = iPlanet Messaging Server 5.1
    applName.2[1] = mailsrv-1 HTTP WebMail server on mailsrv-1.west.sesta.com
    ...
    applName.3 = mailsrv-1 IMAP server on mailsrv-1.west.sesta.com
    ...
    applName.4 = mailsrv-1 POP server on mailsrv-1.west.sesta.com
    ...
    applName.5 = mailsrv-1 SMTP server on mailsrv-1.west.sesta.com
    ...
    applName.6 = mailsrv-1 SMTP Submit server on mailsrv-1.west.sesta.com
    ...
```

注 :

1. 上の例の .1、.2 などの接尾辞は行番号 (applIndex) です。applIndex の値は、MTA に対しては値 1、HTTP サーバに対しては値 2 というように決められています。したがって、上の例では、テーブルの最初の行は MTA のデータを、2 番目の接尾辞がある行は HTTP サーバのデータを提供しています。
2. モニタしている Messaging Server インスタンスの名前です。上の例の場合、インスタンス名は「mailsrv-1」です。
3. これらは SNMP TimeStamp 値で、イベント発生時の sysUpTime の値です。一方 sysUpTime は、SNMP マスターエージェントが起動してから経過した時間で、100 分の 1 秒を単位とする値です。

4. HTTP、IMAP、POP、SMTP、および SMTP 送信サーバの動作ステータスは、それぞれのサーバに設定された TCP ポートを通じて実際にこれらのサーバに接続し、適切なプロトコル (たとえば、HTTP では HEAD 要求と応答、SMTP では HELO コマンドと応答など) で簡単な処理を行うことにより決定されます。この接続試行によって、各サーバのステータス (up (1)、down (2)、または congested (4)) が決定されます。

   これらの試みは、サーバに対する通常の受信接続として認識され、各サーバの applAccumulatedInboundAssociations MIB 変数に影響を与えます。

   MTA の場合、動作ステータスはジョブコントローラのステータスとなります。MTA が稼働中として表示された場合は、ジョブコントローラが起動していることになります。また、MTA が非稼働中として表示された場合は、ジョブコントローラが停止していることになります。この MTA の動作ステータスは、MTA のサービスディスパッチャのステータスには左右されません。MTA の動作ステータスは、up または down の値だけをとります。ジョブコントローラに「congested (混雑)」という概念があるとは言え、MTA のステータスにこの状態が表示されることはありません。

5. HTTP、IMAP、および POP サーバの場合、applRejectedInboundAssociations MIB 変数は、拒否された受信接続の数ではなく、失敗したログイン試行の回数を示します。

applTable の使用法

各サーバをモニタする上で重要なことは、リストされているアプリケーションのそれぞれについてサーバステータス (applOperStatus) をモニタするということです。

applLastInboundActivity に示されている最後の受信アクティビティから長い時間が経過している場合は、何かの不具合が発生して接続が切断されている可能性があります。applOperStatus=2 (down) の場合は、モニタ中のサービスが稼働していません。applOperStatus=1 (up) の場合は、ほかに問題があることが考えられます。

assocTable

このテーブルには、MTA に対するネットワーク接続情報が表示されます。これは 2 次元のテーブルで、アクティブな各ネットワーク接続に関する情報があります。他のサーバに関する接続情報は提供されません。

以下に、applTable (mib-2.27.2.1) のデータ例を示します。

```
assocTable:

    assocRemoteApplication.1.1[1] = 129.146.198.167[2]
    assocApplicationProtocol.1.1[1] = applTCPProtoID.25[3]
```

```
    assocApplicationType.1.1 = peerinitiator(3)[4]
    assocDuration.1.1 = 400[5]
...
```

注 :

1. .x.y という形式の接尾辞では、x はアプリケーションインデックス (applIndex) であり、applTable のどのアプリケーションがレポートされているかを示します。この場合は MTA です。y の部分には、レポートされているアプリケーションの各接続が列挙されます。
2. リモート SMTP クライアントのソース IP アドレスです。
3. ネットワーク接続で使用されているプロトコルを示す OID です。aplTCPProtoID は TCP プロトコルを意味します。.n は使用中の TCP ポートを表す接尾辞で、.25 は TCP ポート 25 で使用されているプロトコルである SMTP を示しています。
4. リモート SMTP クライアントがユーザエージェント (UA) であるか、またはその他の MTA であるかを知ることはできません。このため、サブエージェントは常に peer-initiator をレポートし、ua-initiator をレポートすることはありません。
5. これは SNMP TimeInterval で、その単位は 100 分の 1 秒です。上の例では、接続を開始してから 4 秒が経過しています。

assocTable の使用法

このテーブルは、アクティブな問題を診断するために使用されます。たとえば、急に 200,000 個の受信接続が発生した場合など、このテーブルで接続元を確認することができます。

mtaTable

これは 1 次元のテーブルで、applTable の各 MTA に対してそれぞれ 1 つの行があります。各行には、mtaGroupTable で選択された変数に対し、その MTA 内のすべてのチャネル (グループと呼ばれる) における合計が示されます。

以下に、applTable (mib-2.28.1.1) のデータ例を示します。

```
mtaTable:

    mtaReceivedMessages.1[1] = 172778
    mtaStoredMessages.1 = 19
    mtaTransmittedMessages.1 = 172815
```

```
mtaReceivedVolume.1 = 3817744
mtaStoredVolume.1 = 34
mtaTransmittedVolume.1 = 3791155
mtaReceivedRecipients.1 = 190055
mtaStoredRecipients.1 = 21
mtaTransmittedRecipients.1 = 3791134
mtaSuccessfulConvertedMessages.1 = 0[2]
mtaFailedConvertedMessages.1 = 0
mtaLoopsDetected.1 = 0[3]
```

注 :

1. .x という接尾辞は、applTable におけるアプリケーションの行番号を示します。上の例の .1 は、このデータが applTable 内にある最初のアプリケーションのものであることを意味しています。つまり、このデータは MTA に関するものです。
2. 変換チャネルは、ゼロ以外の値しかとりません。
3. 現在 MTA のメッセージキューに保管されている .HELD メッセージファイルの数をカウントします。

mtaTable の使用法

mtaLoopsDetected がゼロでない場合は、メールのループ問題があります。問題を解決するために、MTA キューの .HELD ファイルを見つけ、診断します。

システムが変換チャネルを使ってウィルススキャンを行い、ウィルスに感染したメッセージを拒否した場合は、mtaSuccessfulConvertedMessages によって、感染したメッセージの数と、その他の変換失敗の数がレポートされます。

mtaGroupTable

この 2 次元のテーブルには、applTable 内の各 MTA に対するチャネル情報があります。この情報には、保存された (キュー内にある) メッセージ数や、配信されたメールメッセージ数などのデータが含まれています。各チャネルに対して保存されたメッセージの数 (mtaGroupStoredMessages) をモニタすることは、とても重要です。この値が通常の範囲を超えて大きくなった場合は、メールがキュー内にたまっています。

以下に、mtaGroupTable (mib-2.28.2.1) のデータ例を示します。

```
mtaGroupTable:

mtaGroupName.1.1[1] = autoreply[2]
        ...
mtaGroupName.1.2[1] = ims-ms
        ...
```

```
mtaGroupName.1.3[1] = tcp_local
    mtaGroupDescription.1.3 = mailsrv-1 MTA tcp_local channel
    mtaGroupReceivedMessages.1.3 = 12154
    mtaGroupRejectedMessages.1.3 = 0
    mtaGroupStoredMessages.1.3 = 2
    mtaGroupTransmittedMessages.1.3 = 12148
    mtaGroupReceivedVolume.1.3 = 622135
    mtaGroupStoredVolume.1.3 = 7
    mtaGroupTransmittedVolume.1.3 = 619853
    mtaGroupReceivedRecipients.1.3 = 33087
    mtaGroupStoredRecipients.1.3 = 2
    mtaGroupTransmittedRecipients.1.3 = 32817
    mtaGroupOldestMessageStored.1.3 = 1103
    mtaGroupInboundAssociations.1.3 = 5
    mtaGroupOutboundAssociations.1.3 = 2
    mtaGroupAccumulatedInboundAssociations.1.3 = 150262
    mtaGroupAccumulatedOutboundAssociations.1.3 = 10970
    mtaGroupLastInboundActivity.1.3 = 1054822
    mtaGroupLastOutboundActivity.1.3 = 1054222
    mtaGroupRejectedInboundAssociations.1.3 = 0
    mtaGroupFailedOutboundAssociations.1.3 = 0
    mtaGroupInboundRejectionReason.1.3 =
    mtaGroupOutboundConnectFailureReason.1.3 =
    mtaGroupScheduledRetry.1.3 = 0
    mtaGroupMailProtocol.1.3 = applTCPProtoID.25
    mtaGroupSuccessfulConvertedMessages.1.3 = 0[3]
    mtaGroupFailedConvertedMessages.1.3 = 0
    mtaGroupCreationTime.1.3 = 0
    mtaGroupHierarchy.1.3 = 0
    mtaGroupOldestMessageId.1.3 = <01IFBV8AT8HYB4T6UA@red.iplanet.com>
    mtaGroupLoopsDetected.1.3 = 0[4]
    mtaGroupLastOutboundAssociationAttempt.1.3 = 1054222
```

注 :

1. .x.y という形式の接尾辞では、x はアプリケーションインデックス (applIndex) であり、applTable のどのアプリケーションがレポートされているかを示します。この場合は MTA です。y には、MTA の各チャネルが列挙されます。この列挙型のインデックス (mtaGroupIndex) は、mtaGroupAssociationTable テーブルと mtaGroupErrorTable テーブルでも使われています。
2. レポートされているチャネルの名前で、この場合は autoreply チャネルです。
3. 変換チャネルは、ゼロ以外の値しかとりません。
4. 現在チャネルのメッセージキューに保管されている .HELD メッセージファイルの数をカウントします。

mtaGroupTable の使用法

Rejected と *Failed* の傾向分析を行うと、チャネルの潜在的な問題を発見できる場合があります。

mtaGroupStoredVolume と mtaGroupStoredMessages の比が突然変化した場合は、キュー付近に大きなジャンクメールがある可能性があります。

mtaGroupStoredMessages が急激に変化した合は、不特定多数宛てのメールが送信されているか、何らかの理由で配信に失敗している可能性があります。

mtaGroupOldestMessageStored の値が、配信不能メッセージの通知時間 (noticesチャネルキーワード) に使用されている値よりも大きい場合、これはバウンスでも処理できないメッセージを示している可能性があります。バウンスは毎晩夜間に行われるため、テストには
mtaGroupOldestMessageStored> (最大時間 + 24 時間) を使用してください。

mtaGroupLoopsDetected がゼロよりも大きい場合は、メールループが検出されています。

mtaGroupAssociationTable

これは 3 次元のテーブルで、各エントリは assocTable へのインデックスを表しています。applTable 内の各 MTA に対し、それぞれ 2 次元のサブテーブルがあります。この 2 次元のサブテーブルには、対応する MTA の各チャネルに対して 1 つの行があります。また、各チャネルに対し、そのチャネルが現在使用しているアクティブなネットワーク接続ごとにエントリが 1 つずつあります。エントリの値は assocTable へのインデックスです (エントリの値と、参照されている MTA の applIndex インデックスによってインデックスが付けられています)。この assocTable 内のエントリは、そのチャネルが保持しているネットワーク接続です。

簡単に言うと、mtaGroupAssociationTable テーブルは assocTable に示されているネットワーク接続を、mtaGroupTable の対応するチャネルに関連付けているものです。

以下に、mtaGroupAssociationTable (mib-2.28.3.1) のデータ例を示します。

```
mtaGroupAssociationTable:

    mtaGroupAssociationIndex.1.3.1[1] = 1[2]
    mtaGroupAssociationIndex.1.3.2 = 2
    mtaGroupAssociationIndex.1.3.3 = 3
    mtaGroupAssociationIndex.1.3.4 = 4
    mtaGroupAssociationIndex.1.3.5 = 5
    mtaGroupAssociationIndex.1.3.6 = 6
    mtaGroupAssociationIndex.1.3.7 = 7
```

注 :

1. .x.y.z という形式の接尾辞では、x はアプリケーションインデックス (applIndex) であり、applTable 内のどのアプリケーションがレポートされているかを示します。この場合は MTA です。y は mtaGroupTable 内のどのチャネルがレポートされているかを示します。上の例で、3 は tcp_local チャネルを表しています。z には、チャネルへ向かってオープンな、またはチャネルからオープンなアソシエーションが列挙されます。
2. この値は assocTable へのインデックスです。特に、x とこの値は、それぞれ applIndex の値と、assocTable への assocIndex インデックスになります。言い換えると、applIndex を無視した場合、assocTable の最初の行は tcp_local チャネルによって制御されているネットワーク接続を表していることになります。

mtaGroupErrorTable

これも 3 次元のテーブルで、メッセージの配信中に各 MTA の各チャネルで発生した一時的および永久的なエラーの数を示します。インデックス値が 4000000 のエントリは一時的なエラー、5000000 のエントリは永久的なエラーです。一時的なエラーの場合は、メッセージが再度キューに入れられ、あとで再び配信が試みられます。永久的なエラーの場合は、メッセージが拒否されるか、配信不能として戻されます。

以下に、mtaGroupErrorTable (mib-2.28.5.1) のデータ例を示します。

```
mtaGroupErrorTable:

    mtaGroupInboundErrorCount.1.1.4000000[1] = 0
    mtaGroupInboundErrorCount.1.1.5000000 = 0
    mtaGroupInternalErrorCount.1.1.4000000 = 0
    mtaGroupInternalErrorCount.1.1.5000000 = 0
    mtaGroupOutboundErrorCount.1.1.4000000 = 0
    mtaGroupOutboundErrorCount.1.1.5000000 = 0

    mtaGroupInboundErrorCount.1.2.4000000[1] = 0
    ...

    mtaGroupInboundErrorCount.1.3.4000000[1] = 0
    ...
```

注：

1. .x.y.z という形式の接尾辞では、x はアプリケーションインデックス (applIndex) であり、applTable 内のどのアプリケーションがレポートされているかを示します。この場合は MTA です。y は mtaGroupTable 内のどのチャネルがレポートされているかを示します。上の例では、1 により autoreply チャネルが、2 により ims-ms チャネルが、3 により tcp_local チャネルが指定されています。z は 4000000 または 5000000 の値をとり、そのチャネルのメッセージ配信中に発生した一時的または永久的なエラーの数を示します。

mtaGroupErrorTable の使用法

エラー数が急激に増加した場合は、異常な配信問題があると考えられます。たとえば、tcp_ channel の値が急激に増加した場合は、DNS またはネットワークの問題が考えられます。ims_ms チャネルの値が急激に増加した場合は、メッセージストアへの配信の問題が考えられます。たとえば、パーティションに空き容量がない、または stored に問題があるなどです。

付録 B

MTA ダイレクト LDAP 操作

iPlanet Messaging Server の 5.2 以前のリリースでは、MTA が使用するユーザおよびグループに関するディレクトリ情報は、多数のファイルやデータベースからアクセスされていました。これらのファイルにあるデータは dirsync プロセスによって更新されており、このプロセスがディレクトリへの変更をモニタし、ファイルとデータベースのデータを適宜更新していました。バージョン 5.2 のデフォルトの動作は以前と同じですが、新しいオプションにより、MTA でディレクトリと直接やりとりすることができるようになりました。このオプションをダイレクト LDAP モードと言います。

MTA をダイレクト LDAP モードで作動させるように設定すると、dirsync プロセスとそのデータベースは使われません。その代わり、MTA は同等の LDAP コールを行い、ドメインが MTA のホストドメインかどうかをまず確認してから、必要な配信情報にアクセスします。ダイレクト LDAP モードが設定可能でメカニズムがより透過的になることを除き、操作を dirsync モードからダイレクト LDAP モードに変更しても、アドレス変換にはほとんど効果はありません。ただし、ホストドメインの動作には変化があり、システムの動作にも大きな影響があります。詳細は、564 ページの「ダイレクト LDAP モードに変更する意味」を参照してください。

この章には、以下の節があります。

ダイレクト LDAP モードを有効にするには

ダイレクト LDAP モードを有効にするには、標準 MTA 設定を以下のように変更します。

1. .../imta/config/imta.cnf ファイルの書き換えセクションに次の行を追加します。

   ```
   $*       $E$F$U%$H$V$H@localhost
   ```

   localhost は、MTA のプライマリホスト名です。

   たとえば、MTA が island.siroe.com に呼び出される場合、.../imta/config/imta.cnf の書き換えセクションの先頭を以下のように変更します。

   ```
   ! Rules to select   local users
   $*                  $E$F$U%$H$V$H@island.siroe.com
   island.siroe.com    $U%$D@island.siroe.com
   siroe.com           $U%$D@island.siroe.com
   ```

2. .../imta/config/imta.cnf の ims-ms チャネルの定義を変更して、filter ssrd:$A を削除します。

   ims-ms チャネル定義が以下のような場合、

   ```
   ! ims-ms
   ims-ms defragment subdirs 20 notices 1 7 14 21 28  ¥
    backoff "pt5m" "pt10m" "pt30m" "pt1h" "pt2h" "pt4h" ¥
    maxjobs 1 pool IMS_POOL fileinto $U+$S@$D filter ssrd:$A
   ims-ms-daemon
   ```

   以下のように変更します。

   ```
   ! ims-ms
   ims-ms defragment subdirs 20 notices 1 7 14 21 28 ¥
    backoff "pt5m" "pt10m" "pt30m" "pt1h" "pt2h" "pt4h" ¥
    maxjobs 1 pool IMS_POOL fileinto $U+$S@$D
   ims-ms-daemon
   ```

3. .../imta/config/option.dat ファイルに以下の行を追加します。

   ```
   ALIAS_MAGIC=8764
   ALIAS_URL0=ldap:///$V?*?sub?$R
   USE_REVERSE_DATABASE=4
   REVERSE_URL=ldap:///$V?mail?sub?$Q
   USE_DOMAIN_DATABASE=0
   ```

   バニティドメインをサポートする場合は、以下のような追加のオプションも設定する必要があります。

```
DOMAIN_MATCH_URL=ldap:///$B?msgVanityDomain?sub? ¥
(msgVanityDomain=$D)
ALIAS_URL1=ldap:///$B?*?sub?(&(msgVanityDomain=$D)$R)
ALIAS_URL2=ldap:///$1V?*?sub?(mailAlternateAddress=@$D)
```

4. .../imta/config/job_controller.cnf から以下の行を削除します。

```
[PERIODIC_JOB=dirsync_incr]
command=IMTA_TABLE:../../imsimta dirsync
time=/00:10
!
[PERIODIC_JOB=dirsync_full]
command=IMTA_TABLE:../../imsimta dirsync -F
time=02:00/24:00
!
```

5. .../imta/config/mappings ファイルの末尾にある SEND_ACCESS マッピングから以下の行を削除します。

```
*|*|inactive|*  $X4.2.1|$NMailbox$ temporarily$ disabled
*|*|deleted|*   $X5.1.6|$NRecipient$ no$ longer$ on$ server
```

6. 以下の MTA データベースを削除するか、または移動します。

```
.../imta/db/aliasesdb.db
.../imta/db/domaindb.db
.../imta/db/reversedb.db
```

7. 変更した MTA 設定をコンパイルします。これは、MTA 設定を有効にする前に行う必要があります。

ダイレクト LDAP モードでの操作

MTA の宛先電子メールアドレスの処理には、基本的に変更はありません。簡単に言うと、MTA はまず書き換え規則を使用して、1) ドメインが認識されるかどうかを確認し、2) 必要に応じてアドレスを書き換え、3) メッセージを該当のチャネルにルーティングします。メッセージが l チャネルにルーティングされると、アドレスはエイリアス検索プロセス (546 ページの「ダイレクト LDAP エイリアス解決」を参照) を使用して変換され、変換されたアドレスは、エイリアスと関連するチャネルにルーティングされるように、書き換え規則を使用して再度書き換えられます。通常これは、ims-ms チャネル、auto_reply チャネル、またはその他の標準 MTA チャネルのいずれかです。

ダイレクト LDAP モードで操作すると、アドレス処理の書き換え規則フェーズとエイリアスフェーズが変更されます。これらの変更については、以下の節で説明します。

- 542 ページの「ダイレクト LDAP 書き換え規則 ($V) でアドレスを解決する」

- 544 ページの「アドレス書き換え中に LDAP エラーを管理する」
- 546 ページの「ダイレクト LDAP エイリアス解決」
- 562 ページの「エイリアスのキャッシング」
- 562 ページの「逆アドレス変換」

ダイレクト LDAP 書き換え規則 ($V) でアドレスを解決する

MTA では、最初にアドレスのドメイン部分 (@ の右の部分) を書き換え規則と照らし合わせてチェックすることによって、アドレスを解決します。書き換え規則は、`.../imta/config/imta.cnf` ファイルの最初の半分にあります。一致するものが見つかると、書き換え規則がメールのルーティング先チャネルを指定します。たとえば、送信インターネットトラフィックまたはローカルチャネルの場合、メールは `tcp_local` にルーティングされ、ディレクトリに規定されているユーザの場合、メールは `l` にルーティングされます。

MTA が `dirsync` モードに設定されていると、規則の評価プロセスはドメインデータベースの情報を使用します。このデータベースは、`dirsync` プロセスが維持するデータベースの 1 つです。MTA がダイレクト LDAP モードに設定されていると、特殊な「try me first」書き換え規則が使われます。この規則は以下のように記述されています。

```
$*      $E$F$U%$H$V$H@localhost
```

この規則の左側にある `$*` パターンは、この規則を最初に、かつすべてのアドレス上で試みることを意味します。右側の意味は以下のとおりです。

- `$E` - エンベロープアドレスのみで使用する
- `$F` - 前方を探す (`To:`) アドレスのみで使用する
- `$U%$H` - アドレスを `user@host` の形式に「書き換える」(この規則は、実際には未変更の元のアドレスが使用されることを指定する)
- `$V$H` - アドレスのホスト部分 (アドレスの `@` 記号の右の部分) がディレクトリに定義されているドメインと一致する場合にのみ、この規則を適用する
- `@localhost` - `l` チャネルにルーティングする

LDAP ドメイン検索のしくみ

書き換え規則プロセスの新しい部分として、$V 照合パラメータがあります。$V は、アドレスがローカルかどうかを確認し、ローカルの場合はディレクトリツリー内でその場所を探すために使われます。$V にはパラメータが必要で、この場合はアドレスのホスト部分である $H です。$V タグは、多くの LDAP 検索を利用します。このプロセスは、DC ツリー内でアドレスのドメイン部分を検索して、ユーザおよびグループツリーの該当するサブツリーを探します。たとえば、以下のアドレスを検索する場合、

robinson.crusoe@desert.island.siroe.com

最初にドメイン desert.island.siroe.com がチェックされ、それに失敗すると、island.siroe.com、siroe.com、および com がチェックされます。この LDAP 検索はディレクトリ内の DC ツリーで行われます (iPlanet Messaging Server ネームスペースと DIT 構造の詳細は、『iPlanet Messaging Server プロビジョニングガイド』を参照)。このツリーは、service.dcroot configutil 属性で指定した場所にルーティングされています。デフォルト値は o=internet です。検索は、以下のような識別名を持つエントリに対して行われます。

```
dc=desert,dc=island,dc=siroe,dc=com,o=internet
dc=island,dc=siroe,dc=com,o=internet
dc=siroe,dc=com,o=internet
dc=com,o=internet
```

ドメイン検索は、見つかったエントリに inetDomain オブジェクトクラスと inetDomainBaseDn 属性、あるいは inetDomainAlias オブジェクトクラスと aliasedObjectName 属性のいずれかがある場合にのみ、成功とみなされます。

上位ドメインをチェックしない場合は、最下位ビットの DOMAIN_UPLEVEL オプションをクリアして、チェックしないようにすることができます (この例では上位ドメインは island.siroe.com、siroe.com、および com)。DOMAIN_UPLEVEL は .../imta/config/option.dat に指定されています。デフォルト値は 1 です。このため、上位レベルをチェックしない場合は、

DOMAIN_UPLEVEL=0 という行を

.../imta/config/option.dat に追加します。

また、$Z という新しいタグがあります。このタグには、$V と正反対の意味があります。ホストがディレクトリ内にある場合は $V が規則を照合し、ホストがディレクトリ内にない場合は $Z が規則を照合します。

バニティドメイン検索

ユーザに対してバニティドメイン (ホストドメインではない) を定義している場合は、これらに対する LDAP チェックも有効にする必要があります。バニティドメインのチェックは、デフォルトでは無効になっています。これを有効にするには、.../imta/config/option.dat に以下の行を追加します。

```
DOMAIN_MATCH_URL=ldap:///$B?msgVanityDomain?sub? ¥
  (msgVanityDomain=$D)
```

バニティドメインのチェックは、ホストドメインのチェックに失敗した場合にのみ行われます。

ドメイン検索キャッシュ

ディレクトリ内のすべてのドメインに対してバニティドメインのチェックを行うには、メールの送信先のインターネットドメインを含むすべてのドメインをチェックする必要があるため、非常にコストがかかります。そこで、コスト削減のために、検索の結果が MTA によってキャッシュに書き込まれます。デフォルトでは、最大 600 秒で、最大 100,000 回の検索結果 (成功または失敗) がキャッシュに書き込まれます。このキャッシュは、`.../imta/config/option.dat` にある以下のオプションによって設定を制御できます。

```
DOMAIN_MATCH_CACHE_SIZE=100000
DOMAIN_MATCH_CACHE_TIMEOUT=600
```

アドレス書き換え中に LDAP エラーを管理する

ディレクトリ内のドメイン検索の結果には、以下の 4 つが考えられます。

- ドメインが見つかり、それが適正である
- ドメインが見つかり、それが不正である
- ドメインが見つからない
- 検索が失敗した (LDAP エラー)

最初のケースは問題ありません。2 番目と 3 番目は同じものとして扱われ、$V 規則は失敗します。最後のケースは、少し難しい状況です。MTA がこのケースの場合に行う適切なアクションには、以下の 2 つがあります。

1. *400 Temporary lookup failure* という SMTP 応答でアドレスを拒否する
2. あとで処理するために `reprocess` チャネルにメールをリダイレクトする

最初のアクションは、メールがリモート MTA から来たものであれば当然のアクションで、正しいアクションです。2 番目のアクションは、メールがユーザエージェントの送信メールであれば、妥当なアクションです。MTA は、この 2 つのアクションの違いを識別し、それに応じて動作する必要があります。これを有効にするメカニズムが、MTA の `DOMAIN_FAILURE` オプションです。`DOMAIN_FAILURE` は、ドメイン検索が失敗した場合に、書き換え規則の未使用部分を置換するための文字列を指定します。このため、`DOMAIN_FAILURE` のデフォルト値が

```
DOMAIN_FAILURE=reprocess-daemon$Mtcp_local$1M$1~-error$4000000?Temp
orary lookup failure
```

であり、処理されている書き換え規則が標準の

```
$*      $E$F$U%$H$V$H@localhost
```

である場合、VH 句によるドメイン検索が失敗すると、処理は書き換え規則が以下のとおりであるかのように続行します。

```
$*      $E$F$U%$H$V$H@reprocess-daemon$Mtcp_local$1M$1~-error$40000
00?Temporary lookup failure
```

この結果の規則の処理は、以下のようになります。

- $E - エンベロープアドレスのみで使用する
- $F - 前方を探す (To:) アドレスのみで使用する
- $U%$H - アドレスを user@host の形式に「書き換える」(この規則は、実際には未変更の元のアドレスが使用されることを指定する)
- VH - アドレスのホスト部分 (アドレスの @ 記号の右の部分) がディレクトリに定義されているドメインと一致する場合にのみ、この規則を適用する。これにより LDAP エラーが発生し、変更された規則が生成される
- @reprocess-daemon - 再処理チャネルにルーティングする
- $Mtcp_local - ソースチャネルが tcp_local でない場合に「失敗」する。このエラーは、ここまでの処理の結果である。規則の処理は続行する
- ($1M) - チャネルが reprocess または conversion などの内部再処理チャネルでない場合に「失敗」する
- $~ - 規則が現在失敗している場合に、照合が成功した処理を停止する
- -error - 宛先チャネルを無効なチャネル reprocess-daemon-error に変更する
- $4000000?Temporary lookup failure - SMTP 拡張エラーコードを 4.0.0 に設定し、エラーテキストを「Temporary lookup failure」に設定する

このため、ソースチャネルが tcp_local (リモート MTA からの接続がすべて可能な場所) であっても、reprocess-daemin-error のチャネルが存在していない場合、アドレスは拒否され、規則に指定されている 400 error code で拒否されます。

ソースチャネルが tcp_intranet (おそらくはユーザエージェント) の場合、規則は reprocess チャネルへのメッセージのルーティングのあとに続きます。

DOMAIN_FAILURE オプションとそれから作成される有効な書き換え規則は、いくつかの新しい書き換えタグを使用します。

$1M は既存の $Mchannel タグと同じで、ソースチャネルが再処理チャネルの場合は規則が失敗します。これは $Mreprocess$Mprocess$Mdefragment$conversion とほとんど同等です。

$~ は、$M または $1M (あるいは $M または $1N) タグに指定されているチャネル照合チェックを実行し、これが失敗した場合は、すぐに処理を正常終了します。

$abbbccc?text は、エラーのイベントで使用するエラーコードとエラーテキストを指定します。エラーコードは、実際には a、bbb、ccc という 3 桁の数字で、a.bbb.ccc という拡張 SMTP 結果コードを生成します。

ダイレクト LDAP エイリアス解決

エイリアス解決の目的は、メッセージの受信アドレス (エイリアス) を取得し、チャネルにメッセージを配信するための電子メールアドレスを生成することです。このアドレスは配信アドレスと呼ばれ、通常その形式は *uid@channel_name* です。

書き換え規則は、アドレスの @ 記号の右の部分だけを調べます。ただし、エイリアス解決は、アドレスの全体を調べることがあります。アドレス解決で使われるメカニズムは、.../imta/config/option.dat の ALIAS_MAGIC オプションによって制御されます。aliases ファイル内で一致をチェックしてから、aliases データベースで一致をチェックするのがデフォルトの動作です。このデータベースは dirsync プロセスによって保守されます (141 ページの「エイリアス」を参照)。

ダイレクト LDAP 操作を有効にするには、.../imta/config/option.dat に以下の行を追加します。

```
ALIAS_MAGIC=8764
```

これによりエイリアス解決は、aliases ファイル (通常はサイトポストマスターだけに使用される) を使用して試行され、次に LDAP ディレクトリで試行されます。LDAP エイリアス解決は、配信チャネルを生成する前に、いくつかのステップを通過します。そのステップは次のとおりです。

1. LDAP ディレクトリ内のアドレスのユーザ / グループエントリを探します。
2. エントリタイプ (ユーザまたはグループ) を判別します。
3. エントリのステータス (たとえば、active、inactive、deleted、hold) を抽出します。
4. uid 属性を抽出します。
5. ユーザの場所を探します。
6. メッセージのサイズが指定した上限を超えていないことを確認します。

7. mailDeliveryOption 属性 (たとえば、メールボックス、自動返信、プログラム、転送) に基づいて、配信アドレスを生成します。

以下の節で、上記のステップの詳細について説明します。

LDAP ディレクトリ内のユーザ / グループエントリを探す

エイリアスアドレスのユーザ / グループエントリを探す LDAP クエリは、以下のオプションによって .../imta/config/option.dat に定義されている URL で定義されます。

```
ALIAS_URL0
ALIAS_URL1
ALIAS_URL2
```

バニティドメインをサポートしていないかぎり、ALIAS_URL0 だけが使用されます。このオプションは以下のように設定することをお勧めします。

```
ALIAS_URL0=ldap:///$V?*?sub?$R
```

$V タグの処理は、543 ページの「LDAP ドメイン検索のしくみ」に記載されている $V タグと同じです。アドレスのドメイン部分の検索が成功した場合、URL の中の $V は、見つかったエントリ内の inetDomainBaseDn または aliasedObjectName 属性がポイントする DN と置き換えられます。検索が失敗すると、エイリアス展開は失敗します。(利用可能な $V タグの変形として $1V があります。これは、検索が失敗した場合にユーザおよびグループのツリーの最上部の DN (local.ugldapbasedn の値) を返します。)

$R は、configutil パラメータの local.imta.schematag で定義されるように、スキーマに適したフィルタによって置き換えられます。一致する電子メールアドレスを検索するために指定できるスキーマ値と属性は、以下のとおりです。

```
ims50    mail,mailalternateAddress,mailEquivalentAddress
nms41    mail,mailalternateAddress
sims40   mail,rfc822mailalias
```

local.imta.schematag で、これらの値のうちの複数をカンマで区切って指定できます。複数のスキーマを指定すると、この属性の和集合の一致が検索されます。ディレクトリスキーマがこれらのスキーマのいずれかと完全に一致しない場合は、configutil にパラメータ local.imta.mailaliases を指定して、検索する属性のリストを変更することができます。たとえば、以下のようになります。

```
local.imta.mailaliases=mail,mailAlternateAddresses,email
```

これにより、mail、mailAlternateAddresses、および email 属性に対して一致が検索されます。

デフォルトでは、$R タグによって生成されるフィルタだけが指定したアドレスを検索します。ただし、上位レベルのドメインにエイリアスを含めたい場合もあります。このため、desert.island.siroe.com ドメインに robinson.crusoe を規定している場合でも、ドメインツリー内のすべてのドメインにあるユーザ名を照合したい場合もあるでしょう。したがって、書き換え規則の評価で一致したドメインが siroe.com の場合、ディレクトリ内で以下のアドレスが検索されることになります。

```
robinson.crusoe@desert.island.siroe.com
robinson.crusoe@island.siroe.com
robinson.crusoe@siroe.com
```

これを実行するには、次のものを DOMAIN_UPLEVEL オプションの最下位ビットに設定する必要があります。これは、たとえば .../imta/config/option.dat ファイルに以下の行を追加することによって設定します。

```
DOMAIN_UPLEVEL=3
```

非標準のディレクトリでのドメイン検索

ユーザおよびグループツリーとは別の DC ツリーで標準の iPlanet ディレクトリ構造を使用できない場合は、エイリアスを検索するツリーのベースを検索するための別のメカニズムを利用することができます。前述のように ALIAS_URL0 の $V を使用する代わりに、マッピングを実行することができます。これを実行するシンタックスでは、URL に、$V ではなく以下のように指定します。

$|/*mapping-name*/*mapping-argument*|

| はコールアウトの始まりと終わりを示します。$| の直後の文字はマッピング名と引数の間の区切り文字で、マッピング名または引数に使用される文字と一致しないものを選ぶ必要があります。*mapping name* は、ドメイン検索マッピングテーブルの名前です。*mapping-argument* はドメインの名前です。たとえば、$D はドメインの名前になります。

バニティドメインエイリアスのドメイン検索

バニティドメインエイリアスをサポートするには、.../imta/config/option.dat に以下のような追加の URL を定義する必要があります。

```
ALIAS_URL1=ldap:///$B?*?sub?(&(msgVanityDomain=$D)$R)
ALIAS_URL2=ldap:///$1V?*?sub?(mailAlternateAddress=@$D)
```

エイリアス解決中の LDAP エラー

ディレクトリ内のエイリアス検索の結果は、何も返されない場合も、1 つまたは複数返される場合もあります。複数のエントリが一致すると、結果が返されなかった場合と同様に検索は失敗したものとみなされ、アドレスは無効として拒否されます。何らかの理由により設定されているディレクトリに達することができない場合、あるいは LDAP クエリでエラーが発生した場合、一時的なエラー (SMTP では 4xx エラー) が表示されてアドレスが拒否されます。送信側の MTA はあとでメールを再試行し、ディレクトリの問題が解決されるまでこの再試行を続けます。

エントリタイプを判別する

ディレクトリ内で一度エントリが見つかったあとであれば、そのエントリを処理して、適切なチャネルにメールを配信することができます。エントリ処理の最初のステップは、そのエントリがユーザ、グループ、またはそれ以外の認識不可能なもののいずれであるかを判別することです。エントリがユーザまたはグループであることがわかった場合、処理は適切に続きます。エントリがユーザでもグループでもない場合、エントリとその結果として処理されるアドレスは、警告なしで無視されます。

エントリタイプは、エントリが属するオブジェクトクラスを確認することによって判別されます。ユーザとグループの必須オブジェクトクラスは、`local.imta.schematag` の設定で定義されるように、ディレクトリ用のスキーマに含まれます。さまざまなスキーマのユーザまたはグループとしてエントリを定義するために指定するオブジェクトクラスは、以下のとおりです。

```
ims50:        ユーザ :     inetLocalMailRecipient + inetmailuser
            グループ :     inetLocalMailRecipient + inetmailgroup

nms41:        ユーザ :     mailRecipient + nsMessagingServerUser
            グループ :     mailGroup

sims40:       ユーザ :     inetMailRouting + inetmailuser
            グループ :     netMailRouting + inetmailgroup
```

ディレクトリスキーマがこれらのスキーマと完全には一致しない場合は、ユーザとグループのディレクトリエントリの違いを見分けるために、独自の判別要因を定義することができます。MTA オプションの `LDAP_USER_OBJECT_CLASSES` と `LDAP_GROUP_OBJECT_CLASSES` を使用して、それぞれユーザまたはグループに分類されるエントリを示すオブジェクトクラスを指定することができます。たとえば、`.../imta/config/option.dat` ファイルに以下の行を追加します。

```
LDAP_USER_OBJECT_CLASSES=inetLocalMailRecipient+inetmailUser,mailRe
cipient+nsMessagingServerUser

LDAP_GROUP_OBJECT_CLASSES=inetLocalMailRecipient+inetmailgroup,mail
Group
```

local.imta.schematag=ims50,nms41 の設定と同じです。つまり、エントリにオブジェクトクラス inetLocalMailRecipient と inetmailUser、またはオブジェクト mailRecipient と nsMessagingServerUser がある場合、そのエントリはユーザであると判別されます。

配信アドレスの作成に使用する属性を抽出する

アドレスのエントリタイプを判別したあと、MTA は、ドメインとユーザまたはグループのエントリから属性セットを抽出して、配信アドレスと配信メッセージを作成する必要があります。ドメインとユーザまたはグループのエントリから、表 B-1、表 B-2、および表 B-3 に示すいくつか、またはすべての属性が抽出されます。以下の表に、使用される必須のデフォルト属性の名前と、それぞれの属性名を選択する際に使用できる MTA オプションを示します。通常、これらのオプションは、標準スキーマに対応するデフォルト値としては設定されません。ただし、ディレクトリでこれらの属性の 1 つ以上に別の属性名を使用する場合は、.../imta/config/option.dat に適切なオプションを設定して変更することができます。

表 B-1　デフォルトのドメイン属性と優先指定オプション

LDAP 属性名	MTA 優先指定オプション
domainUidSeparator	LDAP_DOMAIN_ATTR_UID_SEPARATOR
mailDomainCatchallAddress	LDAP_DOMAIN_ATTR_CATCHALL_ADDRESS
mailDomainConversionTag	LDAP_DOMAIN_ATTR_CONVERSION_TAG
mailDomainMsgMaxBlocks	LDAP_DOMAIN_ATTR_BLOCKLIMIT
mailDomainReportAddress	LDAP_DOMAIN_ATTR_REPORT_ADDRESS
mailDomainSieveRuleSource	LDAP_DOMAIN_ATTR_FILTER
mailDomainStatus	LDAP_DOMAIN_ATTR_STATUS
mailRoutingHosts	LDAP_DOMAIN_ATTR_ROUTING_HOSTS
mailRoutingSmarthost	LDAP_DOMAIN_ATTR_SMARTHOST

表 B-2　デフォルトのユーザ属性と優先指定オプション

LDAP 属性名	MTA 優先指定オプション
mailConversionTag	LDAP_CONVERSION_TAG
mailDeliveryFileURL	LDAP_PROGRAM_INFO
mailDeliveryOption	LDAP_DELIVERY_OPTION

表 B-2　デフォルトのユーザ属性と優先指定オプション (続き)

LDAP 属性名	MTA 優先指定オプション
mailhost	LDAP_MAILHOST
mailMsgMaxBlocks	LDAP_BLOCKLIMIT
mailMsgQuota	LDAP_MESSAGE_QUOTA
mailProgramDeliveryInfo	LDAP_PROGRAM_INFO
mailQuota	LDAP_DISK_QUOTA
mailRoutingAddress	LDAP_ROUTING_ADDRESS
mailSieveRuleSource	LDAP_FILTER
UID	LDAP_UID
	LDAP_SPARE_1*
	LDAP_SPARE_2*

* この 2 つの予備の LDAP オプションは、配信オプションパターンに置き換えるときに使用できるため、重要なものです。これについては、後述の配信オプションの処理に関する節で説明しています。

表 B-3　デフォルトのグループ属性と優先指定オプション

LDAP 属性名	MTA 優先指定オプション
mailRejectText	LDAP_REJECT_TEXT
memberURL	LDAP_GROUP_URL2
mgrpAddHeader	LDAP_ADD_HEADER
mgrpAllowedBroadcaster	LDAP_AUTH_URL
mgrpAllowedDomain	LDAP_AUTH_DOMAIN
mgrpAuthPassword	LDAP_AUTH_PASSWORD
mgrpBroadcasterPolicy	LDAP_AUTH_POLICY
mgrpDeliverTo	LDAP_GROUP_URL1
mgrpDisallowBroadcaster	LDAP_CANT_URL
mgrpDisallowDomain	LDAP_CANT_DOMAIN
mgrpErrorsTo	LDAP_ERRORS_TO
mgrpMsgMaxSize	LDAP_ATTR_MAXIMUM_MESSAGE_SIZE
mgrpMsgPrefixText	LDAP_PREFIX_TEXT

表 B-3　デフォルトのグループ属性と優先指定オプション (続き)

LDAP 属性名	MTA 優先指定オプション
msgpMsgSuffixText	LDAP_SUFFIX_TEXT
mgrpModerator	LDAP_MODERATOR_URL
mgrpRemoveHeader	LDAP_REMOVE_HEADER
mgrpRFC822MailMember*	LDAP_GROUP_RFC822
rfc822MailMember*	LDAP_GROUP_RFC822
uniqueMember	LDAP_GROUP_DN

* デフォルトでは mgrpRFC822MailMember と rfc822MailMember のいずれかを使用できますが、2 つを一緒に使用することはできません。

ユーザ / グループのステータスを抽出する

生成された配信アドレスを制御する主要な属性の 1 つは、ユーザ / グループとドメインのステータスです。mailDomainStatus によって定義されているドメインのステータスが inactive または deleted の場合、これがユーザのステータスとして使用され、ユーザのステータスはチェックされません。ドメインのステータスが active の場合、ユーザまたはグループのエントリのステータスが使用されます。エントリのステータスの定義に使用される属性は、使用するスキーマによって異なります。これは、以下のようになります。

```
ims50:      ユーザ   :     inetuserstatus または mailuserstatus
            グループ :     inetmailgroupstatus
nms41:      ステータス属性なし
sims40:     ユーザ   :     inetsubsscriberstatus
            グループ :     inetmailgroupstatus
```

ユーザとグループのステータスの判別に使われる属性名は、必要に応じて無効にすることができます。ユーザのステータスに使われる属性を指定するときは LDAP_USER_STATUS オプションを使用でき、グループのステータスに使われる属性を指定するときは LDAP_GROUP_STATUS オプションを使用することができます。ユーザまたはグループのステータスがいったん判別されると、ステータスは active、inactive、deleted、または hold のいずれかになります。

active - ユーザまたはグループのステータスがアクティブであることがわかると、554 ページの「ユーザの場所」に記載されているように処理が続行します。

inactive - ユーザまたはグループのステータスが inactive であることがわかると、アドレスは一時的なエラーステータス (4xx SMTP エラーコード) によってすぐに拒否されます。

deleted - ユーザまたはグループのステータスが deleted であることがわかると、アドレスは永久的なエラーステータス (5xx SMTP エラーコード) によってすぐに拒否されます。

hold - ユーザまたはグループのステータスが hold であることがわかると、アドレスが保留チャネルに再度書き込まれるように、エイリアスが生成されます。生成されるエイリアスは、MTA オプションの HOLD_TEMPLATE で指定されたパターンによって制御されます。このテンプレートのデフォルト値は、以下のとおりです。

```
$M?$2I@hold-daemon
```

パターン内のタグの意味については、554 ページの「DELIVERY_OPTIONS を使用して配信アドレスを生成する」で説明しています。アドレスが以下のように指定されていて、

```
robinson.crusoe@desert.island.siroe.com
```

一致するエントリが island.siroe.com のホストドメインにある rcrusoe の UID を指定した場合、以下のようなエイリアスが生成されます。

```
rcusoe?island.siroe.com@hold-daemon
```

このアドレスは、.../imta/config/imta.cnf 内の書き換え規則に一致します。

```
hold-daemon    $U%$H@hold-daemon
```

これは、一致はしてもアドレスを変更しないため、メールは保留チャネルに配信されます。

UID を抽出する

ディレクトリ内のすべての有効なユーザエントリには、uid 属性が含まれている必要があります。グループエントリには、uid 属性が含まれていなくてもかまいません。uid は、配信アドレスを生成するために使用されます。ユーザエントリに uid 属性がない場合、このエントリは無視されます。ユーザエントリに複数の uid 属性がある場合は、最初のエントリだけが使用されます。

ディレクトリ内の uid 属性には、必要以上の情報が含まれていることもあります。たとえば、ホストドメイン内のエントリの形式は、実際の uid、区切り文字 (domainUidSeparator 属性で定義)、次にドメイン (例 : uid=walter@siroe.com) になります。uid の中に区切り文字がある場合、エイリアスの作成に使われるのは区切り文字の前の部分だけです。

配信アドレスの uid として uid 以外の属性を使用する必要がある場合は、LDAP_UID オプションを使用してその他の属性名を指定することができます。

ユーザの場所

ユーザまたはグループがいったんアクティブなユーザとして識別された場合、MTA はそのユーザがこの MTA に対してローカルなユーザであることをチェックする必要があります。ローカルとみなすためには、エントリに、`local.hostname configutil` 属性、または `local.imta.hostnamealiases configutil` 属性で指定されている名前のいずれか 1 つと一致する `mailhost` 属性がなければなりません。ユーザがローカルの場合、MTA は次のステップに進みます。次のステップは、メッセージのサイズの上限を超えないようにすることです。

メールホストをこの MTA の名前と照合できない場合、

```
@mailhost:user@domain
```

という形式の新しいアドレスが生成されます。これはソースルートされた RFC822 アドレスであり、書き換え規則を使って処理されます。ソースルートされたアドレスの場合、書き換え規則はドメイン部分ではなくソースルートアドレスを調べます。

ユーザエントリに `mailhost` 属性がない場合、生成されるアドレスは以下のドメインと関連する `mailRoutingSmarthost` を使用します。

```
@smarthost:user@domain
```

ユーザエントリに `mailhost` 属性がなく、ドメインに `mailRoutingSmartHost` がない場合、アドレスは破棄され、5xx エラーが報告されます。

グループエントリに `mailhost` 属性がない場合、グループはローカルで処理されます。この明らかな矛盾は重要です。それは、グループが特定のサーバ上ではなく受信リレー MTA 上で展開されることがあるからです。

サイズの上限を抽出する

配信アドレスが作成されるまで (ユーザの場合)、またはグループが展開されるまでに MTA が実行する必要のある、最終的なチェックがあります。この最終チェックでは、メールメッセージが単体でユーザの `mailMsgMaxBlocks` 属性を超えていないかどうかを確認します。この属性が設定されていない場合は、ドメインの `mailDomainMsgMaxBlocks` 属性を超えていないかどうかを確認します。メッセージが大きすぎると、アドレスは 5xx サイズ超過エラーによって拒否されます。

DELIVERY_OPTIONS を使用して配信アドレスを生成する

見つかったエントリがユーザエントリの場合はそのまま、書き換え規則を使ってメールを適切なチャネルに返信するための、ユーザの配信アドレスを生成できます。配信アドレス生成の処理はグループの場合も行われますが、グループの場合、その他のいくつかの注意事項があります。これについてはあとの節で説明しています。

配信アドレスは 1 組のパターンによって生成されます。使用されるパターンは、mailDeliveryOption 属性に定義されている値によって異なります。配信アドレスは、有効な mailDeliveryOption ごとに生成されます。パターンは MTA オプションの DELIVERY_OPTIONS によって定義されます。このオプションは .../imta/config/option.dat に定義することができます。DELIVERY_OPTIONS のデフォルト値を以下に示します。

```
DELIVERY_OPTIONS=*mailbox=$M%$2I+$2S@ims-ms-daemon,
        &members=*,
        *native=$M@native-daemon,
        *unix=$M@native-daemon,
        &file=+$F@native-daemon,
        hold=$M?$2I@hold-daemon,
        &$members_offline=*,
        program=$M%$P@pipe-daemon,
        forward=**,
        *autoreply=$M@autoreply-daemon
```

DELIVERY_OPTIONS の値は、カンマで区切られた規則のセットです。各規則の左側は配信方法の名前 (たとえば、mailbox、unix、forward) で、右側は配信アドレス作成のためのパターンです。それぞれの規則の前には、1 つ以上の特殊なフラグ文字を指定することもできます。これは、規則がいつ、どのように適用されるかに影響します。フラグ文字は以下のとおりです。

*　　この規則はユーザのみに適用されます。

&　　この規則はグループのみに適用されます。

$　　このタグはメッセージを reprocess チャネルのキューに入れるため、拡張をオフラインで行うことができます。

このため、ユーザが使用できる配信方法は、mailbox、native、unix、および autoreply だけです。グループが使用できる配信方法は members と members_offline だけで、ユーザとグループの両方が使用できる配信方法は program と forward です。

右側は、単純な代替テキストと、さまざまな LDAP 属性の値を挿入するタグで構成されています。337 ページの「置換タグ (大文字小文字を区別します)」を参照してください。

配信アドレスを生成する - 例

例として、以下のアドレスに送信されたメッセージについて検討します。

robinson.crusoe+goats@desert.island.siroe.com

また、この例では、ディレクトリエントリに以下の属性が含まれていると仮定します。

```
UID: rcrusoe@desert.island.siroe.com
mail: robinson.crusoe@desert.island.siroe.com
mailDeliveryOption: mailbox
mailDeliveryOption: native
mailDeliveryOption: program
mailDeliveryOption: forward
mailDeliveryOption: autoreply
mailProgramDeliveryInfo: capriform.msg
mailForwardingAddress: friday@desert.island.siroe.com
mailForwardingAddress: hulahula@londonbank.siroe.com
```

これにより、元のアドレスは 6 つのエイリアス (配信方法 mailbox、native、program、および autoreply にそれぞれ 1 つずつ、配信方法 forward に 2 つ) を生成します。

mailbox のパターン $M%$2I+$2S@ims-ms-daemon は、より複雑なものの 1 つです。

表 B-4 配信オプション mailbox のパターン拡張

パターンの要素	動作	結果
$M	rcrusoe を生成する	rcrusoe
%	% を生成する	rcrusoe%
$2I	desert.island.siroe.com を生成する	rcrusoe%desert.island.siroe.com
+	+ を生成する	rcrusoe%desert.island.siroe.com+
$2S	goats を生成する	rcrusoe%desert.island.siroe.com+goats
@ims-ms-daemon	@ims-ms-daemon を生成する	rcrusoe%desert.island.siroe.com+goats @ims-ms-daemon

この結果として生成されるアドレスには ims-ms チャネルのチャネルタグと完全に一致するドメイン部分があるため、より詳細な書き換えを行わなくてもそのチャネルにルーティングされます。

native のパターン $M@native-daemon は、より単純なものです。

配信オプション native のパターン拡張

表 B-5 配信オプション native のパターン拡張

パターンの要素	動作	結果
$M	rcrusoe を生成する	rcrusoe

表 B-5　配信オプション native のパターン拡張 (続き)

パターンの要素	動作	結果
@native-daemon	@native-daemon を生成する	rcrusoe@native-daemon

この結果として生成されるアドレスにはパイプチャネルのチャネルタグと完全に一致するドメイン部分があるため、より詳細な書き換えを行わなくてもそのチャネルにルーティングされます。

自動返信のパターン $M@autoreply-daemon は、非常に単純なものです。

表 B-6　配信オプション autoreply のパターン拡張

パターンの要素	動作	結果
$M	rcrusoe を生成する	rcrusoe
@autoreply-daemon	@autoreply-daemon を生成する	rcrusoe@autoreply-daemon

この結果として生成されるアドレスには自動返信チャネルのチャネルタグと完全に一致するドメイン部分があるため、より詳細な書き換えを行わなくてもそのチャネルにルーティングされます。

program のパターン $M%$P@pipe-daemon は、ほとんど同じものです。

表 B-7　配信オプション program のパターン拡張

パターンの要素	動作	結果
$M	rcrusoe を生成する	rcrusoe
%	% を生成する	rcrusoe%
$P	prog を生成する	rcrusoe%prog
@pipe-daemon	@pipe-daemon を生成する	rcrusoe%prog@pipe-daemon

この結果として生成されるアドレスにはパイプチャネルのチャネルタグと完全に一致するドメイン部分があるため、より詳細な書き換えを行わなくてもそのチャネルにルーティングされます。

forward のパターン ** は、使用されている mailForwardingAddress 属性の値になるだけです。結果として、以下のアドレスが生成されます。

```
friday@desert.island.siroe.com
hulahula@londonbank.siroe.com
```

このため、robinson.crusoe に送信されたメッセージが以下の配信アドレスを生成し、以下のチャネルに配信されます。

```
rcrusoe%desert.island.siroe.com+goats@ims-ms-daemon      ims-ms
rcrusoe@native-daemon                                    native
rcrusoe@autoreply-daemon                                 autoreply
rcrusoe%prog@pipe-daemon                                 pipe
friday@desert.island.siroe.com
hulahula@londonbank.siroe.com
```

SIEVE 規則

ユーザのエントリから取得される最終的な LDAP 属性は、mailSieveRuleSource です。これには、ユーザ用の SIEVE フィルタ規則が含まれています。メッセージが配信チャネルのキューに入れられるポイントに来るまで、これらの規則は適用されません。つまり、MTA がエイリアスを展開している間に SIEVE フィルタが取得されても、結果として生成される配信アドレスが展開されて、ims-ms、native、autoreply、またはパイプチャネルに送信されるまで、SIEVE は使用されません。これは dirsync 以外のモードの操作において変更された動作です。dirsync 以外のモードでは、SIEVE 規則を使って処理されるのは ims-ms チャネルに配信されたメールだけでした。

グループエントリを処理する

グループに利用できる 4 つのプログラム配信オプションがあります。program、forward、members、および members_offline です。

program と forward は、ユーザ用の場合と同様に処理されます。

members と members_offline のパターンは両方とも * です。これは、以下の節で説明するグループ展開処理を最大限に活用します。

members_offline の規則の前には $ が付きます。これは、グループ展開が reprocess チャネルで行われることを意味します。メッセージがキューに入れられるチャネルが reprocess チャネル以外のチャネルの場合 (ほとんどの場合、最初にメッセージがキューに入れられるチャネルは tcp_ チャネルの 1 つ)、アドレスの処理は停止し、元のアドレスが受け入れられ、メッセージは reprocess チャネルのキューに入れられます。reprocess チャネルが実行されるときは、アドレスの処理と同じロジックが関与しますが、メッセージがキューに入れられるチャネルは reprocess チャネルであるため、members_offline とその $ は members とまったく同様に処理されます。

原則として、グループの処理は単純明快です。グループのメンバーを電子メールアドレスか識別名のいずれかとして一覧表示するいくつかの属性があります。どちらの場合も、アドレスはグループ展開の結果の一部として使用されます。

実際には、グループの処理は奥が深く、グループエントリの処理に影響する属性は多数あります。

グループエントリの処理の詳細

MTA は、さまざまなグループ処理オプションをそれぞれ順番に考慮することによって、グループエントリを処理します。オプションが処理される順番は重要です。グループ属性は、おおまかに以下の 3 つのタイプに分割できます。

- `mailRejectText` などの処理のためのパラメータを備えた属性。この属性は、何を実行できるか、または何を実行できないかには影響しませんが、プロセスへの入力を行います。
- どの環境でメールをリストに送信できるかを制御する属性。これには、`mgrpAllowDomain` のような属性も含まれます。この属性は、メッセージをグループに送信できるドメインを指定します。これらの属性は、以下の表に記載されている順番で処理されます。
- リストの実際のメンバーを指定する属性。

以下の表に、グループ処理属性を示します。

表 B-8 グループ処理のパラメータを提供する属性

属性	説明
`mailRejectText`	グループに関連する認証メカニズムによってメッセージが拒否される場合に、SMTP 応答として返されるテキストを指定します。この属性は、通常は SMTP のプロトコル規則に準拠する US-ASCII のみの単一値属性です。属性が複数値の場合、最初の属性だけが使用されます。値が複数行の場合、最初の行だけが使用されます。
`mgrpMsgMaxSize`	旧バージョンの属性。代わりに、エントリ処理のはじめの方でチェックされる `mailmsgMaxBlocks` を使用する必要があります。メッセージがこのサイズ (バイト単位で指定) を超えると、メッセージは「メッセージが大きすぎます」というエラーによって拒否されます。
`mgrpAuthPassword`	グループのパスワードを指定し、指定した `mgrpBroadcasterPolicy` にパスワードが必要な場合に使われます。
`mgrpErrorsTo`	これが指定されている場合は、エンベロープの発信元 (MAIL FROM) アドレスがこの属性の値に設定されます。これが指定されていない場合は、メッセージのエンベロープ発信元は変更されません。
`mgrpAddHeader`	(現在はまだサポートされていません)

表 B-8　グループ処理のパラメータを提供する属性 (続き)

属性	説明
mgrpRemoveHeader	(現在はまだサポートされていません)
mgrpMsgPrefixText	(現在はまだサポートされていません)
mgrpMsgSuffixText	(現在はまだサポートされていません)

表 B-9　メールグループアクセス制御属性

属性	説明
mgrpBroadcasterPolicy	グループにメッセージを送信するために必要な認証レベルを指定します。指定可能な値は以下のとおりです。 SMTP_AUTH_REQUIRED または AUTH_REQ のいずれか。グループに送信する前に SMTP の AUTH コマンドを使用して差出人を識別する必要があることを意味します。 PASSWORD_REQUIRED、PASSWD_REQUIRED、または PASSWD_REQ のいずれか。mgrpAuthPassword 属性で指定したリストのパスワードが、メッセージの Approved: header フィールドに表示されている必要があることを意味します。 NO_REQUIREMENTS。属性が指定されていないことと同じで、特別な要件がないことを意味します。
mgrpAllowedDomain	複数値。ユーザがメッセージをグループに送信できるドメインを一覧表示します。指定されていない場合、任意のドメインのユーザがグループに送信できます。
mgrpDisallowedDomain	複数値。ユーザがメッセージをグループに送信できないドメインを一覧表示します。
mgrpAllowedBroadcaster	展開されたときに、リストに送信できるアドレスのリストを生成する URL。URL の展開の結果として生成されるアドレスがグループの場合、そのグループは展開されてより詳細なリストを生成しますが、これは MTA が URL を展開する限界です。メッセージのエンベロープ From アドレスは、この展開によって生成されるアドレスごとにチェックされ、一致するものがある場合にのみメッセージが許可されます。

表 B-9　メールグループアクセス制御属性 (続き)

属性	説明
mgrpDisallowedBroadcaster	展開されたときに、リストに送信できないアドレスのリストを生成する URL。URL の展開の結果として生成されるアドレスがグループの場合、そのグループは展開されてより詳細なリストを生成しますが、これは MTA が URL を展開する限界です。メッセージのエンベロープ From アドレスは、この展開によって生成されるアドレスごとにチェックされ、何も一致しない場合にのみメッセージが許可されます。
mgrpModerator	展開されたときに、リストに送信できるアドレスのリストを生成する URL。URL の展開の結果として生成されるアドレスがグループの場合、そのグループは展開されてより詳細なリストを生成しますが、これは MTA が URL を展開する限界です。メッセージのエンベロープ From アドレスは、この展開によって生成されるアドレスごとにチェックされます。 エンベロープ From アドレスがこれらのアドレスのいずれかと一致する場合、メッセージはモデレータからのものであり、グループへの送信が許可されます。 エンベロープ From アドレスがこれらのアドレスと一致しない場合、メッセージは展開されたばかりのモデレータアドレスのリストに送信され、グループには一切配信されません。つまり、以下のグループメンバー表にある属性はすべて無視され、program または forward 配信オプションは無視されます。

表 B-10　メールグループ展開属性

属性	説明
mgrpDeliverTo	展開されたときに、アドレスのリストを生成する URL。URL の展開の結果として生成されるアドレスがグループの場合、そのグループは展開されてより詳細なリストを生成します。重複するアドレスは削除されますが、それぞれのアドレスを指すグループが作成されて無限再帰が発生することがあります。MTA はこれを解決するために、入れ子にされたリストの展開を 10 レベルまで許可します。
memberURL	mgrpDeliverTo と同じ方法で展開される、もう 1 つの URL のリスト。
uniqueMember	グループメンバーの識別名。それぞれの DN はユーザエントリ、グループエントリ、またはディレクトリのサブツリーのいずれかを指します。ディレクトリのサブツリーの場合、そのツリー内のすべてのエントリが展開されています。

表 B-10　メールグループ展開属性 (続き)

属性	説明
mgrpRFC822MailMember、rfc822MailMember	これらの属性の値は、グループのメンバーのメールアドレスです。指定されているエントリのうち、これらの属性の 1 つだけが許可されます。rfc822MailMember は、Netscape Messaging Server との下位互換のためだけにサポートされています。

エイリアスのキャッシング

すべての LDAP アクティビティは、MTA のパフォーマンスに重大な影響を与えることがあります。これを緩和するために、MTA プロセスは LDAP 検索の結果をキャッシュに書き込みます。このキャッシングは、以下のオプションによって制御されます。示されている値はデフォルト値です。

```
ALIAS_ENTRY_CACHE_SIZE=1000
ALIAS_ENTRY_CACHE_TIMEOUT=600
ALIAS_ENTRY_CACHE_NAGATIVE=0
```

これは、保持されるキャッシュエントリの最大数が 1,000、エントリが保持される時間の最大の長さが 10 分 (600 秒) であることを意味します。キャッシュエントリはドメインキャッシュエントリより大きくなりますが、システムに十分なメモリがあればキャッシュサイズを増やすだけの価値があるかもしれません。
ALIAS_ENTRY_CACHE_NEGATIVE は、エイリアス一致エラーをキャッシュに書き込むかどうかを制御します。デフォルトでは、これらはキャッシュに書き込まれません。エラーをキャッシュに書き込まなければ、新しいユーザの起動は高速になります。また、システムのパフォーマンスに影響するような頻度で、同じユーザへの配信試行が繰り返して失敗する可能性はなくなります。

逆アドレス変換

通常、From: ヘッダーなどの逆アドレスは、MTA でのフローに従って標準化されます。(標準化とは、ヘッダーアドレス内で個人名が前に移動し、コメントが後ろに移動することを意味します。さらに、From: アドレスの場合は、そのアドレスが検索され、mailalternateaddress として見つかると、代わりにそのメールアドレスが使われます。) ユーザのディレクトリエントリにリストされている最初のメールアドレスが使用すべきアドレスであるという原則が適用されます。このプロセスは、DC ツリー内でアドレスのドメイン部分を検索して、アドレスを検索するユーザおよびグループツリーのサブツリーを見つけてから、指定したものと一致する電子メールアドレスを含むエントリを検索し、そのエントリにある最初のメールアドレスを返します。これは、エイリアス処理と非常によく似たプロセスです。

ダイレクト LDAP アドレス変換は、.../imta/config/option.dat に設定されている以下の 2 つのオプションに依存します。

```
USE_REVERSE_DATABASE=4
REVERSE_URL=ldap:///$V?mail?sub?$Q
```

USE_REVERSE_DATABASE=4 は、MTA が古いリバースデータベースを使用せず、ダイレクト LDAP メカニズムを使用することを指定します。REVERSE_URL は、547 ページの「LDAP ディレクトリ内のユーザ / グループエントリを探す」で説明している ALIAS_URL0 URL と非常によく似ています。$V タグは、その節で説明する方法で展開されます。$Q タグは標準の ALIAS_URL0 で使用されている $R タグと似ていますが、このタグは、MTA が照合しようとしている逆アドレスと一致するアドレスを含む属性を検索するフィルタを生成します。$R で生成されるフィルタは、以下の local.imta.schematag configutil オプションの設定によって異なります。

```
ims50          mail,mailalternateAddress
nms41          mail,mailalternateAddress
sims40         mail,rfc822mailalias
```

あとから MTA オプションの LDAP_MAIL_REVERSES に指定して、検索に使用する属性を無効にすることができます。

実際に生成される検索は、エイリアス検索に使用する $R タグで生成される検索と非常によく似ています。この検索では、最初に指定したアドレスだけを検索するのではなく、代替の DC ツリーで実際に見つかったドメインを含むアドレスも検索します。手順については、547 ページの「LDAP ディレクトリ内のユーザ / グループエントリを探す」を参照してください。

逆アドレス検索が失敗すると、逆アドレスには変更は加えられません。

ほかの LDAP 検索と同様、逆アドレス検索の結果はキャッシュに書き込まれます。このキャッシュのサイズとタイムアウトは、以下のオプションによって制御されます。下記の値はデフォルト値です。

```
REVERSE_ADDRESS_CACHE_SIZE=10000
REVERSE_ADDRESS_CACHE_TIMEOUT=600
```

ダイレクト LDAP モードに変更する意味

MTA アドレス変換の場合、プロセスが設定可能でメカニズムがより透過的になることを除き、操作を dirsync モードからダイレクト LDAP モードに変更してもほとんど効果はありません。ただし、ホストドメインの動作には変化があります。dirsync モードではホストドメインのすべてのサブドメインが暗黙的に所有されますが、ダイレクト LDAP モードでは DOMAIN_UPLEVEL=3 を設定することで所有されます。ただし、ダイレクト LDAP モードの DOMAIN_UPLEVEL=0 の設定とは異なり、原則的にドメインに所有されるのは実際に設定するドメインだけです。この二分化された操作モードは、ダイレクト LDAP モードでは使用できません。ドメインの所有権をどちらにするかは、ユーザが決める必要があります。どちらを選んでも違いが生じることはないと思われますが、念のため承知しておいてください。

システムの動作全体には、明らかに影響があります。影響を受けた内容は以下のとおりです。

- LDAP に対する負荷が変化した
- データベースでの冗長性が削減された
- 全体的なメールのスループットが変化した

LDAP に対する負荷が変化した

dirsync プロセスが LDAP ディレクトリに対して作成するクエリは、数は少ないですが、サイズが非常に大きいことがありました。ダイレクト LDAP モードでは、MTA がディレクトリに多数の小さいクエリを作成します。これによる実質的な効果として、キャッシングが使用されていなければ、スループットの大幅な削減が可能です。しかし、ディレクトリに課せられる負荷はよりいっそう標準的なものになってきており、システムはよりスケーラブルになっています。dirsync では MTA を 600 万人以上のユーザに拡大することは困難でしたが、ダイレクト LDAP モードでは、1000 万人のユーザに提供することが可能になっています。

データベースでの冗長性が削減された

dirsync モードでは、MTA は、操作のために多数のデータベース (特に、エイリアスとドメインのデータベース) に依存していました。これらのデータベースはディスク構造が複雑で、システムに突然障害が発生すると破壊されてしまうことがあります。これは、高可用性システムの問題であることが判明しています。ダイレクト LDAP モードでは、MTA はデータベースにはほとんど依存しません。

全体的なメールのスループットが変化した

ディレクトリの使用が増えてデータベースの使用が減ると、スループットに大きく影響します。情報をディレクトリから抽出し、それを必要な形式に処理することは、単にデータベース内の結果を検索するよりも費用がかかります。しかし、エントリがキャッシュ内にあれば、全体的な費用はデータベース内を検索するよりも少なくなります。つまり、ほとんどのメールがエントリをキャッシュに書き込んでいる 2、3 人のユーザに対して処理されるのであれば、スループットは増大します。メールが大規模なユーザ組織全体で分散されるのであれば、スループットは低減します。

ダイレクト LDAP モードのメールスループットのパフォーマンス調整

システムのパフォーマンスは、ALIAS_ENTRY_CACHE_SIZE で設定するエイリアスキャッシュのサイズに影響を受けやすくなっています。エイリアスキャッシュのサイズのデフォルト値は 1000 ですが、おそらくこれはほとんどのシステムには少なすぎるでしょう。これらのキャッシュエントリは大きく (約 2K バイト) なることがあります。このデフォルト値は、小規模な評価システムが過負荷にならないようにするために設定されたものです。この値は 10,000 まで増やすことをお勧めします。大規模なシステムでは 50,000 まで増やします。この変更を有効にするためには、dispatcher.cnf の MAX_LIFE_CONNS の値を増やすことも重要です。キャッシュを有効にするためには、MAX_LIFE_CONNS を ALIAS_ENTRY_CACHE_SIZE の最低 2 倍、通常は 4 倍にする必要があります。アドレス変換については、設定可能になってメカニズムがより透過的になることを除き、操作を dirsync モードからダイレクト LDAP モードに変更してもほとんど効果はありません。

付録 C

iPlanet Messaging Server の Event Notification Service を管理する

この付録では、iPlanet Event Notification Service Publisher (ENS Publisher) を有効にし、iPlanet Messaging Server の iPlanet Event Notification Service (ENS) を管理するために必要な事柄について説明します。

この付録には、以下の節があります。

- iPlanet Messaging Server に ENS Publisher をロードする
- Event Notification Service のサンプルプログラムを実行する
- Event Notification Service を管理する

ENS および ENS API の詳細は、以下の iPlanet Calendar Server および Messaging Server のマニュアルの Web ページにある『iPlanet Messaging and Collaboration イベント通知サービスマニュアル』を参照してください。

iPlanet Messaging Server に ENS Publisher をロードする

Event Notification Service (ENS) は、iPlanet の基礎となる発行 / 購読サービスです。ENS は、iPlanet アプリケーションが関係する特定のタイプのイベントの収集の中心点として使用するディスパッチャとして機能します。イベントは、リソースの 1 つまたは複数のプロパティの値に変更されます。このようなタイプのイベントが発生する時期を知る必要があるアプリケーションを、ENS に登録します。ENS は、イベントを順番に識別し、通知と購読を照合します。

ENS と iBiff (iPlanet Messaging Server の ENS Publisher) は、iPlanet Messaging Server に含まれています。デフォルトでは、ENS は有効になっていますが、iBIFF はロードされていません (iPlanet Messaging Server に ENS Publisher をロードするにはを参照)。

iPlanet Messaging Server で通知を購読するには、iPlanet Messaging Server ホストに `libibiff` ファイルをロードしてから、Messaging Server を停止し、再起動します。

iPlanet Messaging Server に ENS Publisher をロードするには

コマンドラインから以下の手順を実行します。以下の手順では、iPlanet Messaging Server のインストールディレクトリの位置は *server_root* で、iPlanet Messaging Server ユーザは *mailsrv* です。これらの変数の一般的な値は、前者は `/usr/iplanet/server5`、後者は `mailsrv` です。

1. `mailsrv` として、`configutil` ユーティリティを実行して `libibiff` ファイルをロードします。

   `cd` *server_root*`/`**`msg`**`-`*instance*

```
./configutil -o "local.store.notifyplugin" -v "server_root/bin/msg/lib/libibiff"
```

2. `root` として、Messaging Server をいったん停止してから再起動します。

   `cd` *server_root*`/`**`msg`**`-`*instance*

   `./stop-msg`

   `./start-msg`

3. これで、ENS によって通知を受け取る準備ができました。詳細については、「Event Notification Service のサンプルプログラムを実行する」を参照してください。

Event Notification Service のサンプルプログラムを実行する

iPlanet Messaging Server には、通知の受信方法を学習するためのサンプルプログラムが含まれています。これらのサンプルプログラムは、*server_root*/bin/msg/enssdk/examples ディレクトリにあります。

ENS のサンプルプログラムを実行するには

1. *server_root*/bin/msg/enssdk/examples ディレクトリに移動します。
2. C コンパイラを使用して、Makefile.sample ファイルを使用する apub および asub の例をコンパイルします。*server_root*/bin/msg/lib ディレクトリを含むようにライブラリ検索パスを設定します。
3. プログラムをコンパイルしたら、それらを以下のように別々のウィンドウで実行することができます。

   apub localhost 7997

   asub localhost 7997

   apub ウィンドウで入力するものはすべて、asub ウィンドウに表示されます。また、デフォルト設定を使用している場合は、すべての iBiff 通知が asub ウィンドウに表示されます。
4. iBiff が発行した通知を受け取るには、asub.c と同様のプログラムを記述します。

   サンプルプログラムの詳細と ENS のプログラムを独自に記述する方法については、『iPlanet Messaging and Collaboration イベント通知サービスマニュアル』を参照してください。

注　*server_root*/bin/msg/lib ディレクトリを含むようにライブラリ検索パスを設定すると、その後はディレクトリサーバを停止して再起動することはできなくなります。これを回避するには、ライブラリ検索パスからエントリを削除します。

Event Notification Service を管理する

ENS の管理は、サービスの起動と停止、および、ENS の iBiff publisher の動作を制御するための設定パラメータの変更によって行います。

ENS を起動および停止する

ENS サーバを起動および停止するには、`start-msg ens` および `stop-message ens` コマンドを使用します。これらのコマンドは、`root` として実行する必要があります。

ENS を起動および停止するには

- ENS を起動するには、次のコマンドを実行します。

  server_root`/msg-`*instance*`/start-msg ens`
- ENS を停止するには、次のコマンドを実行します。

  server_root`/msg-`*instance*`/stop-msg ens`

iPlanet Event Notification Service 設定パラメータ

いくつかの設定パラメータが iBiff の動作を制御します。これらのパラメータを設定するには、`configutil` ユーティリティプログラムを使用します。

表 C-1 iBiff 設定パラメータ

パラメータ	説明
`local.store.notifyplugin.maxHeaderSize`	通知とともに送信されるヘッダーの最大サイズをバイト単位で指定する。デフォルトは 8192 バイト
`local.store.notifyplugin.maxBodySize`	通知とともに送信される本文の最大サイズをバイト単位で指定する。デフォルトは 100 バイト
`local.store.notifyplugin.`*eventType*`.enable`	指定のイベントタイプが通知を生成するかどうかを指定する。`ReadMsg`、`NewMsg` などのさまざまな *eventTypes* については、『iPlanet Messaging and Collaboration イベント通知サービスマニュアル』を参照。正当な値は 1 (有効にする) および 0 (無効にする)。デフォルト値は 1。つまり、`local.store.notifyplugin.ReadMsg.enable` を 0 に設定すると、`ReadMsg` 通知が無効になる

表 C-1　iBiff 設定パラメータ (続き)

パラメータ	説明
local.store.notifyplugin.ensHost	ENS サーバのホスト名を指定する。デフォルトは 127.0.0.1
local.store.notifyplugin.ensPort	ENS サーバの TCP ポートを指定する。デフォルトは 7997
local.store.notifyplugin.ensEventKey	ENS 通知用に使用するイベントキーを指定する。デフォルトは enp://127.0.0.1/store。イベントキーのホスト名部分は、ENS ホストの判別には使用されない。これは単に、ENS が使用する一意の識別子である。 このキーは、このキーと一致するイベントを通知するために、サブスクライバが購読する

メールユーザとメーリングリストを管理する

この付録では、Console インタフェースを使ってユーザのメールアカウントとメーリングリストを作成および管理する方法について説明します。ユーザとメーリングリストの作成および管理には、ここで説明するように Console インタフェースを使用しないことをお勧めします。ユーザとメーリングリストの作成および変更には、Console インタフェースではなく、iPlanet Delegated Administrator for Messaging for the Delegated Administrator のコマンドラインユーティリティを使用することをお勧めします。ユーザ / グループのコマンドラインユーティリティについては、iPlanet Messaging Server リファレンスマニュアルを参照してください。

警告 ここで説明するように Console インタフェースを使用してユーザやグループを作成すると、Delegated Administrator で表示や修正ができなくなります。ユーザおよびグループの作成や変更の際は、iPlanet Delegated Administrator for Messaging の Delegated Administrator コマンドラインユーティリティを使用するか、『iPlanet Messaging Server プロビジョニングガイド』に記載されている手順を使用することをお勧めします。

この付録には、以下の節があります。

- メールユーザを管理する
- メーリングリストを管理する

注 iPlanet Directory Server 5.1 をインストールしている場合は、同時にインストールした iPlanet Console 5.0 から管理します。iPlanet Messaging Server 5.2 は、同時にインストールした Netscape Console 4.2 から管理します。

メールユーザを管理する

メールユーザにアクセスするには

この項では、ユーザ用のメール管理インタフェースを開く方法について説明します。Messaging Server のメールアカウントは、ユーザエントリの属性として、中央の LDAP ユーザディレクトリ内に保存されているため、メールアカウントを操作するには、ディレクトリ内のユーザエントリにアクセスする必要があります。

新規ユーザを作成するには

新規メールアカウントを作成するには、ディレクトリ内で新規ユーザを作成し、そのユーザのメールアカウントをインストールします。メールアカウントをインストールしないと、そのユーザに対して Console のメール管理機能を使用することはできません (ユーザの作成およびユーザ情報の設定については、『Netscape Console によるサーバの管理』の第 4 章「ユーザおよびグループ」を参照)。

新規メールユーザを作成するには、次の手順に従います。

1. Console のメインウィンドウで「ユーザおよびグループ」タブをクリックします。
2. ドロップダウンリストから「新規ユーザ」を選択し、「作成」をクリックします。
3. ユーザが属する組織単位を選択し、「OK」をクリックします。「ユーザの作成」ウィンドウが開きます。
4. 『Netscape Console によるサーバの管理』の第 4 章「ユーザおよびグループの管理」を参照し、ユーザに関する情報を入力してください。
5. 「ユーザの作成」ウィンドウを開いたままの状態で、「アカウント」タブをクリックします。このユーザアカウントに対して使用できるインストール済み製品が右側のペインに一覧表示されます。
6. 「メールアカウントのインストール」ボックスをクリックします。「ユーザの作成」ウィンドウに「メール」タブが表示されます。
7. 「ユーザの作成」ウィンドウの「メール」タブをクリックしてから、右側のペインにある任意のタブをクリックします。
8. 必要に応じて内容を変更し、「ユーザの作成」ウィンドの下部にある「OK」をクリックします。

注	関連するタブで必要な作業をすべて完了したことを確認してから「OK」をクリックしてください。

既存のユーザにアクセスするには

既存のメールアカウントを変更する場合や、既存のユーザにメール機能を与える場合は、ユーザディレクトリ内でそのユーザにアクセスし、メールアカウントの属性を追加または変更します。

既存のユーザのメール情報にアクセスするには、次の手順に従います。

1. Console のメインウィンドウで「ユーザおよびグループ」タブをクリックします。
2. 「ユーザおよびグループ」のメインウィンドウで「検索」または「高度な検索」をクリックします。
3. 「検索」ウィンドウに検索条件 (ユーザの姓など) を入力し、ユーザディレクトリを検索します。
4. 「ユーザおよびグループ」のメインウィンドウに戻り、検索結果の中から任意のユーザを選択して「編集」をクリックします。
5. 「エントリの編集」ウィンドウに「メール」タブが表示されない場合は、以下の操作を実行します。
   a. 「アカウント」タブをクリックします。インストールされているアカウントが右側のペインに一覧表示されます。
   b. 「メールアカウント」チェックボックスをオンにします。「エントリの編集」ウィンドウに「メール」タブが表示されます。
6. 「エントリの編集」ウィンドウの「メール」タブをクリックしてから、右側のペインで任意のタブをクリックします。
7. 必要に応じて内容を変更し、「エントリの編集」ウィンドウの下部にある「OK」をクリックします。

ユーザの電子メールアドレスを指定するには

メールがユーザに正しく配信されるようにするには、まずユーザのメールアドレス情報を指定する必要があります。アドレス情報は、Messaging Server のホスト名、ユーザのプライマリアドレス、および代替アドレスから構成されています。ホスト名とプライマリアドレスは必ず指定する必要がありますが、代替アドレスは指定しなくてもかまいません。

ユーザのメールアドレス情報を指定するには、次の手順に従います。

1. Console から「ユーザの作成」ウィンドウまたは「エントリの編集」ウィンドウにアクセスします。手順については、574 ページの「メールユーザにアクセスするには」を参照してください。
2. 「メール」タブをクリックします。

3. 「設定」タブがアクティブになっていない場合は、クリックしてアクティブにします。
4. (必須) Messaging Server のホスト名を入力します。

   これは、ユーザのメールを処理する Messaging Server をホストするマシンです。Messaging Server がそのマシンで認識できる完全指定ドメイン名 (FQDN) を入力してください。
5. (必須) ユーザのプライマリ電子メールアドレスを入力します。

   プライマリアドレスは、ユーザのアドレスとして公開される電子メールアドレスです。ユーザが使用できるプライマリアドレスは 1 つだけです。RFC 821 仕様に準拠する有効な形式の SMTP アドレスを使用してください。

   送信メールのヘッダー部分に表示されるユーザアドレスにホスト名を表示しない場合は、プライマリ電子メールアドレスのフィールドにホスト名を入力しないでください。代わりに、以下に示される手順に従って、ホスト名を含む代替アドレスを指定します。
6. (省略可)「代替アドレス」リストにアドレスを入力します。

   代替アドレスとは、基本的にはそのユーザのプライマリアドレスのエイリアスに相当します。代替アドレスは、以下の目的に利用できます。

   - スペルを間違えやすいアドレスにメールが正しく配信されるようにする (たとえば、プライマリアドレスが「Smythe」の場合に、代替アドレスとして「Smith」と指定する)。
   - 送信メールのヘッダーにホスト名を表示しないようにする。ホスト名を非表示にするには、ユーザのプライマリ電子メールアドレスにはホスト名を含めず、代替アドレスにホスト名を含めます。たとえば、プライマリ電子メールアドレスを「`jsmith@siroe.com`」と指定し、代替アドレスを「`jsmith@sesta.com`」と指定します。こうすると、ユーザが送信したメールのヘッダーには `jsmith@siroe.com` と表示されますが、このアドレス宛てのメール (返信を含む) はすべて `jsmith@sesta.com` に配信されます (ただし、`sesta.com` が有効なホスト名である場合のみ)。

   重複しないかぎり、各ユーザに割り当てることができる代替アドレスの数に上限はありません。代替アドレス宛てに送信されたメッセージはすべてプライマリアドレスに配信されます。

   代替アドレスを追加するには、次の手順に従います。

   a. 「代替アドレス」フィールドの下にある「追加」ボタンをクリックします。
   b. 「代替アドレス」ウィンドウで代替アドレスを入力します。アドレス数に上限はありませんが、一度に複数のアドレスを追加することはできません。
   c. 「OK」をクリックして代替アドレスを追加し、「代替アドレス」ウィンドウを閉じます (別のアドレスを追加する場合は、もう一度「追加」ボタンをクリックして「代替アドレス」ウィンドウを表示します)。

7. ユーザのメール情報の変更が完了したら、「エントリの編集」ウィンドウの下部にある「OK」をクリックします。変更作業を続ける場合は、別のタブをクリックします。

配信オプションを設定するには

Messaging Server には 3 種類の主要なメール配信オプションがあり、各ユーザに対して任意の組み合わせのオプションを有効にして構成することができます。配信オプションには、標準 POP/IMAP 配信、プログラム配信、および UNIX 配信 (UNIX Messaging Server ホストのクライアント用) があります。

また、iPlanet Delegated Administrator for Messaging にもエンドユーザ向けの HTML インタフェースがあり、エンドユーザ自身がこれらのオプションを有効にしたり構成したりできるようになっています。Console インタフェースと Delegated Administrator のインタフェースは同じディレクトリ属性を操作するため、どちらか一方のインタフェースを開くと、オプションを設定したのが管理者であるかユーザであるかにかかわらず、最新の設定が表示されます。

ユーザの配信オプションを設定するには、次の手順に従います。

1. Console から「ユーザの作成」ウィンドウまたは「エントリの編集」ウィンドウにアクセスします。手順については、574 ページの「メールユーザにアクセスするには」を参照してください。
2. 「メール」タブをクリックします。
3. 「配信」タブをクリックします。
4. このユーザについて有効にする 1 つまたは複数の配信方法を選択します。
   - POP/IMAP 配信を指定する場合は、577 ページの「POP/IMAP 配信を指定する」を参照してください。
   - プログラム配信を指定する場合は、578 ページの「プログラム配信を指定する」を参照してください。
   - UNIX 配信を指定する場合は、578 ページの「UNIX 配信を指定するには」を参照してください。
5. ユーザのメール情報の変更が完了したら、「エントリの編集」ウィンドウの下部にある「OK」をクリックします。変更作業を続ける場合は、別のタブをクリックします。

POP/IMAP 配信を指定する

このオプションを選択すると、ユーザの標準 POP3 または IMAP4 メールボックスへの配信が可能になります。POP/IMAP 配信を有効にするには、次の手順に従います。

1. 「配信」タブをクリックします。
2. 「POP/IMAP」チェックボックスをオンにし、「プロパティ」ボタンをクリックして「POP/IMAP 配信」ウィンドウを開きます。
3. (省略可) メッセージの配信先および保存先であるメッセージストアパーティションのニックネーム (パス名または絶対物理パス以外) を入力します。このフィールドに何も入力しないと、現在のプライマリパーティションが使用されます。詳細は、343 ページの「メッセージストアを管理する」を参照してください。
4. (省略可) ユーザに割り当てるメール保存ディスク容量の上限を設定します。制限はデフォルト設定 (353 ページの「メッセージストアの制限容量を設定する」を参照)、無制限、または任意の容量 (KB/MB) にすることができます。
5. (省略可) ユーザの保存可能なメッセージ数の上限を設定します。制限はデフォルト設定 (353 ページの「メッセージストアの制限容量を設定する」を参照)、無制限、または任意の数にすることができます。

プログラム配信を指定する

このオプションを指定すると、メールがユーザに配信される前に外部アプリケーションに転送されて処理されるようになります。

注 この項では、ユーザがプログラム配信オプションを選択できるようにする方法について説明します。ただし、ユーザがこのオプションを使用できるようにする前に、まずいくつかの管理タスクを実行して、プログラム配信用のモジュール全体を有効にする必要があります。詳細については、275 ページの「パイプチャネルを使用してメッセージをプログラムに配信するには」を参照してください。

プログラム配信を有効にするには、次の手順に従います。

1. 「配信」タブをクリックします。
2. 「プログラム配信」チェックボックスをオンにし、「プロパティ」ボタンをクリックして「プログラム配信」ウィンドウを開きます。
3. ユーザのメールを処理するための外部アプリケーションコマンドを入力します。
4. 「OK」をクリックします。

UNIX 配信を指定するには

このオプションを指定すると、ユーザのメール配信方法が UNIX 配信に設定されます。つまり、メッセージが指定の UNIX メールボックスに配信されるようになります。このオプションは、ユーザの Messaging Server が UNIX ホストマシン上で稼働している場合にのみ選択できます。

UNIX 配信を有効にするには、次の手順に従います。

1. 「配信」タブをクリックします。
2. 「UNIX 配信」チェックボックスをオンにします。

注	Messaging Server ユーザが UNIX 配信を使用できるようにするには、通常の UNIX メール管理タスクを実行する必要があります。

転送先アドレスを指定するには

Messaging Server のメール転送機能を使用すると、ユーザのプライマリアドレスともう一つのアドレスの両方に、またはもう一つのアドレスにのみメールを転送することができます。

また、iPlanet Delegated Administrator for Messaging にはエンドユーザ向けの HTML インタフェースがあり、ユーザ自身が転送先アドレスを指定できるようになっています。Console インタフェースと Delegated Administrator のインタフェースは同じディレクトリ属性を操作するため、どちらか一方のインタフェースを開くと、オプションを設定したのが管理者であるかユーザであるかにかかわらず、最新の設定が表示されます。

ユーザの転送先アドレス情報を指定するには、次の手順に従います。

1. Console から「ユーザの作成」ウィンドウまたは「エントリの編集」ウィンドウにアクセスします。手順については、574 ページの「メールユーザにアクセスするには」を参照してください。
2. 「メール」タブをクリックします。
3. 「転送」タブをクリックします。

   ユーザの転送先アドレスがすでに指定されている場合は、「転送先アドレス」フィールドに情報が表示されます。
4. 転送先アドレスを追加する場合は、「追加」をクリックします。
5. 「転送先アドレス」ウィンドウで転送先アドレスを入力します。
6. 「OK」をクリックして「メールの転送」タブの「転送先アドレス」フィールドにアドレスを追加し、「転送先アドレス」ウィンドウを閉じます。

7. ユーザのメール情報の変更が完了したら、「エントリの編集」ウィンドウの下部にある「OK」をクリックします。変更作業を続ける場合は、別のタブをクリックします。

注	同一の Messaging Server 上にあり、かつほかの配信方法が設定されていないユーザアカウント間では、互いのアドレスを転送先アドレスに指定しないように注意してください。その場合、配信に支障をきたすことがあります。

自動返信設定を構成するには

iPlanet Messaging Server の自動返信機能を使用すると、受信メールに対して自動的に応答するように設定できます。自動返信には、エコーモード、Vacation モード、自動返信モードの 3 種類を指定できます。

また、iPlanet Delegated Administrator for Messaging にもエンドユーザ向けの HTML インタフェースがあり、エンドユーザ自身が自動返信設定を有効にしたり構成したりできるようになっています。Console インタフェースと Delegated Administrator のインタフェースは同じディレクトリ属性を操作するため、どちらか一方のインタフェースを開くと、オプションを設定したのが管理者であるかユーザであるかにかかわらず、最新の設定が表示されます。

自動返信サービスを有効にするには、次の手順に従います。

1. Console から「ユーザの作成」ウィンドウまたは「エントリの編集」ウィンドウにアクセスします。手順については、574 ページの「メールユーザにアクセスするには」を参照してください。
2. 「メール」タブをクリックします。
3. 「自動返信」タブをクリックします
4. 次のいずれかの自動返信モードを選択します。

   「オフ」: このユーザの自動返信機能を無効にします。

   「エコー」: 受信した各メッセージに対して自動的に返信します。このモードを選択した場合は、「メッセージ」フィールドに任意のメッセージを入力できます。

   「Vacation」: 各差出人から送られた最初のメッセージに対してのみ自動応答が生成されます。同一の差出人から複数のメッセージが送られてきた場合は、自動返信の設定がタイムアウトになるまで 2 通目以降のメッセージに対しては自動応答は生成されません。タイムアウトになると、次のタイムアウトまでの期間に受信した同一差出人からの最初のメッセージに対して、再び自動的に返信メッセージが送信されます。このモードを選択した場合は、「Vacation 開始日」および「Vacation 終了日」オプションを設定し、「返信テキスト」フィールドにメッセージを入力してください。

5. Vacation モードを選択した場合は、自動返信の開始日時と終了日時を設定する必要があります。
   - 「Vacation の開始 / スタート日」チェックボックスをオンにします。
   - 「編集」ボタンをクリックし、表示されたカレンダーで開始日時と終了日時を設定します。
6. タイムアウトを日または時間単位で設定します。
7. エコーモードまたは Vacation モードを選択した場合は、自動返信の件名およびメッセージを入力する必要があります。

   内部の差出人と外部の差出人に対して、それぞれ異なるメッセージを設定することができます。内部の差出人に対してのみ自動返信を設定すると、同じドメイン内の差出人だけにメッセージが送信されます。

   また、メッセージテキスト領域の上にあるドロップダウンリストから使用可能な言語を選択し、言語別のメッセージを作成することができます。
8. ユーザのメール情報の変更が完了したら、「エントリの編集」ウィンドウの下部にある「OK」をクリックします。変更作業を続ける場合は、別のタブをクリックします。

認証済みサービスを設定するには

ユーザがアクセスできるメールサービスを有効にするには、次の手順に従います。

1. Console から「ユーザの作成」ウィンドウまたは「エントリの編集」ウィンドウにアクセスします。手順については、574 ページの「メールユーザにアクセスするには」を参照してください。
2. 「メール」タブをクリックします。
3. 「認可されているサービス」タブをクリックします。

   「認可されているサービス」ウィンドウに、該当ドメインで使用できるサービスが表示されます。
4. サービスを追加、編集、削除するには、「追加」、「編集」、「削除」ボタンをそれぞれクリックします。いずれかのボタンをクリックすると、「認証済みサービスの規則を変更」ウィンドウが表示されます。
5. ドロップダウンリストから、規則を作成するサービス (IMAP、POP、SMTP、HTTP、またはすべて) を選択します。
6. 「許可」または「拒否」を選択し、規則を適用するドメインを指定します。
7. 「OK」をクリックして変更内容を反映させます。

メーリングリストを管理する

メーリングリストにアクセスするには

この項では、管理インタフェースからメーリングリストにアクセスする方法について説明します。Messaging Server のメーリングリストは、グループエントリの属性として LDAP ユーザディレクトリに保存されているため、メーリングリストを管理するには、ディレクトリグループにアクセスして修正する必要があります。

新規グループを作成するには

新規メーリングリストを作成するには、ディレクトリ内で新規グループを作成し、そのグループ用のメールアカウントをインストールします。メールアカウントをインストールしないと、そのグループに対して Console のメール管理機能を使用することはできません (グループの作成およびグループ情報の設定については、『Netscape Console によるサーバの管理』の第 4 章「ユーザおよびグループ」を参照)。

新規メーリングリストを作成するには、次の手順に従います。

1. Console のメインウィンドウで「ユーザおよびグループ」タブをクリックします。
2. ドロップダウンリストから「新規グループ」を選択し、「作成」をクリックします。
3. グループが属する組織単位を選択し、「OK」をクリックします。
4. 詳細については、『Netscape Console によるサーバの管理』の第 4 章「ユーザおよびグループの管理」を参照してください。

   注 : メーリングリストの作成だけを目的とする場合は、「ユーザおよびグループのメンバー」タブからメンバーを追加する必要はありません。「Mail account Email-Only Members」タブを使用できます。

   - グループの正規メンバーには、メーリングリストに関する完全な権限だけでなく、グループのメンバーに指定されているほかのすべての権限が与えられます。正規メンバー (静的または動的) を追加するには、「メンバー」タブを使用します。
   - メーリングリストメンバーには、グループの作成目的がメーリングリストの使用だけであるかどうかにかかわらず、グループのメーリングリストに関する権限しか与えられません。メーリングリストメンバーは、電子メール専用メンバーと呼ばれます。電子メール専用メンバーを追加するには、「メンバー」タブを使用します。
5. 「グループの作成」ウィンドウを開いたままの状態で、「アカウント」タブをクリックします。

   このグループアカウントに対して使用できるインストール済み製品が右側のペインに一覧表示されます。

6. 「メールアカウント」チェックボックスをオンにします。

   「グループの作成」ウィンドウに「メール」タブが表示されます。

7. 「グループの作成」ウィンドウの「メール」タブをクリックしてから、右側のペインにあるタブをクリックします。

8. 必要に応じて内容を変更し、「グループの作成」ウィンドウの下部にある「OK」をクリックします。

   エントリが作成され、「グループの作成」ウィンドウが閉じます。

注　メール管理用の各ウィンドウの下部にある「OK」ボタンをクリックすると、メール管理用の各タブを使って設定した情報がすべて有効になります。必要な作業をすべて完了したことを確認してから「OK」をクリックしてください。

既存のグループにアクセスするには

既存のメーリングリストに変更する場合や、既存のグループにメーリングリスト機能を与える場合は、ユーザディレクトリ内でそのグループにアクセスし、メールアカウントの属性を追加または変更します。

既存のグループのメーリングリスト情報にアクセスするには、次の手順に従います。

1. Console のメインウィンドウで「ユーザおよびグループ」タブをクリックします。

2. 「ユーザおよびグループ」のメインウィンドウで「検索」または「高度な検索」をクリックします。

3. ウィンドウに検索条件 (グループ名など) を入力し、ユーザディレクトリを検索します。

4. 「ユーザおよびグループ」のメインウィンドウに戻り、検索結果の中から任意のグループを選択して「編集」をクリックします。

5. 「エントリの編集」ウィンドウに「メール」タブが表示されない場合は、以下の操作を実行します。

   - 「アカウント」タブをクリックします。インストールされているアカウントが右側のペインに一覧表示されます。
   - 「メールアカウント」チェックボックスをオンにします。「エントリの編集」ウィンドウに「メール」タブが表示されます。

6. 「エントリの編集」ウィンドウで「メール」タブをクリックしてから、右側のペインで任意のタブをクリックします。

   これらのタブは、「グループの作成」ウィンドウからアクセスできるタブと同一のものです。

7. 必要に応じて内容を変更し、「エントリの編集」ウィンドウの下部にある「OK」をクリックします。

メーリングリスト設定を指定するには

メールがメーリングリストに正しく配信されるようにするには、まずリストのメールアドレス情報を指定する必要があります。メールアドレス情報は、グループのプライマリアドレス、およびプライマリアドレスのエイリアスである代替アドレスから構成されます。さらに、メーリングリストの所有者、説明、メンバー、属性、制約、返信に関するアクションなどを指定することもできます。

メーリングリスト情報を指定するには、次の手順に従います。

1. Console から「グループの作成」ウィンドウまたは「エントリの編集」ウィンドウにアクセスします。手順については、582 ページの「メーリングリストにアクセスするには」を参照してください。
2. 「メール」タブをクリックします。
3. 「設定」タブがアクティブになっていない場合は、クリックしてアクティブにします。
4. (必須) メーリングリストのプライマリ電子メールアドレスを入力します。

   プライマリアドレスは、このメーリングリストのアドレスとして公開されるアドレスです。各メーリングリストに複数のプライマリアドレスを設定することはできません。また、プライマリアドレスには RFC 821 に準拠する有効な形式の SMTP アドレスを使用してください。
5. (省略可) メーリングリストの代替アドレスを指定します。

   代替アドレスとは、グループのプライマリアドレスのエイリアスに相当します。代替アドレスは、以下の目的に利用できます。

   - スペルを間違えやすいアドレスにメールが正しく配信されるようにする。
   - 送信メールのヘッダーにホスト名を表示しないようにする。ホスト名を非表示にするには、グループのプライマリ電子メールアドレスにはホスト名を含めず、代替アドレスにホスト名を含めます。

   重複しないかぎり、各グループに割り当てることができる代替アドレスの数に上限はありません。代替アドレス宛てに送信されたメッセージはすべてプライマリアドレスに配信されます。

   代替電子メールアドレスを追加するには、次の手順に従います。

   a. 「代替電子メールアドレス」フィールドの下にある「追加」ボタンをクリックします。

b. 「代替電子メールアドレス」ウィンドウで代替アドレスを入力します。アドレス数に上限はありませんが、一度に複数のアドレスを追加することはできません。

c. 「OK」をクリックして代替アドレスを追加し、「代替電子メールアドレス」ウィンドウを閉じます (別のアドレスを追加する場合は、もう一度「追加」ボタンをクリックして「代替電子メールアドレス」ウィンドウを表示します)。

6. (省略可) 「Errors-to」フィールドに、メーリングリスト宛てに送信されたメッセージが配信不能の場合に、エラーメッセージの送信先となる電子メールアドレスを入力します。

7. (省略可)「Messaging Server のホスト名」フィールドにメーリングリストをホストするマシンのホスト名を入力します。

「プライマリ電子メールアドレス」フィールドにホスト名が含まれている場合は、このフィールドは空白でもかまいません。プライマリ電子メールアドレスでホスト名を省略した場合は、必ずここでホスト名を指定してください。

ユーザのメールアカウントの場合とは異なり、メーリングリストのホスト名を指定しない場合は、そのリストの LDAP エントリにアクセスできるすべてのホストがリストを処理できることになります (多くの場合は、故意にそのような設定が使われます)。特定ホストのみがリストを処理できるように設定する場合は、ホスト名を指定する必要があります。たとえば、大規模なリストを負荷の小さいサーバで処理するように設定すれば、ほかのサーバの負荷を軽減できます。

注意 : このウィンドウで一度に複数のホスト名を入力することはできません。複数のホスト名を入力するには、`ldapmodify` コマンドラインユーティリティを使用してください。

8. (省略可) メーリングリストの所有者を入力します。

リスト所有者には、ユーザの追加や削除、設定の変更、リストの削除などの管理権限が与えられます。

メーリングリストの所有者を指定するには、「所有者」タブをクリックして、以下のいずれかの操作を実行します。

- 「追加」をクリックし、「リスト所有者の DN を入力」ウィンドウで新しい所有者の識別名 (DN) を入力し (例 : `uid=jsmith`、`ou=people`、`o=siroe.com`)、「OK」をクリックします。
- 「検索」をクリックして、「ユーザおよびグループを検索」ウィンドウを開き、所有者を検索します。

注意 : このウィンドウで所有者を選択すると、自動的に適切な DN のシンタックスが表示されます。「ユーザおよびグループを検索」ウィンドウの詳細については、『Netscape Console によるサーバの管理』の第 4 章「ユーザおよびグループの管理」を参照してください。

9. (省略可) 説明を追加します。

Messaging Server が使用するためではなく、説明としてテキストや URL を入力するには、「説明」タブをクリックし、以下のいずれかまたは両方を行います。

- メーリングリストの目的や特徴に関する説明を入力します。
- メーリングリストについての追加情報が記載されている HTML ページの URL を入力します。この情報は参考用であり、Messaging Server が使用するためのものではないことに注意してください。

10. メーリングリスト情報の設定が完了したら、「エントリの編集」ウィンドウの下部にある「OK」をクリックします。変更作業を続ける場合は、別のタブをクリックします。

リストメンバーを指定するには

メーリングリストに電子メール専用メンバーを追加するには、以下のいずれかまたは両方を行います。

- メンバーを 1 人ずつメーリングリストに追加します。
- グループのメンバーを決定するフィルタとして、ユーザディレクトリに適用する動的検索条件を定義します。

ここでは、Console の「ユーザおよびグループ」インタフェース上で電子メール専用メンバーと呼ばれるメーリングリストメンバーについて説明します。電子メール専用メンバーには、グループのメーリングリストに関する権限のみが与えられます。正規メンバーを追加する手順については、『Netscape Console によるサーバの管理』を参照してください。通常、正規メンバーには、電子メール専用メンバーより多くの権限や責任が与えられます。グループの詳細については、『Netscape Console による Managing Server の管理』の 第 4 章「ユーザおよびグループの管理」を参照してください。

メンバーの動的検索条件を定義するには

動的検索条件は、ユーザディレクトリ内でメンバーを検索する際にフィルタとして適用される LDAP 検索 URL によって構成されています。グループ宛てにメッセージが届くと、このメカニズムによって、名前の静的なリストではなく、ディレクトリ検索に基づいて、メッセージが配信されるユーザが決まります。そのため、各メンバーの情報を詳細にたどらなくても、大規模で複雑なグループを作成して管理することができます。

LDAP 検索フィルタには、必ず LDAP URL のシンタックスの形式を使用してください。LDAP フィルタの作成の詳細については、『Netscape Console によるサーバの管理』の第 4 章「ユーザおよびグループの管理」を参照してください。iPlanet Directory Server マニュアルおよび RFC 1959 も参照してください。

LDAP URL のシンタックスは、次のとおりです。

```
ldap://hostname:port/base_dn?attributes?scope?filter
```

URL の各オプションには、以下の意味があります。

表 D-1 LDAP URL オプション

オプション	説明
hostname	Directory Server のホスト名 (デフォルトは Messaging Server が使用する Directory Server のホスト名)
port	LDAP サーバのポート番号。ポート番号を指定しない場合は、Messaging Server が使用するデフォルトの標準 LDAP ポートが使用される
base_dn	検索ベースとして使用されるディレクトリエントリの識別名。必ず指定する必要がある
attributes	検索結果として返される属性。これらの属性は、Messaging Server によって返される
scope	検索範囲
	「`base`」を指定すると、検索ベース (*base_dn*) レベルの情報のみが検索対象になる
	「`one`」を指定すると、検索ベースの 1 つ下のレベルの情報が検索対象になる (検索ベースレベルは含まれない)
	「`sub`」を指定すると、検索ベースおよびその下のレベルにあるすべての情報が検索対象になる
filter	検索範囲内のエントリに適用される検索フィルタ。フィルタを指定しない場合は、(`objectclass=*`) が使用される

以下に、「Sunnyvale」をメールホストとするユーザをフィルタリングする LDAP 検索 URL の例を示します。

```
ldap:///o=Siroe Corp,c=US??sub?(&(mailHost=sunnyvale.siroe.com)
  (objectClass=inetLocalMailRecipient))
```

この URL は、組織名が Siroe (`o=Siroe`)、所在地が米国 (`c=US`)、メールホスト名が Sunnyvale (`mailHost=sunnyvale`) のユーザをフィルタリングするためのものです。`objectClass` 属性は、検索対象のエントリの種類を定義するもので、この場合は `inetLocalMailRecipient` (`objectClass=inetLocalMailRecipient`) となっています。

Console を使用して検索フィルタを作成した場合、グループ名はすべて無視され、検索結果にはユーザ名だけが表示されることに注意してください。これは、グループメンバーでもあるユーザの名前が重複して表示されることを避けるための設定です。コマンドライン設定ユーティリティ (`configutil`) を使うとこの設定を無効にすることができますが、コマンドラインの使用はできるかぎり避けてください。

次の項で説明しているとおり、検索 URL は、Console のテンプレートウィンドウ (「LDAP 検索 URL の作成」ウィンドウ) を使用して作成できます。

メーリングリストにメンバーを追加するには

メーリングリストに (電子メール専用) メンバーを追加するには、次の手順に従います。

1. Console から「グループの作成」ウィンドウまたは「エントリの編集」ウィンドウにアクセスします。手順については、582 ページの「メーリングリストにアクセスするには」を参照してください。
2. 「メール」タブをクリックします。
3. 「電子メール専用メンバー」タブをクリックします。
   - (省略可) メンバーの検索に LDAP 検索 URL を使用する場合は、「Dynamic criteria for email-only membership」フィールドの下にある「追加」ボタンをクリックし、「Add Dynamic Criterion」ウィンドウで次の手順を実行します。
   - フィールドに LDAP 検索 URL を入力するか、または「構築」ボタンをクリックして「LDAP 検索 URL の作成」ウィンドウ (検索 URL の構築に使用するテンプレート) を開きます。
   - 「OK」をクリックして「Dynamic criteria for email-only membership」フィールドに入力した条件を有効にし、「Add Dynamic Criterion」ウィンドウを閉じます。

   LDAP 検索 URL の作成については、586 ページの「メンバーの動的検索条件を定義するには」を参照してください。
4. (省略可) メーリングリストに個々のメンバーを追加するには、「電子メール専用のメンバー」フィールドの下にある「追加」ボタンをクリックし、「電子メール専用メンバーの追加」ウィンドウで次の手順を実行します。
   - フィールドに新規メンバーのプライマリアドレスを入力します。RFC 821 に準拠する有効な形式の SMTP アドレスを入力してください。グループに制約を設定する場合は特に、代替アドレスは指定しないでください。フィールドに複数のアドレスを入力することはできないため、このウィンドウで一度に複数のメンバーを追加することはできません。
   - 「OK」をクリックしてリストにメンバーを追加し、「電子メール専用メンバーの追加」ウィンドウを閉じます。別のアドレスを入力するには、もう一度「追加」をクリックして、「電子メール専用メンバーの追加」ウィンドウを開きます。

5. メーリングリスト情報の設定が完了したら、「エントリの編集」ウィンドウの下部にある「OK」をクリックします。変更作業を続ける場合は、別のタブをクリックします。

メッセージ送信に関する制約を定義するには

メーリングリスト宛てに送信されるメッセージにさまざまな制約を設けることができます。たとえば、特定のユーザだけにリストへの送信を許可する、差出人の認証を要求する、メッセージの送信元を制限する、メッセージのサイズを制限する、などの制約を設けることができます。制約に違反するメッセージは拒否されます。

注	これらの制約は、リスト宛てに送信されるメッセージを制御するためには便利ですが、安全性の高いアクセス制御を保証するものではありません。

グループに対するメッセージ送信の制約を定義するには、次の手順に従います。

1. Console から「グループの作成」ウィンドウまたは「エントリの編集」ウィンドウにアクセスします。手順については、582 ページの「メーリングリストにアクセスするには」を参照してください。
2. 「メール」タブをクリックします。
3. 「制約」タブをクリックします。
4. (省略可) 次のいずれかのオプションを選択して、送信を許可する差出人を定義します。
   - **「すべて」:** 差出人を制限しません (デフォルトの設定)。ただし、このオプションを選択すると、次の手順で説明している SMTP 認証を選択できなくなることに注意してください。
   - **「メーリングリストのすべて」:** メーリングリストメンバー (電子メール専用メンバー以外のグループメンバーも含む) だけにリストへのメッセージ送信を許可します。
   - **「次のリストのすべて」:** フィールドに明示的に指定されたユーザだけにリストへのメッセージ送信を許可します。

   「次のリストのすべて」を選択した場合、リストに差出人を追加するには、「許可された差出人」フィールドの下にある「追加」をクリックするか、または「検索」をクリックして、「ユーザおよびグループを検索」ウィンドウを開きます。「追加」をクリックすると、「許可された差出人の追加」ウィンドウが開きます。フィールドに許可する差出人の電子メールアドレスまたは識別名 (DN) を入力します。「OK」をクリックして「許可された差出人」フィールドにユーザを追加し、「許可された差出人の追加」ウィンドウを閉じます。上記の手順を繰り返して許可する差出人をすべて追加します。

「ユーザおよびグループを検索」ウィンドウの詳細については、『Netscape Console によるサーバの管理』を参照してください。

5. (省略可) 送信元を制限するために、許可された差出人のドメインを定義します。
   - 「許可された差出人ドメイン」フィールドの下にある「追加」ボタンをクリックします。
   - 「許可された差出人ドメインの追加」ウィンドウでドメイン名を入力し、「OK」をクリックしてドメインをリストに追加します。

   入力したドメインにサブドメインがある場合は、それらのサブドメインもすべて自動的に含まれることに注意してください。たとえば、`siroe.com` には `sales.siroe.com` が含まれます。
6. (省略可) メッセージサイズの上限を指定します。

   サイズをバイト単位で入力してください。
7. メーリングリスト情報の設定が完了したら、「エントリの編集」ウィンドウの下部にある「OK」をクリックします。変更作業を続ける場合は、別のタブをクリックします。

モデレータを定義するには

メーリングリストには、1 人または複数のモデレータを追加できます。

モデレータが転送メッセージを受信すると、その処理方法はモデレータが決定します (モデレータが複数存在する場合は、最初のモデレータが処理方法を決定します)。処理には、メッセージの承認とリストへのメッセージの転送 (通常、パスワードを使用)、またはメッセージの削除が含まれます。

メーリングリストのモデレータを定義するには、次の手順に従います。

1. Console から「グループの作成」ウィンドウまたは「エントリの編集」ウィンドウにアクセスします。手順については、582 ページの「メーリングリストにアクセスするには」を参照してください。
2. 「メール」タブをクリックします。
3. 「モデレータ」タブをクリックします。
4. 「モデレータのリスト」フィールドの下にある「追加」ボタンをクリックします。
5. 「モデレータの追加」ウィンドウで、モデレータのプライマリ電子メールアドレスまたは識別名 (DN) を入力します。アドレスを入力するか、または「検索」をクリックして「ユーザおよびグループを検索」ウィンドウを開き、アドレスを検索します。「モデレータの追加」ウインドウでは、一度に複数のモデレータを追加することはできません。

「ユーザおよびグループを検索」ウィンドウの詳細については、『Netscape Console によるサーバの管理』を参照してください。

6. 「OK」をクリックして「モデレータのリスト」にモデレータを追加し、「モデレータの追加」ウィンドウを閉じます (別のアドレスを追加するには、もう一度「追加」をクリックして「モデレータの追加」ウィンドウを開きます)。
7. メーリングリスト情報の設定が完了したら、「エントリの編集」ウィンドウの下部にある「OK」をクリックします。変更作業を続ける場合は、別のタブをクリックします。

用語集

/var/mail　新しいメールメッセージを順番に単一のフラットテキストファイル内に格納する Berkeley 方式の受信箱を示すために使用される名前。

A レコード (A record)　ホスト名および関連付けられた IP アドレスを含む DNS レコードの一種。A レコードは、Messaging Server がインターネット上で電子メールをルーティングするために使用します。**ドメイン名システム (DNS)**、**MX レコード**も参照してください。

Administration Server 管理者 (administration server administrator)　Directory Server に接続していない場合でも、サーバの起動および停止を行う管理権限を持つユーザ。
Administration Server 管理者は、ローカルサーバグループ内のすべてのサーバに対する制限されたサーバタスク (通常はサーバの再起動と停止のみ) を実行できます。
Administration Server をインストールすると、この管理者のエントリが自動的にローカルに作成されます (この管理者はユーザディレクトリ内のユーザではありません)。

APOP　Authenticated Post Office Protocol の略。POP (Post Office Protocol) に似ていますが、認証にはプレーンテキストによるパスワードではなく、暗号化したパスワードとチャレンジ文字列を使用します。

attributes　LDAP データは、属性と値のペアとして表されます。個々の情報は、記述属性に関連付けられています。**使用可能な属性**、**必須の属性**も参照してください。

AUTH　SMTP コマンドの 1 つ。SMTP クライアントがサーバに対して認証方法を指定したり、認証プロトコル交換を実行したり、必要に応じて後続プロトコルの相互対話で使用するセキュリティ層をネゴシエートしたりできるようにします。

Berkeley DB　トランザクション用のデータベースストアで、読み取りと書き込みの同時実行の負荷が大きく、さらにトランザクションと回復可能性が要求されるアプリケーションで使用します。

CA　認証局。デジタル証明書 (デジタルの識別子) を発行し、その公開鍵を対象者が広く利用できるようにする組織。

capability　クライアントに提供され、特定の IMAP サービスで利用可能な機能を定義する文字列。

cipher　暗号化で使用されるアルゴリズム。

ciphertext (暗号文)　暗号化されたテキスト。**cleartext (平文)** の対語です。

cleartext (平文)　暗号化されていないテキスト。

CLI　**コマンドラインインタフェース**を参照してください。

cn　共通名を表す LDAP エイリアス。

CNAME レコード (CNAME record)　ドメイン名のエイリアスをドメイン名にマップする DNS レコードの一種。

Configuration Directory Server　単一サーバまたはサーバのセットの構成情報を保持する Directory Server。

cookie　特定の Web サイトを訪れたときにブラウザのメモリに自動的に入力されるテキストのみの文字列。cookie は、Web ページ作成者によってプログラムされます。ユーザは、cookie を受け入れることも、拒否することもできます。cookie を受け入れると、Web ページを高速に読み込むことができます。ユーザのマシンのセキュリティを脅かすものではありません。

CRAM-MD5　RFC 2195 に記述されている軽量な標準化過程の認証方法。ネットワークでユーザのログインパスワードだけを保護する場合に、TLS (SSL) の代わりに使用できます。TLS より高速ですが、やや強度が落ちます。

cronjob　UNIX 専用。指定した時間に cron デーモンによって自動的に実行されるタスク。**crontab ファイル**も参照してください。

crontab ファイル (crontab file)　UNIX 専用。指定した時間に自動的に実行されるコマンドのリスト。1 行に 1 つずつ記述されています。

DC ツリー (DC Tree)　ドメインコンポーネント (Domain Component) ツリー。DNS ネットワーク構造を反映するディレクトリ情報ツリー。DC ツリー内の識別名は、`cn=billbob,dc=bridge,dc=net,o=internet` のようになります。

Delegated Administration Server　ホストドメインによるディレクトリへのアクセス制御を処理するデーモンプログラム。

Delegated Administrator Console　Web ブラウザベースのソフトウェアコンソール。ドメイン管理者はこれを使用して、ホストドメインに対してユーザやグループの追加または変更を行うことができます。また、エンドユーザは、これを使用して、自分のパスワードの変更、メッセージ転送規則の設定、Vacation 規則の設定、メールリスト購読の一覧表示などを行うことができます。

Delegated Administrator for Messaging and Collaboration ドメイン管理者がホストドメインに対してユーザやグループの追加または変更を行うために使用する一連のインタフェース (GUI とユーティリティ)。

DIGEST-MD5　CRAM-MD5 より安全で軽量な標準化過程の認証方法。RFC 2831 に記述されています。RFC 2831 には、TLS (SSL) のような設定の手間をかけずに接続全体を保護するオプションも記述されています。

Directory Manager　ディレクトリサーバデータベースの管理権限を持つユーザ。アクセス制御はこのユーザには適用されません。Directory Manager はディレクトリのスーパーユーザと考えることができます。

Directory Server　LDAP に基づく iPlanet のディレクトリサービス。**ディレクトリサービス**、**Lightweight Directory Access Protocol**、**Configuration Directory Server**、**Users and Groups Directory Server** も参照してください。

DIT　**ディレクトリ情報ツリー**を参照してください。

DN　**識別名**を参照してください。

dn　識別名のための LDAP エイリアス。**識別名**も参照してください。

DNS　**ドメインネームシステム**を参照してください。

DNS エイリアス (DNS alias)　DNS サーバが、別のホスト (DNS CNAME レコード) へのポインタとして認識するホスト名。マシンの実際の名前は 1 つだけですが、1 つまたは複数のエイリアスを持つことができます。たとえば、`www.siroe.domain` を現在サーバが置かれている `realthing.siroe.domain` という実際のマシンをポイントするエイリアスとすることができます。

DNS スプーフィング (DNS spoofing)　DNS サーバが不正情報を提供するように仕向けるネットワーク攻撃の形態。

DNS データベース (DNS database)　ドメイン名 (ホスト名) および対応する IP アドレスのデータベース。

DNS ドメイン (DNS domain)　共通の接尾辞 (ドメイン名) の付いたホスト名を持つコンピュータのグループ。構文的には、ピリオド (ドット) で区切られた一連の名前 (ラベル) から成るインターネットドメイン名です。たとえば `corp.mktng.siroe.com` などです。**ドメイン**も参照してください。

DSN　配信ステータス通知を参照してください。

dsservd　ディレクトリ情報が格納されたデータベースにアクセスし、LDAP プロトコルを使用してディレクトリクライアントと通信するデーモン。

dssetup　既存の Directory Server を iPlanet Messaging Server で使用できるように準備する Directory Server 準備ツール。

EHLO コマンド (EHLO command)　サーバが拡張 SMTP コマンドをサポートするかどうかをサーバに照会するための SMTP コマンド。RFC 1869 に定義されています。

ESMTP　Extended Simple Mail Transfer Protocol を参照してください。

ESP　Enterprise Service Provider (エンタープライズサービスプロバイダ) の略。

ETRN　クライアントからサーバに対して、サーバ上でクライアントマシンを待機しているメッセージのメールキューの処理を開始するように要求する SMTP コマンド。RFC 1985 に定義されています。

EXPN　メールリストを展開するための SMTP コマンド。RFC 821 に定義されています。

Extended Simple Mail Transfer Protocol (ESMTP)　インターネットメッセージ転送プロトコルの一種。ESMTP では、SMTP コマンドセットにオプションのコマンドを追加することで、その機能が拡張されています。ESMTP サーバが、リモートサイトで実装されているコマンドを検出する機能などが含まれます。

facility (機能)　Messaging Server ログファイルエントリ内での、ログエントリを生成したソフトウェアサブシステム (ネットワークやアカウントなど) の指定。

FQDN　完全指定ドメイン名を参照してください。

GUI　グラフィカルユーザインタフェース。

HA　高可用性を参照してください。

hashdir　特定ユーザのメッセージストアが含まれるディレクトリを調べるためのコマンドラインユーティリティ。

HTTP　HyperText Transfer Protocol を参照してください。

HyperText Transfer Protocol (HTTP)　Web 上でハイパーテキストドキュメントの転送を可能にするための標準プロトコル。iPlanet Messaging Server は、Web ベースの電子メールをサポートするために HTTP サービスを提供しています。**Messenger Express** も参照してください。

IDENT　Identification Protocol を参照してください。

Identification Protocol　特定の TCP 接続のリモート端末を制御するリモートプロセスを識別できるようにするプロトコル。RFC 1413 に定義されています。

IMAP4　**Internet Message Access Protocol Version 4** を参照してください。

imsadmin コマンド (imsadmin commands)　ドメイン管理者、ユーザ、およびグループを管理するためのコマンドラインユーティリティのセット。

imsimta コマンド (imsimta commands)　MTA (Message Transfer Agent) の各種の保守、テスト、管理を行うためのコマンドラインユーティリティのセット。

INBOX　メール配信用のユーザのデフォルトメールボックス用に予約されている名前。INBOX は、大文字と小文字が区別されない唯一のフォルダ名です。たとえば、以下のようになります。INBOX、Inbox、inbox は、すべてユーザのデフォルトのメールボックスとして有効な名前です。

instance_root　**インスタンスディレクトリ**を参照してください。

Internet Message Access Protocol Version 4 (IMAP4)　ユーザがメインのメッセージ送信システムから切断された場合でもメールを処理できるようにする標準プロトコル。IMAP 仕様により、切断されたユーザの管理制御が可能になるとともに、メッセージングシステムに再接続したときにユーザのメッセージストアの同期化が可能になります。

IP　**インターネットプロトコル**を参照してください。

IP アドレス (IP address)　`198.93.93.10` のような、ドットで区切られた一連の数値で、イントラネットまたはインターネット上でのマシンの実際の場所を示します。TCP/IP を使用するホストには、32 ビットのアドレスが割り当てられます。

iPlanet Setup　すべての iPlanet サーバおよび iPlanet Console 用のインストールプログラム。

ISP　Internet Service Provider (インターネットサービスプロバイダ) の略。電子メール、電子カレンダー、World Wide Web へのアクセス、Web ホスティングなどのインターネットサービスを顧客に提供する会社です。

LDAP　**Lightweight Directory Access Protocol** を参照してください。

LDAP Data Interchange Format (LDIF)　Directory Server エントリをテキスト形式で表すために使用する形式。

LDAP 検索文字列 (LDAP search string)　ディレクトリ検索に使用される属性を定義する、置換可能なパラメータを含む文字列。たとえば、「uid=%s」という LDAP 検索文字列は、検索の基準がユーザ ID 属性であることを意味します。

LDAP サーバ (LDAP Server)　LDAP ディレクトリを管理し、そのディレクトリに対する照会サービスを提供するソフトウェアサーバ。iPlanet ディレクトリサービスは LDAP サーバの実装です。

LDAP サーバフェイルオーバー (LDAP server failover)　LDAP サーバのバックアップ機能。1 つの LDAP サーバに障害が発生した場合、システムは、別の LDAP サーバに切り替えることができます。

LDAP 参照 (LDAP referrals)　別の LDAP エントリへのシンボリックリンク (参照) から成る LDAP エントリ。LDAP 参照は、LDAP ホストと識別名から構成されます。通常、LDAP 参照は、データを複製せずに、既存の LDAP データを参照するために使用されます。また、移動された特定のエントリに依存するプログラムの互換性を維持するためにも使用されます。

LDAP フィルタ (LDAP filter)　特定の属性または属性値の有無に基づいて一連のエントリを指定する方法。

LDBM　LDAP Data Base Manager の略。

LDIF　**LDAP Data Interchange Format** を参照してください。

Legato Networker　Legato® が提供するサードパーティ製バックアップユーティリティ。

Lightweight Directory Access Protocol (LDAP)　TCP/IP を介して複数のプラットフォーム上で実行できるように設計されたディレクトリサービスプロトコル。X.500 Directory Access Protocol (DAP) を簡素化したもので、ユーザプロファイル、メールリスト、複数の iPlanet サーバ上の設定データなどの情報の格納、検索、および配布を単一の場所で管理できるようにします。iPlanet Directory Server は、LDAP プロトコルを使用します。

MD5　RSA Data Security によって提供されるメッセージダイジェストアルゴリズム。MD5 を使用すると、一意になる確率が高い短い形式のダイジェストデータを生成できます。同一のメッセージダイジェスト電子メールが生成されるようなデータを作成することは数学的に非常に困難です。

Message Handling System (MHS)　接続されている MTA、ユーザエージェント、およびメッセージストアのグループ。

Message Transfer Agent (MTA)　メッセージのルーティングと配信専用のプログラム。複数の MTA が連携してメッセージを転送し、目的の受取人に配信します。MTA は、メッセージをローカルのメッセージストアに配信するのか、リモート配信のために別の MTA にルーティングするのかを決定します。

Messaging Multiplexor　複数のメールサーバに対する単一接続ポイントとして機能し、複数のメールボックスホストを利用する多数のユーザへの配信を円滑に行うための特別な iPlanet Messaging Server。

Messaging Server 管理者 (Messaging Server administrator)　iPlanet Messaging Server インスタンスのインストールや管理などの権限を持つ管理者。

Messenger Express Multiplexor　マルチプレクサとして機能するメッセージ処理用プロキシサーバで、ユーザが iPlanet Messaging Server の HTTP サービス (Messenger Express) に接続できるようにします。Messenger Express Multiplexor を使用すると、複数のサーバマシンにユーザを分散できるようになります。

MessengerExpress　ユーザがブラウザベース (HTTP) のインタフェースを介してメールボックスにアクセスできるようにするメールクライアント。メッセージ、フォルダ、その他のメールボックス情報は、HTML 形式でブラウザのウィンドウに表示されます。**Web メール**も参照してください。

MHS　**Message Handling System** を参照してください。

MIME　**Multipurpose Internet Mail Extension** を参照してください。

MMP　**Messaging Multiplexor** を参照してください。

MTA　**Message Transfer Agent** を参照してください。

MTA 設定ファイル (MTA configuration file)　Messaging Server のすべてのチャネル定義と、ルーティングのためのアドレス書き換え規則を含むファイル (`imta.cnf`)。**チャネル**、**書き換え規則**も参照してください。

MTA ディレクトリキャッシュ (MTA directory cache)　ユーザおよびグループに関するディレクトリサービス情報のスナップショットで、MTA がメッセージを処理するために必要とします。**ディレクトリの同期**も参照してください。

MTA ホップ (MTA hop)　MTA 間でメッセージをルーティングする処理。

MUA　**ユーザエージェント**を参照してください。

Multiplexor　**Messaging Multiplexor** を参照してください。

Multipurpose Internet Mail Extension (MIME)　電子メールメッセージ内にマルチメディアファイルを追加できるようにするために使用されるプロトコル。

MX レコード (MX record)　メール交換レコード。ホスト名を別のホスト名にマップする、DNS レコードの一種。

NDN　**非配信通知**を参照してください。

NOTARY メッセージ (notary messages)　RFC 1892 の NOTARY 仕様に準拠した非配信通知 (NDN) および配信ステータス通知 (DSN)。

OSI ツリー (OSI tree)　Open Systems Interconnect ネットワーク構造を反映するディレクトリ情報ツリー。OSI ツリー内の識別名は、`cn=billt,o=bridge,c=us` のようになります。

POP3　**Post Office Protocol Version 3** を参照してください。

Post Office Protocol Version 3 (POP3)　標準の配信方法を提供するプロトコル。このプロトコルを使用する場合、MTA (Message Transfer Agent) はユーザのメールフォルダへのアクセス権を持っている必要はありません。アクセス権が不要なことは、メールクライアントと MTA が別のコンピュータに置かれることが多いネットワーク環境で利点となります。

queue　**メッセージキュー**を参照してください。

RC2　RSA Data Security によって提供される可変鍵サイズによるブロック暗号化方式。

RC4　RSA Data Security によって提供されるストリーム暗号化方式。RC2 よりも高速に処理できます。

RDN　相対識別名。実際のエントリ自体の名前。この文字列にエントリの祖先を付加すると完全な識別名になります。

RFC　Request For Comments の略。1969 年に開始されたドキュメントシリーズで、インターネットの一連のプロトコルと、関連する実験について記述されています。インターネット標準について記述した RFC の数はわずかですが、すべてのインターネット標準が RFC として公開されています。`http://www.imc.org/rfcs.html` を参照してください。

SASL　**Simple Authentication and Security Layer** を参照してください。

SCM　**Service Control Manager** を参照してください。

Secure Sockets Layer (SSL)　クライアントとサーバの間での安全な接続を確立するソフトウェアライブラリ。

sendmail　UNIX マシンで使用される一般的な MTA。ほとんどのアプリケーションでは、sendmail の代わりに iPlanet Messaging Server を使用できます。

server_root　特定のホスト上にある Administration Server に関連付けられたすべての iPlanet サーバがインストールされているディレクトリ。通常、*server-root* と記述します。**インストールディレクトリ**、**インスタンスディレクトリ**も参照してください。

Service Control Manager　サービスを管理するための Windows NT の管理プログラム。

Sieve　メールのフィルタリング言語。

Simple Authentication and Security Layer (SASL)　POP、IMAP、または SMTP クライアントがサーバから識別されるようにするためのメカニズムを制御する手段。iPlanet Messaging Server での SMTP SASL の使用は、RFC 2554 (ESMTP AUTH) に準拠しています。SASL は、RFC 2222 に定義されています。

Simple Mail Transfer Protocol (SMTP)　インターネットでもっとも一般的に使用される電子メールプロトコルで、iPlanet Messaging Server でもサポートされています。RFC 821 に定義されています。また関連するメッセージ形式が RFC 822 に記述されています。

SIMS　Sun Internet Mail Server の略。

SIZE　クライアントが特定のメッセージのサイズをサーバに対して宣言できるようにする SMTP 拡張機能。サーバは、宣言されたメッセージサイズに基づいて、メッセージ受信の可否をクライアントに示すことができます。サーバは、受信可能なメッセージの最大サイズをクライアントに宣言できます。RFC 1870 に定義されています。

SMTP　**Simple Mail Transfer Protocol** を参照してください。

SMTP AUTH　**AUTH** を参照してください。

sn　苗字を表すエイリアスディレクトリ属性。

SSL　**Secure Sockets Layer** を参照してください。

SSR　**サーバ側規則**を参照してください。

TCP　**Transmission Control Protocol** を参照してください。

TCP/IP　**Transmission Control Protocol/Internet Protocol** を参照してください。

TLS　**Transport Layer Security** を参照してください。

Transmission Control Protocol (TCP)　2 つのホスト間での信頼性の高い接続指向のストリームサービスを提供するインターネットプロトコル群内の基本転送プロトコル。

Transmission Control Protocol/Internet Protocol (TCP/IP)　インターネットプロトコルで使用される複数のネットワークプロトコルの総称。この名前は、トランスポート層のプロトコルである TCP (Transmission Control Protocol) とネットワーク層のプロトコルである IP (Internet Protocol) の 2 つの主要ネットワークプロトコルを指します。

Transport Layer Security (TLS)　SSL を標準化したもの。**Secure Sockets Layer** も参照してください。

UA　**ユーザエージェント**を参照してください。

UBE　**不特定多数宛てのメール**を参照してください。

UID　(1) ユーザ識別子。システムでユーザを識別するための一意の文字列。ユーザ ID とも呼ばれます。(2) ユーザ ID (ログイン名) のエイリアスディレクトリ属性。

Users and Groups Directory Server (User/Groups Directory Server)　組織内のユーザおよびグループに関する情報を保持する Directory Server。

UUCP　UNIX to UNIX Copy Program (UNIX から UNIX へのコピープログラム) の略。UNIX システム間での通信に使用されるプロトコルです。

Veritas Cluster Server　iPlanet Messaging Server と統合できる Veritas Software 製の高可用性クラスタリングソフトウェア。

VRFY　ユーザ名を確認するための SMTP コマンド。RFC 821 に定義されています。

Web サーバ (Web server)　World Wide Web アクセスを提供するために導入されるソフトウェアプログラムまたはサーバコンピュータ。Web サーバは、ユーザからの要求を受け取り、要求されたファイルやアプリケーションを検索し、さらにエラーメッセージを発行します。

Web メール (webmail)　ブラウザベースの電子メールサービスを示す一般的な用語。ブラウザベースのクライアントは、多くの処理をサーバに任せるので、「シンクライアント」とも呼ばれ、常にサーバ上に格納されるメールにアクセスします。**Messenger Express** も参照してください。

X.400　メッセージ処理システムの標準。

アカウント (account)　特定のユーザまたはユーザグループを定義する情報。この情報には、ユーザやグループの名前、1 つまたは複数の有効な電子メールアドレス、および電子メールの配信方法と配信先が含まれます。

アクセス制御規則 (access control rules)　特定のディレクトリエントリまたは属性のセットに対するユーザの権限を指定する規則。

アクセス制御情報 (access control information)　ACI。アクセス制御リストの単一の情報項目。

アクセス制御リスト (access control list)　ACL。ディレクトリに対するユーザやグループのアクセス権を定義するためにディレクトリに関連付けられた一連のデータ。

アクセスドメイン (access domain)　指定したドメイン内から利用できる Messaging Server 操作を制限します。たとえば、アクセスドメインを使用すると、特定のアカウント宛てのメールを収集できる場所を制限できます。

アクセスの制御 (access control)　サーバ、またはサーバ上のフォルダやファイルへのアクセスを制御する方法。

アドレス (address)　電子メールメッセージの送信先と送信方法を決定するメッセージ内の情報。アドレスはメッセージヘッダーとメッセージエンベロープの両方に表示されます。エンベロープアドレスは、メッセージのルーティング方法と配信方法を決定します。ヘッダーアドレスは表示専用です。

アドレス指定プロトコル (addressing protocol)　電子メールの利用を可能にするアドレス指定規則。RFC 822 は、インターネット上でもっとも幅広く使用されているプロトコルで、iPlanet Messaging Server でサポートされています。その他のプロトコルには、X.400 や UUCP (UNIX to UNIX Copy Protocol) などがあります。

アドレス処理 (address handling)　アドレス指定のエラーを検出し、必要に応じてアドレスを書き換え、アドレスと受取人の照合を行うために MTA によって実行される処理。

アドレストークン (address token)　書き換え規則パターンのアドレス要素。

暗号化 (encryption)　コードキーを持つ特定の受取人以外には解読できないように情報を隠すプロセス。

安全なファイルシステム (safe file system)　システムがクラッシュした場合に、データをクラッシュ前の状態にロールバックし、すべてのデータをリストアできるようにログを記録しているファイルシステム。安全なファイルシステムの例として、Veritas File System (VxFS) などがあります。

一時的な失敗 (transient failure)　メッセージ処理中に発生するエラー状態。リモート MTA が、配信時にメッセージを処理できない場合でも、あとで処理可能になることがあります。ローカル MTA は、メッセージをキューに戻し、あとで再転送されるようにスケジュールします。

インスタンス (instance)　個別に実行可能なサーバの設定、または特定のホスト上にあるその他のソフトウェアエンティティ。インストール済みの 1 組のバイナリファイルから、独立して実行およびアクセスできる複数の iPlanet サーバのインスタンスを作成できます。

インスタンスディレクトリ (instance directory)　サーバの特定のインスタンスを定義するファイルを含むディレクトリ。Messaging Server の場合は、サーバルートのサブディレクトリ (*server_root*/msg-*instance*/) です。*instance* は、インストール時に指定したサーバの名前です。**インストールディレクトリ**、**サーバルート**も参照してください。

インストールディレクトリ (installation directory)　サーバのバイナリ (実行可能) ファイルがインストールされるディレクトリ。Messaging Server の場合は、サーバルートのサブディレクトリ (*server-root*/bin/msg/) です。**インスタンスディレクトリ**、**サーバルート**も参照してください。

インターネット (Internet)　TCP/IP プロトコルを使用する、世界規模のネットワークのネットワーク。

インターネットプロトコル (IP) (Internet Protocol)　インターネットおよびイントラネットの基礎となる基本ネットワークレイヤープロトコル。

インターネットプロトコルアドレス (internet protocol address)　**IP アドレス**を参照してください。

イントラネット (intranet)　企業や組織内における複数の TCP/IP ネットワークのネットワーク。イントラネットでは、World Wide Web で使われているものと同種のサーバおよびクライアントソフトウェアを、企業 LAN 上で提供される社内アプリケーションとして使用できます。インターネットと通信するイントラネット上の機密情報は、通常はファイアウォールで保護されます。**ファイアウォール**、**エクストラネット**も参照してください。

永続的な失敗 (permanent failure)　メッセージ処理中に発生するエラー状態。この状態が発生すると、メッセージストアは電子メールメッセージのコピーを削除します。MTA はメッセージを差出人に戻し、メッセージのコピーを削除します。

エイリアス (alias)　電子メールアドレスの別名。

エイリアスの参照解除 (dereferencing an alias)　バインドまたは検索で、ディレクトリサービスがエイリアス識別名をエントリの実際の識別名に変換するように指定すること。

エイリアスファイル (alias file)　ポストマスターエイリアスなど、ディレクトリ内に設定されていないエイリアスを設定するために使用されるファイル。

エクストラネット (extranet)　企業イントラネットで顧客や供給業者がアクセスできる部分。**イントラネット**も参照してください。

エクスパンダ (expander)　メッセージをアドレスのリストに配信できるようにする、電子メール配信システムの一部。メールエクスパンダは、メーリングリストを実装するために使用されます。ユーザが 1 つのアドレス (`hacks@somehost.edu` など) にメッセージを送信すると、メールエクスパンダがリスト内のメールボックスへの配信を行います。メールエクスプローダとも呼ばれます。**EXPN** も参照してください。

エクスパンド (expansion)　この用語は、MTA によるメールリストの処理で使用されます。メールリスト宛てのメッセージを、各メールリストのメンバーに必要な数のコピーに変換することです。

エラーハンドラ (error handler)　エラーを処理するプログラム。Messaging Server では、エラーメッセージを発行し、ポストマスターが入力したエラーアクションフォームを処理します。

エラーハンドラアクションフォーム (Error-Handler Action form)　Messaging Server が処理できない受信メッセージとともにポストマスターアカウントに送信されるフォーム。ポストマスターは、フォームに入力して、メッセージの処理方法をサーバに指示します。

エラーメッセージ (error message)　エラーやその他の状況をレポートするメッセージ。iPlanet Messaging Server は、処理できない電子メールメッセージを受け取った場合など、さまざまな状況でメッセージを生成します。また、情報の通知だけを目的とする通知エラーと呼ばれるメッセージもあります。

エンタープライズネットワーク (enterprise network)　地理的に分散している相互接続されたネットワークの集合で構成されるネットワーク。エンタープライズネットワークは、広範囲に分散している企業のニーズを満たすもので、企業のミッションクリティカルなアプリケーションで使用されます。

エンベロープ (envelope)　電子メールメッセージの差出人と受取人に関する情報を転送するためのコンテナ。これらの情報は、メッセージヘッダーには含まれません。エンベロープは、さまざまな電子メールプログラムで、メッセージを別の場所に移動するときに使用します。ユーザには、メッセージのヘッダーと本文だけが表示されます。

エンベロープフィールド (envelope field)　メッセージエンベロープ内の名前付きの情報項目。`RCPT TO` などがあります。

オブジェクトクラス (object class)　エントリが記述するオブジェクトの種類と、そのエントリに含まれる属性のセットを指定するテンプレート。たとえば、iPlanet Directory Server では、`commonname`、`mail` (電子メールアドレス)、`mailHost`、`mailQuota` などの属性を持つ `emailPerson` オブジェクトクラスが指定されます。

オフライン状態 (off-line state)　メールクライアントがサーバシステムからクライアントシステムにメッセージをダウンロードして、メッセージの表示や返信の作成ができる状態。サーバ上のメッセージは、削除される場合と削除されない場合があります。

オンライン状態 (online state)　メッセージをサーバ上に残したまま、メールクライアントがリモートから返信する状態。

書き換え規則 (rewrite rules)　ドメイン書き換え規則とも呼ばれます。MTA が配信メッセージを正しいホストにルーティングするために使用するツールです。書き換え規則には、以下の機能があります。(1) 受信メッセージのアドレスからホストまたはドメインの仕様を抽出する。(2) ホストまたはドメイン仕様を書き換え規則のパターンと照合する。(3) ドメインテンプレートに基づいてホストまたはドメイン仕様を書き換える。(4) メッセージを置くチャネルキューを決定する。

鍵データベース (key database)　サーバの証明書用の鍵のペアを含むファイル。鍵ファイルとも呼ばれます。

仮想ドメイン (virtual domain)　(1) ISP ホストドメイン。(2) Messaging Multiplexor によってクライアントのユーザ ID に追加されるドメイン名。LDAP 検索やメールボックスサーバへのログインで使用します。**ドメイン**、**ホストドメイン**も参照してください。

完全指定ドメイン名 (FQDN) (fully-qualified domain name)　特定のインターネットホストを識別する一意の名前。**ドメイン名**も参照してください。

管理権限 (administration privileges)　ユーザの管理に関する役割を定義する権限のセット。

管理コンソール (administration console)　**コンソール**を参照してください。

管理者 (administrator)　定義済みの一連の管理権限を持つユーザ。**構成管理者**、**Directory Manager**、**Administration Server 管理者**、**サーバ管理者**、**メッセージストア管理者**、**トップレベル管理者**、**ドメイン管理者**、**組織管理者**、**ファミリーグループ管理者**、**メールリスト所有者**も参照してください。

管理対象オブジェクト (managed object)　設定可能な属性の集まり。たとえば、ディレクトリサービスの属性の集まりです。

管理ドメイン (administration domain)　管理制御の対象範囲。**ドメイン**も参照してください。

共有フォルダ (shared folder)　複数のユーザが読み取り可能なフォルダ。共有フォルダに対しては所有者が指定されます。所有者は、フォルダに対する読み取りアクセス権を指定したり、共有フォルダからメッセージを削除したりできます。共有フォルダにはモデレータを指定することもできます。モデレータは、受信メッセージの編集、ブロック、転送などを行うことができます。共有できるのは IMAP フォルダだけです。**個人用フォルダ**も参照してください。

許可フィルタ (Allow filter)　次のサービスへのアクセスを許可されているクライアントを識別するための、Messaging Server のアクセス制御規則。POP、IMAP、または HTTP。**拒否フィルタ**も参照してください。

拒否フィルタ (Deny filter)　次のサービスへのアクセスを拒否されているクライアントを識別するための、Messaging Server アクセス制御規則。POP、IMAP、または HTTP。**許可フィルタ**も参照してください。

クライアント (client)　サーバにサービスまたは情報を要求するソフトウェアエンティティ。

クライアントサーバモデル (client-server model)　ネットワーク接続されたコンピュータがほかのクライアントコンピュータに特定のサービスを提供する処理モデル。例として、DNS の ネームサーバとネームリゾルバのパラダイム、NFS やディスクレスホストなどのファイルサーバとファイルクライアントの関係などがあります。

クラスパス (class path)　サーブレットエンジンとサーブレットテンプレートを実行するために必要なディレクトリおよび `.jar` ファイルへのパス。

グリーティングフォーム (greeting form)　ユーザのアカウントが作成されたときにユーザに送信されるメッセージ。このフォームは、新しいアカウントを確認し、その内容を検証するために使用されます。

グループ (group)　識別名によって編成された LDAP メールエントリのグループ。通常は、メールリストとして使用されますが、グループのメンバーに特定の管理権限を与えるために使用される場合もあります。**動的グループ**、**静的グループ**も参照してください。

グループフォルダ (group folders)　これらのフォルダには、共有フォルダとグループフォルダが含まれます。**共有フォルダ**も参照してください。

ゲートウェイ (gateway)　ゲートウェイおよびアプリケーションゲートウェイという用語は、1 つのネイティブフォーマットから別のフォーマットへの変換を行うシステムを指します。例として、X.400 と RFC 822 間の送受信を行う電子メールゲートウェイがあります。複数の電子メールシステム (特に、2 つの異なるネットワーク上の類似性のないメールシステム) を接続し、その間でメッセージを転送するマシンです。マッピングと変換は複雑になることもあり、一般的に、あるシステムからメッセージを完全に受け取ってから適切な変換を行って次のシステムに送信するようなストアアンドフォワードのしくみが必要です。

検索ベース (search base)　**ベース DN** を参照してください。

公開鍵暗号化 (public key encryption)　公開コンポーネントと非公開コンポーネントの 2 つの部分から成る鍵 (コード) を使用する暗号化方式。メッセージの暗号化には、受取人の公開鍵が使われます。メッセージを解読する場合は、受取人が、自分だけが知っている非公開の鍵を使用します。

高可用性 (High Availability)　サービスの中断を検出できるようにし、システム障害やプロセス失敗時の回復メカニズムを提供します。さらに、一次システムに障害が発生した場合には、バックアップシステムがサービスを引き継ぐことができるようにします。

構成管理者 (configuration administrator)　iPlanet トポロジ全体のサーバおよび構成ディレクトリデータの管理権限を持つユーザ。構成管理者は、iPlanet トポロジ内のすべてのリソースに無制限にアクセスできます。ほかの管理者にサーバアクセス権を割り当てることができる唯一の管理者です。構成管理者は、管理者グループとそのメンバーが配置されるまで初期の管理構成を管理します。

個人用フォルダ (personal folder)　所有者だけが読み取り可能なフォルダ。**共有フォルダ**も参照してください。

コマンドラインインタフェース (command line interface)　コマンドラインから実行できるコマンド。ユーティリティとも呼ばれます。

コメント文字 (comment character)　行の最初に配置することで、その行を実行されないコメントに変換する文字。

コンソール (Console)　多くの iPlanet コンポーネントの設定、監視、管理、およびトラブルシューティングを行うことができる GUI (グラフィカルユーザインタフェース)。

サーバインスタンス (server instance)　インストールされた特定のサーバソフトウェアを表す、ディレクトリ、プログラム、およびユーティリティ。

サーバ側規則 (SSR) (server side rules)　サーバ側でメールをフィルタリングできるようにする規則のセット。Sieve メールフィルタリング言語に基づいています。

サーバ管理者 (server administrator)　サーバ管理タスクを実行するユーザ。サーバ管理者は、タスク ACI に基づいて、特定のサーバのタスクに制限付きのアクセス権を提供します。構成管理者が、ユーザにサーバへのアクセス権を割り当てる必要があります。サーバへのアクセス権を与えられたユーザは、サーバ管理者となり、サーバへのアクセス権をほかのユーザに与えることができます。

サービス (service)　(1) サーバが提供する機能。たとえば、iPlanet Messaging Server は、SMTP、POP、IMAP、HTTP などのサービスを提供します。(2) ユーザインタフェースを持たない Windows NT 上のバックグラウンドプロセス。iPlanet サーバは、Windows NT プラットフォーム上ではサービスとして稼働します。UNIX プラットフォーム上の**デーモン**と同じです。

サービス拒否攻撃 (denial of service attack)　個人が意図的にまたは誤ってメッセージを大量に送信したために、メールサーバが処理不能になる状態。サーバのスループットに著しい悪影響を与えたり、サーバ自体が過負荷状態になって機能しなくなることがあります。

サーブレット (servlet)　Web サーバがクライアントの要求に応じてコンテンツを生成するために実行するサーバ側の Java プログラム。サーブレットは、サーバ側で実行されますが、ユーザインタフェースを使用しないという点でアプレットに似ています。

再組立 (defragmentation)　MIME (Multipurpose Internet Mail Extension) の機能で、大きいサイズのメッセージが小さなメッセージ (断片) に分割された場合に、そのメッセージを再現します。各断片の Message Partial Content-Type ヘッダーフィールドには、断片を 1 つのメッセージに再組立するために使用する情報が含まれています。**断片化**も参照してください。

サブドメイン (subdomain)　ドメインの一部。たとえば、`corp.siroe.com` というドメイン名では、`corp` はドメイン `siroe.com` のサブドメインを示します。**ホスト名**、**完全指定ドメイン名**も参照してください。

サブネット (subnet)　ホスト ID のブロックを識別する IP アドレスの一部分。

参照 (referral)　Directory Server が情報要求を送信したクライアントに対し、そのクライアントが通信する必要がある DSA (Directory Service Agent) に関する情報とともに情報要求を返すプロセス。**知識情報**も参照してください。

識別名 (distinguished name)　ディレクトリ情報ツリー内のエントリの位置を一意に指定する、カンマで区切られた一連の属性と値。通常、DN と略記されます。

自動返信オプションファイル (autoreply option file)　Vacation 通知ファイルなどの自動返信オプションを設定するために使用するファイル。

自動返信ユーティリティ (AutoReply utility)　自動返信機能が有効になっているアカウント宛てに送信されたメッセージに対し、自動的に返信するためのユーティリティ。iPlanet Messaging Server 内のすべてのアカウントは、受信メッセージに対して自動的に返信するように設定できます。

従属参照 (subordinate reference)　ディレクトリサーバによって保持される名前付きコンテキストの子の名前付きコンテキスト。**知識情報**も参照してください。

上位参照 (upper reference)　ディレクトリ情報ツリー (DIT) 内で、ディレクトリサーバの名前付きコンテキストの上位にある名前付きコンテキストを保持するディレクトリサーバを示します。

使用可能な属性 (allowed attributes)　特定のオブジェクトクラスを使用するエントリについて指定できるが、必須ではない属性。**属性**、**必須の属性**も参照してください。

証明書データベース (certificate database)　サーバのデジタル証明書 (1 つまたは複数) が含まれているファイル。証明書ファイルとも呼ばれます。

証明書に基づく認証 (certificate-based authentication)　クライアントが提供したデジタル証明書によるユーザの識別。**パスワード認証**も参照してください。

証明書名 (certificate name)　証明書とその所有者を特定する名前。

ジョブコントローラ (Job Controller)　ほかのさまざまな MTA コンポーネントの要求に応じてタスクをスケジュールおよび実行する MTA コンポーネント。

シングルサインオン (single sign-on)　ユーザを一度認証するだけで、複数のサービス (メール、ディレクトリ、ファイルサービスなど) にアクセスできるようにする機能。

スキーマ (schema)　iPlanet Directory Server 内にエントリとして格納できる情報のタイプの定義 (構造と構文を含む)。スキーマと一致しない情報がディレクトリに格納されていると、ディレクトリにアクセスするクライアントが適切な結果を表示できない場合があります。

スプーフィング (spoofing)　ネットワーク攻撃の形態の 1 つで、サーバにアクセスまたはメッセージ送信しようとしているクライアントに、不正なホスト名を使用させること。

スマートホスト (smart host)　ほかのメールサーバが受取人を認識できない場合に、メッセージの転送先となる、ドメイン内のメールサーバ。

スレーブチャネルプログラム (slave channel program)　リモートシステムによって開始された転送を受け入れるチャネルプログラム。**マスターチャネルプログラム**も参照してください。

スレッド (thread)　プロセス内の小さな実行インスタンス。

正規表現 (regular expression)　パターンマッチングのために、文字の範囲またはクラスを表す特殊文字を使った文字列。

静的グループ (static group)　各グループメンバーを列挙することにより静的に定義されたメールグループ。**動的グループ**も参照してください。

セキュリティモジュールデータベース (security-module database)　SSL 暗号化方式用のハードウェアアクセラレータを記述する情報を含むファイル。`secmod` とも呼ばれます。

セッション (session)　クライアントサーバ接続のインスタンス。

切断状態 (disconnected state)　メールクライアントはサーバに接続し、選択したメッセージのキャッシュコピーを作成してからサーバとの接続を切断します。

設定ファイル (configuration file)　iPlanet Messaging システムの特定のコンポーネントに対する設定パラメータが含まれているファイル。

相対識別名 (relative distinguished name)　**RDN** を参照してください。

組織管理者 (organization administrator)　Delegated Administrator for Messaging and Collaboration の GUI または CLI を使用して、組織またはサブ組織内のメールユーザとメールリストの作成、変更、および削除を行う管理権限を持つユーザ。

代替アドレス (alternate address)　アカウントの二次的なアドレス。通常はプライマリアドレスを変化させたものです。1 つのアカウントに複数のアドレスがあると便利な場合があります。

単一フィールド置換文字列 (single field substitution string)　書き換え規則において、ホストまたはドメインアドレスの指定アドレストークンを動的に書き換えるドメインテンプレートの一部分。**ドメインテンプレート**も参照してください。

断片化 (fragmentation)　大きなメッセージを複数の小さなメッセージに分割できるようにする Multiple Internet Mail Extensions (MIME) 機能。**再組立**も参照してください。

知識情報 (knowledge information)　ディレクトリサービスインフラストラクチャ情報の一部。Directory Server は、知識情報を使用して、情報要求をほかのサーバに渡します。

チャネル (channel)　メッセージを処理する基本的な MTA コンポーネント。チャネルは、別のコンピュータシステムまたはシステムグループとの接続を表します。各チャネルは、1 つまたは複数のチャネルプログラムと 1 つの送信メッセージキューから構成されます。送信メッセージキューには、そのチャネルに関連付けられている 1 つまたは複数のシステム宛てのメッセージが格納されます。**チャネルブロック**、**チャネルホストテーブル**、**チャネルプログラム**も参照してください。

チャネルプログラム (channel program)　次の機能を実行するチャネルの一部。(1) メッセージをリモートシステムに送信し、送信後にメッセージをキューから削除する。(2) リモートシステムからメッセージを受信して適切なチャネルキューに置く。**マスターチャネルプログラム**、**スレーブチャネルプログラム**も参照してください。

チャネルブロック (channel block)　単一のチャネル定義。**チャネルホストテーブル**も参照してください。

チャネルホストテーブル (channel host table)　チャネル定義のセット。

通知メッセージ (notification message)　Messaging Server によって送信されるメッセージの一種で、メッセージ配信処理のステータスと、配信に関する問題や障害の理由などを知らせます。このメッセージは、情報提供を目的とし、ポストマスターに対してアクションを要求するものではありません。**配信ステータス通知**も参照してください。

次のホップリスト (next-hop list)　メール経路で、メッセージの転送先を判別するために使用される近接システムのリスト。次のホップリスト内のシステムの順序によって、メール経路内でシステムにメッセージが転送される順序が決まります。

データストア (data store)　ディレクトリ情報の保存場所。通常はディレクトリ情報ツリー全体の情報が含まれます。

デーモン (daemon)　端末から独立してバックグラウンドで動作し、必要に応じて機能を実行する UNIX プログラム。デーモンプログラムの一般的な例として、メールハンドラ、ライセンスサーバ、印刷デーモンなどがあります。Windows NT マシンの場合、この種のプログラムはサービスと呼ばれます。**サービス**も参照してください。

ディスパッチャ (Dispatcher)　定義済み TCP ポートへの接続要求を処理する MTA コンポーネント。ディスパッチャは、複数のマルチスレッドサーバが特定のサービスを分担できるようにするマルチスレッド接続ディスパッチエージェントです。ディスパッチャを使用すると、複数のマルチスレッド SMTP サーバプロセスを同時に実行できるようになります。

ディレクトリエントリ (directory entry)　識別名で特定されるディレクトリ属性とその値のセット。各エントリには、エントリが記述するオブジェクトの種類を指定し、エントリに含まれる属性のセットを定義するオブジェクトクラス属性が含まれています。

ディレクトリ検索 (directory lookup)　ユーザやリソースの名前またはその他の特性を基準として、ディレクトリ内で特定のユーザやリソースに関する情報を検索するプロセス。

ディレクトリコンテキスト (directory context)　メッセージストアへのアクセスに対して、ユーザとパスワードの認証に使用するエントリの検索を開始するディレクトリツリー情報内のポイント。**ベース DN** も参照してください。

ディレクトリサービス (directory service)　組織内の人材とリソースに関する、論理的に集中化された情報のリポジトリ。**Lightweight Directory Access Protocol** も参照してください。

ディレクトリ情報ツリー (directory information tree)　ディレクトリエントリを編成する、ツリー状の階層構造。DIT とも呼ばれます。DIT は DNS (DC ツリー) または Open Systems Interconnect ネットワーク (OSI ツリー) に従って編成できます。

ディレクトリスキーマ (directory schema)　ディレクトリに保存できるデータを定義する一連の規則。

ディレクトリ同期 (directory synchronization)　MTA ディレクトリキャッシュをディレクトリサービスに保存された現在のディレクトリ情報で更新 (同期化) するプロセス。**MTA ディレクトリキャッシュ**も参照してください。

転送 (forwarding)　**メッセージの転送**を参照してください。

転送プロトコル (transport protocols)　SMTP や X.400 など、MTA 間でのメッセージ転送手段を提供するプロトコル。

統一メッセージング (unified messaging)　電子メール、ボイスメール、FAX、およびその他の通信形態に関して単一のメッセージストアを使用するという概念。iPlanet Messaging Server では、完全な統一メッセージングソリューションの基盤を提供します。

同期 (synchronization)　(1) マスターディレクトリサーバのデータによる複製ディレクトリサーバのデータの更新。(2) MTA ディレクトリキャッシュの更新。

動的グループ (dynamic group)　LDAP 検索 URL で定義されるメールグループ。通常、ユーザはディレクトリエントリ内で LDAP 属性を設定することによってグループに参加します。

ドキュメントルート (document root)　iPlanet Web Server にアクセスするユーザに対して表示されるファイル、イメージ、データを含むサーバマシン上のディレクトリ。

トップレベル管理者 (top-level administrator)　Delegated Administrator for Messaging and Collaboration の GUI または CLI を使用して、Messaging Server ネームスペース全体のメールユーザ、メールリスト、ファミリーアカウント、およびドメインの作成、変更、および削除を行う管理権限を持つユーザ。デフォルトでは、このユーザは、トポロジ内のすべての Messaging Server のメッセージストア管理者として作業することができます。

ドメイン (domain)　単一のコンピュータシステムの制御下にあるリソース。**管理ドメイン**、**DNS ドメイン**、**ホストドメイン**、**仮想ドメイン**も参照してください。

ドメインエイリアス (domain alias)　別のドメインを指すドメインエントリ。ホストドメインはエイリアスを使用することにより、複数のドメイン名を持つことができます。

ドメイン書き換え規則 (domain rewrite rules)　**書き換え規則**を参照してください。

ドメイン管理者 (domain administrator)　Delegated Administrator for Messaging and Collaboration の GUI または CLI を使用して、ホストドメイン内のメールユーザ、メールリスト、およびファミリーアカウントの作成、変更、および削除を行うための管理権限を持つユーザ。デフォルトでは、このユーザは、トポロジ内のすべての Messaging Server のメッセージストア管理者として作業することができます。

ドメイン制限容量 (domain quota)　電子メールメッセージ用にドメインに割り当てられる容量で、システム管理者によって設定されます。

ドメイン組織 (domain organization)　組織ツリー内でホストドメインの下にあるサブドメイン。ドメイン組織は、企業内でユーザとグループのエントリを部門別に編成する場合に有用です。

ドメインテンプレート (domain template)　書き換え規則の一部で、アドレスのホスト部分とドメイン部分の書き換え方法を定義します。テンプレートは、完全に静的なホストアドレスおよびドメインアドレス、または単一フィールド置換文字列、あるいはその両方を含む場合があります。

ドメインネームシステム (DNS) (Domain Name System)　コンピュータが、ネットワークまたはインターネット上のほかのコンピュータをドメイン名で見つけることができるようにする分散型名前解決ソフトウェア。システムは、標準 IP アドレスをホスト名 (`www.siroe.com` など) に関連付けます。通常、各マシンはこの情報を DNS サーバから取得します。DNS サーバは、ホスト名をインターネットアドレスに変換するための、複製された分散型のデータ照会サービスを提供します。**A レコード**、**MX レコード**、**CNAME レコード**も参照してください。

ドメイン部分 (domain part)　電子メールアドレスの @ 記号の右側にある部分。たとえば、`siroe.com` は、電子メールアドレス `dan@siroe.com` のドメイン部分です。

ドメインホスティング (domain hosting)　共有 Messaging Server 上で 1 つまたは複数のドメインをホストする機能。たとえば、`siroe.com` と `sesta.org` の両方のドメインを `siroe.net` メールサーバ上でホストできます。ユーザは、ホストドメインとの間でメールの送受信を行います。メールサーバの名前は、電子メールアドレスには表示されません。

ドメイン名 (domain name)　(1) 電子メールアドレス内で使用されるホスト名。(2) 管理組織を定義する一意の名前。ドメインにはほかのドメインを含めることができます。ドメイン名は右から左の方向に解釈されます。たとえば、`siroe.com` は、Siroe Company のドメイン名であり、かつトップレベルの `com` ドメインのサブドメインです。`siroe.com` ドメインをさらに `corp.siroe.com` などのサブドメインに分割することもできます。**ホスト名**、**完全指定ドメイン名**も参照してください。

名前解決 (name resolution)　IP アドレスを対応する名前にマップするプロセス。**DNS** も参照してください。

名前付きコンテキスト (naming context)　DN によって識別されるディレクトリ情報ツリーの特定の接尾辞。iPlanet Directory Server では、特定のタイプのディレクトリ情報が名前付きコンテキストに格納されます。たとえば、Siroe Corporation のボストンオフィスのマーケティング部門の社員のすべてのエントリを格納する名前付きコンテキストは、`ou=mktg, ou=Boston, o=siroe, c=US` のようになります。

名前付き属性 (naming attribute)　ディレクトリ情報ツリーの識別名の最後の属性。**相対識別名**も参照してください。

認証 (authentication)　(1) iPlanet Messaging Server に対し、クライアントユーザであることを立証するプロセス。(2) クライアントまたは別のサーバに対し、iPlanet Messaging Server であることを立証するプロセス。

認証局 (Certificate Authority)　**CA** を参照してください。

認証証明書 (authentication certificate)　相手を検証し認証するために、サーバからクライアント、またはクライアントからサーバに送信されるデジタルファイル。証明書は、その所有者 (クライアントまたはサーバ) の信頼性を保証します。証明書は譲渡できません。

ネームスペース (namespace)　LDAP ディレクトリのツリー構造。**ディレクトリ情報ツリー**も参照してください。

ネットワークマネージャ (network manager)　SNMP データの読み取り、フォーマット、および表示を行うプログラム。SNMP クライアントとも呼ばれます。

ノード (node)　DIT 内のエントリ。

パーティション (partition)　**メッセージストアパーティション**を参照してください。

配信 (delivery)　**メッセージの配信**を参照してください。

配信ステータス通知 (delivery status notification)　受取人に配信中のメッセージに関するステータス情報を示すメッセージ。たとえば、ネットワークが停止したために配信が遅れていることを知らせるメッセージなどがあります。

配布リスト (distribution list)　**メールリスト**を参照してください。

配布リスト所有者 (distribution list owner)　**メールリスト所有者**を参照してください。

バインド DN (bind DN)　操作実行時に Directory Server に対する認証に使用される識別名。

パスワードの認証 (password authentication)　ユーザ名とパスワードによるユーザの識別。**証明書に基づく認証**も参照してください。

パターン (pattern)　許可フィルタや拒否フィルタなどで、マッチングのために使用される文字列表現。

バックアップ (backup)　メッセージストアのフォルダの内容をバックアップデバイスにバックアップするプロセス。**リストア**も参照してください。

バックエンドサーバ (backend server)　電子メールメッセージの保管と取り出しの機能だけを持つ電子メールサーバ。メッセージストアサーバとも呼ばれます。

バックボーン (backbone)　分散システムの主要な接続メカニズム。バックボーン上の中間システムに接続するすべてのシステムは、相互に接続されます。バックボーンがある場合でも、コスト、パフォーマンス、セキュリティなどの理由から、バックボーンを迂回するようにシステムを設定することができます。

バニティドメイン (vanity domain)　特定のサーバまたはホストドメインではなく、個別のユーザに関連付けられているドメイン名。`MailAlternateAddress` 属性を使用して指定されます。バニティドメインのドメイン名には LDAP エントリが含まれません。バニティドメインは、個人または小さな組織が、独自のホストドメインを持つための管理負荷をかけずに、カスタマイズしたドメイン名を使用する場合に便利です。カスタムドメインとも呼ばれます。

ハブ (hub)　システムの単一接続ポイントとして機能するホスト。たとえば、2 つのネットワークがファイアウォールで分離されている場合は、しばしばファイアウォールコンピュータがメールハブとして機能します。

必須の属性 (required attributes)　特定のオブジェクトクラスを使用するエントリ内に存在している必要がある属性。**使用可能な属性**、**属性**も参照してください。

非配信通知 (nondelivery notification)　メッセージ転送中に、アドレスパターンと書き換え規則の間に一致するものが見つからない場合、MTA は、オリジナルのメッセージとともに非配信レポートを差出人に返します。

ファイアウォール (firewall)　ネットワーク構成の 1 つで、通常はハードウェアおよびソフトウェアの両方を使用して、組織内のネットワーク接続されたコンピュータと組織外のコンピュータの間の防護壁を構成します。一般に、ファイアウォールは物理的な建物または組織のサイト内にある、ネットワークの電子メール、ディスカッショングループ、データファイルなどの情報を保護するために使用されます。

ファミリーグループ管理者 (family group administrator)　ファミリーグループ内のファミリーメンバーの追加と削除を行う管理権限を持つユーザ。このユーザは、グループのほかのメンバーに管理アクセス権を与えることができます。

フェイルオーバー (failover)　冗長バックアップを提供するために、あるシステムから別のシステムにコンピュータサービスを自動転送すること。

フォルダ (folder)　メッセージの名前付きのコレクション。フォルダにはほかのフォルダを含めることができます。メールボックスとも呼ばれます。**個人用フォルダ**、**共有フォルダ**、**INBOX** も参照してください。

複製ディレクトリサーバ (replica directory server)　データのすべてまたは一部のコピーを受信するディレクトリ。

輻輳しきい値 (congestion thresholds)　システム管理者が設定できるディスク容量の上限。システムリソースが不足しているときに新しい操作を制限することによって、データベースへの過重負荷を防ぐことができます。

不特定多数宛てのメール (UBE)(Unsolicited Bulk Email (UBE))　一般に宣伝目的でメール送信業者から大量に送信される迷惑メール。

プレーンテキスト (plaintext)　データの転送方法を表します。意味は状況によって異なります。たとえば、SSL のプレーンテキストパスワードは暗号化され、cleartext (平文) としては送信されません。SASL では、プレーンテキストパスワードはハッシュされ、パスワードのハッシュだけがテキストとして送信されます。**SSL**、**SASL** も参照してください。

プレーンテキスト認証 (plaintext authentication)　**パスワード認証**を参照してください。

プロキシ (proxy)　1 つのシステムが別のシステムの代理でプロトコルの要求に応答するメカニズム。プロキシシステムをネットワーク管理で使用すると、モデムなどの単純なデバイスに完全なプロトコルスタックを実装する必要がなくなります。

プロセス (process)　オペレーティングシステムによって設定される、独立して完全に機能する実行環境。通常、アプリケーションの各インスタンスは個別のプロセスで実行されます。**スレッド**も参照してください。

プロトコル (protocol)　情報を交換する 2 つ以上のシステムが従う必要がある規則と、交換されるメッセージに関する公式の記述。

プロビジョニング (provisioning)　iPlanet Directory Server のエントリを追加、変更、または削除するプロセス。これらのエントリには、ユーザ、グループ、およびドメイン情報が含まれます。

ベース DN (base DN)　検索が開始されるディレクトリ内の識別名エントリ。検索ベースとも呼ばれます。例 : `ou=people, o=siroe.com`

ヘッダー (header)　電子メールメッセージで本文の前にある部分。ヘッダー内では、フィールド名のあとにコロンと値が続きます。ヘッダーには、電子メールプログラムとユーザにとって、メッセージが意味をなすようにするために有用な情報が含まれてます。たとえば、配信情報、内容の概要、トレース、MIME 情報などが含まれます。これらは、メッセージの受取人、差出人、送信日時、内容を示します。ヘッダーは、電子メールプログラムが読み取れるように RFC 822 に従って記述されている必要があります。

ヘッダーフィールド (header field)　メッセージヘッダー内の名前付きの情報項目。From:、To: などがあります。ヘッダー行と呼ばれることもあります。

ポート番号 (port number)　ホストマシン上の個々の TCP/IP アプリケーションを指定する番号。転送されたデータの宛先を提供します。

ホスト (host)　1 つ以上のサーバが置かれているマシン。

ホストドメイン (hosted domain)　ISP にアウトソースされた電子メールドメイン。ISP は、企業の電子メールドメインのホスティングを提供し、その企業の電子メールサービスの運営および管理を行います。ホストドメインは、ほかのホストドメインと同一の Messaging Server ホストを共有します。初期の LDAP ベースの電子メールシステムでは、1 つのドメインが 1 つまたは複数の電子メールサーバホストによってサポートされていました。Messaging Server では、複数のドメインを単一のサーバ上でホストできます。各ホストドメインには、そのドメインのユーザとグループのコンテナを指す LDAP エントリがあります。ホストドメインは、仮想ホストドメインまたは仮想ドメインとも呼ばれます。**ドメイン**、**仮想ドメイン**も参照してください。

ポストマスターアカウント (postmaster account)　Messaging Server からのシステム生成メッセージを受信する電子メールグループおよび電子メールアドレスのエイリアス。ポストマスターアカウントには、1 つ以上の有効なメールボックスを指定する必要があります。

ホスト名 (host name)　ドメイン内の特定マシンの名前。ホスト名は、IP ホスト名です。IP ホスト名としては、「短縮形」のホスト名 (mail など) または完全指定ホスト名が使用されます。完全指定ホスト名は、ホスト名とドメイン名の 2 つの部分から成ります。たとえば、`mail.siroe.com` は、ドメイン `siroe.com` 内のマシン `mail` を表します。ホスト名は、ドメイン内で一意にする必要があります。組織内の異なるサブドメイン内にある場合は、複数のマシンに `mail` という名前と付けることができます。たとえば、`mail.corp.siroe.com` と `mail.field.siroe.com` を使用できます。ホスト名は、常に、特定の IP アドレスにマップされます。**ドメイン名**、**完全指定ドメイン名**、**IP アドレス**も参照してください。

ホスト名の非表示 (host name hiding)　特定の内部ホスト名を含まないドメインベースの電子メールのアドレスを使用すること。

ホップ (hop)　2 台のコンピュータ間での転送。

本文 (body)　電子メールメッセージの一部分。ヘッダーとエンベロープは標準書式に従う必要がありますが、メッセージの本文は、テキスト、グラフィックス、マルチメディアなどを使って差出人が自由に作成できます。構造化された本文は MIME 標準に従う必要があります。

マスターチャネルプログラム (master channel program)　リモートシステムへの転送を開始するチャネルプログラム。**スレーブチャネルプログラム**も参照してください。

マスターディレクトリサーバ (master directory server)　複製されるデータを含むディレクトリサーバ。

見出し (banner)　クライアントがはじめて接続したときに IMAP などのサービスによって表示されるテキスト文字列。

無効なユーザ (invalid user)　メッセージ処理中に発生するエラー状態。この状態が発生すると、メッセージストアは、MTA と通信して、メッセージのコピーを削除します。MTA はメッセージを差出人に戻し、メッセージのコピーを削除します。

メーリングリスト (mailing list)　**メールリスト**を参照してください。

メーリングリスト所有者 (mailing list owner)　**メールリスト所有者**を参照してください。

メールクライアント (mail client)　ユーザが電子メールを送受信する際に利用するプログラム。さまざまなネットワークやメールプログラムの一部で、ユーザがもっとも頻繁に使用する部分です。メールクライアントは、配信するメッセージを作成して送信し、新たに受信したメールを確認し、受信メールを受理して整理します。

メール交換レコード (mail exchange record)　**MX レコード**を参照してください。

メールボックス (mailbox)　メッセージの格納と表示を行う場所。**フォルダ**も参照してください。

メールリスト (mail list)　電子メールアドレスのリスト。メールリストのアドレスを指定することによってそれらの電子メールアドレス宛てにメッセージを送信できます。グループと呼ばれることもあります。

メールリスト所有者 (mail list owner)　メールリストのメンバーの追加と削除を行う管理権限を持つユーザ。

メールリレー （mail relay）　MUA または MTA からのメールを受け取り、そのメールを受取人のメッセージストアや別のルーターに中継するメールサーバ。

メールルーター (mail router)　**メールリレー**を参照してください。

メッセージ (message)　電子メールの基本単位。メッセージは、ヘッダーと本文で構成され、多くの場合、差出人から受取人に転送される間はエンベロープに格納されます。

メッセージアクセスサービス (message access services)　Messaging Server メッセージストアへのクライアントアクセスをサポートするプロトコルサーバ、ソフトウェアドライバ、およびライブラリ。

メッセージキュー (message queue)　クライアントやほかのメールサーバから受け取ったメッセージを (即時または指定日に) 配信するために保管するディレクトリ。

メッセージストア (message store)　Messaging Server インスタンスに対してローカルに配信されたすべてのメッセージのデータベース。メッセージは、単一の物理ディスクに格納することも、複数の物理ディスクに格納することもできます。

メッセージストア管理者 (message store administrator)　Message Server のメッセージストアを管理する管理権限を持つユーザ。このユーザは、メールボックスの表示と監視、およびストアへのアクセス制御の指定を行うことができます。プロキシ認証の権限を使用して、ストアを管理するための特定のユーティリティを実行できます。

メッセージストアパーティション (message store partition)　単一の物理ファイルシステムパーティション上に置かれたメッセージストアまたはメッセージストアのサブセット。

メッセージ制限容量 (message quota)　特定のフォルダが消費できるディスク容量を定義する制限。

メッセージの削除 (delete message)　削除するメッセージにマークを付けること。削除したメッセージは、別の処理で消去 (パージ) するまで、メッセージストアからは削除されません。**メッセージのパージ**、**メッセージの消去**も参照してください。

メッセージの消去 (expunge message)　メッセージに削除マークを付け、その後 INBOX から永久に削除すること。**メッセージの削除**、**メッセージのパージ**も参照してください。

メッセージの送信 (message submission)　クライアントのユーザエージェント (UA) は、メールサーバにメッセージを転送し、配信を要求します。

メッセージの転送 (message forwarding)　MTA が、特定のアカウントに配信されたメッセージを、アカウントの属性で指定された 1 つまたは複数の新しい宛先に送信するときの処理。転送は、ユーザが設定できます。**メッセージの配信**、**メッセージのルーティング**も参照してください。

メッセージのパージ (purge message)　ユーザおよびグループフォルダ内で削除マークを付け、参照することのなくなったメッセージを永久に削除し、使用していた領域をメッセージストアのファイルシステムに戻すプロセス。**メッセージの削除**、**メッセージの消去**も参照してください。

メッセージの配信 (message delivery)　MTA がメッセージをローカルの受取人 (メールフォルダまたはプログラム) に配信するときの処理。

メッセージのルーティング (message routing)　最初の MTA が、受取人がローカルアカウントではなくほかの場所にいると判断したときに、別の MTA にメッセージを転送する処理。通常、ルーティングを設定できるのはネットワーク管理者だけです。**メッセージの転送**も参照してください。

メンバー (member)　メールリスト宛ての電子メールのコピーを受け取るユーザまたはグループ。**メールリスト**、**エクスパンド**、**モデレータ**、**所有者**も参照してください。

モデレータ (moderator)　メールリスト宛てのすべての電子メールを最初に受信して、以下の処理を選択実行するユーザ。(A) 配布リストにメッセージを転送する。(B) メッセージを編集してからメールリストに転送する。(C) メッセージをメールリストに転送しない。**メールリスト**、**エクスパンド**、**メンバー**も参照してください。

ユーザアカウント (user account)　サーバにアクセスするためのアカウント。ディレクトリサーバ上のエントリとして管理されます。

ユーザエージェント (UA) (user agent)　ユーザがメールメッセージを作成、送信、受信できるようにするクライアントコンポーネント。Netscape Communicator などがあります。

ユーザエントリまたはユーザプロファイル (user entry or user profile)　各ユーザに関する必須および任意の情報を記述するフィールド。識別名、氏名、役職、電話番号、ポケットベルの番号、ログイン名、パスワード、ホームディレクトリなどがあります。

ユーザ制限容量 (user quota)　電子メールメッセージ用にユーザに割り当てられる容量で、システム管理者によって設定されます。

ユーザフォルダ (user folders)　ユーザの電子メールのメールボックス。

リストア (restore)　フォルダの内容をバックアップデバイスからメッセージストアに復元するプロセス。**バックアップ**も参照してください。

リッスンポート (listen port)　サーバがクライアントやその他のサーバと通信するために使用するポート。

リバース DNS 検索 (reverse DNS lookup)　数値 IP アドレスを対応する完全指定ドメイン名に解釈するために DNS に照会するプロセス。

リレー (relaying)　メッセージサーバ間でメッセージを渡すプロセス。

ルーター (router)　複数のネットワークトラフィック経路から利用する経路を決定するシステム。ルーターは、ネットワークに関する情報を取得するためのルーティングプロトコルを使用し、さらに、「ルーティングマトリクス」と呼ばれるいくつかの条件に基づいて最善の経路を決定するアルゴリズムを使用します。OSI の用語では、ルーターはネットワークレイヤーの中間システムになります。**ゲートウェイ**も参照してください。

ルーティング (routing)　**メッセージのルーティング**を参照してください。

ルートエントリ (root entry)　ディレクトリ情報ツリー (DIT) 階層のトップレベルのエントリ。

ルックアップ (lookup)　検索の同義語。特定のパラメータを使ってデータを並べ替えます。

レベル (level)　ログの詳細度の指定。ログファイルに記録するイベントの種類の相対的な数を意味します。たとえば、Emergency レベルでは、ログに記録されるイベントはわずかですが、Informational レベルでは数多くのイベントがログに記録されます。

ローカル部分 (local part)　電子メールアドレス内の受取人を識別する部分。**ドメイン部分**も参照してください。

ログディレクトリ (log directory)　サービスのすべてのログファイルが保存されているディレクトリ。

ログ有効期限 (log expiration)　有効期間が過ぎたログファイルは、ログディレクトリから削除されます。

ログローテーション (log rotation)　現在のログファイルとして使用する新しいログファイルを作成すること。以後のログイベントは、新しいログファイルに書き込まれます。以前のログファイルはログディレクトリ内に残りますが、ログが書き込まれることはありません。

ワイルドカード (wildcard)　1 つまたは複数のほかの文字または文字範囲を表すことができる検索文字列内の特殊文字。

索引

記号

数字

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

X

あ

い

う

え

お

か

き

く

け

こ

さ

し

す

せ

そ

た

ち

つ

て

と

な

に

ね

は

ひ

ふ

へ

ほ

ま

み

め

も

ゆ

よ

り

る

ろ

わ